O'REILLY

Анализ поведенческих данных на R и Python

Как улучшить бизнес-результаты на основе данных клиентов

Флоран Бюиссон

ДМК ПРЕСС
ИЗДАТЕЛЬСТВО

Флоран Бюиссон

Анализ поведенческих данных на R и Python

Behavioral Data Analysis with R and Python

Customer-Driven Data for Real Business Results

Florent Buisson

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Анализ поведенческих данных на R и Python

Как улучшить бизнес-результаты на основе данных клиентов

Флоран Бюиссон

ДМК ПРЕСС
издательство
Москва, 2022

УДК 004.43
ББК 32.372.1
Б98

Бюиссон Ф.
Б98 Анализ поведенческих данных на R и Python / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2022. – 368 с.: ил.

ISBN 978-5-97060-992-7

Задействуйте всю мощь поведенческих данных в своей компании, используя инструменты, специально разработанные для их анализа. Автор, эксперт в области экономики и бихевиористики, показывает, как повысить ценность и результаты аналитических проектов за счет понимания того, что движет поведением людей. Практическая часть книги содержит полные примеры и упражнения на языках R и Python, которые помогут вам получать более глубокую информацию о данных.

Издание предназначено для бизнес-аналитиков и других специалистов, исследующих данные и владеющих программированием на R или Python. Для чтения требуется минимальное знакомство с линейной и логистической регрессией.

УДК 004.43
ББК 32.372.1

ISBN 978-1-492-06137-3 (англ.)
ISBN 978-5-97060-992-7 (рус.)

Положительные отзывы на книгу «Анализ поведенческих данных на R и Python»

«В отличие от некоторых книг по науке о данных, в которых авторы стремятся научить своих читателей новым техническим приемам, цель Флорана – иная и более глубокая. Он стремится научить нас мудрости, ориентированной на данные: как строить подробное и тонкое понимание данных, содержащих следы человеческого поведения».

– *Стив Вендель*,
руководитель отдела бихевиористики, Morningstar

«Книга “Анализ поведенческих данных” поможет вам разбираться в данных, даже если вы не можете проводить контролируемые эксперименты».

– *Колин Макфарланд*,
директор платформы экспериментирования, Netflix

«Мы переполнены данными, и эта книга является давно востребованным ресурсом, который направляет практиков в том, как использовать эти данные для строительства достоверных причинно-следственных моделей, которые предсказывают и объясняют поведения в реальном мире».

– *Дэвид Льюис*, президент научно-иссследовательского
института BEworks в компании BEworks

«Для всех, кто хочет применять бихевиористику в качестве проводника в принятии деловых решений, эта книга представляет собой ценное подробное введение в принципы эффективного использования причинно-следственных диаграмм во время экспериментирования и в поведенческом анализе».

– *Мэтт Райт*, директор по бихевиористике, WiderFunnel

«Часть того, что делает бихевиористику работоспособной, заключена в бесшовном сочетании количественных и качественных выводов в поддержку нашего понимания причин, почему люди делают то, что они делают. Эта книга поможет любому человеку, обладающему несколькими базовыми навыками работы с данными, принимать осмысленное участие в этом процессе бихевиористики».

– *Мэтт Уоллерт*,
руководитель отдела бихевиористики в frog,
автор книги «Начало в конце: как создавать продукты,
которые создают изменения»

Содержание

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и O'Reilly очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

Статистика является предметом удивительно многих применений и инструментом удивительно немногих эффективных практиков.

– Брэдли Эфрон и Р. Дж. Тибширани, «Введение в бутстрап» (1993)

Добро пожаловать в «Анализ поведенческих данных на R и Python»! Высказывание о том, что мы живем в век данных, уже стало банальным. Инженеры теперь регулярно используют сенсорные данные на машинах и турбинах, чтобы предсказывать время, когда они выйдут из строя, и проводят превентивное техническое обслуживание. Аналогичным образом маркетологи используют массивы данных, начиная с вашей демографической информации и заканчивая вашими прошлыми покупками, чтобы определять вид объявления, которое вам следует показывать, и время его показа. Как говорится, «данные – это новая нефть», а алгоритмы – это новый двигатель внутреннего сгорания, двигающий нашу экономику вперед.

В большинстве книг по аналитике, машинному обучению и науке о данных авторы неявно предполагают, что задачи, которые пытаются решать инженеры и маркетологи, могут решаться с помощью одних и тех же подходов и инструментов. Разумеется, переменные имеют разные имена, и необходимо приобретать некоторые знания, относящиеся к конкретной области, но кластеризация *k*-средних – это кластеризация *k*-средних, независимо от того, кластеризуете вы данные о турбинах или сообщения в социальных сетях. Принимая на вооружение инструменты машинного обучения в таком ключе, компании нередко могли точно предсказывать поведения, но ценой более глубокого и богатого понимания того, что на самом деле происходит. Это привело к критике моделей науки о данных как «черных ящиков».

Вместо того чтобы стремиться к точным, но непрозрачным предсказаниям, эта книга стремится ответить на вопрос «Что движет поведением?». Если мы решим отправить электронное письмо потенциальным клиентам, то купят ли они подписку на нашу службу в результате отправки этого электронного письма? И какие группы клиентов должны получать это электронное письмо? Склонны ли пожилые клиенты покупать разные товары, потому что они старше? Как влияет опыт клиентов на лояльность и удержание клиентов? Изменив нашу точку зрения с предсказания поведений на их объяснение и измерение причин, мы сможем снять проклятие «корреляция не есть каузация», которое мешало поколениям аналитиков быть уверенными в результатах своих моделей.

Этот сдвиг не будет связан с введением новых аналитических инструментов: мы будем использовать только два инструмента анализа данных: старую добрую линейную регрессию и ее логистического собрата. Указанные две

модели по своей сути читаются намного легче, чем другие типы моделей. Определенно, это нередко происходит ценой более низкой предсказательной точности (т. е. они допускают все больше и больше ошибок в предсказании), но здесь для нашей цели измерения взаимосвязей между переменными это не имеет значения.

Вместо этого мы потратим много времени на то, чтобы научиться разбираться в данных. В своей роли специалиста, проводящего собеседование по науке о данных, я повидал немало кандидатов, которые были способны использовать сложные алгоритмы машинного обучения, но не развили в себе сильное чувство данных: у них мало интуиции относительно того, что, собственно, происходит в их данных, кроме того что им говорят их алгоритмы.

Я твердо убежден, что вы можете развить эту интуицию и попутно повысить ценность и результаты ваших аналитических проектов – нередко значительно, – приняв следующие меры:

- бихевиористский менталитет, который взирает на данные не как на самоцель, а как на линзу для изучения психологии и поведений людей;
- инструментарий причинно-следственной (каузальной) аналитики, который позволяет нам уверенно утверждать, что одна вещь обусловливает другую, и определять силу этой взаимосвязи.

Хотя каждая из них может приносить большие выгоды сама по себе, я считаю, что они являются естественными дополнениями, которые лучше всего использовать вместе. Учитывая, что словосочетание «бихевиористский менталитет с использованием инструментария причинно-следственной аналитики» трудно выговорить, вместо него я будут называть его причинно-поведенческим подходом, или каркасом. Указанный каркас имеет дополнительную выгоду: он в равной степени применим к экспериментальным и историческим данным, используя при этом различия между ними. Это контрастирует с традиционной аналитикой, которая манипулирует ими с помощью совершенно других инструментов (например, ANOVA и T-тест для экспериментальных данных), и наукой о данных, которая не трактует экспериментальные данные отлично от исторических данных.

Для кого эта книга предназначена

Если вы анализируете данные в бизнесе на R или Python, то эта книга для вас. Я использую слово «бизнес» в широком смысле для обозначения любой коммерческой, некоммерческой или правительственной организации, где важны правильные идеи и практические выводы, которые движут действиями.

С точки зрения математики и статистики, не имеет значения, кем вы являетесь: деловым аналитиком, строящим ежемесячные прогнозы, исследователем опыта пользователей (UX), изучающим поведения на основе кликабельности, или исследователем данных, строящим модели машинного обучения. У этой книги есть одно фундаментальное условие: вы должны быть хотя бы немного знакомы с линейной и логистической регрессией. Если вы понимаете регрессию, то вы сможете проследить за аргументами этой кни-

ги и извлечь из нее большую пользу. С другой стороны, я убежден, что даже опытные исследователи данных с докторскими степенями в области статистики или компьютерных наук найдут этот материал новым и полезным, при условии что они еще не являются специалистами в области поведенческой или причинно-следственной аналитики.

С точки зрения подготовленности в качестве программиста, вы должны уметь читать и писать исходный код на R или Python, в идеале на том и другом. Я не буду показывать вам, как определять функцию или как манипулировать структурами данных, такими как кадры данных в pandas. Уже есть отличные книги, которые справляются с этим лучше, чем я, например «Python для анализа данных» Уэса Маккинни (*Python for Data Analysis*, Wes McKinney, O'Reilly)[1] и «R для науки о данных» Гарретта Гролемунда и Хэдли Уикхэма (*R for Data Science*, Garrett Grolemund and Hadley Wickham, O'Reilly)[2]. Если вы читали какую-либо из этих книг, посещали вводные занятия или использовали хотя бы один из двух языков на работе, то здесь вы будете подготовлены к излагаемому материалу. Точно так же я обычно не буду представлять и обсуждать исходный код, используемый для создания многочисленных рисунков в книге, хотя он будет размещен в репозитории книги на GitHub[3].

Для кого эта книга не предназначена

Если вы работаете в академических кругах или в области, которая требует от вас соблюдения академических норм (например, фармацевтические испытания), то эта книга все еще может представлять для вас интерес, но рецепты, которые я описываю, могут вызывать у вас проблемы с вашим консультантом/редактором/менеджером.

Эта книга не является обзором традиционных методов анализа поведенческих данных, таких как T-тест или ANOVA. Мне еще не приходилось сталкиваться с ситуацией, когда регрессия была менее эффективной, чем эти методы для предоставления ответа на деловой вопрос, поэтому я намеренно ограничиваю эту книгу линейной и логистической регрессией. Если вы хотите изучать другие методы, то вам придется поискать в другом месте (например, в книге «Практическое машинное обучение с помощью Scikit-Learn, Keras и TensorFlow» Орельена Жерона (*Hands-On Machine Learning with Scikit-Learn*, Keras, and TensorFlow, Aurélien Géron, O'Reilly)[4] в отношении алгоритмов машинного обучения).

Понимание и изменение поведений в прикладных условиях требует как анализа данных, так и качественных навыков. В этой книге основное внимание уделяется первому, в первую очередь по соображениям пространства. В дополнение к этому уже есть отличные книги, которые охватывают послед-

---

[1] См. https://www.oreilly.com/library/view/python-for-data/9781491957653/.

[2] См. https://www.oreilly.com/library/view/r-for-data/9781491910382/.

[3] См. https://oreil.ly/BehavioralDataAnalysisCh8.

[4] См. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/.

нее, такие как «Толчок в верном направлении: совершенствование решений о здоровье, богатстве и счастье» Ричарда Талера и Касса Санштейна (*Nudge: Improving Decisions About Health, Wealth, and Happiness*, Richard Thaler and Cass Sunstein, Penguin) и «Дизайн для изменения поведения: применение психологии и поведенческой экономики» Стивена Венделя (*Designing for Behavior Change: Applying Psychology and Behavioral Economics*, Stephen Wendel, O'Reilly)[1]. Тем не менее я дам введение в концепции бихевиористики, чтобы вы могли применять инструменты из этой книги, даже если вы – новичок в данной области.

Наконец, если вы – абсолютный новичок в анализе данных на R или Python, то эта книга не для вас. Я рекомендую начать с нескольких отличных введений, таких как те, которые упомянуты в этом разделе.

Исходный код на R и Python

Почему именно R и Python? Почему бы не выбрать один язык из перечисленных? Дебаты по теме «R против Python» все еще оживленны и продолжаются в интернете. Этот вопрос, по моему скромному мнению, в сущности, тоже не имеет значения. Реальность такова, что вам придется применять любой язык, который используется в вашей организации, и точка. Однажды я работал в медицинской компании, где по техническим и нормативным причинам доминирующим языком был SAS. Я регулярно использовал R и Python для своих собственных аналитических расчетов, но так как я не мог избежать работы с унаследованным исходным кодом SAS, в течение первого месяца работы я заставил себя усвоить SAS настолько, насколько мне было нужно. Если вы не проведете всю свою карьеру в компании, в которой не используется R или Python, то вы, скорее всего, в любом случае подхватите некоторые основы и того, и другого, так что с таким же успехом вы могли бы постичь двуязычие. Я еще не встречал никого, кто заявил бы, что «обучение чтению исходного кода на [другом языке] было пустой тратой моего времени».

Если исходить из допущения, что вам повезло работать в организации, в которой используется и то, и другое, с каким языком вам следует работать? Я думаю, что это на самом деле зависит от вашего контекста и задач, которые вам приходится выполнять. Например, я лично предпочитаю выполнять разведывательный анализ данных (EDA) на R, но нахожу, что Python намного проще использовать для создания веб-страниц. Советую выбирать, исходя из специфики вашей работы и опираясь на актуальную информацию: оба языка постоянно совершенствуются, и то, что было верно для предыдущей версии R или Python, может оказаться неверным для текущей версии. Например, Python становится гораздо более дружественной средой для EDA, чем когда-либо. Лучше потратить свою энергию на изучение обоих языков, чем на изучение форумов, посвященных выбору лучшего из двух.

---

[1] См. https://www.oreilly.com/library/view/designing-for-behavior/9781492056027/.

Среды исходного кода

В начале каждой главы я буду называть пакеты R и Python, которые необходимо загружать специально для каждой отдельной главы. В дополнение к этому я также буду использовать несколько стандартных пакетов по всей книге; во избежание повторов они называются только здесь (они уже включены во все скрипты в репозитории на GitHub). Вы всегда должны начинать свой исходный код с них, а также с нескольких параметрических настроек:

```
## R
library(tidyverse)
library(boot)       #Требуется для бутстрап-симуляций
library(rstudioapi) #Для загрузки данных из локальной папки
library(ggpubr)     #Для генерирования мультиграфиков

# Задание начального значения случайного числа
# будет обеспечивать воспроизводимость случайных чисел
set.seed(1234)
# Я лично нахожу используемую по умолчанию научную числовую нотацию
# (т. е. с экспонентами) менее удобной для чтения в распечатках, поэтому я ее отменяю
options(scipen=10)

## Python
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols
import matplotlib.pyplot as plt # Для графики
import seaborn as sns           # Для графики
```

Условные обозначения в исходном коде

Я использую R в RStudio. R 4.0 был запущен, когда я писал эту книгу, и я принял эту версию за основу, чтобы сделать книгу как можно более актуальной.

Исходный код R пишется шрифтом, специально предназначенным для исходного кода, с комментарием, указывающим используемый язык, вот так:

```
## R
> x <- 3
> x
[1] 3
```

Я использую Python в среде интерактивной разработки Spyder дистрибутива Anaconda. Обсуждение темы «Python 2.0 против 3.0», надеюсь, уже позади (по меньшей мере, в отношении нового исходного кода; унаследованный исходный код – это уже другая история), и я буду использовать Python 3.7. Условные обозначения, принятые для исходного кода Python, несколько похожи на условные обозначения для R:

```
## Python
In [1]: x = 3
In [2]: x
Out[2]: 3
```

Мы часто будем смотреть на результаты регрессий. Они бывают довольно многословными, с большим объемом диагностики, которая не имеет отношения к аргументам этой книги. Вы не должны пренебрегать ими в реальной жизни, но данный вопрос лучше освещен в других книгах. Поэтому я буду сокращать результат следующим образом:

```
## R
> model1 <- lm(icecream_sales ~ temps, data=stand_dat)
> summary(model1)

...
Coefficients:

             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4519.055    454.566  -9.941   <2e-16 ***
temps        1145.320      7.826 146.348   <2e-16 ***
...
```

```
## Python
model1 = ols("icecream_sales ~ temps", data=stand_data_df)
print(model1.fit().summary())

...
                 coef   std err         t     P>|t|      [0.025         0.975]
------------------------------------------------------------------------------
Intercept  -4519.0554   454.566    -9.941     0.000    -5410.439     -3627.672
Temps       1145.3197     7.826   146.348     0.000     1129.973      1160.666
...
```

Программирование в функциональном стиле

Один из шагов перехода от начинающего программиста к программисту среднего уровня состоит в том, чтобы перестать писать скрипты, в которых ваш исходный код представляет собой просто длинную последовательность инструкций, и вместо этого структурировать свой исходный код в функции. В этой книге мы будем писать и многократно использовать функции в разных главах, наподобие приведенных ниже, для строительства бутстраповских[1] интервалов уверенности[2]:

---

[1] Термин «бутстрап» (bootstrap) дословно означает «вытягивание себя за шнурки ботинок». Неплохой аналогией является история барона Мюнхгаузена, который вытянул себя вместе с лошадью из болота за волосы. – *Прим. перев.*

[2] Указанный термин (confidence interval), обозначающий вычисляемый из наблюдаемых данных диапазон, ограниченный нижним и верхним пределами, переведен в книге именно как интервал уверенности, поскольку речь идет об уверенности (confidence) исследователя в своих данных, а не о доверии к ним (trust), а это, как говорят в Одессе, две большие разницы. – *Прим. перев.*

```
## R
boot_CI_fun <- function(dat, metric_fun, B=20, conf.level=0.9){

  boot_vec <- sapply(1:B, function(x){
    cat("итерация бутстрапа ", x, "\n")
    metric_fun(slice_sample(dat, n = nrow(dat), replace = TRUE))})
  boot_vec <- sort(boot_vec, decreasing = FALSE)
  offset = round(B * (1 - conf.level) / 2)
  CI <- c(boot_vec[offset], boot_vec[B+1-offset])
  return(CI)
}
```

```
## Python
def boot_CI_fun(dat_df, metric_fun, B = 20, conf_level = 9/10):

  coeff_boot = []
  # Вычислить коэффициент, представляющий интерес для симуляции
  for b in range(B):
      print("Номер итерации " + str(b) + "\n")
      boot_df = dat_df.groupby("rep_ID").sample(n=1200, replace=True)
      coeff = metric_fun(boot_df)
      coeff_boot.append(coeff)

  # Извлечь интервал уверенности
  coeff_boot.sort()
  offset = round(B * (1 - conf_level) / 2)
  CI = [coeff_boot[offset], coeff_boot[-(offset+1)]]

  return CI
```

Функции также имеют добавочное преимущество в лимитировании остатков непонимания: даже если вы не понимаете, как работают приведенные выше функции, вы все равно можете считать само собой разумеющимся, что они возвращают интервалы уверенности, и следовать остальным рассуждениям, откладывая более глубокое погружение в их исходный код на потом.

Использование примеров исходного кода

Дополнительные материалы (примеры исходного кода и т. д.) доступны для скачивания по адресу https://oreil.ly/BehavioralDataAnalysis.

Адаптированный вариант примеров в виде электронного архива вы можете скачать со страницы книги на веб-сайте https://dmkpress.com/.

Навигация по книге

Стержневая интуитивная мысль книги состоит в том, что эффективный анализ данных основывается на постоянном взаимодействии между тремя компонентами:

- фактическими поведениями в реальном мире и связанными с ними психологическими явлениями, такими как намерения, мысли и эмоции;
- причинно-следственной аналитикой и в особенности причинно-следственными диаграммами;
- данными.

Книга разделена на пять частей:

часть I «Понимание поведений».
Эта часть закладывает основу для причинно-поведенческого каркаса и взаимосвязей между поведениями, причинно-следственным рассуждением и данными;

часть II «Причинно-следственные диаграммы и распутывание».
В этой части вводится понятие спутывания и объясняется, каким образом причинно-следственные диаграммы позволяют нам распутывать наши аналитические расчеты на данных;

часть III «Устойчивый анализ данных».
Здесь мы занимаемся разведкой инструментов для работы с пропущенными данными и знакомим с бутстраповскими симуляциями, поскольку в остальной части книги мы будем широко опираться на бутстраповские интервалы уверенности.
Данные, которые малы по объему, неполные или имеют неправильную форму (например, с несколькими пиками или выбросами), не являются новой проблемой, но она бывает особенно острой с поведенческими данными;

часть IV «Дизайн и анализ экспериментов».
В этой части мы обсудим вопросы дизайна и анализа экспериментов;

часть V «Расширенные инструменты анализа поведенческих данных».
Наконец, мы сводим все вместе, чтобы разведать модерацию, опосредование и инструментальные переменные.

Различные части книги в некоторой степени основаны друг на друге, и поэтому я рекомендую читать их по порядку, по меньшей мере при вашем первом подходе к книге.

Условные обозначения в книге

В книге используются следующие типографические условные обозначения.

Курсивный шрифт
Обозначает новые термины, URL-адреса, адреса электронной почты, имена файлов и расширения файлов.

`Моноширинный шрифт`
Используется для листингов программ, а также внутри абзацев для ссылки на элементы программ, такие как переменные или имена функций, базы данных, типы данных, переменные среды, инструкции и ключевые слова.

`Жирный моноширинный шрифт`
Показывает команды либо другой текст, который должен быть набран пользователем.

`Моноширинный шрифт курсивом`
Показывает текст, который должен быть заменен значениями, передаваемыми пользователем, либо значениями, определяемыми по контексту.


Этот элемент обозначает общее замечание.


Данный элемент обозначает предупреждение или предостережение.

Благодарности

Авторы часто благодарят своих супругов за терпение и называют особенно проницательных рецензентов. Мне посчастливилось иметь и то, и другое в одном человеке. Я не думаю, что кто-либо другой осмелился бы или сумел бы так много раз отправлять меня обратно за «чертежную доску», и по этой причине данная книга стала намного лучше. Поэтому моя первая благодарность – моему партнеру по жизни и менталитету.

Несколько моих коллег и соратников – ученых-бихевиористов – были достаточно великодушны, чтобы посвятить свое время чтению и комментариям к более раннему черновику. Данная книга стала от этого только лучше. Спасибо (в обратном алфавитном порядке) Джин Утке, Джессике Якубоуски, Чинмайе Гупте и Федре Дайфе!

Особая благодарность Бетани Винкель за ее помощь в написании.

Теперь я съеживаюсь при воспоминании о том, насколько грубыми и запутанными были самые первые наброски. Мой редактор по разработке и технические рецензенты терпеливо подталкивали меня к тому, где эта книга сейчас находится, делясь своим богатым опытом и знаниями. Спасибо вам, Гэри О'Брайен, и спасибо вам, Сюань Инь, Шеннон Уайт, Джейсон Стэнли, Мэтт Лемей и Андреас Кальтенбруннер.

Об авторе

Флоран Бюиссон – поведенческий экономист с 10-летним опытом работы в бизнесе, аналитике и бихевиористике. Еще недавно он основал и в течение четырех лет возглавлял научную группу по бихевиористике в страховой компании Allstate.

Ранее он работал во французской консалтинговой фирме по стратегиям, где использовал экономику и анализ данных для ответа на сложные вопросы эконометрии, например для построения индекса, измеряющего стабильность сельскохозяйственной политики в развивающихся странах от имени Продовольственной и сельскохозяйственной организации ООН. Он также работал в области специализированной медицинской аналитики, анализируя поведение пациентов с тяжелыми заболеваниями.

Флоран публикует научные статьи в таких журналах, как рецензируемый журнал Journal of Real Estate Research, посвященный исследованиям в сфере недвижимости. Он имеет степень магистра эконометрии, а также степень доктора философии в области поведенческой экономики в Университете Сорбонны в Париже.

Об иллюстрации на обложке (колофон)

На обложке книги «Анализ поведенческих данных на R и Python» изображена южноамериканская гремучая змея (Crotalus durissus). Этот вид очень ядовитой гадюки обитает в районах по всей Южной Америке, за исключением высоких Анд и крайнего юга. Его также можно найти на нескольких Карибских островах.

Эти гремучие змеи изменчивы по внешнему виду, как правило, с бледным подбрюшьем и более темно-коричневыми ромбовидными формами или полосами на спине, выделяющимися на более бледном фоне. Они питаются как грызунами, так и ящерицами. Взрослые особи могут вырастать до 6 футов в длину, а в неволе жить до 20 лет. Они размножаются сезонно, и самки рожают до 14 живых детенышей одновременно.

По оценкам, в Северной и Южной Америке от укуса этой змеи умирают около 400 человек в год, а укус южноамериканской гремучей змеи, как известно, особенно смертоносен. Ее яд содержит четыре основных токсина: кротоксин, конвульсин, гироксин и кротамин, – которые змея использует для захвата и переваривания своей добычи.

Гремучие змеи часто используют свой загадочный камуфляж в качестве первой защиты и остаются неподвижными при приближении более крупного животного; в результате этой стратегии иногда случаются укусы людей, потому что они подходят слишком близко к змее или даже наступают на нее. Еще одна защита является источником их общепринятого названия: уникальная предупреждающая функция «погремушек» на их хвостах. Они состоят из кератиновых чешуек с несколькими рыхлыми слоями, и когда змея использует набор уникальных мышц хвоста для вибрации своего хвоста, сухие слои ударяются друг о друга и издают характерный звук. Каждый раз, когда змея сбрасывает кожу, добавляется набор погремушек, что делает число сегментов одним из потенциальных индикаторов (наряду с размером и длиной) возраста змеи.

Южноамериканская гремучая змея занесена Международным союзом по охране природы IUCN в список животных, вызывающих наименьшее беспокойство. Многие животные на обложках издательства O'Reilly находятся под угрозой исчезновения, и все они важны для мира.

Цветная иллюстрация на обложке выполнена Карен Монтгомери на основе черно-белой гравюры из Малой энциклопедии Мейерса.

Часть I

ПОНИМАНИЕ ПОВЕДЕНИЙ

В этой первой части книги дается объяснение причины, почему анализ поведенческих данных требует нового подхода.

В главе 1 будет описан этот новый подход – причинно-поведенческий каркас анализа данных. Мы рассмотрим конкретный пример, показывающий, как даже самые простые аналитические расчеты на данных бывают сорваны присутствием спутывающего фактора. Решение этой проблемы в лучшем случае осложнено, а в худшем – невозможно при использовании традиционных подходов, но новый каркас обеспечивает простой процесс.

В главе 2 будет продолжено изучение особенностей поведенческих данных, обеспечивая при этом осторожное введение в бихевиористику и процесс обеспечения того, чтобы наши данные адекватно отражали соответствующие реально существующие поведения.

Глава 1

Причинно-поведенческий каркас для анализа данных

Как мы обсуждали в предисловии, понимание того, что именно движет поведениями, для того чтобы их изменять, является одной из ключевых целей прикладной аналитики, будь то в коммерческой, некоммерческой или общественной организации. Мы хотим докопаться, почему кто-то купил ту или иную вещь и почему кто-то другой ее не купил. Мы хотим понять, почему кто-то продлил подписку, связался с кол-центром, вместо того чтобы оплатить онлайн, зарегистрировался в качестве донора органа или пожертвовал его некоммерческой организации. Обладание этими знаниями позволяет нам предсказывать, что конкретно люди будут делать в разных сценариях, и помогает нам определять, что именно наша организация может сделать, чтобы побуждать (или не побуждать) их делать это снова. Я считаю, что указанная цель лучше всего достигается путем комбинирования анализа данных с бихевиористским менталитетом и инструментарием причинно-следственной аналитики для создания интегрированного подхода, который я назвал «причинно-поведенческим каркасом». В этом каркасе *поведения* находятся на вершине, потому что их понимание является нашей конечной целью. Это понимание достигается посредством *причинно-следственных диаграмм* и *данных*, которые образуют два опорных столпа треугольника (рис. 1.1).

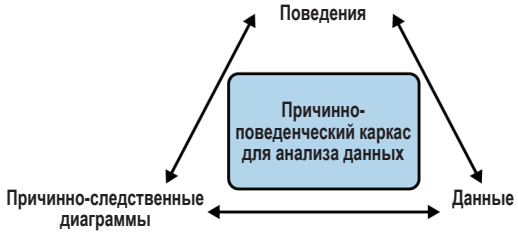

Рис. 1.1 ❖ Причинно-поведенческий каркас для анализа данных

По ходу изложения в этой книге мы разведаем каждую часть треугольника и посмотрим, как они соединяются друг с другом. В заключительной главе мы увидим, как вся наша работа сходится вместе, достигая с помощью одной строки кода того, что при традиционных подходах было бы неимоверно сложной задачей: измерения степени, в которой удовлетворенность клиентов увеличивает будущие расходования клиентов. В дополнение к выполнению таких экстраординарных задач этот новый каркас также позволит вам эффективнее проводить распространенные аналитические расчеты, такие как определение эффекта проводимой по электронной почте рекламной кампании или свойства продукта на поведение клиентов при осуществлении покупок.

Прежде чем перейти к этой теме, читатели, знакомые с предсказательной аналитикой, возможно, зададутся вопросом, почему я вместо нее выступаю за причинно-следственную аналитику. Ответ кроется в том, что, несмотря на то что предсказательная аналитика была (и останется) очень успешной в рамках бизнеса, она бывает недостаточной, когда ваши аналитические расчеты касаются поведений людей. В частности, применение причинно-следственного подхода помогает нам выявлять и устранять «спутывание», очень распространенную проблему с поведенческими данными. Я подробно остановлюсь на этих моментах в остальной части первой главы.

Почему для объяснения человеческого поведения нужна причинно-следственная аналитика

Понимание того, как причинно-следственная аналитика вписывается в аналитический ландшафт, поможет нам лучше уяснить, почему она необходима в рамках бизнеса. Как мы увидим, эта потребность проистекает из сложности человеческого поведения.

Различные типы аналитики

Существует три разных типа аналитики: описательная, предсказательная и причинно-следственная. Описательная аналитика обеспечивает *описание* данных. Проще говоря, я думаю о ней как об аналитике «каким является то или это» или «что именно мы измерили». Под этот зонтик подпадает деловая отчетность. Сколько клиентов отменили свои подписки в прошлом месяце? Сколько прибыли мы получили в прошлом году? Всякий раз, когда мы вычисляем среднее значение или другие простые метрики, мы неявно используем описательную аналитику. Описательная аналитика – это самая простая форма аналитики, но ее также недооценивают. Многие организации на самом деле изо всех сил пытаются получать четкое и единое представление о своей деятельности. Для того чтобы увидеть масштабы этой проблемы

в организации, просто задайте один и тот же вопрос финансовому отделу и операционному отделу и измерьте степень, с которой ответы будут различаться[1].

Предсказательная аналитика обеспечивает *предсказание*. Я думаю о ней как об аналитике «что будет, если сохранятся текущие условия» или «что именно мы еще не измерили». Большинство методов машинного обучения (например, нейронные сети и модели градиентного бустинга) относятся к этому типу аналитики и помогают нам отвечать на такие вопросы, как «Сколько клиентов отменят подписку в следующем месяце?» и «Является ли этот заказ мошенническим?». За последние несколько десятилетий предсказательная аналитика изменила мир; легионы занятых в бизнесе исследователей данных являются свидетельством ее успеха.

Наконец, причинно-следственная аналитика предоставляет *причины* данных. Я думаю о ней как об аналитике «что будет, если?» или «что будет при других условиях». Она отвечает на такие вопросы, как «Сколько клиентов отменят свою подписку в следующем месяце, если мы не отправим им уведомление?». Наиболее известным инструментом причинно-следственной аналитики является A/B-тест, т. н. рандомизированный эксперимент, или рандомизированное контролируемое испытание (randomized controlled trial, аббр. RCT). Это связано с тем, что самый простой и эффективный способ ответить на предыдущий вопрос состоит в том, чтобы отправить купон заранее отобранной группе клиентов и посмотреть, сколько из них отменят подписку по сравнению с контрольной группой.

Мы охватим эксперименты в части IV книги, но перед этим, в части II, мы рассмотрим еще один инструмент из этого инструментария, а именно причинно-следственные диаграммы, которые можно использовать даже тогда, когда мы не можем экспериментировать. И действительно, одна из моих целей состоит в том чтобы побудить вас думать о причинно-следственной аналитике шире, а не просто приравнивать ее к экспериментированию.

Хотя эти ярлыки, возможно, создают впечатление четкой категоризации, на самом деле между этими тремя категориями существует больший градиент, и вопросы и методы между ними размыты. Вы также можете столкнуться с другими терминами, такими как *предписывающая аналитика*, которые еще больше размывают границы и добавляют другие нюансы, не меняя общую картину кардинально.

Люди – сложные существа

Если предсказательная аналитика была настолько успешной, а причинно-следственная аналитика использует те же инструменты анализа данных, что и регрессия, почему бы не придерживаться предсказательной анали-

---

[1] Справедливости ради во многих обстоятельствах они просто *должны* быть разными, потому что данные используются для разных целей и подчиняются разным правилам. Но даже вопросы, на которые вы ожидали бы получить единственно верный ответ (например, «Сколько у нас сейчас сотрудников?»), как правило, обнаруживают расхождения.

тики? Если коротко, потому что люди сложнее, чем ветряные турбины. Поведение человека:

имеет несколько причин.
Поведение турбины не зависит от ее личности, социальных норм сообщества турбин или обстоятельств ее воспитания, в то время как предсказательная сила любой отдельной переменной на поведение человека почти всегда разочаровывает из-за этих факторов;

зависит от контекста.
Незначительные или косметические изменения в окружающей среде, такие как изменение принятого по умолчанию варианта выбора, могут оказывать большое влияние на поведение. Эта ситуация является благословением с точки зрения поведенческого *дизайна*, поскольку она позволяет нам подстегивать изменения в поведениях, но является проклятием с точки зрения поведенческой *аналитики*, потому что это означает, что каждая ситуация уникальна настолько, что становится трудной для предсказания;

является переменным (ученые сказали бы, что поведение недетерминированно).
Один и тот же человек может вести себя совершенно по-другому, когда его неоднократно помещают в одну и ту же ситуацию, которая внешне выглядит совершенно одинаковой, даже после учета косметических факторов. Это может быть связано с преходящими эффектами, такими как настроение, либо долговременными эффектами, такими как скука от приема одного и того же ежедневного завтрака. Оба этих фактора могут радикально менять поведение, но их трудно улавливать;

является инновационным.
Когда условия в окружающей среде меняются, человек может переключаться на поведение, которого он буквально никогда раньше не демонстрировал, и это происходит даже при самых обыденных обстоятельствах. Например, впереди на вашем обычном пути следования происходит автомобильная авария, и поэтому вы в последнюю минуту решаете повернуть направо;

является стратегическим.
Люди делают выводы и реагируют на поведения и намерения других людей. В некоторых случаях это может означать «восстановление» сотрудничества, которое было нарушено внешними обстоятельствами, что делает его устойчиво предсказуемым. Но в других случаях это может предусматривать добровольное запутывание своего поведения, чтобы сделать его непредсказуемым во время соревновательной игры, такой как шахматы (или мошенничество!).

Все эти аспекты человеческого поведения делают его менее предсказуемым, чем поведение физических объектов. В целях отыскания регулярностей, более надежных для анализа, мы должны спуститься на один уровень глубже, чтобы понять и измерить причины поведения. Тот факт, что кто-то

съел овсянку на завтрак и выбрал определенный маршрут в понедельник, не означает, что он сделает то же самое во вторник, но вы можете быть более уверены в том, что он хоть как-то позавтракает и отправится по какому-то маршруту на работу.

Экстраполяция в аналитике, проклятие размерности и критика Лукаса

Читатели с квантитативным опытом, возможно, будут не совсем удовлетворены моим высказыванием о том, что «поведение человека трудно предсказывать, потому что оно сложное», поэтому вот математическая версия этого аргумента. Я начну с описания разницы между интерполяцией и экстраполяцией. На рис. 1.2 показано немного симулированных данных с линейной взаимосвязью между двумя переменными.

Линия на рисунке – это регрессионная линия наилучшей подгонки, т. е. линия, соответствующая линейной регрессии между двумя переменными, с наклоном, приближенно равным 3. Мы можем использовать ее для предсказания неизвестных значений Y на основе известного значения X (и наоборот). Например, имея значение X = 50, мы бы предсказали, что Y равно 150. Слева от этого значения есть наблюдаемые точки, то есть точки, для которых X < 50, а также точки справа от этого значения, для которых X > 50. Этот процесс предсказывания называется интерполяцией, потому что наша точка находится между наблюдаемыми точками (приставка «интер» означает «между»; например, international = «межнациональный»). И наоборот, если бы мы использовали линию регрессии с X = 0, чтобы предсказать, что Y = 0, это было бы названо экстраполяцией, поскольку точка, которую мы пытаемся предсказать, находится за пределами облака наблюдаемых точек («экстра» означает «снаружи»; например, экстраординарное = «вне обычного»). В статистике и в повседневной жизни экстраполировать – значит покидать область наблюдаемого и известного, чтобы делать предсказания. В то время как интерполяция обычно безопасна и надежна, экстраполяция всегда несколько умозрительна: требуется «рывок веры», чтобы допустить, что правила, применяемые внутри неких границ, по-прежнему будут соблюдаться за их пределами.

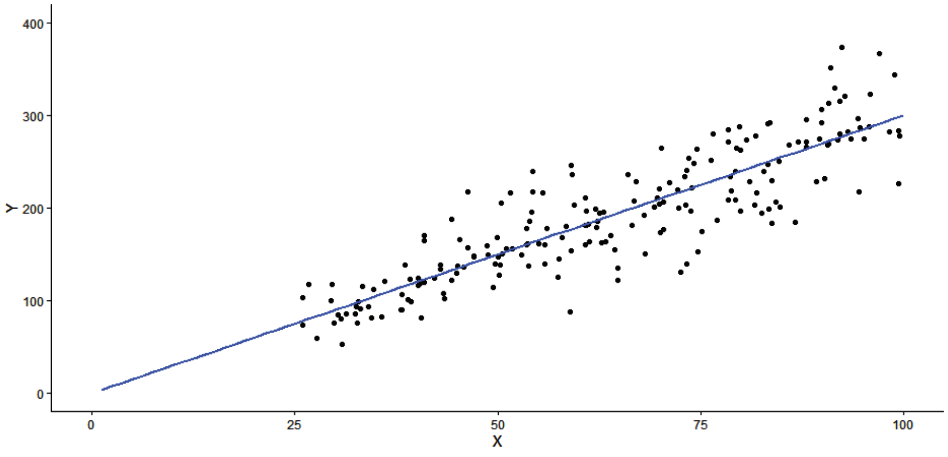

Рис. 1.2 ❖ Линейная связь между двумя переменными, с линией регрессии

Физические объекты, такие как ветряная турбина, находятся под воздействием достаточно малого и постоянного числа факторов (это не похоже на то, как некоторые законы физики берут выходные или новые дни и делают их похожими на случайные). Следовательно, у нас есть много точек данных относительно размерностей пространства задачи, а значит, мы почти всегда интерполируем. Для простоты модель может пренебрегать вторичными или редкими явлениями, такими как шторм 1 раз в 100 лет, но даже когда подобные выбросы происходят, результат остается в какой-то мере предсказуемым: бритвенное лезвие сломается и упадет в воду, а не улетит.

С другой стороны, поведение человека находится под воздействием большого числа разных факторов, которые могут быть или не быть релевантными в данный день и могут расти или затухать с течением времени. Следовательно, у нас обычно оказывается мало точек данных относительно размеров пространства задачи, а значит, мы гораздо чаще экстраполируем – наталкиваясь на проблему, известную в статистике под названием «проклятия размерности». В дополнение к этому незначительные изменения в окружающей среде могут приводить к серьезным изменениям в поведении, что делает попытку предсказывать будущее поведение человека, основываясь только на прошлом поведении, азартной игрой с плохими шансами на успех.

Для людей, интересующихся генеалогией поведенческой экономики, макроэкономист Роберт Лукас сформулировал этот аргумент в 1970-х годах («критика Лукасом» кейнсианских моделей). Вместо этого он рекомендовал выявлять более глубокие параметры человеческих поведений, такие как потребительские предпочтения, – еще одна версия аргумента, который я приводил ранее.

Чтоб ей пусто было! Скрытые опасности, когда разбирательства отданы на усмотрение регрессии

В предыдущем разделе я упомянул, что причинно-следственная аналитика часто использует те же инструменты, что и предсказательная аналитика. Однако, поскольку у них разные цели, инструменты используются по-разному. Так как регрессия является одним из главных инструментов для обоих типов аналитики, она бывает отличным средством иллюстрирования разницы между предсказательной и причинно-следственной аналитикой. Регрессия, подходящая для предсказательной аналитики, зачастую приводила бы к ужасной регрессии для целей причинно-следственной аналитики, и наоборот.

Регрессия в предсказательной аналитике используется для оценивания неизвестного значения (часто, но не всегда, в будущем). Она делает это, беря известную информацию и используя различные факторы для триангулирования наилучшей догадки в отношении данной переменной. Важным является предсказанное значение и его точность, а не то, почему или как оно было предсказано.

В причинно-следственной аналитике регрессия тоже используется, но фокус внимания лежит не на оценивании значения целевой переменной. Вместо этого основное внимание уделяется причине этого значения. В терминах

регрессии нас больше интересует не сама зависимая переменная, а ее связь с данной независимой переменной. При правильно структурированной регрессии коэффициент корреляции может быть переносимой мерой причинно-следственного эффекта независимой переменной на зависимую переменную.

Но что значит иметь правильно структурированную регрессию для этой цели? Почему мы не можем просто взять регрессии, которые мы уже используем для предсказательной аналитики, и рассматривать предоставленные коэффициенты как меры причинно-следственной связи? Мы не можем этого сделать, потому что каждая переменная в регрессии может модифицировать коэффициенты других переменных. Следовательно, наша смесь переменных должна быть искусно изготовлена не для создания наиболее точного предсказания, а для создания наиболее точных коэффициентов. Два набора переменных, как правило, различаются, потому что переменная может быть сильно коррелирована с нашей целевой переменной (и, следовательно, быть очень предсказуемой), фактически не влияя на эту переменную.

В этом разделе мы проведем разведку вопросов, почему эта разница имеет важность в перспективе и почему отбор переменных – это более чем половина битвы в поведенческой аналитике. Мы сделаем это на конкретном примере C-Mart, вымышленной сети супермаркетов с магазинами по всей территории Соединенных Штатов. Первая из двух вымышленных компаний, используемых на протяжении всей книги, C-Mart, поможет нам понять возможности и трудности анализа данных для традиционных компаний в цифровую эпоху.

Данные

Папка этой главы в репозитории на GitHub[1] содержит два CSV-файла, *chap1-stand_data.csv* и *chap1-survey_data.csv*, с наборами данных для двух примеров этой главы.

В табл. 1.1 показана информация, содержащаяся в CSV-файле *chap1-stand_data.csv*, о продажах мороженого и холодного кофе на ежедневном уровне в киосках C-Mart.

Таблица 1.1. Информация о продажах в файле chap1-stand_data.csv

Имя переменной	Описание переменной
IceCreamSales (ПродажиМороженого)	Ежедневные продажи мороженого в киосках C-Mart
IcedCoffeeSales (ПродажиХолодногоКофе)	Ежедневные продажи холодного кофе в киосках C-Mart
SummerMonth (ЛетнийМесяц)	Двоичная переменная, которая относит день к летним месяцам
Temp (Температура)	Средняя температура за этот день и у этого киоска

В табл. 1.2 показана информация, содержащаяся в CSV-файле *chap1-survey_data.csv*, полученная в результате опроса прохожих за пределами киосков C-Mart.

[1] См. https://oreil.ly/BehavioralDataAnalysisCh1.

Таблица 1.2. Информация об опросе в файле chap1-survey_data.csv

Имя переменной	Описание переменной
VanillaTaste (ПредпочтениеВанильногоВкуса)	Пристрастие опрашиваемого к ванильному вкусу, 0–25
ChocTaste (ПредпочтениеШоколадногоВкуса)	Пристрастие опрашиваемого к шоколадному вкусу, 0–25
Shopped (Покупал)	Двоичная переменная, которая указывает на то, что опрашиваемый когда-либо совершал покупки в местном киоске C-Mart

Почему корреляция не есть каузация: спутывающий фактор в действии

В каждом магазине C-Mart есть киоск с мороженым. Компания считает, что на ежедневные продажи мороженого влияет погода – или, выражаясь на жаргоне причинно-следственных связей, что погода является причиной продаж. Другими словами, при прочих равных условиях мы исходим из допущения, что люди с большей вероятностью будут покупать мороженое в более жаркие дни, что имеет интуитивно понятный смысл. Это мнение подтверждается сильной корреляцией исторических данных между температурой и продажами, как показано на рис. 1.3 (соответствующие данные и исходный код находятся в репозитории книги на GitHub).

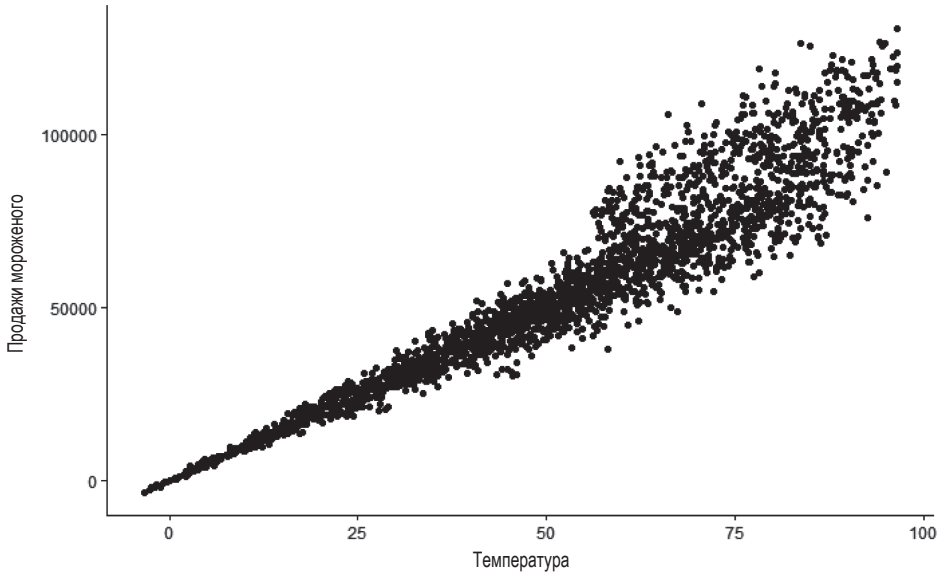

Рис. 1.3 ❖ Продажи мороженого как функция от наблюдаемой температуры

Как указано в предисловии, мы будем использовать регрессию в качестве нашего главного инструмента анализа данных. Выполнение линейной регрессии продаж мороженого на температуре занимает одну строку исходного кода:

```
## Python (результат не показан)
print(ols("icecream_sales ~ temps", data=stand_data_df).fit().summary())

## R
> summary(lm(icecream_sales ~ temps, data=stand_dat))
...
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4519.055    454.566  -9.941   <2e-16 ***
Temps        1145.320      7.826 146.348   <2e-16 ***
...
```

Для наших целей в этой книге наиболее важной частью результата на выходе является раздел *коэффициентов*, в котором говорится, что оценочное пересечение (коэффициент сдвига) – теоретическое среднее значение продаж мороженого при температуре ноль градусов – составляет –4519, что, очевидно, является бессмысленной экстраполяцией. Он также говорит нам о том, что оценочный коэффициент для температуры составляет 1145, а значит, каждый дополнительный градус температуры, как ожидается, будет увеличивать продажи мороженого на 1145 долларов.

Теперь давайте вообразим, что мы находимся в конце особенно теплой недели октября и, основываясь на предсказаниях модели, компания заранее увеличила запасы в киосках с мороженым. Тем не менее еженедельные продажи, хотя и были выше, чем обычно на этой неделе октября, сильно не дотянули до объема, предсказанного моделью. Та-ак! И что же случилось? Следует ли уволить аналитика данных?

Произошло то, что модель не учитывает важнейший факт: бóльшая часть продаж мороженого приходится на летние месяцы, когда дети не ходят в школу. Регрессионная модель сделала свое лучшее предсказание на основе имеющихся данных, но часть причины увеличения продаж мороженого (летние каникулы у учеников) была ошибочно отнесена к температуре, так как летние месяцы положительно коррелируют с температурой. Поскольку повышение температуры в октябре не привело к внезапным летним каникулам (вы уж извините, детки!), мы увидели более низкие продажи, чем в летние дни при такой температуре.

С технической точки зрения, месяц года – это спутывающий фактор в нашей взаимосвязи между температурой и продажами. *Спутывающий фактор*[1] – это переменная, которая вносит систематическое смещение в регрессию; когда в ситуации, которую вы анализируете, присутствует спутывание, это означает, что интерпретация коэффициента регрессии как причинно-следственного будет приводить к ненадлежащим выводам.

---

[1] Для справки вот выжимка определения указанного термина в переводе с нескольких языков. Спутывающий фактор (confounder), или повреждающий, помеховый, мешающий, смешивающий фактор, – это переменная, которая влияет как на зависимую переменную, так и на независимую переменную, вызывая ложную ассоциацию. Являясь причинно-следственным понятием, указанная переменная имеет связь как с интересующей причиной, так и с интересующим следствием. – *Прим. перев.*

Давайте подумаем о таком месте, как Чикаго, где присутствует континентальный климат: зима – очень холодная, а лето – очень жаркое. Сравнивая продажи в случайный жаркий день с продажами в случайный холодный день без учета соответствующего месяца года, вы, скорее всего, будете сравнивать продажи в жаркий летний день, когда дети не ходят в школу, с продажами в холодный зимний день, когда дети учатся в школе; эта ситуация раздувает очевидную взаимосвязь между температурой и продажами.

В приведенном примере мы также можем ожидать неуклонного занижения предсказания продаж в более холодную погоду. По правде говоря, в летние месяцы происходит сдвиг парадигмы, и когда этим сдвигом приходится управлять исключительно с помощью температуры в линейной регрессии, коэффициент регрессии для температуры неизменно будет слишком высоким для более теплых температур и слишком низким для более холодных температур.

Слишком много переменных может испортить всю обедню

Потенциальным решением проблемы спутывающих факторов было бы добавление в регрессию всех переменных, которые можно добавить. Менталитет в стиле «все, что есть, и кухонная раковина в придачу» все еще имеет сторонников среди статистиков. В книге «Книга вопросов почему» Джуди Перл и Дана Макензи (*The Book of Why*, Judea Pearl and Dana Mackenzy) упоминают, что «даже ведущий статистик недавно написал, что “избегание обусловленности на некоторых наблюдаемых ковариатах[1]… является ненаучной сиюминутной эквилибристикой”» (Pearl & Mackenzie 2018, стр. 160)[2]. Это также довольно распространено среди исследователей данных. Справедливости ради, если ваша цель состоит только в том, чтобы предсказывать переменную, и у вас есть модель, обстоятельно продуманная для обобщения за пределы ваших тестовых данных, и вас не волнует, почему предсказанная переменная принимает некоторое значение, то это совершенно правильная позиция. Но это не сработает, если ваша цель состоит в том, чтобы понять причинно-следственные связи, дабы действовать в соответствии с ними. По этой причине простое добавление как можно большего числа переменных в вашу модель не только является неэффективным, но и может стать совершенно контрпродуктивным и вводить в заблуждение.

Давайте продемонстрируем это на нашем примере, добавив переменную, которую мы могли бы включить, но которая будет систематически смещать нашу регрессию. Я создал переменную *ПродажиХолодногоКофе*, чтобы она коррелировала с *Температурой*, но не с *ЛетнимМесяцем*. Давайте посмотрим,

---

[1] Ковариат (covariate) – это объясняющая переменная, естественным образом существующая в исследуемой модели и могущая являться предсказательной. – *Прим. перев.*

[2] На всякий случай, если вам интересно, вышеупомянутого статистика зовут Дональдом Рубином (Donald Rubin).

что произойдет с нашей регрессией, если мы добавим эту переменную в дополнение к *Температуре* и *ЛетнимМесяцам* (двоичной переменной, обозначающей месяц июль или август как 1 и любой другой месяц как 0):

```
## R (результат не показан)
> summary(lm(icecream_sales ~ iced_coffee_sales + temps + summer_months))

## Python
print(ols("icecream_sales ~ temps + summer_months + iced_coffee_sales",
          data=stand_data_df).fit().summary())
...
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept         24.5560    308.872     0.080      0.937    -581.127     630.239
Temps          -1651.3728   1994.826    -0.828      0.408   -5563.136    2260.391
summer_months   1.976e+04    351.717    56.179      0.000    1.91e+04    2.04e+04
iced_coffee_sales 2.6500       1.995     1.328      0.184      -1.262       6.562
...
```

Мы видим, что коэффициент *Температуры* резко изменился по сравнению с нашим предыдущим примером до такой степени, что теперь он – отрицательный. Высокие *p*-значения для *Температуры* и *ПродажХолодногоКофе* обычно воспринимаются как признаки того, что что-то не так, но поскольку *p*-значение для *Температуры* «хуже», аналитик может прийти к выводу, что он должен удалить его из регрессии. Как это возможно?

Истина, лежащая в основе данных (которая по понятным причинам известна, так как я сфабриковал связи вручную и рандомизировал данные вокруг этих связей), заключается в том, что, когда становится жарко, люди с большей вероятностью покупают холодный кофе. В жаркие дни люди также с большей вероятностью покупают больше мороженого. Но покупка холодного кофе сама по себе не делает покупателей более или менее склонными покупать мороженое. Летние месяцы также не коррелируют с покупками холодного кофе, поскольку школьники не являются существенным фактором спроса на холодный кофе (подробную информацию о применяемой математике см. во врезке).

Техническое более глубокое погружение: что здесь произошло?

Уравнение для продаж мороженого, которое я использовал для генерирования симулированных данных, выглядит следующим образом:

$$\textit{ПродажиМороженого} := 1000 \cdot \textit{Температура} + 20\,000 \cdot \textit{ЛетнийМесяц} + \varepsilon_1,$$

где ε_1 представляет некий случайный шум со средним значением, равным нулю, а знак «:=» обозначает, что это уравнение представляет то, как переменная слева, *ПродажиМороженого*, определяется или строится.

Однако уравнение, которое мы оцениваем в нашей линейной регрессии, таково:

$$\textit{ПродажиМороженого} = \beta_T \cdot \textit{Температура} + \beta_S \cdot \textit{ЛетнийМесяц} + \beta_C \cdot \textit{ПродажиХолодногоКофе}.$$

Истинное уравнение, которое использовалось для генерирования продаж холодного кофе, таково:

$$ПродажиХолодногоКофе := 1000 \cdot Температура + \varepsilon_2.$$

Приведенное выше означает, что мы можем переписать предыдущее уравнение следующим образом:

$$ПродажиМороженого = \beta_T \cdot Температура + \beta_S \cdot ЛетнийМесяц + \beta_C \cdot (1000 \cdot Температура + \varepsilon_2) = (\beta_T + 1000\ \beta_C) \cdot Температура + \beta_S \cdot ЛетнийМесяц.$$

За исключением некого случайного везения, наш коэффициент β_S должен быть близок к истинному значению, равному 20 000. Но в случае температуры наша программа попытается решить уравнение, которое приведено ниже:

$$\beta_T + 1000 \cdot \beta_C = 1000.$$

Это одно уравнение с двумя неизвестными, поэтому оно имеет бесконечное число решений. Будут работать $\beta_T = 0$ и $\beta_C = 1$, но будут работать и $\beta_T = 500$ и $\beta_C = 0.5$ либо $\beta_T = 5000$ и $\beta_C = -4$. Алгоритм наименьших квадратов выберет комбинацию, которая обеспечивает наибольшее значение R^2, но она не будет надежной (хотя на практике ненадежность, как правило, будет намного меньше, чем в этом симулированном примере). В техническом плане мы ввели мультиколлинеарность.

На рис. 1.4 показана положительная корреляция между продажами холодного кофе и продажами мороженого, поскольку и то, и другое увеличивается, когда становится теплее, однако любое увеличение продаж холодного кофе в летние месяцы можно объяснить совместной корреляцией с переменной температуры. Когда регрессионный алгоритм пытается объяснить продажи мороженого с использованием трех имеющихся переменных, объяснительная сила температуры на продажах холодного кофе была добавлена в переменную температуры, тогда как холодный кофе был вынужден компенсировать придание избыточной силы температуре. Несмотря на то что продажи холодного кофе статистически не значимы и коэффициент этой переменной относительно невелик, величина продаж в долларах намного выше, чем величина в градусах температуры, поэтому в конечном счете продажи холодного кофе нивелируют инфляцию коэффициента температуры.

В предыдущем примере добавление переменной *ПродажиХолодногоКофе* в регрессию запутало взаимосвязь между температурой и продажами мороженого. К сожалению, верно и обратное: включение в регрессию неправильной переменной может создавать иллюзию взаимосвязи, когда ее нет.

Придерживаясь нашего примера с мороженым в C-Mart, предположим, что категорийный менеджер заинтересован в понимании вкусов покупателей, поэтому он просит сотрудника встать у входа в магазин и опрашивать проходящих мимо людей о том, насколько им нравится ванильное мороженое и насколько им нравится шоколадное мороженое (оба по шкале от 0 до 25), а также о том, покупали ли они когда-либо мороженое в киоске. В целях обеспечения простоты мы будем исходить из того, что в киоске продается только шоколадное и ванильное мороженое.

Будем считать для примера, что вкус ванильного мороженого и вкус шоколадного мороженого совершенно не связаны. Некоторым людям нравится

одно, но не другое, некоторым одинаково нравится и то, и другое, некоторым одно нравится больше, чем другое, и так далее. Но все эти предпочтения влияют на то, покупает человек в киоске или нет, т. е. на двоичную переменную (Да/Нет).

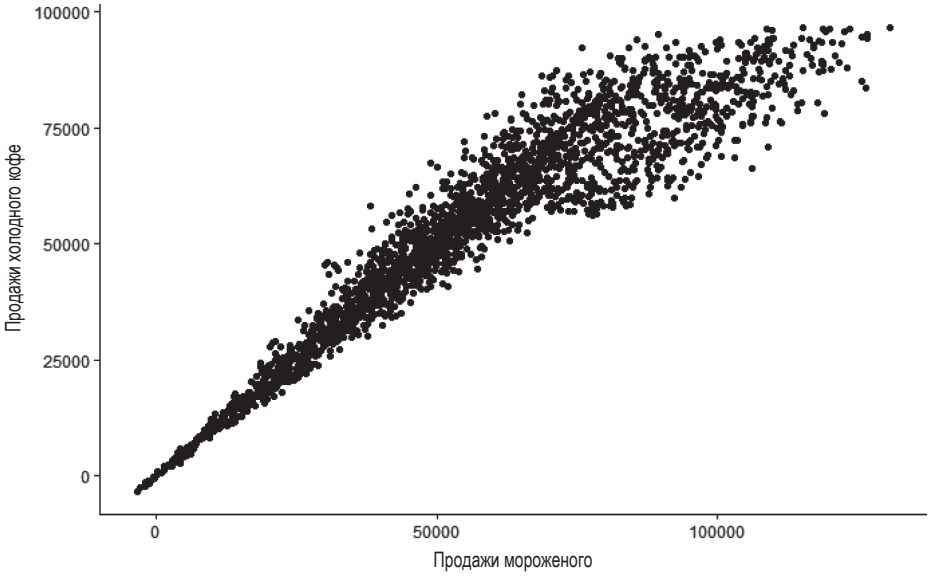

Рис. 1.4 ❖ График продаж холодного кофе в сопоставлении с продажами мороженого

Поскольку переменная *Покупал* является двоичной, мы бы использовали логистическую регрессию, если бы хотели измерить влияние любой из переменных *Вкуса* на поведение при осуществлении покупок. Поскольку две переменные *Вкуса* некоррелированы, мы бы увидели обычное облако без видимой корреляции, если бы сопоставили их друг с другом; однако каждая из них влияет на вероятность покупок в киоске с мороженым (рис. 1.5).

На первом графике я добавил линию наилучшей подгонки, которая является почти идеально горизонтальной, что отражает отсутствие корреляции между переменными (коэффициент корреляции равен 0.004, который отражает ошибку отбора). На втором и третьем графиках мы видим, что предпочтение ванильного вкуса и шоколадного вкуса в среднем выше у покупателей (*Покупал* = 1), чем у непокупателей, что имеет смысл.

Пока что все идет хорошо. Допустим, после того как вы получаете данные опроса, ваш деловой партнер сообщает вам о том, что он подумывает о введении поощрительного купона для киоска с мороженым: когда вы покупаете мороженое, вы получаете купон на случай будущих посещений. Этот фактор лояльности не повлияет на респондентов, которые никогда не делали покупки в киоске, поэтому релевантной популяцией являются те, кто делал покупки в магазине. Деловой партнер рассматривает возможность использования вкусовых ограничений в купонах для балансировки запасов, но не знает,

насколько можно повлиять на выбор того или иного вкуса покупателем. Если бы кто-то, купивший ванильное мороженое, получил купон на скидку 50 % на шоколадное мороженое, то сделает это что-то, кроме добавления большего количества бумаги в корзину для мусора? В любом случае, насколько благосклонно люди, которые любят ванильное мороженое, смотрят на шоколадное мороженое?

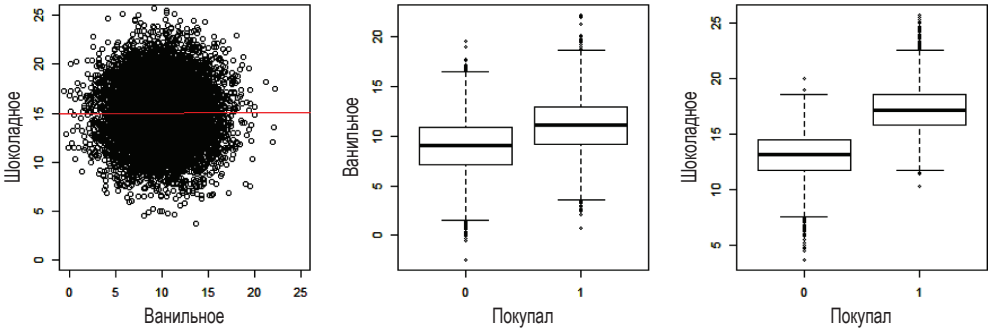

Рис. 1.5 ❖ Левая панель: пристрастия к ванильному и шоколадному вкусам не коррелируют в совокупной популяции; средняя панель: пристрастия к ванильному вкусу выше у людей, которые покупают в киоске с мороженым, чем для людей, которые этого не делают; правая панель: тот же результат для пристрастия к шоколадному вкусу

Вы снова строите тот же график, на этот раз ограничивая данные людьми, которые на вопрос о покупках ответили «Да» (рис. 1.6).

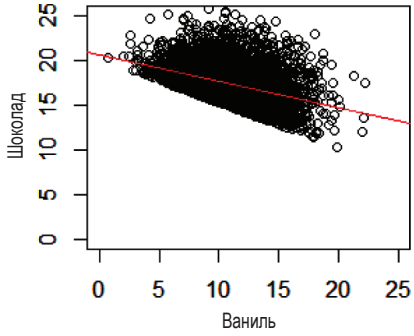

Рис. 1.6 ❖ Предпочтение ванильного вкуса и шоколадного вкуса среди покупателей

В настоящее время между этими двумя переменными существует сильная отрицательная корреляция (коэффициент корреляции равен –0.39). Что случилось? Неужели любители ванили, которые подходят к вашему киоску, превращаются в ненавистников шоколада, и наоборот? Конечно же нет. Эта корреляция была создана искусственно, когда вы сдерживали себя по отношению к покупателям.

Давайте вернемся к нашим истинным причинно-следственным связям: чем сильнее у кого-то пристрастие к ванильному вкусу, тем больше вероятность того, что он будет делать покупки в вашем киоске, и то же самое для шоколадного вкуса. Это означает, что существует кумулятивный эффект этих двух переменных. Если у кого-то слабое вкусовое пристрастие и к ванильному, и к шоколадному мороженому, то он вряд ли будет делать покупки в вашем киоске; другими словами, большинство людей со слабым пристрастием к ванильному вкусу среди ваших покупателей имеют сильное пристрастие к шоколадному вкусу. С другой стороны, если у кого-то есть сильное пристрастие к ванильному вкусу, то он, возможно, будет делать покупки в вашем киоске, даже если у него нет сильного пристрастия к шоколадному вкусу. Отражение этого факта можно увидеть на предыдущем графике: для высоких значений пристрастия к ванильному вкусу (скажем, выше 15) существуют точки данных с более низкими значениями пристрастия к шоколадному вкусу (ниже 15), тогда как для низких значений пристрастия к ванильному вкусу (ниже 5) единственные точки данных на графике имеют высокое значение пристрастия к шоколадному вкусу (выше 17). Ничьи пристрастия не изменились, но люди со слабым пристрастием и к ванильному вкусу, и к шоколадному вкусу исключены из вашего набора данных.

Указанное явление имеет свой технический термин – *парадокс Берксона*[1], но Джуди Перл и Дана Маккензи дают ему более интуитивное название: «эффект оправдания» (explain-away effect, или эффект отмазки). Если у одного из ваших покупателей есть сильное пристрастие к ванильному вкусу, то это полностью объясняет, почему он делает покупки в вашем киоске, и ему «не нужно» иметь сильное пристрастие к шоколадному вкусу. С другой стороны, если у одного из ваших покупателей есть слабое пристрастие к ванильному вкусу, то это не может объяснить, почему он делает покупки в вашем киоске, и у него должно быть более сильное, чем в среднем, пристрастие к шоколадному вкусу.

Парадокс Берксона противоречит здравому смыслу, и поначалу его трудно понять. Он может приводить к систематическому смещению в ваших данных, в зависимости от того, как они были собраны, даже до того, как вы начнете какой-либо анализ. Классическим примером того, как эта ситуация может создавать искусственные корреляции, является то, что некоторые заболевания демонстрируют более высокую степень корреляции, если рассматривать популяцию пациентов больниц в сопоставлении с общей популяцией. На самом деле, конечно же, происходит то, что и той, и другой болезни недостаточно для того, чтобы попасть в больницу; чье-то состояние здоровья становится настолько неважнецким, что оправдывает госпитализацию только тогда, когда обе присутствуют[2].

---

[1] См. https://oreil.ly/KwJ1R.

[2] С технической точки зрения, это несколько иная ситуация, поскольку вместо двух линейных (или логистических) связей существует пороговый эффект, но основополагающий принцип, согласно которому включение неправильной переменной может приводить к искусственным корреляциям, по-прежнему применим.

Выводы

Предсказательная аналитика была чрезвычайно успешной в течение последних нескольких десятилетий и останется таковой. С другой стороны, при попытке понять и – что важнее – изменить поведение человека причинно-следственная аналитика предлагает убедительную альтернативу.

Причинно-следственная аналитика, однако, требует иного подхода, чем тот, к которому мы привыкли с предсказательной аналитикой. Надеюсь, примеры в этой главе убедили вас в том, что вы не можете просто вбросить кучу переменных в линейную или логистическую регрессию и надеяться на лучшее (что мы могли бы рассматривать как подход «включи все, что есть, и Бог подскажет свой вариант»). Однако вы все еще, возможно, задаетесь вопросом о других типах моделей и алгоритмов. Являются ли модели градиентного бустинга или глубокого обучения каким-то образом невосприимчивыми к спутывающим факторам, мультиколлинеарности и ложным корреляциям? К сожалению, ответ отрицательный. Во всяком случае, из «черно-ящичной» природы этих моделей вытекает, что отлавливать спутывающие факторы становится труднее.

В следующей главе мы разведаем вопрос о том, как думать о самих поведенческих данных.

Глава 2

Понимание поведенческих данных

В главе 1 мы выяснили, что стержневая цель этой книги состоит в использовании анализа данных для понимания того, что именно движет поведением. Это требует понимания взаимосвязи между данными и поведениями, которая была представлена в главе 1 стрелкой в причинно-поведенческом каркасе (рис. 2.1).

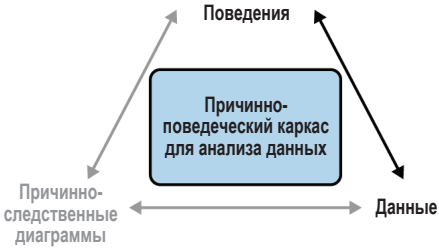

Рис. 2.1 ❖ Причинно-поведенческий каркас с выделенной стрелкой для этой главы

Простите за ссылку на поп-культуру, но если вы видели фильм «Матрица», то помните, что главный герой мог смотреть на окружающий его мир и видеть стоящие за ним цифры. Так вот, в этой главе вы научитесь смотреть на свои данные и видеть стоящие за ними поведения.

Первый раздел в основном адресован читателям, имеющим опыт работы с бизнесом или анализом данных, и предлагает базовое введение в стиле «бихевиористики для начинающих» в ключевые понятия бихевиористики. Если вы – бихевиорист по образованию, то вы, вероятно, не найдете здесь много нового, но, возможно, вам захочется пролистать эту главу, просто чтобы понять, какие именно термины я использую.

Основываясь на этом общем понимании, второй раздел покажет вам, как смотреть на ваши данные через поведенческую призму и выявлять поведенческое понятие, связанное с каждой переменной. Во многих случаях,

к сожалению, переменная сначала лишь слабо связана с соответствующим поведением, поэтому мы также узнаем о том, как «бихевиоризировать» такие капризные переменные.

Базовая модель человеческого поведения

«Поведение» – одно из тех слов, которые хорошо знакомы нам из-за многократного контакта с ними речи, но которые редко, если вообще когда-либо, имеют правильное определение. Однажды я спросил деловую партнершу, какое поведение она пытается поощрять, и ее ответ начался со слов «мы хотим, чтобы они знали, что...». В тот момент я осознал две вещи: (1) помогая прояснять поставленные цели, я смог бы добавить в этот проект больше ценности, чем ожидалось изначально, и (2) введение в бихевиористику, через которое я провел ее ранее, было действительно отстойным, если она по-прежнему считала, что знание чего-то – это и есть поведение. Надеюсь, на этот раз я справлюсь с работой лучше, и вы выйдете из этого раздела, испытав радость от знания того, как добавлять ценность в вашу организацию.

И действительно, я твердо убежден, что одна из ключевых выгод бихевиористского менталитета состоит в том, что он побуждает людей думать точнее о том, что они пытаются сделать. Изменить чье-то мнение – это не то же самое, что повлиять на его действия, и наоборот. Для этой цели я предложу упрощенную, но, надеюсь, действенную модель человеческого поведения, которую я сначала проиллюстрирую примером того, как компания по уходу за собой может извлекать выгоду из кризиса среднего возраста клиента (рис. 2.2).

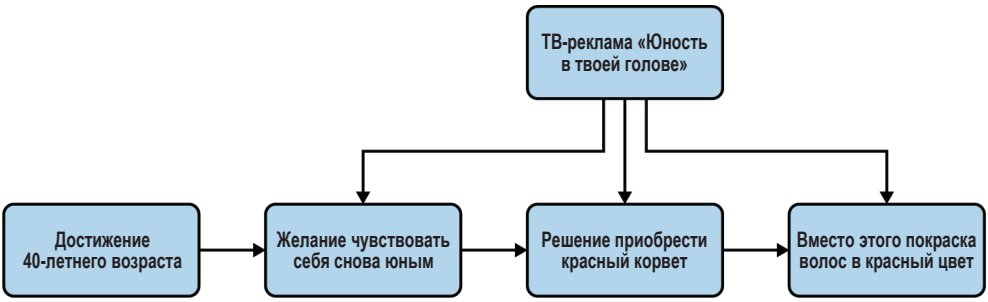

Рис. 2.2 ❖ Наша модель поведения человека в условиях кризиса среднего возраста

В этом примере личностные характеристики (достижение 40-летнего возраста) приводят к эмоциям и мыслям (желание снова почувствовать себя юным), которые, в свою очередь, приводят к намерению (решению приобрести красный корвет). Это намерение может выливаться в соответствующее поведение или может приводить к другому действию (вместо этого покрасить волосы в красный цвет), в зависимости от поведений бизнеса (например, проведение телевизионной рекламы).

В некоторых обстоятельствах мы, возможно, попытаемся повлиять не на поведения наших клиентов, а на поведения наших сотрудников, поставщиков и т. д. Нам пришлось бы эту модель соответствующим образом скорректировать, но интуиция осталась бы прежней: с одной стороны, есть человек, на поведение которого мы пытаемся повлиять, а с другой – существуют все процессы, правила и решения, которые контролирутся нашим бизнесом (рис. 2.3).

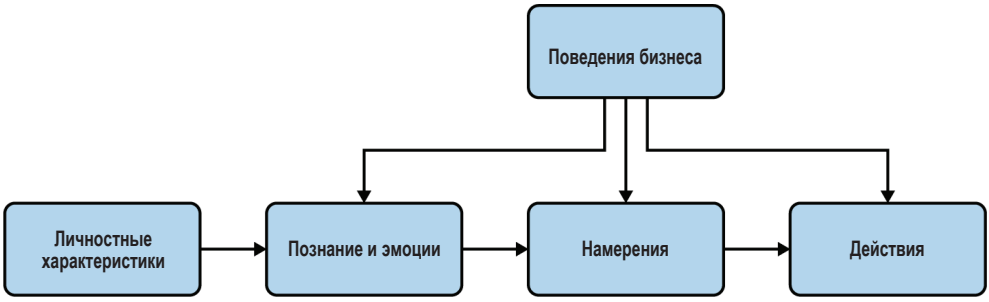

Рис. 2.3 ❖ Наша общая модель человеческого поведения

В этой модели личностные характеристики влияют на познание и эмоции, которые, в свою очередь, влияют на намерения, которые влияют на действия. Поведения бизнеса в форме процессов, правил и решений, которые мы контролируем, влияет на все три категории.

Существует много разных моделей, и в этой нет ничего решающего или волшебного. Но на стадии анализа данных, которая является темой этой книги, я считаю, что эти пять компонентов лучше всего соответствуют типам данных, с которыми вы столкнетесь. Давайте рассмотрим их один за другим, проанализировав, чем конкретно каждый компонент определяется, и разведаем то, как собирать и обрабатывать данные о каждом из них этичным образом.

Личностные характеристики

Сбор и использование демографических переменных, таких как возраст, пол и семейное положение, являются сутью прикладной аналитики, и не без оснований: они часто бывают неплохими предсказателями того, что человек будет делать. Однако как аналитики поведенческих данных мы захотим думать о личностных характеристиках и шире, и точнее. Для наших целей мы дадим определение термину «личностные характеристики» как вся имеющаяся у нас информация о человеке, которая меняется редко либо очень постепенно в течение соответствующего периода времени, охватываемого нашим анализом.

Возвращаясь к примеру с киосками с мороженым C-Mart, это означает включение таких вещей, как черты характера и привычки жизненного уклада. Предположим, что один из ваших покупателей обладает высокой степенью

открытости для опыта и всегда готов попробовать какую-нибудь новую вкусовую комбинацию, такую как черника и сыр; в своем анализе вы можете разумно трактовать эту черту как стабильную личностную характеристику, хотя психология говорит нам, что личность предсказуемым образом меняется в течение жизни человека. С другой стороны, градостроитель, заглядывающий на десятилетия вперед, возможно, захочет учесть не только постепенные изменения в привычках жизненного уклада, но и потенциальные расхождения, в результате которых направление этих изменений само меняется. В этом смысле я думаю о личностных характеристиках как о «первичных причинах», определенных для того, чтобы избегать бесконечной регрессии: мы отслеживаем их изменения, но игнорируем их собственные причины.

Хотя демографические переменные соответствуют приведенному выше определению с точки зрения относительной стабильности и первичности, возможно, будет не сразу ясно, что значит рассматривать их как причины познания, эмоций, намерений и действий. Некоторые читатели могут даже почувствовать себя неловко при мысли, что возраст или пол становится для кого-то причиной что-то делать, поскольку это может подозрительно походить на социальный детерминизм.

Я бы аргументировал тем, что мы можем находить решения этих трудностей, определяя причину как «способствующий фактор» в вероятностном смысле. Достижение 40-летнего возраста не является ни необходимой, ни достаточной причиной кризиса среднего возраста, а кризис среднего возраста не является ни необходимой, ни достаточной причиной покупки красного корвета. На самом деле обе причинно-следственные связи очень тесно переплетены с другими способствующими факторами: вклад возраста в экзистенциальные сомнения во многом зависит от социальных шаблонов, таких как профессиональные и семейные траектории (например, возраст, в котором человек выходит на рынок труда или заводит первого ребенка, если допустить, что он делает и то, и другое), и урегулирование таких сомнений с помощью потребления становится возможным только при наличии достаточного располагаемого дохода. Оно также зависит от степени, с которой на вас влияет реклама.

Как гласит мантра бихевиористики, «поведение есть функция личности и окружающей среды», и социальные факторы часто, возможно, имеют больший вес, чем демографические переменные. С точки зрения причинно-следственного моделирования и анализа данных, это взаимодействие между социальными явлениями и личностными характеристиками может быть зафиксировано с помощью модерации и опосредования (которые мы разведаем в главах 11 и 12).

В этом смысле демографические переменные часто полезны при анализе поведенческих данных, поскольку их предсказательная сила указывает на наличие других личностных характеристик (способствующим фактором которых они являются), носящих более психологический и практический характер. Например, в США и многих других странах правоохранительная деятельность и уход за больными по-прежнему в подавляющем большинстве являются гендерно-дифференцированными областями. Следует отметить, что эмпирическая регулярность вряд ли принесет вам какое-либо похло-

пывание по спине. Гораздо более интересный вопрос заключается в том, как происходит этот эффект. Например, одна из гипотез может заключаться в том, что это проистекает из социальных представлений и норм об авторитете и заботе. В качестве альтернативы можно было бы утверждать, что это явление самосохраняется из-за отсутствия альтернативных ролевых моделей. Определение того, какая гипотеза верна (или какая имеет больший вес, если верны обе), может привести к гораздо более эффективному консультированию старшеклассников по вопросам карьеры.

Сбор данных и этические соображения

Демографическая информация широко доступна и используется (можно даже сказать, используется чрезмерно) в деловой аналитике. Главная трудность состоит в выявлении и измерении других личностных характеристик, чтобы иметь возможность правильно относить их эффекты на поведения, вместо того чтобы необоснованно приписывать их демографическим переменным. Здесь бывают очень полезны исследования рынка и пользователей.

Неверное приписывание эффектов также имеет значение с этической точки зрения: во многих случаях преступность можно рассматривать как сознательное или неосознанное неверное приписывание поведений, текущих или прогнозируемых, демографическим переменным. Даже если ваш анализ верен, вам также необходимо обеспечивать, чтобы он непреднамеренно не приводил к подкреплению шаблонов придания преимуществ и недостатков.

Познание и эмоции

Познание и эмоции – это фраза, которую я использую для обозначения ментальных состояний, таких как эмоции, мысли, ментальные модели и мнения. Вы можете думать об этом как обо всем, что происходит в мозгу клиента, помимо намерений и более постоянных личностных характеристик. Например, осведомлен ли потенциальный клиент о вашей компании и ее продуктах или услугах? Если да, то что он о них думает? Какова его ментальная модель принципа работы банка? Испытывает ли он укол зависти всякий раз, когда видит на дороге новую элегантную Tesla?

Познание и эмоции охватывают все это, а также более туманные деловые словечки и словосочетания, такие как удовлетворенность клиентов (customer satisfaction, аббр. CSAT) и клиентский опыт (customer experience, аббр. CX). Тема удовлетворенности клиентов стала деловой мантрой: во многих компаниях есть коллектив по изучению удовлетворенности клиентов, и имеются конференции, консультанты и посвященные этой теме книги. Но что именно это такое? Можно измерить их причины и следствия? Да, можно, но это требует интеллектуального смирения и готовности уйти с головой в сыскную работу.

Сбор данных и этические соображения

Познание и эмоции похожи на психологические личностные характеристики в двух отношениях: их, как правило, нельзя наблюдать напрямую, и соответ-

ствующие данные собираются с помощью опросов или во время наблюдения за пользовательским опытом (user experience, аббр. UX). Важно отметить, что если вы не используете физиологические меры, вы будете опираться на заявленные или наблюдаемые меры.

Это подводит нас к одному из самых больших различий между пользовательским опытом, или человекоцентрированным дизайном, и бихевиористикой: пользовательский опыт начинается с предположения, что люди знают, чего они хотят, что конкретно они чувствуют по отношению к кому-то и почему, тогда как бихевиористика начинается с предположения, что мы не знаем о многих вещах, происходящих даже в наших собственных головах. Если использовать юридическую метафору, то бихевиорист часто будет трактовать высказываемые человеком мысли как подозрительные, до тех пор, пока не будет доказано, что они заслуживают доверия, в то время как исследователь пользовательского опыта будет трактовать их как откровенные, до тех пор, пока не будет доказано, что они дезориентируют.

Однако я не хочу преувеличивать разницу, и на практике различие нередко становится размытым в зависимости от конкретной ситуации. Если клиент скажет исследователю пользовательского опыта, что веб-сайт вызывает путаницу и разочарование при его использовании, то тот возьмет страницу из книги по исследованию пользовательского опыта и поверит, что клиент действительно испытал негативные эмоции. И наоборот, при проведении фундаментальных исследований продукта квалифицированный бихевиорист часто выходит за рамки заявленных намерений и пытается выявлять более глубокие потребности клиента.

С этической точки зрения попытка изменить чье-либо познание и эмоции, безусловно, является серой территорией. Моя рекомендуемая лакмусовая бумажка в этом отношении состоит в тесте газеты «Нью-Йорк таймс»: достаточно ли доброжелательны и прозрачны ваши намерения и методы, чтобы босс вашего босса не возражал увидеть их на первой странице «Нью-Йорк таймс»? Реклама стремится влиять на мысли и эмоции людей, но я не ожидаю, что когда-нибудь увижу заголовок «Компании тратят миллиарды, чтобы заставить людей желать их вещи»; в любом случае это было бы встречено большим зевком. С другой стороны, *шламы* (sludges, или отбросы) – манипулятивные бихевиористские трюки, типа «еще трое человек прямо сейчас рассматривают этот объект недвижимости!» – и старая добрая ложь указанный тест не прошла бы. Это по-прежнему оставляет много искренних возможностей для применения бихевиористики: например, для объяснения преимуществ вашего продукта в терминах, которые лучше резонируют с ментальными моделями ваших клиентов, или для придания опыту от покупок большей легкости и удовольствия. Поскольку темой этой книги является анализ данных, а не поведенческий дизайн, мы рассмотрим только то, как анализировать такие вмешательства, а не как их развивать.

Намерения

Когда кто-то говорит: «Я собираюсь сделать X», независимо от того, чем является X: покупкой продуктов на неделю либо бронированием отпуска, он

выражает намерение. В силу этого намерения приближают нас на один шаг к действиям.

Технически говоря, намерение – это психическое состояние, которое можно было бы включить в предыдущую категорию познания и эмоций, но я чувствовал, что оно заслуживает отдельного упоминания из-за его важности в прикладном анализе поведенческих данных. Если вы не планируете принуждать клиентов к определенному пути (например, удалите номер телефона с вашего веб-сайта, чтобы они вам не звонили), то вам, как правило, придется использовать их намерение в качестве средства для изменения поведения.

В дополнение к этому у людей часто не получается воплощать то, что они хотят сделать. Эта идея известна в бихевиористике как *разрыв между намерением и действием*. (В качестве примера подумайте о планах на Новый год и о том, как часто они нарушаются.) Следовательно, ключ к подстегиванию поведения клиентов кроется в ответе на вопрос, почему не возникает потенциальный отклик: то ли потому, что клиенты не хотят предпринимать это действие, то ли потому, что нечто происходит между намерением и действием.

Сбор данных и этические соображения

В общем течении бизнеса у нас обычно нет прямых данных о намерениях. Поэтому бóльшая часть этих данных будет поступать из высказываний клиентов, как правило, из двух источников:

опросы.
: Они могут быть либо «в моменте» (например, в конце посещения веб-сайта или магазина), либо асинхронными (например, администрируемыми по обычной почте, электронной почте или телефону);

наблюдения за пользовательким опытом.
: В этих наблюдениях за пользовательским опытом исследователь пользовательского опыта просит испытуемого пройти определенный опыт и собирает информацию о его намерениях (прося его думать вслух в ходе процесса либо задавая вопросы позже, например: «Что вы пытались сделать в тот момент?»).

В качестве альтернативы мы часто пытаемся вывести намерения логически из наблюдаемых поведений, что в аналитике именуется «моделированием намерений».

С этической точки зрения влияние на чьи-либо намерения является целью рекламы, маркетинга и темных искусств убеждения. Это возможно, и я искренне убежден, что в ряде ситуаций это даже возможно без нарушения этических норм, как стремятся делать исследователи пользовательского опыта и поведения. Но простая истина заключается в том, что это, как правило, трудновыполнимо и может иметь неприятные последствия. Гораздо проще и безопаснее определять места, где вы способны помогать клиентам преодолевать разрыв между намерением и действием. Переформулируя распро-

страненную деловую фразу нашей модели, словосочетание «болевая точка» нередко отражает препятствие на пути к достижению намерения.

Действия

Действие – это базовая единица *поведения*, и я нередко буду использовать эти два слова взаимозаменяемо. Я часто рекомендую людям эмпирическое правило, которое состоит в том, что действие или поведение – это то, что вы должны были бы наблюдать, если бы находились в комнате в тот момент, не спрашивая человека. «Купить что-нибудь на Amazon» – это действие. Как и «чтение отзыва о товаре на Amazon». Но «знать что-то» или «решить купить что-то на Amazon» им не является. Невозможно узнать, что кто-то принял решение, если вы не спросите его либо не увидите, как он действует в соответствии с этим решением (что является следствием, но не одним и тем же).

Действие или поведение часто можно определить на разных уровнях гранулярности. «Поход в спортзал» заметен, и во многих ситуациях это приемлемый уровень гранулярности. Если вы отправляете людям купон на бесплатное посещение и они появляются, значит, вы достигли своей цели, и нет необходимости думать глубже. Однако если вы работаете над оздоровительным приложением и хотите обеспечить, чтобы люди следовали своей программе тренировок, то вам нужно быть гораздо более гранулярным, чем это. Однажды я присутствовал на презентации, которую проводил бихевиорист Стив Вендель; в ней он шаг за шагом рассказывал аудитории о том, как «ходить в спортзал» (надевать спортивную одежду, садиться в машину, выбирать упражнения, которые нужно делать, будучи в тренажерном зале, и так далее). Это тот уровень гранулярности, который вам часто потребуется задействовать, если вы пытаетесь выявить и устранять поведенческие блокаторы. То же самое относится и к навигации клиента по веб-сайту или в приложении. Что значит «зарегистрироваться»? Какую информацию необходимо предоставить или ввести? Из каких вариантов вы просите его выбирать?

Сбор данных и этические соображения

Данные о действиях или поведении часто представляют собой самую большую категорию располагаемых данных о клиентах и, как правило, подпадают под рубрику «транзакционные данные». Однако во многих случаях эти данные нуждаются в некоторой дополнительной обработке, чтобы по-настоящему отражать поведения. Мы рассмотрим это подробнее во втором разделе данной главы.

Если вы ориентируетесь на поведение сотрудников (например, в аналитике персонала либо с целью уменьшения выбытия), то данные могут быть намного более разреженными и труднодоступными, поскольку их часто нужно будет извлекать из программно-информационного обеспечения деловых процессов и, возможно, они не будут регистрироваться как само собой разумеющиеся.

Давайте назовем вещи своими именами: модифицирование поведения не является конечной целью поведенческого аналитика. В некоторых случаях все может быть очень просто: если вы удалите свой номер телефона службы поддержки клиентов со своего веб-сайта, то вам будет звонить меньше людей. Хорошая это идея или нет – это уже другая история. И очевидно, что модификация поведения иногда бывает чрезвычайно трудной, о чем свидетельствует большая и растущая литература по изменению поведения[1].

В любом случае, цель модифицирования поведения сопровождается этической ответственностью. Здесь снова вопрос теста газеты «Нью-Йорк таймс» является моей рекомендуемой лакмусовой бумажкой.

Поведения бизнеса

Последняя категория данных, которую нам необходимо учитывать, – это данные о *поведениях бизнеса*: действия организации или одного из ее сотрудников, которые влияют на клиента (или другого сотрудника, если вы сосредоточены на поведении сотрудников). К ним относятся:

- общение, такое как электронные письма и обычные письма;
- изменения языка веб-сайта или разговорного пути для представителя кол-центра по обслуживанию клиентских телефонных звонков;
- деловые правила, такие как критерии вознаграждения клиентов (например, «если клиент потратил X долларов за последние шесть месяцев, то отправить ему купон») или для принятия решений о найме;
- решения отдельных сотрудников, такие как маркировка учетной записи клиента как потенциально мошеннической или продвижение еще одного сотрудника.

Сбор данных

Для аналитиков поведенческих данных поведения бизнеса бывают как благословением, так и проклятием. С одной стороны, с точки зрения дизайна вмешательства, они являются одним из наших главных рычагов, которые подстегивают поведения индивидуумов: например, мы можем изменить содержимое письма или частоту отправки электронной почты, чтобы убедить клиента оплачивать счет вовремя. В силу этого поведения бизнеса являются чрезвычайно ценным инструментом, без которого наша работа была бы значительно труднее, и когда именно мы вносим изменения в поведения бизнеса, то, как правило, достаточно легко собирать соответствующие данные. Еще одним преимуществом поведений бизнеса является то, что в отношении сбора данных присутствует меньше этических соображений: организации, как правило, могут свободно собирать данные о своих собственных операциях

[1] Талер и «Толчок в верном направлении» Санштейна (2009) и «Зацепленный: как создавать продукты, формирующие привычки» Нира Эйяла *(Hooked: How to Build Habit-Forming Products*, Eyal, 2014) являются двумя неплохими справочными материалами среди многих по этой теме.

и поведении своих сотрудников в ходе обычной деятельности (хотя определенно существуют обстоятельства, при которых это может рассматриваться как вторжение в частную жизнь).

С другой стороны, с точки зрения сбора и анализа данных, поведения бизнеса могут стать худшим кошмаром аналитика: подобно воде для рыбы, они бывают невидимы для организации, и их эффект на поведения индивидуумов тогда становится непреодолимым шумом. Это происходит по двум причинам.

Во-первых, многие организации, если они вообще отслеживают поведения бизнеса, просто не отслеживают его на том же уровне детализации, что и поведения клиентов. Предположим, что C-Mart столкнулась с сокращением рабочего времени летом 2018 года, что привело к временному сокращению продаж. Удачно, если это будет выяснено только из располагаемых данных! Многие деловые правила, даже если они имплементированы в программно-информационном обеспечении, просто нигде не регистрируются в машиночитаемом формате. Если соответствующие данные действительно были зарегистрированы, то они часто хранятся только в базе данных подразделения (или, что еще хуже, в файле Excel), а не в корпоративном озере данных.

Во-вторых, поведения бизнеса могут влиять на интерпретацию переменных, относящихся к поведениям клиентов. Самым ярким примером тому могут быть *шламы* – преднамеренные трения и запутывающая передача сообщений, вводящая клиентов в заблуждение. Представьте себе форму на вебсайте, которая при вводе вами своего адреса электронной почты автоматически устанавливает флажок «Я хочу получать маркетинговые электронные письма», который вы сняли в начале формы. Будет ли этот флажок указывать на то, что клиент действительно хочет получать маркетинговые электронные письма? Помимо таких очевидных примеров, можно обнаружить поведения бизнеса, скрывающиеся за многими поведениями клиентов, в особенности в сфере продаж. Многие модели предрасположенности к покупкам должны иметь оговорку «среди людей, которым наш торговый персонал решил позвонить». Парадоксально, но хотя структура вознаграждения торговых представителей часто является одним из рычагов, которые интересуют руководителей бизнеса больше всего, она редко включается в модели поведений клиентов при осуществлении покупок.

В конечном счете получение надежных данных о поведениях бизнеса, в особенности с течением времени, может бросить серьезный вызов аналитикам поведенческих данных, но это означает, что оно также является одним из путей создания добавочной ценности для своей организации до проведения любого анализа.

Как соединять поведения и данные

Личностные характеристики, познание и эмоции, намерения, действия и поведения бизнеса – все это является понятиями, которыми мы располагаем, чтобы представлять и понимать наш мир, будучи аналитиками поведенче-

ских данных. Однако соединение поведений и данных – это не просто вопрос назначения имеющихся переменных одной из этих корзин. Для переменной просто «касаться поведений» недостаточно, чтобы ее квалифицировать как «поведенческие данные»; как мы увидели, например, в предыдущем разделе, переменная, предположительно касающаяся поведения клиентов, на самом деле может отражать только деловое правило. В этом разделе я дам вам список советов по бихевиоризированию ваших данных и обеспечению того, чтобы они максимально плотно вписывались в ту качественную реальность, которую они должны представлять.

В целях придания вещам большей конкретности мы обратимся к нашей второй вымышленной компании, Air Coach and Couch (AirCnC), компании по онлайновому бронированию путешествий и жилья. Руководство AirCnC попросило своего аналитика измерить эффект удовлетворенности клиентов (CSAT) на поведение клиентов при осуществлении покупок в будущем, а именно на сумму, расходуемую в течение шести месяцев после бронирования, *6МесячныйРасход* (*M6Spend*), один из ключевых индикаторов результативности. Мы ответим на этот распространенный, но непрослеживаемый деловой вопрос к концу книги, в главе 12. А пока мы просто посмотрим на то, как приступать к работе, начав с его формулирования.

Развивать бихевиористски целостный менталитет

Поскольку бихевиористика для бизнеса столь нова, вы, как правило, будете первым человеком, применившим эту линзу к данным вашей организации, которые, скорее всего, будут содержать десятки, если не сотни или даже тысячи переменных. Эта задача обескураживающе смелая, но принятие на вооружение правильного менталитета поможет вам в них разобраться и приступить к работе.

Вообразите на секунду, что вы инженер по монтажу технологических конструкций, на которого недавно была возложена ответственность за обслуживание моста. Вы могли бы начать с одного конца и оценивать его целостность дюйм за дюймом (или сантиметр за сантиметром) до тех пор, пока не дойдете до другого конца, а затем разработать 10-летний план достижения идеальной структурной целостности. В то же время выбоины становятся все хуже, подвергая опасности автомобилистов и повреждая их транспортные средства, поездка за поездкой. Более разумный подход состоял бы в том, чтобы быстро осмотреться, определить первостепенные задачи, которые можно быстро решить, и определить места, где подойдут временные расходы до тех пор, пока у вас не появится время и бюджет для внесения дальнейших структурных изменений[1].

---

[1] В этом месте хотел бы заранее извиниться перед инженерами-строителями, и инженерами по монтажу технологических конструкций в частности, если, что вполне вероятно, данная метафора серьезно искажает их работу. Это всего лишь поэтическая вольность и все такое.

Тот же самый менталитет относится и к вашим данным. Если вы не окажетесь в числе самых первых сотрудников стартапа, то вам придется иметь дело с существующими данными и унаследованными процессами. Не паникуйте и не начинайте просматривать свой список таблиц в алфавитном порядке. Начните с конкретной деловой задачи и выявите переменные, которые, скорее всего, будут неточными, в порядке уменьшения их важности для деловой задачи:

1) интересующие причины и следствия;
2) посредники и модераторы, если они релевантны;
3) любая потенциальная спутывающая переменная (или фактор);
4) другие неспутывающие независимые переменные (т. н. ковариаты).

Вам придется принимать решения по ходу дела: например, следует ли вам включать некоторую переменную в свой анализ, или же она настолько плохо определена, что вам лучше обойтись без нее? К сожалению, нет четко обозначенного критерия для вынесения правильных суждений; вам придется полагаться на свой деловой здравый смысл и опыт. Однако есть четко обозначенный путь к вынесению этих суждений неправильно: притворяться, что их не существует. Переменная либо будет включена в ваш анализ, либо нет, и обойти этот факт невозможно. Если ваш инстинкт склоняется к включению, то вполне вероятно, что вам следует задокументировать причину, описать потенциальные источники ошибок и указать на то, как изменились бы результаты, если бы переменная была опущена. Как однажды выразился один исследователь пользовательского опыта в дружеской беседе со мной, быть исследователем в бизнесе означает постоянно думать о том, «что конкретно тебе может сойти с рук».

Не доверять и проверять

К сожалению, во многих обстоятельствах способ регистрации данных диктуется деловыми и финансовыми правилами и ориентирован на транзакции, а не на клиента. Это означает, что вы должны считать переменные подозрительными до тех пор, пока не будет доказана их невиновность: другими словами, не исходить автоматически из того, что переменная *КлиентСделалХ* означает, что клиент сделал Х. Это может означать что-то совершенно другое. Например:

- клиент поставил галочку, не читая мелкий шрифт, в котором говорилось, что он соглашается на Х;
- клиент ничего не сказал, поэтому мы по умолчанию транслировали это в Х;
- клиент заявил, что он сделал Х, но мы не можем перепроверить;
- мы купили данные у поставщика, указывающие на то, что клиент в какой-то момент своей жизни регулярно делал Х.

Даже если клиент действительно сделал Х, мы не можем брать за основу его намерение. Возможно, он сделал это:

- потому что мы отправили ему электронное письмо с напоминанием;

- четыре раза подряд, потому что страница не обновлялась;
- ошибочно, когда он на самом деле хотел сделать Y;
- неделю назад, но из-за нормативных ограничений мы зафиксировали это только сегодня.

Другими словами, перефразируя популярную строчку из фильма «Принцесса-невеста»: «Ты продолжаешь использовать эту переменную. Я не думаю, что она означает то, о чем ты думаешь».

Выявлять категорию

Как уже говорилось, руководство компании AirCnC попросило своих аналитиков разобраться в *эффекте удовлетворенности клиентов* (CSAT) на поведения при осуществлении покупок, что является чрезмерно сложным поручением. Наш первый шаг состоит в том, чтобы понять, о чем мы говорим.

На первом курсе колледжа наш профессор философии давал нам подсказки для эссе, такие как «Что такое прогресс?» и «Человек и машина». Вместе с ними он давал несколько отличных советов, если мы заходили в тупик: по его словам, мы должны выяснить, «из какой книги эта глава». Парадоксально, но когда вопрос кажется слишком большим и непрослеживаемым, часто помогает решение о том, к какой более широкой категории он относится.

К счастью, как аналитикам поведенческих данных нам не нужно бродить по библиотеке данных в поисках вдохновения; у нас есть индивидуальная классификация, которая была описана ранее в этой главе:

- личностные характеристики;
- познание и эмоции;
- намерения;
- действия (т. н. поведения);
- поведения и процессы бизнеса.

Давайте продолжим путем исключения. Удовлетворенность клиентов не является фиксированной и постоянной, поэтому она не является частью личностных характеристик. Это не то, что люди делают, и не то, что люди собираются сделать, и, значит, это не поведение и не намерение. Наконец, это не то, что происходит исключительно на деловой стороне, и, значит, это не поведение или процесс бизнеса. Следовательно, удовлетворенность клиентов является частью познания и эмоций, как и его двоюродный брат – *клиентский опыт*. Вторую переменную в нашей деловой задаче, «поведения при осуществлении покупок», гораздо легче категоризировать: очевидно, что это поведение клиента.

По моему опыту, многие проекты деловой аналитики оказываются безуспешными или дают неутешительные результаты по той причине, что аналитик не уточнил, о чем идет речь в проекте. У организаций всегда есть всеохватывающая целевая метрика – прибыль в случае компаний, результаты работы с клиентами в случае некоммерческих организаций и т. д. На более низком уровне отделы часто имеют свои собственные целевые показатели,

такие как чистый балл промоутера[1] в случае коллектива по работе с клиентами, процент простоев в случае ИТ и т. д. Если деловой партнер просит вас измерить или улучшить переменную, которая выглядит не связанной с одной из этих целевых метрик, то это обычно означает, что он имеет в виду неявную и, возможно, ошибочную поведенческую теорию, соединяющую эти две переменные.

«Вовлеченность клиентов», еще одно модное понятие, усовершенствовать которое часто просят бихевиористов, является хорошим примером этого явления. Только вот не ясно, где ему место, потому что оно на самом деле может относиться к двум разным вещам:

- поведению, а именно широкому шаблону взаимодействий с бизнесом: клиент А считается более вовлеченным, чем клиент В, если клиент А чаще заходит на веб-сайт и тратит больше времени на навигацию по нему;
- познанию или эмоции, например когда аудитория «увлечена» фильмом или ходом событий, потому что они поглощены потоком и хотят знать, что будет дальше.

И действительно, я твердо убежден, что путаница между этими двумя вещами объясняет бóльшую часть привлекательности метрик вовлеченности для стартапов и более широкого цифрового мира, даже если они, возможно, будут дезориентировать. Например, в первом смысле этого слова я больше увлечен своей стиральной машиной, когда она перестает работать; это не переводится в удовольствие и рвение во втором смысле данного слова. Организации, которые пытаются повысить вовлеченность как поведение, часто бывают разочарованы результатами. Когда вовлеченность как поведение не преобразовывается в вовлеченность как эмоцию, это не приводит к желаемым результатам, таким как более высокая лояльность и удержание.

В качестве личного примера: одна деловая партнерша однажды попросила меня о помощи, чтобы побудить сотрудников пройти определенное обучение. После некоторого обсуждения стало ясно, что на самом деле она хотела, чтобы сотрудники соблюдали деловое правило; она считала, что они его не соблюдали, потому что были недостаточно информированы об этом правиле. Мы сориентировали проект в сторону понимания того, почему сотрудники не соблюдали требования и как их побудить к этому. Одним словом: следует остерегаться самодиагностирующих пациентов!

Теперь мы можем изменить нашу деловую задачу в нашей модели человеческого поведения: руководители AirCnC хотят знать, влияют ли когнитивные способности/эмоции на поведение клиентов, и если да, то насколько сильно. И действительно, в подавляющем большинстве задач деловой аналитики, по меньшей мере, одной из связанных с этим переменных является

---

[1] *Чистый балл промоутера* (Net Promoter Score) – это индекс в диапазоне от –100 до 100, который измеряет готовность клиентов рекомендовать продукты или услуги компании другим. Он используется в качестве косвенного индикатора для оценивания общей удовлетворенности клиента продуктом или услугой компании и лояльности клиента к бренду. – *Прим. перев.*

поведение либо клиента, либо бизнеса. Будут ли клиенты более удовлетворены, если мы отправим им последующее электронное письмо? Склоняются ли довольные клиенты к покупке больше или нет?

Если после определения категорий соответствующих переменных ни одна из них не подпадает под категории поведений клиентов или бизнеса, то это должно вызвать тревогу. Это наводит на мысль о деловой задаче без четкого возгласа «Ну и что дальше?». Предположим, что пожилые клиенты более удовлетворены. Ну и что дальше? Что мы будем делать с этой информацией? Конечные результаты бизнеса диктуются поведениями, как нашими, так и наших клиентов.

Определив соответствующие поведения, теперь самое время подробно остановиться на соответствующих переменных.

Уточнять поведенческие переменные

Как я упоминал ранее, переменная, которая «касается поведения», – это не то же самое, что поведенческая переменная. Вам часто придется их преобразовывать, чтобы делать их поистине поведенческими.

Давайте сосредоточимся на поведениях клиентов, поскольку они интуитивнее, но эта логика применима и к поведениям бизнеса. Хорошая поведенческая переменная будет обладать следующими характеристиками:

наблюдаема.

Как упоминалось в главе 1, поведение можно наблюдать, по меньшей мере в принципе. Если бы вы находились в комнате с клиентом, то могли бы вы увидеть, как он себя ведет? Отказ от бронирования в середине процесса можно наблюдать; изменение своего мнения – нет. Хорошей подсказкой будет следующее: если у нее нет метки времени, то она, вероятно, недостаточно конкретна и гранулярна;

отдельна.

Предприятия часто опираются на агрегатные метрики, такие как степени или доли (например, доля клиентов, которые отменяют свою учетную запись). Агрегатные метрики могут предлагать полезные моментальные снимки для целей отчетности, но они могут становиться жертвой систематических смещений и спутывающих факторов, таких как изменения в составе популяции (т. н. клиентской смеси), в особенности когда они рассчитываются на основе временных интервалов.

Например, давайте вообразим, что успешная маркетинговая кампания привлекает много новых пользователей на веб-сайт AirCnC. Давайте также допустим, что в этой сфере бизнеса значительная доля новых клиентов отменяет свою учетную запись в первый месяц. Отсюда степень ежедневной отмены AirCnC может тревожно взмыть вверх в течение месяца, следующего за кампанией, даже если ничего не было, что пошло бы не так. Хорошее эмпирическое правило состоит в том, чтобы прочные агрегатные переменные основывались на прочных индивидуальных переменных. Если переменная имеет смысл только в агрегате и не имеет содержатель-

ной интерпретации на отдельном уровне, то это красный флажок. В нашем примере содержательной отдельной противоположностью степени отмены будет вероятность отмены. При контролировании отдельных характеристик и стажа взаимодействия с компанией этот показатель будет оставаться стабильным, несмотря на приток новых клиентов;

атомарна.
Схожим образом предприятия часто агрегируют воедино самые разные поведения, делящие между собой общее намерение. Например, у клиента может быть три разных способа изменить свой платежный адрес с помощью AirCnC: перейдя в настройки своей учетной записи, отредактировав информацию по завершении бронирования и связавшись с колцентром. Для того, кто наблюдает за клиентом в данный момент, они будут выглядеть по-разному, но могут регистрироваться аналогичным образом в базе данных. Опять же, я не хочу утверждать, что агрегатные поведенческие переменные изначально плохи. Безусловно, существуют аналитические расчеты, которые требуют использования двоичной переменной *ИзмененнаяПлатежнаяИнформация*. Но по меньшей мере вы должны знать о конкретных способах, которыми это можно делать, и не помешает по возможности проверять, применимо ли к каждому из них совокупное заключение, к которому вы пришли.

Во многих обстоятельствах выявление или создание удовлетворяющей поведенческой переменной предусматривает «марание рук». Базы данных для аналитических или исследовательских целей часто предлагают «очищенную» версию истины, в том виде, как она будет появляться в базах транзакционных данных, в которых перечисляется только самая актуальная, проверенная информация. В большинстве случаев это имеет смысл: если клиент сделал бронирование, а затем отменил его и получил назад свои средства, то мы бы не хотели, чтобы эта сумма учитывалась в переменной *ИзрасходованнаяСумма*. В конце концов, с точки зрения бизнеса, AirCnC не смогла бы удерживать эти деньги. Однако с поведенческой точки зрения этот клиент отличается от клиента, который не делал никаких бронирований в течение того же периода времени, и существуют аналитические расчеты, для которых было бы уместно принимать это во внимание. Не следует изучать древний язык программирования, такой как COBOL, только для того, чтобы получать доступ к базам данных самого низкого уровня, но стоит немного покопаться за пределами ваших обычных красивых таблиц.

Понимать контекст

Следует подчеркнуть, что высказывание «поведение есть функция человека и окружающей среды» является фундаментальным принципом бихевиористики. Хотя личностные переменные, безусловно, важны, я чувствую, что аналитики часто слишком на них опираются, потому что указанные переменные легкодоступны, и, как следствие, аналитики недостаточно задумываются о контекстуальных переменных.

Нередко понять контекст(ы), в котором(ых) люди себя ведут лучше всего посредством качественных исследований, таких как собеседования и опросы, результаты которых могут использоваться для генерирования новых переменных. Здесь я сосредоточусь на том, каким образом извлекать контекстную информацию из существующих данных.

Касается времени

Как упоминалось ранее, поведение можно наблюдать. Вы можете точно засечь момент времени, когда это произошло, по меньшей мере в теории, а часто и на практике, благодаря *временным меткам*. Временные метки являются золотыми самородками для поведенческих аналитиков, потому что они обеспечивают интуитивное и нередко легко имплементируемое понимание. Неудивительно, что для их извлечения нет готового алгоритма, есть только эмпирические правила, основанные на деловом смысле и специфике решаемой задачи. Я дам вам наиболее распространенные подсказки, которые нужно искать.

Частота

Естественная склонность, имея дело с поведенческими и, в более широком смысле, событийными данными, – смотреть на частоту, численность событий/поведений в единицу времени. К сожалению, частотные данные иногда могут вести себя неправильно и демонстрировать искусственные разрывы, которые не отражают изменения в поведениях. Например, предположим, что клиент AirCnC берет отпуск каждое лето и каждую зиму, что транслируется в два бронирования в год. Однако в один год он берет зимние каникулы в январе вместо декабря. Мы бы засчитали один отпуск в предыдущем году и три в последнем, даже если поведения на самом деле не изменились. Такие явления наверстывания – длительные периоды, за которыми следуют более короткие, лучше всего понимать, отслеживая продолжительности непосредственно. В случае других поведений прошлое осталось в прошлом, и частота будет более устойчивой метрикой.

Продолжительность

Продолжительность также предлагает естественный способ измерения затухающих эффектов. Вещи, которые вы сделали или которые произошли давным-давно, как правило, имеют меньшие эффекты, чем более недавние события. Это часто делает продолжительность хорошей предсказательной переменной. Если клиент не покинул AirCnC после плохого опыта пять лет назад, то это, вероятно, больше не сильно влияет на его решения, и было бы лучше оценивать метрику удовлетворенности клиента (CSAT) прошлых поездок по тому, как давно они произошли, а не просто использовать среднее значение.

Смежность

Точно так же поведения, которые очень близки друг к другу, часто не являются случайными и могут давать ключ к пониманию того, что происходит. Клиент, звонящий в кол-центр AirCnC, чтобы изменить свою

платежную информацию после попытки изменить ее онлайн, ведет себя иначе, чем клиент, который звонит напрямую. Разрозненные данные делают проекты омниканальной аналитики обескураживающе сложной задачей, но успешное сведение данных из разных каналов будет приносить большие дивиденды. Один из лучших путей к агрегатным поведенческим данным лежит через создание переменных для правила «делать Z после того, как сделано X».

Социальные расписания
Люди с большей вероятностью будут на работе или по дороге на работу или с работы с понедельника по пятницу в течение дня[1]. Клиент, неторопливо просматривающий пункты назначения выходного дня в субботу утром, возможно, будет на спортивной тренировке своего ребенка. Современная жизнь имеет свои ритмы и расписания, которые общеизвестны. Из-за их гранулярности нередко лучше начинать с переменной «час недели», а не с отдельных переменных «час дня» и «день недели» (по местному времени, конечно). В зависимости от вашей сферы бизнеса вы можете агрегировать вещи дополнительно еще глубже в переменные, такие как «вечера в будние дни» и т. д.

Информация и «известные неизвестности»
Если дерево падает в лесу и это регистрируется в ваших данных, но никто из клиентов этого не слышал, издавало ли оно звук? Ваша организация и ваши клиенты часто знают разные вещи в разные моменты времени. Переменные, используемые в отношении поведения, всегда должны отражать информацию, доступную человеку, который делает это, пока он это делает. Это бывает так же просто, как заменить «дату отправки» на «ожидаемую дату получения» для почты, которую вы отправляете своему клиенту, и наоборот – для почты, которую он отправляет вам. Люди не знают содержимого писем, которые они не открывали. Как всегда, это отчасти здравый смысл и отчасти поведенческая логика, но это поможет обеспечивать, чтобы ваши переменные плотно укладывались в соответствующие поведения.

Собака, которая не лаяла
Иногда люди делают одно вместо другого, а иногда у них нет выбора (либо они его не видят). То, чего люди не делают, нередко бывает таким же интересным, как и то, что они делают. Выявлять альтернативные поведения можно путем поиска развилок: если поведение B часто происходит после поведения A, то какое другое поведение C так же часто происходит после A? И наоборот, если поведение D часто происходит после поведения B, то какие другие поведения приводят к поведению D? Это может обеспечивать вам возможность выявлять «счастливые пути» либо клиентов, которые заблудились на пути к выполнению работы.

Более глубокое понимание контекста поведения является одним из способов, с помощью которого поведенческий аспект приносит пользу про-

[1] Если, конечно, не случится глобальной пандемии.

ектам деловой аналитики, таким как AirCnC. Хочет ли руководство AirCnC оценивать удовлетворенность клиентов своим опытом онлайнового бронирования, своим пребыванием либо представителями службы AirCnC (среди многих других возможностей)? Вы можете спросить клиента о степени его удовольствия в любой момент, начиная с самого первого взаимодействия с AirCnC, и он даст вам ответ, но это не гарантирует, что ответы означают одно и то же или даже что они что-то значат.

В нашем случае руководство AirCnC действительно хочет понять, насколько значима удовлетворенность клиента после звонка в службу. Это поможет им определять, следует ли им больше инвестировать в наем и обучение высококлассных представителей или же, наоборот, можно ли им передать услуги на аутсорсинг в страну с более низкой заработной платой (подсказка: вероятно, им не следует этого делать).

Выводы

Одна из радостей и проблем прикладной бихевиористики заключается в том, что она предусматривает постоянное переключение между качественным и количественным анализом. Моя цель в этой главе состояла в том, чтобы вооружить вас базовой моделью человеческого поведения и полезными советами, чтобы вы могли начать напрягать серое вещество и связывать поведения с данными. Это означает, что еще до выполнения любого анализа данных вы можете привнести добавочную ценность в свою организацию, улучшив поведенческую целостность ее данных и выяснив вопрос о том, как подходить к деловым задачам.

В следующей части книги мы познакомим вас с третьим столпом причинно-поведенческого каркаса, а именно с причинно-следственными (или каузальными) диаграммами, которые позволят нам строить взаимосвязи между поведениями.

Часть II

ПРИЧИННО-СЛЕДСТВЕННЫЕ ДИАГРАММЫ И РАСПУТЫВАНИЕ

В части I мы увидели, как спутывание способно ставить под угрозу даже самые простые аналитические расчеты на данных. Во второй части мы научимся строить причинно-следственные диаграммы для представления, понимания и распутывания взаимосвязей между переменными.

Прежде всего глава 3 содержит введение в причинно-следственные диаграммы и их строительные блоки.

В главе 4 мы увидим, как создавать причинно-следственную диаграмму с нуля для нового анализа. Причинно-следственная диаграмма, которую мы увидели в нашем примере с мороженым, была по дизайну очень простой. Но в реальной жизни часто бывает сложно понять, какие переменные следует включать в нашу причинно-следственную диаграмму, помимо интересующих нас причины и следствия, и как определять взаимосвязи между ними.

Подобным же образом устранить спутывание в нашем примере с мороженым было просто: нам лишь нужно было включить в регрессию общую причину интересующих нас переменных. В случае же более сложных причинно-следственных диаграмм бывает трудно определить переменные, которые следует включать в регрессию. В главе 5 мы рассмотрим правила, которые вы сможете применять даже к самой сложной причинно-следственной диаграмме.

Глава 3

Введение в причинно-следственные диаграммы

На самом деле, за редким исключением, из корреляции действительно вытекает причинно-следственная связь. Если мы наблюдаем систематическую взаимосвязь между двумя переменными, и мы исключили вероятность того, что это просто случайное совпадение, то что-то должно быть причиной этой взаимосвязи. Когда зритель в малайском театре теней видит на экране сплошную круглую тень, он знает, что ее отбросил какой-то трехмерный объект, хотя он может и не знать, чем является этот объект: шаром или рисовой миской в профиль. Более точной хлесткой фразой для вводной статистики было бы, что из простой корреляции вытекает неурегулированная причинно-следственная структура.

– Билл Шипли, «Причина и корреляция в биологии» (2016)

Причинно-следственные диаграммы вполне могут быть одним из самых мощных инструментов анализа, о котором большинство людей никогда не слышали. Как таковые они являются одним из трех экстремумов (вершин) причинно-поведенческого каркаса (рис. 3.1). Они обеспечивают язык для выражения и анализа причинно-следственных связей, который работает особенно хорошо при выполнении аналитических расчетов на поведенческих данных.

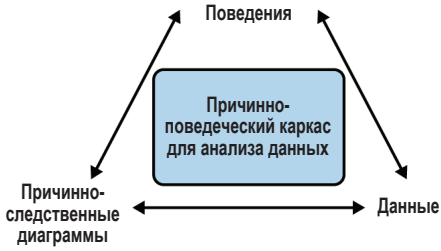

Рис. 3.1 ❖ Причинно-поведенческий каркас для анализа данных

В первом разделе этой главы я покажу, как причинно-следственные диаграммы вписываются в указанный каркас с концептуальной точки зрения, то есть как они связаны с поведениями и данными. Во втором разделе я опишу три фундаментальные структуры в причинно-следственных диаграммах: цепочки, развилки и сталкиватели. Наконец, в третьем разделе мы увидим несколько распространенных преобразований, которые могут применяться к причинно-следственным диаграммам.

Причинно-следственные диаграммы и причинно-поведенческий каркас

Прежде всего давайте дадим определение термину «причинно-следственная диаграмма». *Причинно-следственная диаграмма* (causal diagram, аббр. CD), или *каузальная диаграмма*, – это визуальная демонстрация переменных, изображаемых в виде прямоугольников, и их взаимосвязей друг с другом, изображаемых в виде стрелок, переходящих из одного прямоугольника в другой.

В нашем примере с C-Mart из главы 1 переменная *ПродажиХолодногоКофе* находилась под влиянием одной причины – *Температуры*. На рис. 3.2 показана соответствующая причинно-следственная диаграмма.

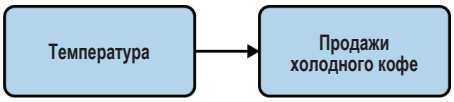

Рис. 3.2 ❖ Наша самая первая причинно-следственная диаграмма

Каждый прямоугольник изображает переменную, которую мы можем наблюдать (ту, которую мы имеем в нашем наборе данных), а стрелка между ними изображает существование и направление причинно-следственной связи. Здесь стрелка между *Температурой* и *ПродажамиХолодногоКофе* указывает на то, что первая является причиной последних.

Однако иногда возникает дополнительная переменная, которую мы не можем наблюдать. Если мы все же хотим показать ее на причинно-следственной диаграмме, то мы можем представить ее затененным прямоугольником[1] (рис. 3.3).

На рис. 3.3 *ПристрастиеПокупателейКСладкому* является причиной *ПродажХолодногоКофе*, из чего следует, что покупатели с более сильным пристрастием к сладкому покупают больше холодного кофе. Однако мы не можем наблюдать степень пристрастия покупателя к сладкому. Позже мы обсудим важность ненаблюдаемых спутывающих факторов и, в более общем случае, ненаблюдаемых переменных в причинно-следственном анализе. А пока

[1] Наиболее распространенным способом изображения ненаблюдаемых переменных на причинно-следственной диаграмме являются овалы вместо прямоугольников.

давайте просто отметим, что даже если у нас нет возможности наблюдать конкретную переменную, ее все равно можно включить в причинно-следственную диаграмму, изобразив ее в виде овала.

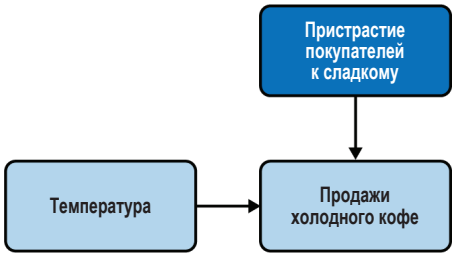

Рис. 3.3 ❖ Причинно-следственная диаграмма с ненаблюдаемой переменной

Причинно-следственные диаграммы представляют поведения

Первый вариант рассмотрения причинно-следственных диаграмм состоит в том, чтобы трактовать их как представления причинно-следственных взаимосвязей между поведениями, а также другими явлениями в реальном мире, которые влияют на поведения (рис. 3.4). С этой точки зрения элементы причинно-следственных диаграмм представляют собой реальные «вещи», которые существуют и имеют эффекты друг на друга. Аналогией из области физических наук был бы магнит, железный брусок и магнитное поле вокруг магнита. Увидеть магнитное поле невозможно, но оно тем не менее существует, и оно влияет на железный брусок. Возможно, у вас нет никаких данных о магнитном поле, и, возможно, вы никогда не видели описывающих его уравнений, но вы способны его чувствовать, когда двигаете брусок, и вы способны развить интуицию относительно того, что он делает.

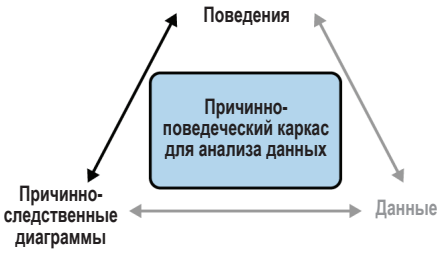

Рис. 3.4 ❖ Причинно-следственные диаграммы связаны с поведениями в нашем каркасе

Та же точка зрения применима, когда мы хотим понять, что именно движет поведениями. Мы интуитивно понимаем, что у людей есть привычки,

предпочтения и эмоции, и мы относимся к ним как к причинам, даже если у нас часто нет никаких числовых данных о них. Когда мы говорим: «Джо купил арахис, потому что был голоден», мы опираемся на наши знания, опыт и мнения о людях в целом и Джо в частности. Мы относимся к голоду как к реальной вещи, даже если не измеряем уровень сахара в крови или активность мозга Джо.

Здесь мы делаем причинно-следственное высказывание о реальности: мы говорим, что если бы Джо не был голоден, то он бы не купил арахис. Причинно-следственная связь настолько важна для нашего интуитивного понимания реальности, что даже маленькие дети способны делать правильные причинно-следственные выводы (о чем свидетельствует использование ими слова «потому что») задолго до того, как они подверглись контакту с каким-либо научным методом или анализом данных. Конечно, интуиция подвержена множеству систематических смещений, хорошо известных бихевиористам, даже когда она принимает более образованную форму здравого смысла или опыта. Но чаще всего интуиция неплохо нами руководит в нашей повседневной жизни даже в отсутствие количественных данных.

Вас, возможно, беспокоит, что использование причинно-следственных диаграмм для представления интуиции и мнений о мире привносит субъективность, и это, безусловно, верно. Но поскольку причинно-следственные диаграммы являются инструментами мышления и анализа, они не обязательно должны быть «правдивыми». У нас с вами могут быть разные идеи относительно того, почему Джо купил арахис, из чего следует, что мы будем чертить разные причинно-следственные диаграммы. Даже если бы мы полностью согласились с тем, что именно является причиной чего-либо, мы не смогли бы представить все вещи и их взаимосвязи на одной диаграмме; всегда существует личное суждение, связанное с определением того, какие переменные и связи включать или исключать. В некоторых случаях, когда у нас есть данные, это будет помогать: мы сможем отклонить причинно-следственную диаграмму, потому что она несовместима с располагаемыми данными. Но в других случаях очень разные причинно-следственные диаграммы будут совместимы с данными одинаково, и мы не будем способны выбирать между ними, в особенности если у нас нет экспериментальных данных.

Эта субъективность, возможно, выглядит как (вероятно, фатальный) недостаток причинно-следственных диаграмм, но на самом деле она является функциональной особенностью, а не дефектом. Причинно-следственные диаграммы не создают неопределенности; они просто отражают неопределенность, которая уже существует в нашем мире. Если есть несколько возможных интерпретаций рассматриваемой ситуации, которые выглядят одинаково верными, то вы должны говорить об этом прямо. Альтернативой было бы позволить людям, у которых в голове разные ментальные модели, мнить, что они знают правду, а другим соглашаться с ними, когда на самом деле это не так. По меньшей мере, открытое обсуждение неопределенности позволит проводить принципиальную дискуссию и направить ваш анализ.

Причинно-следственные диаграммы представляют данные

Хотя в строительстве и интерпретации причинно-следственных диаграмм присутствует некое искусство, в них есть и наука, и мы можем использовать причинно-следственные диаграммы для представления взаимосвязей между переменными в наших данных (рис. 3.5). Когда эти взаимосвязи полностью линейны или примерно таковы, причинно-следственные диаграммы имеют четкие эквиваленты в линейной алгебре. Это означает, что мы можем использовать правила и инструменты линейного алгоритма для подтверждения «законности» того, как мы манипулируем и преобразовываем причинно-следственные диаграммы, тем самым обеспечивая правильность выводимых нами заключений.

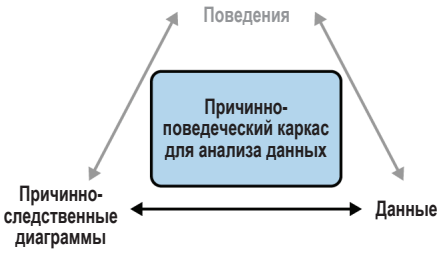

Рис. 3.5 ❖ Причинно-следственные диаграммы также связаны с данными

Требование линейности может показаться очень ограничительным. Однако некоторые правила и инструменты линейной алгебры продолжают оставаться применимыми, когда некоторые из этих взаимосвязей не являются линейными, но по-прежнему относятся к широкой категории моделей, именуемых *обобщенными линейными моделями* (generalized linear models, аббр. GLM). Например, логистическая регрессионная модель является обобщенной линейной моделью. Из этого вытекает, что мы можем представлять и обрабатывать причинно-следственную связь, в которой переменная эффекта является двоичной, с помощью причинно-следственных диаграмм. Как видно на врезке, в этом случае математика становится извилистее, но большинство наших интуиций о причинно-следственных диаграммах остаются истинными.

Техническое более глубокое погружение: причинно-следственные диаграммы и логистическая регрессия

Все взаимосвязи, которые мы встречали до сих пор между переменными, являются линейными. Когда зависимая переменная является двоичной (т. е. она может принимать только два значения, обычно «Да» и «Нет»), вместо линейной регрессии мы используем логистическую регрессию. Логистическая регрессия является примером обобщенной линейной модели, и в силу этого она не линейна, но имеет некоторые линейные характеристики. Из этого вытекает, что логика причинно-следственных диаграмм по-

прежнему работает, при некотором размахивании руками. Если вам интересно, давайте заглянем под капот, чтобы убедиться, что вы понимаете, что происходит.

Давайте посмотрим на один из наших предыдущих примеров, в котором предпочтение ванильного вкуса и предпочтение шоколадного вкуса являются причиной покупки мороженого. На этот раз мы будем представлять покупки мороженого в виде двоичной переменной: 1, если покупатель купил хотя бы одно мороженое в тот день, и 0, если он этого не сделал (рис. 3.6).

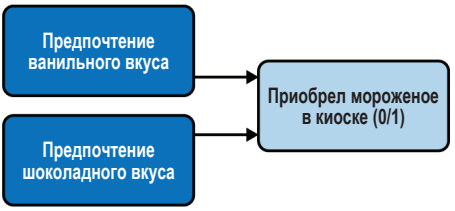

Рис. 3.6 ❖ Представление покупок мороженого в виде двоичной переменной на причинно-следственной диаграмме

Вместо линейной регрессии, которую мы использовали ранее, стрелки на этой причинно-следственной диаграмме теперь представляют логистическую регрессию, означающую, что вероятность того, что наша целевая переменная равна 1, является преобразованием линейной комбинации объясняющих переменных:

$$P(\textit{ПокупкаМороженого} = 1) = (1 + e^{-(\beta_0 + \beta_V \cdot \textit{ПредпочтениеВанильногоВкуса} + \beta_C \cdot \textit{ПредпочтениеШоколадногоВкуса})})^{-1}.$$

$F(x) = 1/(1 + e^{-x})$ называется логистической функцией, и она продуцирует S-образную кривую со значениями от нуля до единицы. Поэтому приведенное выше уравнение можно переписать, вставив в него функцию f:

$$P(\textit{ПокупкаМороженого} = 1) = f(\beta_0 + \beta_V \cdot \textit{ПредпочтениеВанильногоВкуса} + \beta_C \cdot \textit{ПредпочтениеШоколадногоВкуса}).$$

Это означает, что мы не можем выполнить трансляцию увеличения на 1 «единицу» предпочтения ванильного мороженого напрямую в фиксированное увеличение вероятности покупки мороженого. По этой причине коэффициенты логистической регрессии, как известно, трудно интерпретировать. Однако взаимосвязи между коэффициентами и переменными внутри логистической функции остаются линейными, означая, что алгебраические преобразования, которые мы увидим в следующем разделе, все равно будут правильными.

Например, мы могли бы концептуально разрезать *ПредпочтениеШоколадногоВкуса* на переменные, представляющие вкус различных шоколадных ароматов. Приведенное выше уравнение тогда приняло бы следующий вид:

$$P(\textit{ПокупкаМороженого} = 1) = F(\beta_0 + \beta_V \cdot \textit{ПредпочтениеВанильногоВкуса} + \beta_{DC} \cdot \textit{ПредпочтениеВкусаТемногоШоколада} + \beta_{FC} \cdot \textit{ПредпочтениеВкусаМолочногоШоколада}).$$

Внутри логистической функции объясняющие переменные по-прежнему остаются линейными, и линейные преобразования по-прежнему могут применяться. Все это означает, что я в основном буду ссылаться на линейную регрессию, потому что она более распространена, но вы можете быть уверены, что все, что я говорю, также будет применимо и к логистической регрессии, плюс/минус некоторые незначительные математические преобразования.

С этой точки зрения причинно-следственная диаграмма на рис. 3.3, соединяющая *Температуру* с *ПродажамиХолодногоКофе*, означала бы, что

ПродажиХолодногоКофе = β · *Температура* + ε.

Эта линейная регрессия означает, что если бы температура повысилась на один градус при прочих равных условиях, то продажи холодного кофе увеличились бы на β долларов. Каждый прямоугольник на причинно-следственной диаграмме представляет столбец данных, как и в случае с симулированными данными в табл. 3.1.

Таблица 3.1. Симулированные данные, иллюстрирующие взаимосвязь на причинно-следственной диаграмме

Дата	Температура	ПродажиХолодногоКофе	β · Температура	ε = ПродажиХолодногоКофе – β · Температура
6/1/2019	71	$70 945	$71 000	$55
6/2/2019	57	$56 969	$57 000	$31
6/3/2019	79	$78 651	$79 000	–$349

Для людей, знакомых с линейно-алгебраической нотацией, приведенное выше уравнение можно переписать следующим образом:

$$\begin{pmatrix} 70,945 \\ 56,969 \\ 78,651 \end{pmatrix} = 1000 * \begin{pmatrix} 71 \\ 57 \\ 79 \end{pmatrix} + \begin{pmatrix} 55 \\ 31 \\ -349 \end{pmatrix}.$$

Причинно-следственные диаграммы и члены ошибки

Последний столбец в табл. 3.1, в котором показан остаточный или член ошибки в нашей регрессии, не имеет аналогов на причинно-следственной диаграмме рис. 3.2. Для полноты описания члены ошибки иногда изображаются на причинно-следственных диаграммах в виде пустых окружностей (рис. 3.7). Не стесняйтесь их использовать, если вы сочтете это полезным; я не буду делать этого в остальной части книги, потому что считаю, что это делает причинно-следственные диаграммы менее удобными для чтения. Я просто буду исходить из допущения, что любые связи на причинно-следственной диаграмме, которые мы хотели бы оценить, сопровождаются неявным членом ошибки.

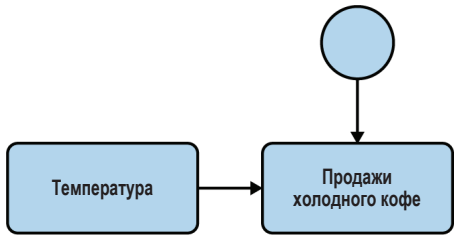

Рис. 3.7 ❖ Добавление члена ошибки на причинно-следственную диаграмму

С этой точки зрения причинно-следственные диаграммы всецело касаются данных – переменных и взаимосвязей между ними. Они сразу же обобщают на несколько причин. Давайте начертим причинно-следственную диаграмму, показывающую, что *Температура* и *ЛетнийМесяц* являются причинами *ПродажМороженого* (рис. 3.8).

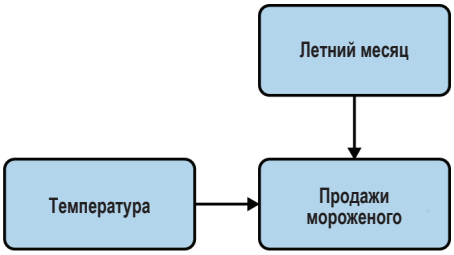

Рис. 3.8 ❖ Причинно-следственная диаграмма более чем с одной причиной

Транслирование этой причинно-следственной диаграммы в математические члены позволяет получить следующее ниже уравнение:

$$ПродажиМороженого = \beta_T \cdot Температура + \beta_C \cdot ЛетнийМесяц + \varepsilon.$$

Очевидно, что это уравнение является стандартной множественной линейной регрессией, но тот факт, что оно основано на причинно-следственной диаграмме, меняет его интерпретацию. Вне причинно-следственного каркаса мы могли бы сделать из нее единственный вывод, а именно что «повышение температуры на один градус связано с увеличением продаж мороженого на β_T долларов». Поскольку корреляция не есть каузация, было бы неправомерно делать какие-либо дальнейшие выводы. Однако когда регрессия подкрепляется причинно-следственной диаграммой, как в данном случае, мы можем сделать значительно более сильное заявление, а именно: «если эта причинно-следственная диаграмма не ошибается, то повышение температуры на один градус будет причиной увеличения продаж мороженого на β_T долларов», что и находится в фокусе внимания бизнеса.

Если у вас есть квантитативный опыт, такой как наука о данных, то у вас, возможно, возникнет соблазн сосредоточиться на взаимосвязи между причинно-следственными диаграммами и данными в ущерб соединения с поведениями. Это определенно жизнеспособный путь, и он породил целую категорию статистических моделей, именуемых *вероятностными графическими моделями*. Например, имеются и до сих пор разрабатываются алгоритмы для выявления причинно-следственных связей в данных, не опираясь на человеческий опыт или суждения. Однако эта область все еще находится в зачаточном состоянии, и при применении к реально существующим данным эти алгоритмы часто не способны выбирать между несколькими возможными причинно-следственными диаграммами, которые приводят к совершенно разным последствиям для бизнеса. Бизнес и здравый смысл

нередко способны лучше справляться с выбором наиболее разумного варианта. Поэтому я твердо убежден, что вам лучше использовать смешанный подход, показанный в рамках этой книги, и принять идею о том, что вам нужно будет использовать свое собственное суждение. *Перепасовка*, которую причинно-следственные диаграммы обеспечивают между вашей интуицией и вашими данными, – буквально во многих случаях – является тем местом, где лежат деньги.

Фундаментальные структуры причинно-следственных диаграмм

Причинно-следственные диаграммы могут принимать ошеломляющее разнообразие форм. К счастью, исследователи уже достаточно долго работают над причинно-следственной связью (т. н. каузальностью), и они навели в ней некоторый порядок:

- существуют только три фундаментальные структуры – цепочки, развилки и сталкиватели, и все причинно-следственные диаграммы могут быть представлены в виде их комбинаций;
- рассматривая причинно-следственные диаграммы так, как если бы они были генеалогическими древами, мы можем легко описывать взаимосвязи между переменными, которые находятся на диаграмме далеко друг от друга, например говоря, что одна из них является «потомком» или «дочерним элементом» другой.

И действительно, это все, что нужно сделать! Теперь мы посмотрим на эти фундаментальные структуры подробнее, и после того, как вы ознакомитесь с ними и с тем, как называть взаимосвязи между переменными, вы сможете полностью описывать любую причинно-следственную диаграмму, с которой вы будете работать.

Цепочки

Цепочка – это причинно-следственная диаграмма с тремя прямоугольниками, представляющими три переменные, и двумя стрелками, соединяющими эти прямоугольники прямой линией. Для того чтобы показать вам одну из них, мне придется ввести в наш пример с C-Mart новое лакомство, а именно всемогучий пончик. Для простоты давайте допустим, что только одна из переменных, которые мы уже видели, влияет на продажи пончиков: *ПродажиХолодногоКофе*. Тогда *Температура*, *ПродажиХолодногоКофе* и *ПродажиПончиков* связаны причинно-следственной связью (рис. 3.9).

Эта причинно-следственная диаграмма принимает вид цепочки за счет двух стрелок, которые идут «в одном направлении», т. е. первая стрелка идет от одного прямоугольника к другому, а вторая стрелка идет от этого второго прямоугольника к последнему. Указанная причинно-следственная диаграм-

ма является расширением причинно-следственной диаграммы из рис. 3.3. Она отражает тот факт, что температура является причиной продаж холодного кофе, которые, в свою очередь, являются причиной продаж пончиков.

Рис. 3.9 ❖ Причинно-следственная диаграмма в виде цепочки

Давайте дадим определение нескольким терминам, которые позволят нам характеризовать взаимосвязи между переменными. На этой диаграмме *Температура* называется родителем *ПродажХолодногоКофе*, а *ПродажиХолодногоКофе* является дочерним элементом *Температуры*. Но *ПродажиХолодногоКофе* при этом также является родителем для *ПродажПончиков*, которая является его дочерним элементом. Когда переменная имеет взаимосвязь родитель/ребенок с другой переменной, мы называем ее *прямой взаимосвязью*. Когда между ними существуют промежуточные переменные, мы называем ее *косвенной взаимосвязью*. Фактическая численность переменных, которые делают связь косвенной, как правило, не имеет значения, поэтому подсчитывать число прямоугольников, чтобы описать фундаментальную структуру взаимосвязей между ними, не требуется.

В дополнение к этому мы говорим, что переменная является *предком* другой переменной, если первая переменная является родителем другой, которая может быть родителем еще одной, и так далее, заканчивая тем, что наша вторая переменная является дочерней. В нашем примере *Температура* является предком *ПродажПончиков*, потому что она является родителем *ПродажХолодногоКофе*, которые сами по себе являются родителем *ПродажПончиков*. Вполне логично, что это делает *ПродажиПончиков* потомком *Температуры* (рис. 3.10).

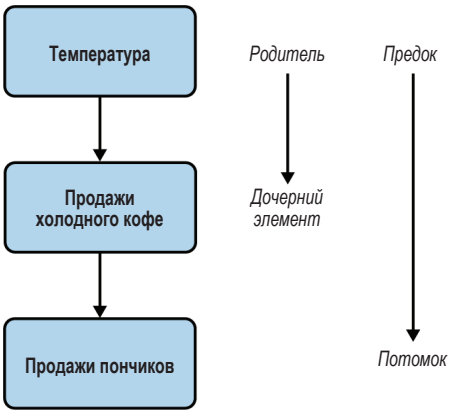

Рис. 3.10 ❖ Взаимосвязи между переменными по цепочке

В этой ситуации *ПродажиХолодногоКофе*, кроме того, являются *посредником* во взаимосвязи между *Температурой* и *ПродажамиПончиков*. Мы рассмотрим посредничество подробнее в главе 12. А пока же давайте просто отметим, что если значение посредника не меняется, то переменные, расположенные выше по цепочке, не будут влиять на переменные, расположенные ниже по цепочке, если они не связаны иным образом. В нашем примере если бы C-Mart испытывала нехватку холодного кофе, то мы ожидали бы, что в течение всего периода этой нехватки изменения в температуре не имеют никакого эффекта на продажи пончиков.

Сворачивание цепочек

Приведенная выше причинно-следственная диаграмма транслируется в следующие ниже регрессионные уравнения:

$$ПродажиХолодногоКофе = \beta_T \cdot Температура;$$

$$ПродажиПончиков = \beta_I \cdot ПродажиХолодногоКофе.$$

Мы можем заменить *ПродажиХолодногоКофе* его выражением во втором уравнении:

$$ПродажиПончиков = \beta_I \cdot (\beta_T Температура) = (\beta_I \beta_T) Температура.$$

Но $\beta_I\beta_T$ – это просто произведение двух постоянных коэффициентов, поэтому мы можем рассматривать его как новый коэффициент сам по себе: $ПродажиПончиков = \beta_T \cdot Температура$. Нам удалось выразить *ПродажиПончиков* в виде линейной функции *Температуры*, которая, в свою очередь, может быть транслирована в причинно-следственную диаграмму (рис. 3.11).

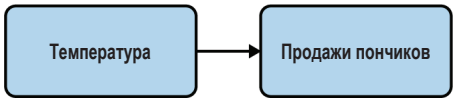

Рис. 3.11 ❖ Сворачивание причинно-следственной диаграммы в другую причинно-следственную диаграмму

Здесь мы свернули цепочку, то есть удалили переменную в середине и заменили ее стрелкой, идущей от первой переменной к последней. Поступая таким образом, мы практически упростили нашу исходную причинно-следственную диаграмму, чтобы сосредоточиться на интересующей нас взаимосвязи. Это бывает полезно, когда последняя переменная в цепочке является интересующей нас деловой метрикой, а по первой из них можно предпринимать действия. В некоторых обстоятельствах, например, нас могут интересовать промежуточные взаимосвязи между *Температурой* и *ПродажамиХолодногоКофе*, а также между *ПродажамиХолодногоКофе* и *ПродажамиПончиков* для управления ценообразованием или рекламными акциями. В других обстоятельствах нас могла бы интересовать только взаимосвязь

между *Температурой* и *ПродажамиПончиков*, например для планирования материальных запасов.

Здесь также применимо *свойство транзитивности* линейной алгебры: если *ПродажиПончиков* стали причиной еще одной переменной, то эта цепочка также может быть свернута вокруг *ПродажПончиков* и т. д.

Расширение цепочек

Очевидно, что операцию сворачивания можно обратить вспять: мы можем перейти от нашей последней причинно-следственной диаграммы к предыдущей, добавив в середине переменную *ПродажиХолодногоКофе*. В более общем случае мы говорим, что расширяем (или разворачиваем) цепочку всякий раз, когда вводим промежуточную переменную между двумя переменными, соединенными в данный момент стрелкой. Например, предположим, что мы начнем со взаимосвязи между *Температурой* и *ПродажамиПончиков* (рис. 3.11). Эта причинно-следственная связь транслируется в уравнение *ПродажиПончиков* = $\beta \cdot$ *Температура*. Давайте допустим, что *Температура* влияет на *ПродажиПончиков* только через *ПродажиХолодногоКофе*. Мы можем добавить эту переменную на нашу причинно-следственную диаграмму, что возвращает нас к исходной причинно-следственной диаграмме рис. 3.8 (рис. 3.12).

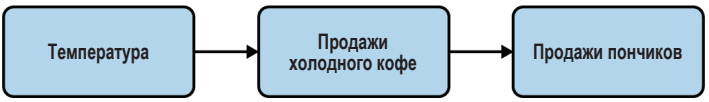

Рис. 3.12 ❖ Расширение причинно-следственной диаграммы в еще одну причинно-следственную диаграмму

Расширение цепочек бывает полезно для более глубокого понимания того, что происходит в данной ситуации. Например, предположим, что температура повысилась, но продажи пончиков не выросли. Для этого могут быть две потенциальные причины:

- прежде всего повышение температуры не увеличило продажи холодного кофе, возможно потому, что менеджер магазина был более агрессивен с кондиционером. Другими словами, первая стрелка на рис. 3.11 исчезла или ослабла;
- в качестве альтернативы: повышение температуры, напротив, увеличило продажи холодного кофе, но увеличение продаж холодного кофе не увеличило продажи пончиков, например потому, что вместо этого люди покупают недавно появившееся печенье. Другими словами, на рис. 3.11 первая стрелка не изменилась, но вторая исчезла или ослабла.

В зависимости от истинности одной из альтернатив вы можете предпринять самые разные корректирующие действия – отключить кондиционер либо изменить цену печенья. Во многих случаях рассмотрение переменной в середине цепочки, а именно посредника, позволит вам принимать более эффективные решения.

Поскольку цепочки можно сворачивать либо расширять по желанию, в целом мы явно не указываем, когда это было сделано. Тут всегда исходят из того, что любая стрелка потенциально может быть расширена, чтобы высветить промежуточную переменную на этом пути.

Из этого также следует, что определение упомянутых ранее «прямых» и «косвенных» взаимосвязей относится к конкретному представлению причинно-следственной диаграмы: когда вы сворачиваете цепочку, две переменные, которые имели косвенную связь, теперь имеют прямую связь.

Развилки

Когда переменная является причиной двух или более эффектов, взаимосвязь создает развилку. Температура является причиной как *ПродажХолодного-Кофе*, так и *ПродажМороженого*, а изображение этой развилки показано на рис. 3.13.

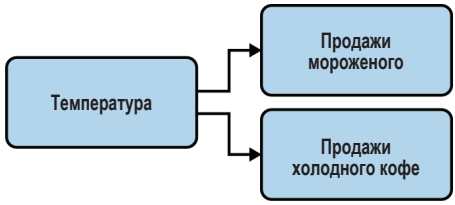

Рис. 3.13 ❖ Развилка между тремя переменными

Приведенная выше причинно-следственная диаграмма показывает, что *Температура* влияет как на *ПродажиМороженого*, так и на *ПродажиХолодногоКофе*, но что они не имеют причинно-следственной связи друг с другом. Если на улице жарко, то спрос как на мороженое, так и на холодный кофе увеличивается, но покупка одного не вызывает у вас желания купить другое и не снижает вероятности покупки другого.

Такая ситуация, когда две переменные имеют общую причину, встречается очень часто, но также является потенциально проблематичной, поскольку создает корреляцию между этими двумя переменными. Вполне резонно, что когда будет жарко, мы увидим увеличение продаж и того, и другого, а когда будет холодно, меньше людей захотят и того, и другого. Линейная регрессия, предсказывающая *ПродажиМороженого* из *ПродажХолодногоКофе*, была бы в известной степени предсказывающей, но здесь корреляция не равна каузации, и поскольку мы знаем, что причинно-следственное воздействие равно 0, предоставляемый моделью коэффициент не будет точным.

На эту взаимосвязь можно взглянуть и по-другому, как на то, что если бы C-Mart испытывала нехватку холодного кофе со льдом, то мы бы не ожидали увидеть изменения в продажах мороженого. В более общем случае было бы лишь небольшим преувеличением сказать, что развилки являются одним из главных корней зла в мире анализа данных. Всякий раз, когда мы наблюдаем корреляцию между двумя переменными, которая не отражает прямой

причинно-следственной взаимосвязи между ними (т. е. ни одна из них не является причиной другой), то чаще всего это происходит потому, что у них есть общая причина. С этой точки зрения, одно из главных преимуществ использования причинно-следственных диаграмм заключается в том, что они способны очень четко и интуитивно понятно показывать, что конкретно в таких случаях происходит и как это исправлять.

Развилки также типичны для ситуаций, когда мы смотрим на *демографические переменные*: возраст, пол и место жительства, – все они являются причинами самых разных других переменных, которые могут быть или не быть причинами друг друга. Демографическую переменную, такую как возраст, можно изобразить как располагающуюся у корня развилки, имеющей много ответвлений.

Когда у вас есть развилка в середине причинно-следственной диаграммы, иногда возникает вопрос о том, сможете ли вы по-прежнему сворачивать цепочку вокруг нее. Например, предположим, что мы заинтересованы в анализе взаимосвязи между *ЛетнимМесяцем* и *ПродажамиХолодногоКофе*, используя причинно-следственную диаграмму рис. 3.14.

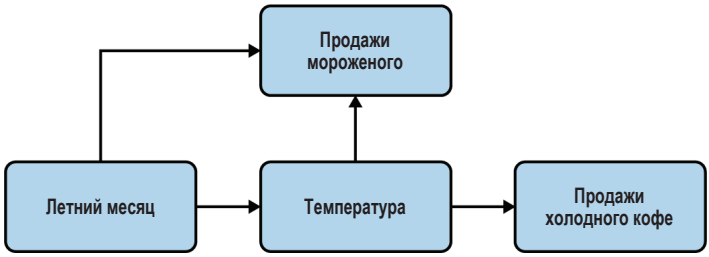

Рис. 3.14 ❖ Причинно-следственная диаграмма с развилкой и цепочкой

На этой причинно-следственной диаграмме есть развилка между *ЛетнимМесяцем*, с одной стороны, и *ПродажамиМороженого* и *Температурой* – с другой, но также есть цепочка *ЛетнийМесяц* → *Температура* → *ПродажиХолодногоКофе*. Сможем ли мы свернуть эту цепочку?

В данном случае да. В главе 5 мы увидим, как определять, когда именно переменная является спутывающим фактором взаимосвязи; здесь достаточно будет сказать, что *ПродажиМороженого* не является спутывающим фактором взаимосвязи между *ЛетнимМесяцем* и *ПродажамиХолодногоКофе*, которая нас интересует. Следовательно, мы можем упростить причинно-следственную диаграмму (рис. 3.15).

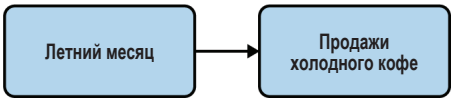

Рис. 3.15 ❖ Свернутая версия предыдущей причинно-следственной диаграммы

Схожим образом, если бы нас интересовала взаимосвязь между *ЛетнимМесяцем* и *ПродажамиМороженого* рис. 3.14, то мы могли бы пренебречь *ПродажамиХолодногоКофе*, но не *Температурой*.

Поскольку развилки столь важны для причинно-следственного анализа, иногда у нас возникнет желание их изображать, даже если мы не знаем общей причины. В этом случае мы будем представлять неизвестную развилку двуглавой стрелкой (рис. 3.16).

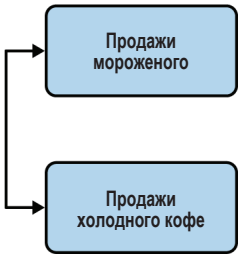

Рис. 3.16 ❖ Развилка с неизвестной общей причиной

Двуглавая стрелка при этом выглядит так, как будто две переменные являются причинами друг друга. Это сделано специально, и мы будем использовать двуглавую стрелку, также когда мы наблюдаем корреляцию между двумя переменными, но мы не знаем, какая из них является причиной. Отсюда двуглавая стрелка охватывает три возможные причины, по которым две переменные A и B будут казаться коррелированными: A является причиной B, B является причиной A и/или A и B имеют общую причину. Иногда мы будем использовать двуглавую стрелку в качестве местозаполнителя до тех пор, пока не выясним истинную причину; если нас не волнует причина, то мы можем просто оставить двуглавую стрелку на окончательной причинно-следственной диаграмме.

Сталкиватели

Очень немногие вещи в мире имеют только одну причину. Когда две или более переменных являются причиной одного и того же результата, то взаимосвязь создает *сталкивателя* (collider). Поскольку в концессионном киоске C-Mart продаются только два вида мороженого, шоколадное и ванильное, причинно-следственная диаграмма, представляющая вкусовое пристрастие и поведение при покупке мороженого, покажет, что пристрастие к любому из них было бы причиной прошлых покупок мороженого в киоске (рис. 3.17).

Сталкиватели являются обычным явлением, и при этом они иногда создают трудности в анализе данных. Сталкиватель в некотором смысле противоположен развилке, и проблемы с ними, в свою очередь, имеют симметричный характер: развилка проблематична, если мы не отслеживаем общую причину, тогда как сталкиватель является проблемой, если мы, напротив,

отслеживаем общее следствие (общий эффект). Мы разведаем эти вопросы подробнее в главе 5.

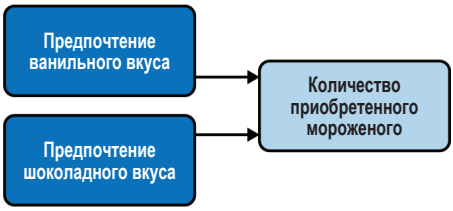

Рис. 3.17 ❖ Причинно-следственная диаграмма в виде сталкивателя

Резюмируя этот раздел: цепочки, развилки и сталкиватели представляют собой три единственно возможных пути соотнесения трех переменных друг с другом на причинно-следственной диаграмме. Однако они не исключают друг друга, и на самом деле довольно часто встречаются три переменные, которые демонстрируют все три структуры одновременно, как это было в нашем самом первом примере (рис. 3.18).

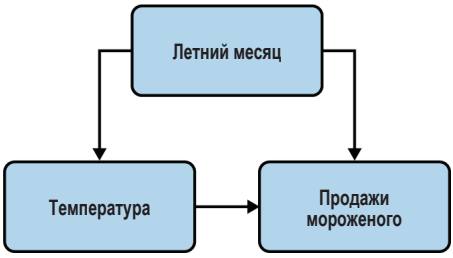

Рис. 3.18 ❖ Причинно-следственная диаграмма с тремя переменными, содержащая цепочку, развилку и сталкиватель одновременно

Здесь *ЛетнийМесяц* влияет на *ПродажиМороженого*, а также на *Температуру*, которая сама влияет на *ПродажиМороженого*. Учавствующие в игре причинно-следственные связи достаточно просты и понятны, но этот граф также содержит все три типа базовых взаимосвязей:

- цепочка: *ЛетнийМесяц* → *Температура* → *ПродажиМороженого*;
- развилка, в которой *ЛетнийМесяц* является причиной как *Температуры*, так и *ПродажМороженого*;
- сталкиватель, в котором *ПродажиМороженого* обусловливаются как *Температурой*, так и *ЛетнимМесяцем*.

В подобной ситуации следует отметить еще одну вещь, а именно переменные имеют более чем одну взаимосвязь друг с другом. Например, *ЛетнийМесяц* является родителем *ПродажМороженого*, потому что есть стрелка, которая проходит непосредственно от первого ко вторым (прямая взаимосвязь); но в то же время *ЛетнийМесяц* также косвенно является предком *ПродажМо-*

роженого по цепочке *ЛетнийМесяц* → *Температура* → *ПродажиМороженого* (косвенная взаимосвязь). Поэтому хорошо видно, что они не являются эксклюзивными!

Хотя причинно-следственная диаграмма всегда состоит из этих трех структур, она не является статичной. Причинно-следственную диаграмму можно преобразовывать путем модифицирования самих переменных, а также их взаимосвязей, как мы сейчас увидим.

Распространенные преобразования причинно-следственных диаграмм

Цепочки, развилки и сталкиватели принимают переменные на причинно-следственной диаграмме как данность. Но точно так же, как цепочка может сворачиваться или расширяться, переменные сами по себе могут нарезаться или агрегироваться, чтобы «увеличивать и уменьшать» масштаб конкретных поведений и категорий. Мы также можем принимать решение модифицировать стрелки, например когда мы встречаем в прочих условиях непрослеживаемые циклы.

Нарезка/дезагрегирование переменных

Развилки и сталкиватели часто создаются, когда вы разрезаете или дезагрегируете переменную, чтобы выявить ее компоненты. В приведенном ранее примере мы рассмотрели взаимосвязь между *Температурой* и *ПродажамиПончиков*, где посредником были *ПродажиХолодногоКофе* (рис. 3.19).

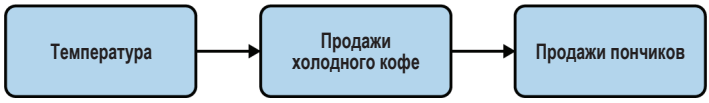

Рис. 3.19 ❖ Цепочка, которую мы разрежем/дезагрегируем

Но, возможно, мы хотим разрезать *ПродажиХолодногоКофе* в разбивке по типам, чтобы лучше понять динамику спроса. Это то, что я подразумеваю под «нарезкой» переменной. Она разрешена в соответствии с правилами линейной алгебры, потому что мы можем выразить суммарный объем продаж холодного кофе как сумму продаж в разбивке по типу, скажем американо и латте:

ПродажиХолодногоКофе = *ПродажиХолодногоАмерикано* + *ПродажиХолодногоЛатте*.

Наша причинно-следственная диаграмма теперь стала бы как на рис. 3.20, с развилкой слева и сталкивателем справа.

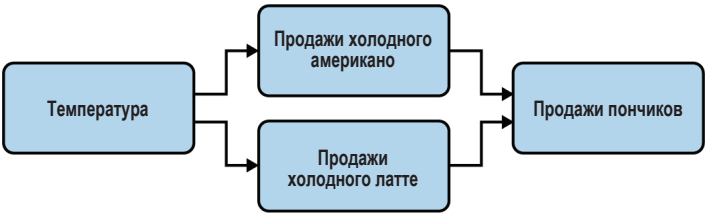

Рис. 3.20 ❖ Цепочка, в которой посредник был разложен на срезы

Каждый срез переменной теперь будет иметь свое собственное уравнение:

$$ПродажиХолодногоАмерикано = \beta_{TA} \cdot Температура;$$

$$ПродажиХолодногоЛатте = \beta_{TL} \cdot Температура.$$

Поскольку эффект *Температуры* полностью опосредуется нашими срезами *ПродажХолодногоКофе*, мы можем создать единую множественную регрессию для *ПродажПончиков* следующим образом:

$$ПродажиПончиков = \beta_{IA} \cdot ПродажиХолодногоАмерикано + \beta_{IL} \cdot ПродажиХолодногоЛатте.$$

Это позволит вам точнее понимать, что происходит – следует ли вам планировать одинаковый рост продаж в обоих типах при повышении температуры? Оказывают ли они оба одинаковый эффект на *ПродажиПончиков* или вам следует попытаться отдать предпочтение одному из них?

Агрегирование переменных

Как вы, возможно, догадались, нарезка переменных может поворачиваться вспять, и в более общем случае мы можем агрегировать переменные, которые имеют одни и те же причины и следствия. Это может использоваться для агрегирования и дезагрегирования анализа данных в разбивке по продуктам, регионам, направлениям деятельности и т. д. Но это также может использоваться свободнее для представления важных причинно-следственных факторов, которые не определены прецизионно точно. Например, предположим, что *Возраст* и *Пол* влияют как на *ПредпочтениеВанильногоВкуса*, так и на склонность покупать мороженое в концессионных киосках C-Mart, *ПриобрелМороженое* (рис. 3.21).

Поскольку *Возраст* и *Пол* имеют одинаковые причинно-следственные связи, их можно агрегировать в переменную *ДемографическиеХарактеристики* (рис. 3.22).

В этом случае у нас, очевидно, нет ни одного столбца в данных под названием «Демографические характеристики» или «Демография»; мы используем эту переменную на нашей причинно-следственной диаграмме просто в качестве укороченного обозначения для множества переменных, которые мы,

возможно, захотим или не захотим разведать подробнее позже. Допустим, мы хотим провести A/B-тест и понять имеющиеся причинно-следственные связи. Как мы увидим позже, рандомизация позволяет нам контролировать демографические факторы, чтобы нам не пришлось включать их в наш анализ, но мы, возможно, захотим включить их в наше описание ситуации без рандомизации. При необходимости мы всегда можем расширить нашу диаграмму, чтобы точно представить соответствующие демографические переменные. Однако помните, что любая переменная может быть разрезана, но агрегировать можно только переменные, имеющие одинаковые прямые и косвенные связи.

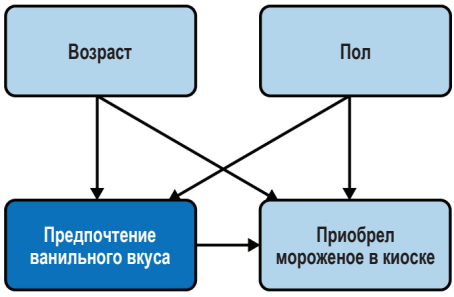

Рис. 3.21 ❖ Возраст и пол указаны отдельно

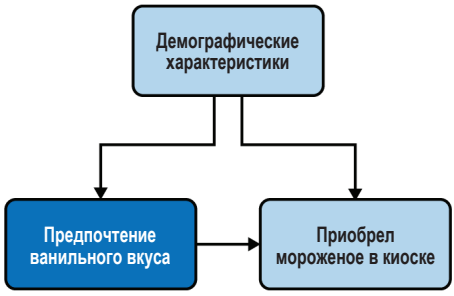

Рис. 3.22 ❖ Причинно-следственная диаграмма, где *Возраст* и *Пол* агрегированы в единую переменную

Техническое более глубокое погружение: агрегирование переменных и линейная алгебра

Использование сводной демографической переменной, возможно, покажется сомнительной ловкостью рук, но на самом деле это совершенно разумно с точки зрения линейной алгебры. Если вы поверите мне на слово или ваша линейная алгебра проржавела насквозь, то вы можете просто пропустить эту врезку и не беспокоиться об этом. Но если вы хотите проверить математику, то вот она.

Мы можем трактовать наши переменные как векторы:

$$\textit{ПредпочтениеВанильногоВкуса} = \begin{pmatrix} 3 \\ 15 \\ 8 \\ \vdots \\ 17 \end{pmatrix}, \textit{Возраст} = \begin{pmatrix} 23 \\ 78 \\ 52 \\ \vdots \\ 41 \end{pmatrix} \text{ и } \textit{Пол} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \end{pmatrix}.$$

Теперь у нас есть соответствующее уравнение:

$$\textit{ПредпочтениеВанильногоВкуса} = \beta_A \cdot \textit{Возраст} + \beta_G \cdot \textit{Пол}.$$

Но мы также можем скрепить наши векторы в матрице (технически именно так традиционно решаются линейные регрессионные модели):

$$\textit{Демография} = (\textit{Возраст} \quad \textit{Пол}) = \begin{pmatrix} 23 & 0 \\ 78 & 1 \\ 52 & 1 \\ \vdots & \vdots \\ 41 & 0 \end{pmatrix}.$$

Это позволяет нам переписать предыдущее уравнение следующим образом:

$$\textit{ПредпочтениеВанильногоВкуса} = (\textit{Возраст} \quad \textit{Пол}) \cdot \begin{pmatrix} \beta_a \\ \beta_g \end{pmatrix} = \textit{Демография} \cdot \vec{\beta}, \text{ где } \vec{\beta} = \begin{pmatrix} \beta_a \\ \beta_g \end{pmatrix}.$$

Теперь мы выразили *ПредпочтениеВанильногоВкуса* как линейную функцию *Демографии* в векторной нотации. Другими словами, до тех пор, пока мы агрегируем только те переменные, которые имеют точно такие же взаимосвязи с другими переменными, мы находимся на твердой математической основе.

А что делать с циклами?

В трех фундаментальных структурах, которые мы увидели, между двумя заданными прямоугольниками была только одна стрелка. В более общем случае, следуя по направлению стрелок (например, A → B → C → A), невозможно было достигать одной и той же переменной дважды. Переменная может быть следствием одной переменной и причиной другой, но она не может быть одновременно причиной и следствием одной переменной.

Однако в реальной жизни мы часто встречаем переменные, которые влияют друг на друга причинно-следственным образом. Этот тип причинно-следственной диаграммы называется циклом. Циклы могут возникать по разным причинам; двумя наиболее распространенными в анализе поведенческих данных являются эффекты замещения и циклы обратной связи. В частности, существуют некоторые обходные пути, которые будут позволять вам справляться с циклами, когда вы будете с ними сталкиваться.

Понимание циклов: эффекты замещения и циклы обратной связи

Эффекты замещения являются краеугольным камнем экономической теории: потребители могут заменять один продукт другим в зависимости от доступности и цены продуктов, а также желания потребителей их разнообразить. Например, покупатели, приходящие в концессионный магазин C-Mart, возможно, будут выбирать между холодным кофе и горячим кофе, основываясь не только на температуре, но и на специальных акциях и на том, как часто они пили кофе на этой неделе. Следовательно, существует причинно-следственная связь между покупкой холодного кофе и покупкой горячего кофе, а также еще одна причинно-следственная связь в противоположном направлении (рис. 3.23).

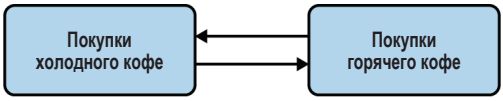

Рис. 3.23 ❖ Причинно-следственная диаграмма с эффектом замещения, генерирующего цикл

Следует отметить, что направление стрелок показывает направление причинно-следственной связи (что является причиной, а что – следствием), а не знак следствия. Во всех причинно-следственных диаграммах, которые мы рассматривали ранее, переменные имели положительную связь, когда увеличение в одной становилось причиной увеличения в другой. Взаимосвязи являются отрицательными в том случае, когда увеличение в одной переменной будет причиной уменьшения в другой. Знак следствия не имеет значения для причинно-следственных связей, и регрессия сможет правильно отсортировывать знак коэффициента, при условии если вы правильно выявляете соответствующие причинно-следственные связи.

Еще одним распространенным циклом является *цикл обратной связи*, когда человек изменяет свое поведение в ответ на изменения в окружающей среде. Например, менеджер магазина C-Mart, возможно, присматривает за длиной очередей и открывает новые линии обслуживания, если существующие становятся слишком длинными, чтобы покупатели не отказывались ждать и не уходили (рис. 3.24).

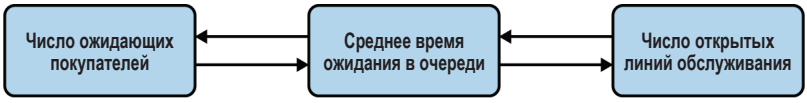

Рис. 3.24 ❖ Пример цикла обратной связи, генерирующего цикл

Управление циклами

Циклы отражают ситуации, которые часто сложно изучать и которыми сложно управлять, вследствие чего для этой цели выросла целая область исследований, именуемая *системным мышлением*[1]. Сложные математические методы, такие как моделирование структурных уравнений, были разработаны для точного описания циклов, но их анализ вывел бы нас за рамки этой книги. Однако я был бы неосторожен, если бы не дал вам никакого решения, поэтому я приведу два эмпирических правила, которые должны уберечь вас от увязывания в циклах.

Первое эмпирическое правило состоит в том, чтобы обращать пристальное внимание на выбор правильного момента времени. Почти во всех случаях требуется некоторое время, чтобы одна переменная повлияла на другую, из чего вытекает, что вы можете «прерывать цикл» и превращать его в «ациклическую» причинно-следственную диаграмму, т. е. диаграмму без циклов (которую затем вы сможете проанализировать с помощью инструментов, представленных в этой книге), глядя на свои данные на более гранулярном уровне времени. Например, предположим, что менеджеру магазина требуется 15 минут, чтобы отреагировать на увеличение времени ожидания, открытием новых линий обслуживания, и точно так же покупателям требуется 15 минут, чтобы скорректировать свое восприятие времени ожидания. В этом случае, прояснив временной порядок вещей, мы можем разрезать переменную времени ожидания на нашей причинно-следственной диаграмме (рис. 3.25).

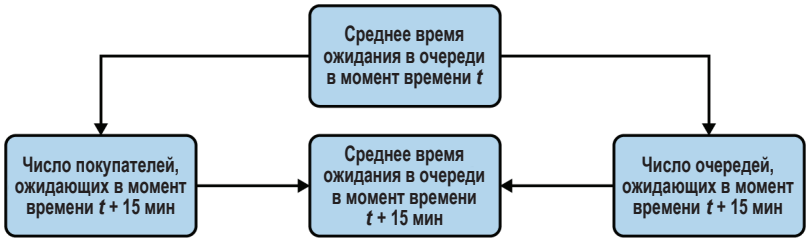

Рис. 3.25 ❖ Разбиение цикла обратной связи на временные приращения

Я буду объяснять эту причинно-следственную диаграмму по одной части за раз. Слева у нас есть стрелка от среднего времени ожидания к числу ожидающих покупателей:

$$ЧислоОжидающихПокупателей(t + 15 \text{ мин}) = \beta_1 \cdot СреднееВремяОжидания(t).$$

Это означает, что число покупателей, ожидающих, скажем, в 9:15 утра, будет выражено как функция среднего времени ожидания в 9:00 утра. Тогда

[1] Заинтересованным читателям предлагается обратиться к книге «Мышление в системах: руководство для начинающих» Медоуза и Райта (Thinking in Systems: A Primer, Meadows and Wright, 2008), а также к книге «Пятая дисциплина: искусство и практика обучающейся организации» Сенге (The Fifth Discipline: The Art & Practice of the Learning Organization, Senge, 2010).

число покупателей, ожидающих в 9:30 утра, будет иметь такую же связь со средним временем ожидания в 9:15 утра и т. д.

Схожим образом, справа у нас есть стрелка от среднего времени ожидания к числу открытых линий:

$$\textit{ЧислоОткрытыхЛиний}(t + 15 \text{ мин}) = \beta_2 \cdot \textit{СреднееВремяОжидания}(t).$$

Это означает, что число линий, открытых в 9:15 утра, будет выражено как функция среднего времени ожидания в 9:00 утра. Тогда число линий, открытых в 9:30 утра, будет иметь такую же связь со средним временем ожидания в 9:15 утра и т. д.

Затем в середине у нас есть причинно-следственные стрелки от числа ожидающих покупателей и от числа открытых линий к среднему времени ожидания. Ради простоты, исходя из наличия здесь линейных связей, это транслировалось бы в следующее ниже уравнение:

$$\textit{СреднееВремяОжидания}(t) = \beta_3 \cdot \textit{ЧислоОжидающихПокупателей}(t) + \beta_4 \cdot \textit{ЧислоОткрытыхЛиний}(t).$$

На самом деле допущение о линейности связей в данном случае вряд ли будет верным. Были разработаны специальные модели для очередей или для переменных времени-до-события (например, анализ выживаемости). Указанные модели являются частью более широкой категории обобщенных линейных моделей, и поэтому хорошим эмпирическим правилом является то, что для наших целей они будут вести себя как логистические регрессии.

Из этого вытекает, что среднее время ожидания для покупателей, достигающих кассовых линий в 9:15 утра, зависит от числа уже присутствующих покупателей и числа кассовых линий, открытых в 9:15 утра. Тогда среднее время ожидания для покупателей, достигающих кассовых линий в 9:30 утра, зависит от числа уже присутствующих покупателей и числа кассовых линий, открытых в 9:30 утра, и т. д.

Разбивая переменные на временные приращения, мы смогли создать причинно-следственную диаграмму, на которой нет цикла в строгом смысле этого слова. Мы можем оценивать три приведенных выше линейно-регрессионных уравнения без введения какой-либо круговой логики.

Второе эмпирическое правило для работы с циклами состоит в том, чтобы упрощать вашу причинно-следственную диаграмму и оставлять стрелки только вдоль причинно-следственной связи, которая вас интересует больше всего. Эффектов обратной связи (когда переменная влияет на переменную, которая только что повлияла на нее), как правило, меньше, а часто и намного меньше, чем первого эффекта, и их можно игнорировать в первом приближении.

В нашем примере с холодным и горячим кофе вы, возможно, будете обеспокоены тем, что увеличение в продажах холодного кофе, когда на улице жарко, снизит продажи горячего кофе; это беспокойство является разумным, и его следует проанализировать. Однако маловероятно, что снижение в продажах горячего кофе, в свою очередь, станет спусковым крючком для дальнейшего

увеличения в продажах холодного кофе, и вы можете игнорировать этот эффект обратной связи на своей причинно-следственной диаграмме (рис. 3.26).

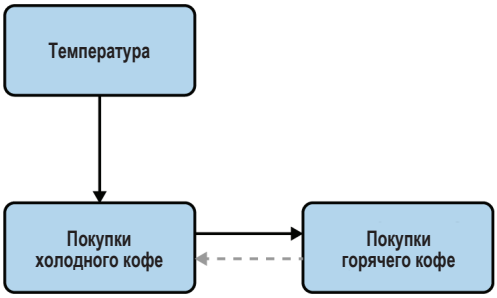

Рис. 3.26 ❖ Упрощение причинно-следственной диаграммы за счет пренебрежения некоторыми связями

На рис. 3.26 мы удаляем стрелку от покупок горячего кофе к покупкам холодного кофе и игнорируем эту взаимосвязь в качестве разумного приближения.

Еще раз, это всего лишь эмпирическое правило и определенно не общее приглашение к игнорированию циклов и эффектов обратной связи. На вашей законченной причинно-следственной диаграмме они должны быть представлены полностью, чтобы направлять будущие аналитические расчеты.

Пути

Рассмотрев различные пути взаимодействия переменных, мы можем теперь ввести последнее понятие, которое охватывает их все: *пути*. Мы говорим, что между двумя переменными существует путь, если между ними есть стрелки, независимо от направления стрелок, и если ни одна переменная не появляется на этом пути дважды. Давайте посмотрим, как это выглядит на причинно-следственной диаграмме, которую мы встречали ранее (рис. 3.27).

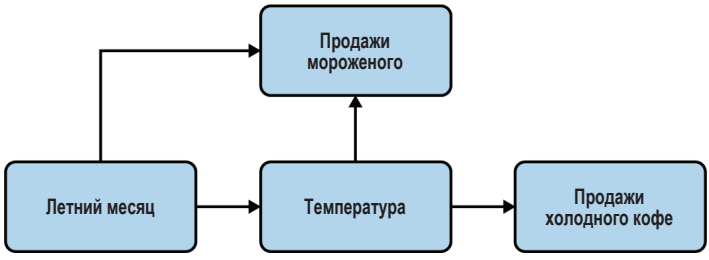

Рис. 3.27 ❖ Пути в причинно-следственной диаграмме

На приведенной выше причинно-следственной диаграмме есть два пути от *ЛетнегоМесяца* к *ПродажамХолодногоКофе*:

- один путь по цепочке *ЛетнийМесяц* → *Температура* → *ПродажиХолодногоКофе*;
- второй путь через *ПродажиМороженого*, *ЛетнийМесяц* → *ПродажиМороженого* ← *Температура* → *ПродажиХолодногоКофе*.

Из этого следует, что цепочка является путем, но им же является и развилка или сталкиватель! Также обратите внимание на то, что при этом два разных пути между двумя переменными могут иметь совместные стрелки, если между ними есть хотя бы одно различие, как в данном случае: стрелка от *Температуры* к *ПродажамХолодногоКофе* появляется в обоих путях.

Однако следующее ниже не является допустимым путем между *Температурой* и *ПродажамиХолодногоКофе*, поскольку *Температура* появляется дважды:

- *Температура* ← *ЛетнийМесяц* → *ПродажиМороженого* ← *Температура* → *ПродажиХолодногоКофе*.

Одним из последствий этих определений является то, что если на причинно-следственной диаграмме вы выбираете две разные переменные, то между ними всегда есть по меньшей мере один путь. Определение путей может показаться настолько широким, что оно бесполезно, но, как мы увидим в главе 5, пути на самом деле будут играть важную роль в выявлении спутывающих факторов на причинно-следственной диаграмме.

Выводы

Корреляция не есть каузация, потому что спутывающие факторы могут привносить систематическое смещение в наши аналитические расчеты. К сожалению, как мы увидели на примерах, простого включения в регрессию всех до единой располагаемых переменных (вместе с кухонной раковиной в придачу) для распутывания недостаточно. Хуже того, контролирование на неправильных переменных может приводить к ложным корреляциям и создавать новые систематические смещения.

В качестве первого шага на пути к несмещенной регрессии я представил инструмент, именуемый причинно-следственными диаграммами. Причинно-следственные диаграммы могут быть наилучшим аналитическим инструментом, о котором вы никогда не слышали. Они могут использоваться для представления наших интуитивных представлений о причинно-следственных связях в реальном мире, а также причинно-следственных связей между переменными в наших данных; но они наиболее эффективны в качестве моста между ними, позволяя нам соединять нашу интуицию и экспертные знания с наблюдаемыми данными, и наоборот.

Причинно-следственные диаграммы бывают извилистыми и сложными, но они основаны на трех простых строительных блоках: цепочках, развилках и сталкивателях. Кроме того, их можно сворачивать или расширять, разрезать или агрегировать в соответствии с простыми правилами, которые согласуются с линейной алгеброй. Если вы хотите узнать о причинно-след-

ственных диаграммах больше, то книга Перл и Маккензи (2018) представляет собой очень удобное для чтения и приятное введение.

Полная мощь причинно-следственных диаграмм станет очевидной в главе 5, в которой мы увидим, что они позволяют нам оптимально справляться со спутывающими факторами в регрессии, даже с неэкспериментальными данными. Но причинно-следственные диаграммы полезны и в более широком смысле, помогая нам лучше думать о данных. В следующей главе, когда мы приступим к очистке и подготовке данных для анализа, они позволят нам смягчать систематические смещения в наших данных до проведения любого анализа. Это даст вам возможность ознакомиться с причинно-следственными диаграммами подробнее в простой обстановке.

Глава 4

Строительство причинно-следственных диаграмм с нуля

В этом месте вам, возможно, будет интересно узнать, откуда взялась [причинно-следственная диаграмма]. Отличный вопрос. Возможно, это наиважнейший из вопросов. [Причинно-следственная диаграмма] предположительно должна быть теоретическим представлением современных знаний о явлениях, которые вы изучаете. Это то, что эксперт назвал бы самой вещью, и это экспертное знание исходит из самых разных источников. Примеры включают экономическую теорию, другие научные модели, беседы с экспертами, ваши собственные наблюдения и опыт, литературные обзоры, а также вашу собственную интуицию и гипотезы.

– Скотт Каннингем, «Причинно-следственный вывод: микстейп» (2021)

Наша цель в этой книге неизменно состоит в том, чтобы измерять воздействие одной переменной на другую, которое можно представить в виде «стартовой» причинно-следственной диаграммы (рис. 4.1).

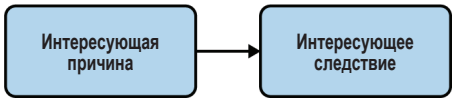

Рис. 4.1 ❖ Самая простая причинно-следственная диаграмма из возможных

После того как вы начертили эту связь, встает вопрос: что делать дальше? Как узнать, какие другие переменные вы должны включать или не включать? Многие авторы говорят, что вы должны опираться на экспертные знания, и это нормально, если вы работаете в такой устоявшейся области, как экономика или эпидемиология. Но моя точка зрения в этой книге заключается в том, что в своей организации вы, скорее всего, являетесь «бихевиористом

под номером один», и поэтому вам нужно уметь начинать с табула раза (или чистого листа).

В этой главе я опишу рецепт перевода вас с базовой причинно-следственной диаграммы рис. 4.1 на работоспособную. Проходя через этот процесс, пожалуйста, имейте в виду нашу конечную цель – понять, что именно движет поведениями, чтобы иметь возможность делать релевантные и практические выводы для нашего бизнеса. Наша целевая задача совсем не в том, чтобы создать полное и прецизионно точное знание всего мира. Сокращения и приближения – это часть честной игры, и все должно оцениваться в соответствии с одним-единственным критерием: помогает ли это мне в достижении моей деловой цели?

В дополнение к этому рецепт, который я изложу, не является механическим алгоритмом, которому вы могли бы слепо следовать, чтобы добираться до нужной причинно-следственной диаграммы. Напротив, будут иметь решающее значение деловая хватка, здравый смысл и понимание данных. Мы будем ходить взад и вперед между нашим качественным пониманием текущей причинно-следственной ситуации и количественными связями, присутствующими в данных, сверяя одно с другим до тех пор, пока не почувствуем, что получили удовлетворительный результат. Здесь слово «удовлетворительный» имеет важный смысл: в прикладной обстановке вы обычно не сможете сказать своему менеджеру, что дадите ему правильный ответ через три года. Вам нужно дать ему как можно менее плохой ответ в короткий срок, при этом планируя работу по сбору данных, которая будет улучшать ваш ответ с годами.

В следующем далее разделе я представлю деловую задачу этой главы и соответствующие интересующие нас переменные. Затем мы будем поступательно выстраивать соответствующую причинно-следственную диаграмму по следующему ниже рецепту, по одному разделу в каждом шаге:

- выявить переменные, которые потенциально могут / должны быть включены в причинно-следственную диаграмму;
- определить необходимость включения переменных;
- повторять процесс итеративно согласно потребности;
- упростить диаграмму.

Давайте начнем!

Деловая задача и настройка данных

В этом разделе мы будем работать с набором реально существующих данных о бронировании гостиничных номеров в двух гостиницах, расположенных в одном городе[1]. Данные и пакеты, которые мы будем использовать, описаны

[1] *Нуно Антонио, Ана де Алмейда и Луис Нуньес.* Наборы данных о спросе на бронирование гостиниц. Данные в кратком изложении (Nuno Antonio, Ana de Almeida, and Luis Nunes, *Hotel booking demand data sets, Data in Brief*), https://doi.org/10.1016/j.dib.2018.11.126.

в следующем ниже подразделе, а затем мы углубимся, чтобы понять интересующую нас взаимосвязь.

Данные и пакеты

Папка этой главы в репозитоии на GitHub[1] содержит CSV-файл *chap4-hotel_booking_case_study.csv* с переменными, указанными в табл. 4.1.

Таблица 4.1. Переменные в файле данных

Имя переменной	Описание переменной
NRDeposit, *NRD* (НевозвратныйДепозит)	Двоичная 0/1, имела ли бронь невозвратный депозит
IsCanceled (БроньАннулировалась)	Двоичная 0/1, была бронь аннулирована или нет
DistributionChannel (КаналРаспределения)	Категориальная переменная со значениями «Прямой», «Корпоративный», «Турагент/турорганизация», «Другой»
CustomerType (ТипКлиента)	Категориальная переменная со значениями «Транзитный», «Транзитная сторона», «Контрактный», «Групповой»
MarketSegment (СегментРынка)	Категориальная переменная со значениями «Прямой», «Корпоративный», «Онлайновый турагент» «Офлановый турагент/турорганизация», «Групповой», «Другой»
Children (Дети)	Целочисленная, число детей в брони
Average Daily Rate, *ADR* (СреднесуточнаяСтоимостьНомера)	Числовая, среднесуточная стоимость номера, итоговая сумма брони / число дней
PreviousCancellation (ПредыдущееАннулирование)	Двоичная 0/1, аннулировал клиент бронь раньше или нет
IsRepeatedGuest (ПостоянныйГость)	Двоичная 0/1, бронировал ли клиент раньше номер в гостинице
Country (Страна)	Категориальная, страна происхождения заказчика
Quarter (Квартал)	Категориальная, квартал бронирования
Year (Год)	Целочисленная, год бронирования

В этой главе мы будем использовать следующие ниже пакеты в дополнение к стандартным, указанным в предисловии:

```
## R
library(rcompanion) # Для функции коэффициента корреляции V Крамера
library(car)        # Для диагностической функции VIF

## Python
from math import sqrt                     # Для калькуляции V Крамера
from scipy.stats import chi2_contingency # Для калькуляции V Крамера
```

Понимание интересующей взаимосвязи

Как показано на рис. 4.2, мы попытаемся ответить на вопрос «Влияет ли тип депозита на частоту аннулирования брони?».

[1] См. https://oreil.ly/BehavioralDataAnalysisCh4.

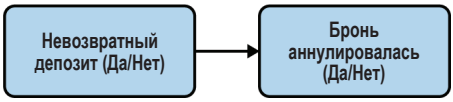

Рис. 4.2 ❖ Интересующая причинно-следственная связь

Давайте начнем с того, что посмотрим на *базовую частоту аннулирования* в разбивке по типу депозита (мне нравится смотреть как на абсолютные цифры, так и на проценты, если в некоторых категориях их очень мало):

```
## R (результат не показан)
with(dat, table(NRDeposit, IsCanceled))
with(dat, prop.table(table(NRDeposit, IsCanceled), 1))

## Python
table_cnt = dat_df.groupby(['NRDeposit', 'IsCanceled']).\
agg(cnt = ('Country', lambda x: len(x)))
print(table_cnt)

table_pct = table_cnt.groupby(level=0).apply(lambda x: 100 * x/float(x.sum()))
print(table_pct)

                        cnt
NRDeposit IsCanceled
0         0           63316
          1           23042
1         0              55
          1             982
                           cnt
NRDeposit IsCanceled
0         0           73.318048
          1           26.681952
1         0            5.303761
          1           94.696239
```

Мы видим, что подавляющее большинство броней не имеют депозита, а частота аннулирования составляет около 27 %. С другой стороны, при бронировании с невозвратными депозитами (NRDeposit) очень высока частота аннулирования брони. На первый взгляд, эта корреляция удивляет. Приведет ли замена нашей политики на «бездепозитную» для всех клиентов к сокращению частоты аннулирования брони? Поведенческий здравый смысл подсказывает, что с большей долей вероятности гостиницы запрашивают невозвратные депозиты в ситуациях «высокорисковых» броней и присутствует *спутывающий фактор*, как показано на рис. 4.3.

Мы довольно быстро перешли от рис. 4.2 к рис. 4.3, но этот шаг был важным: причинно-следственная диаграмма на рис. 4.2 демонстрирует базовый вопрос деловой аналитики: «Какова причинно-следственная связь между типом депозита и частотой аннулирования?» С другой стороны, причинно-следственная диаграмма на рис. 4.3 представляет более информированную

поведенческую гипотезу: «Невозвратный депозит, по-видимому, увеличивает частоту аннулирования, но эта взаимосвязь, вероятно, спутывается факторами, которые нам нужно будет определить».

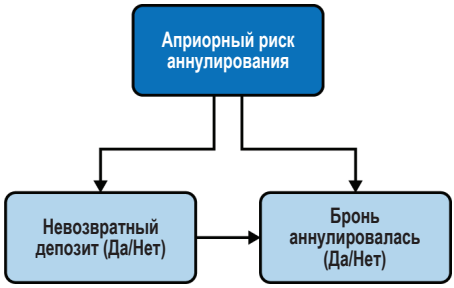

Рис. 4.3 ❖ Причинно-следственная связь, скорее всего, спутана

Приятный аспект использования причинно-следственных диаграмм для анализа поведенческих данных состоит в том, что они являются отличным инструментом для совместной работы. Любой сотрудник вашей организации, обладающий минимальными знаниями о причинно-следственных диаграммах, может взглянуть на рис. 4.3 и сказать: «Ну да, нам требуются невозвратные депозиты для праздничных заказов, и они часто аннулируются из-за погоды», или любую другую любопытную деталь поведенческих знаний, до которых вы не смогли бы дойти в противном случае.

В этом месте самым лучшим следующим шагом был бы рандомизированный эксперимент: назначить возвратные или невозвратные депозиты случайной выборке клиентов, и вы сможете подтвердить или отклонить свою поведенческую гипотезу. Однако вы, пожалуй, не сможете этого сделать или пока еще не сможете. Тем временем мы попытаемся распутать взаимосвязь путем выявления переменных, релевантных для включения в диаграмму.

Выявление переменных-кандидатов на включение

Пытаясь выявить переменные, потенциально пригодные для включения в диаграмму, совершенно естественно стремиться начинать с имеющихся у вас данных. Эта склонность вводит в заблуждение, сродни пьяному человеку, который ищет ключи от дома не там, где он их потерял, а под уличным фонарем, потому что там больше света. Поступая таким образом, вы, возможно, будете игнорировать наиболее важные переменные просто потому, что они не у вас под носом. Кроме того, вы с большей вероятностью будете принимать переменные в своих данных за чистую монету и не будете за-

даваться вопросом, являются ли они наилучшим представлением того, что происходит в реальном мире.

Например, категориальные переменные в ваших данных, скорее всего, будут представлять точку зрения, ориентированную на бизнес, а не на клиента, и, возможно, целесообразнее будет агрегировать некоторые категории или даже объединить разные переменные в новые. В нашем случае у нас есть переменная *СегментРынка* и еще одна для числа детей в брони. Мы можем подтвердить, посмотрев на данные, что очень немногие корпоративные клиенты приводят с собой детей. Следовательно, мы могли бы подумать о создании новой категориальной переменной с категориями «корпоративный без детей», «некорпоративный без детей» и «некорпоративный с детьми», выделив корпоративных клиентов с детьми в качестве выбросов, заслуживающих отдельного исследования (может быть, в качестве отправной точки для специализированных услуг?).

Вместо того чтобы поддаваться систематическому смещению в стиле «Есть только то, что ты видишь»[1,2], мы начнем с поведенческих категорий, описанных в главе 2, двигаясь от действия в обратную сторону:

- действия;
- намерения;
- познание и эмоции;
- личностные характеристики;
- поведения бизнеса.

Наконец, переменные в каждой из этих категорий могут находиться под воздействием временных трендов, таких как линейные тренды или сезонность, поэтому мы добавим их по умолчанию в конце этого раздела. В целях усиления фокуса внимания на качественной интуиции мы не будем рассматривать какие-либо данные до следующего раздела, посвященного подтверждению взаимосвязей. Заменив наш априорный риск аннулирования (а также другие потенциальные спутывающие факторы) этими категориями, наша причинно-следственная диаграмма теперь выглядит как на рис. 4.4, с кучей ненаблюдаемых переменных, которые были добавлены к двум интересующим нас переменным.

Для каждой из этих категорий мы теперь будем искать переменные, которые могут быть причиной любой из двух интересующих нас переменных.

---

[1] В оригинале «What You See Is All There Is», или дословно «То, что ты видишь, – это все, что есть». – *Прим. перев.*

[2] Этот красочный ярлык был популяризирован бихевиористом Дэниелом Канеманом.

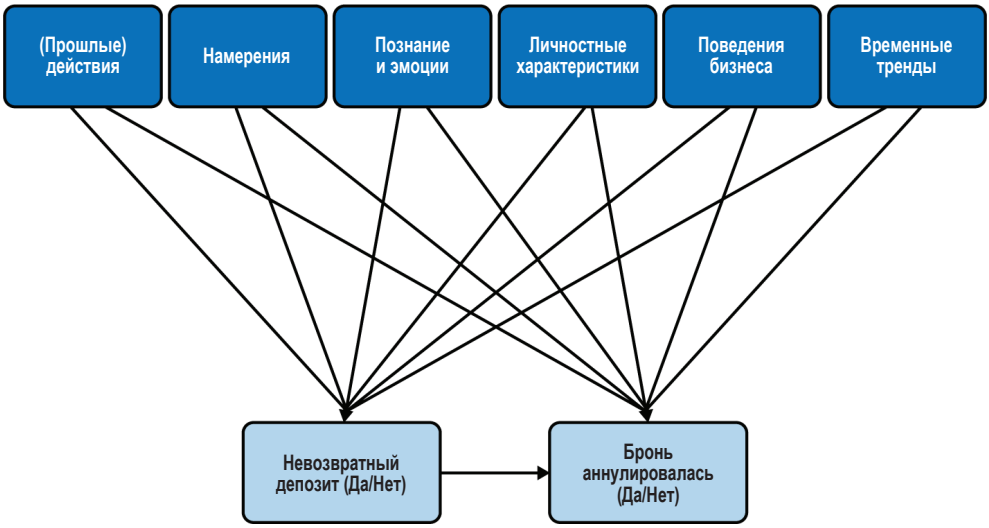

Рис. 4.4 ❖ Обновленная причинно-следственная диаграмма с категориями переменных, потенциально пригодных для включения

Действия

При поиске переменных для включения в категорию действий мы обычно пытаемся выявлять поведения клиента в прошлом, которые, возможно, повлияли на требование гостиницей невозвратого депозита (nonrefundable deposit, аббр. NRD).

Очевидным кандидатом в этом случае являются факты прошлых аннулирований брони клиентом. Возможно, гостиница с большей вероятностью будет запрашивать невозвратный кредит у клиентов, которые отказывались от брони в прошлом. Кроме того, возможно, что то, что заставило их аннулировать в прошлом, также с большей вероятностью заставит их аннулировать и в будущем.

В более общем случае, когда одна из интересующих нас переменных сама является действием, прошлое поведение часто является неплохой предсказательной переменной для включения, даже просто в качестве косвенного индикатора ненаблюдаемых личностных характеристик. В наших данных есть две переменные, которые связаны с прошлым поведением: *ПредыдущееАннулирование* и *ПостоянныйГость*. На рис. 4.5 показана обновленная причинно-следственная диаграмма, в которой неизмененные части выделены серым цветом.

Это не значит, что указанные прошлые поведения являются единственными релевантными; просто они оказались единственными, о которых я думал и для которых у нас были данные. Будем надеяться, что вы сможете подумать о других!

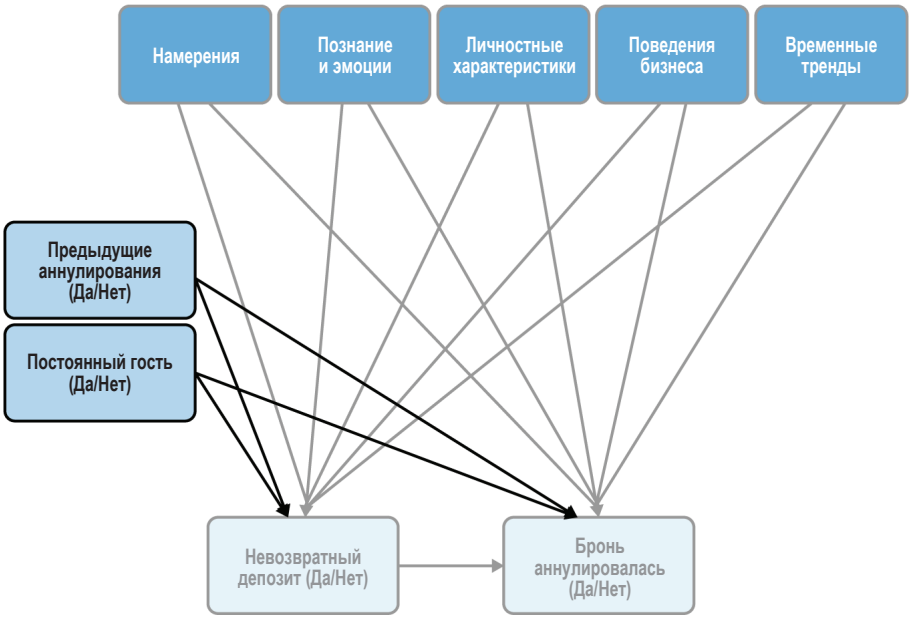

Рис. 4.5 ❖ Обновленная причинно-следственная диаграмма в конце шага действий

Намерения

В анализе данных намерения легко проглядеть, потому что они нередко пропущены в существующих данных. Однако они являются одной из наиболее важных движущих сил поведений, и их часто можно выявлять путем проведения собеседований с клиентами и сотрудниками. Таким образом, они являются одной из лучших иллюстраций выгод не просто рассмотрения существующих располагаемых данных, но и принятия подхода «поведения в первую очередь».

В данном случае я могу подумать о двух намерениях: основание для поездки и основания для аннулирования (рис. 4.6).

Обратите внимание, что я представил *ПричинуПоездки* как потенциальный спутывающий фактор, т. е. со стрелкой в сторону обеих интересующих нас переменных, тогда как *ОснованиеДляАннулирования* воздействует только на *БроньАннулировалась*. Пока что мы имеем всего лишь поведенческую догадку, а именно что основание для аннулирования не влияет на тип депозита. Моя аргументация заключается в том, что основание для аннулирования на момент внесения депозита неизвестно.

Рисунок 4.6 также показывает разносторонность причинно-следственных диаграмм для поведенческого анализа: мы можем поместить эти две потен-

циальные переменные на причинно-следственную диаграмму, даже не зная фактического списка оснований в любом случае; эти переменные мы определим позже в ходе собеседований. На данный момент мы можем отметить, что три имеющиеся в наших данных переменные, по-видимому, находятся под воздействием основания для поездки, и в силу этого их можно включить: *ТипКлиента*, *СегментРынка* и *КаналРаспространения* (рис. 4.7). Мы также вернемся к этим переменным в подразделе, посвященном личностным характеристикам.

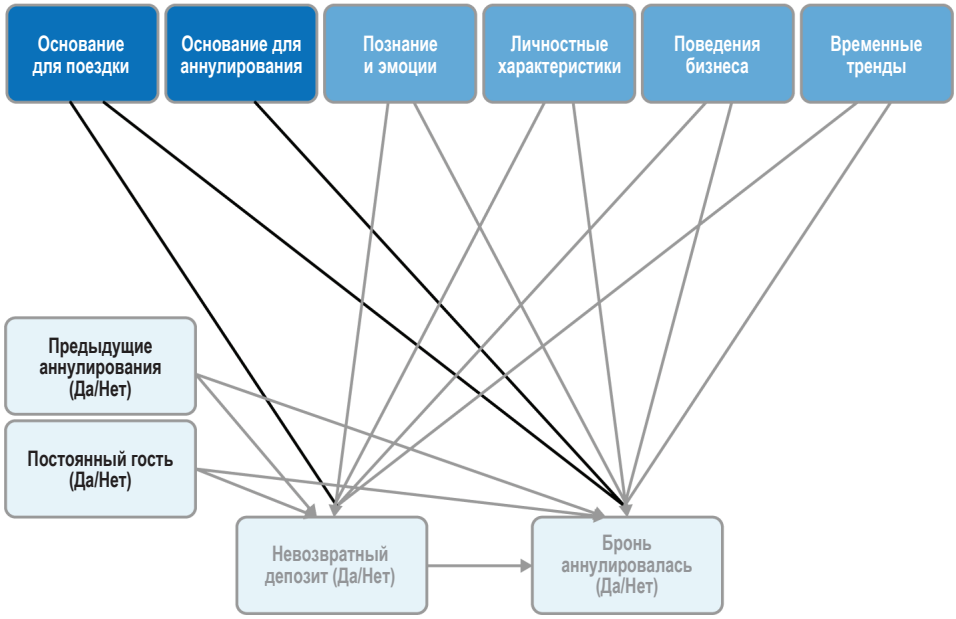

Рис. 4.6 ❖ Добавление намерений в причинно-следственную диаграмму

Познание и эмоции

Когда я пытаюсь выявить релевантные социальные, психологические или когнитивные явления для анализа, мне нравится увеличивать масштаб конкретных точек принятия решений. Здесь это будет, когда клиент бронирует номер и когда он аннулирует бронь.

В первой точке принятия решения клиенты, возможно, не понимают, что их депозит не подлежит возврату, или они, возможно, об этом забыли. Во второй точке принятия решения они, возможно, трактуют свой депозит как потерянные затраты и не прилагают усилий для сохранения своих броней (рис. 4.8).

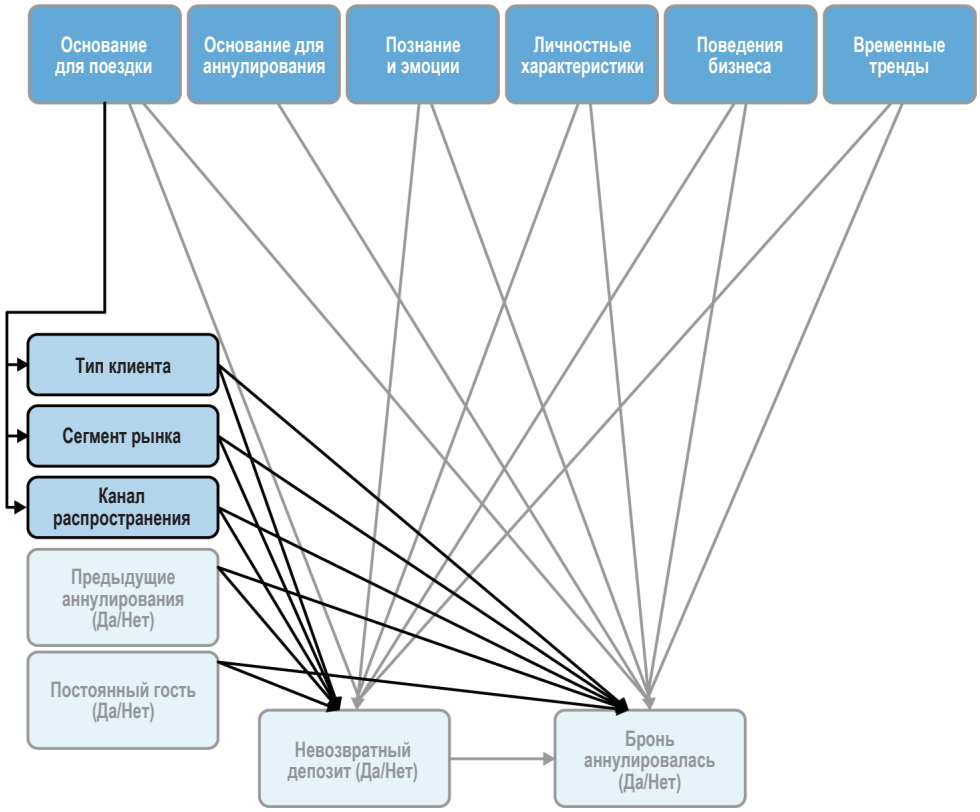

Рис. 4.7 ❖ Обновленная причинно-следственная диаграмма в конце шага намерения

Личностные характеристики

Как упоминалось в главе 2, демографические переменные нередко представляют ценность не столько сами по себе, сколько в качестве посредников для других личностных характеристик, таких как черты характера. Поэтому трудностью этого шага является противостояние влиянию любых демографических переменных, присутствующих в наших данных, и соблюдение приверженности нашему причинно-поведенческому образу мыслей. Для этого неплохо сначала подумать о чертах характера и только потом обратиться к демографическим переменным.

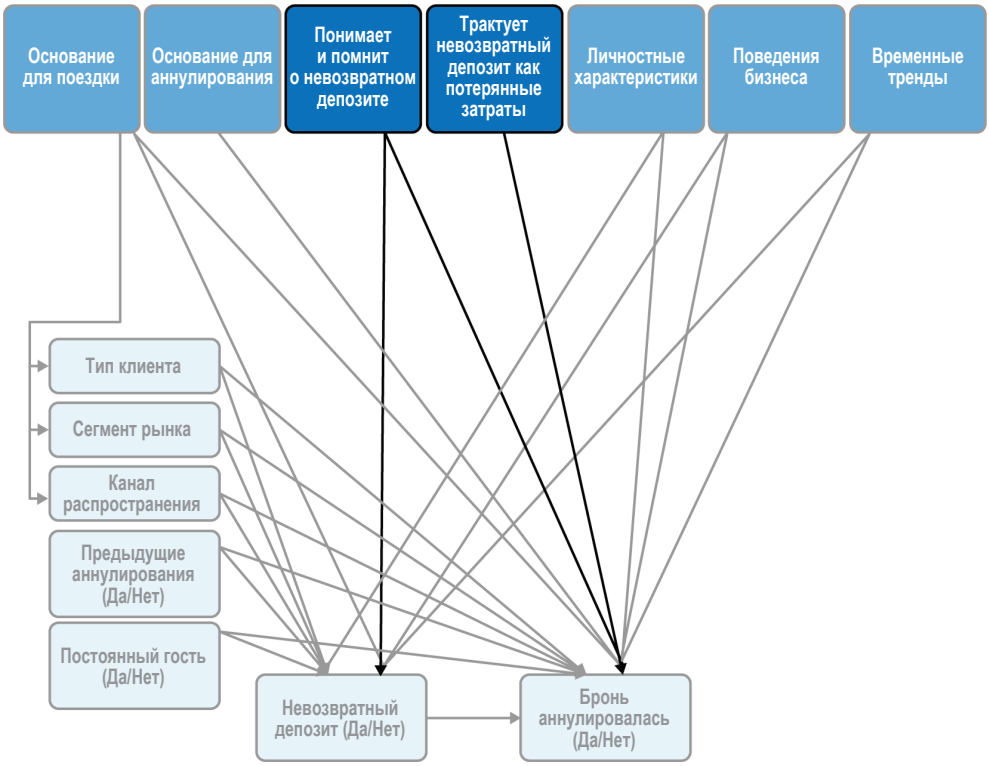

Рис. 4.8 ❖ Обновленная причинно-следственная диаграмма в конце шага познания и эмоций

Черты

Основываясь на наших знаниях психологии личности, неплохими кандидатными чертами, которые могут стать причиной нашего поведения по аннулированию, являются добросовестность и невротичность: высказывание о том, что менее организованные и более беззаботные люди с большей вероятностью в итоге аннулируют бронь, выглядит правдоподобным (рис. 4.9).

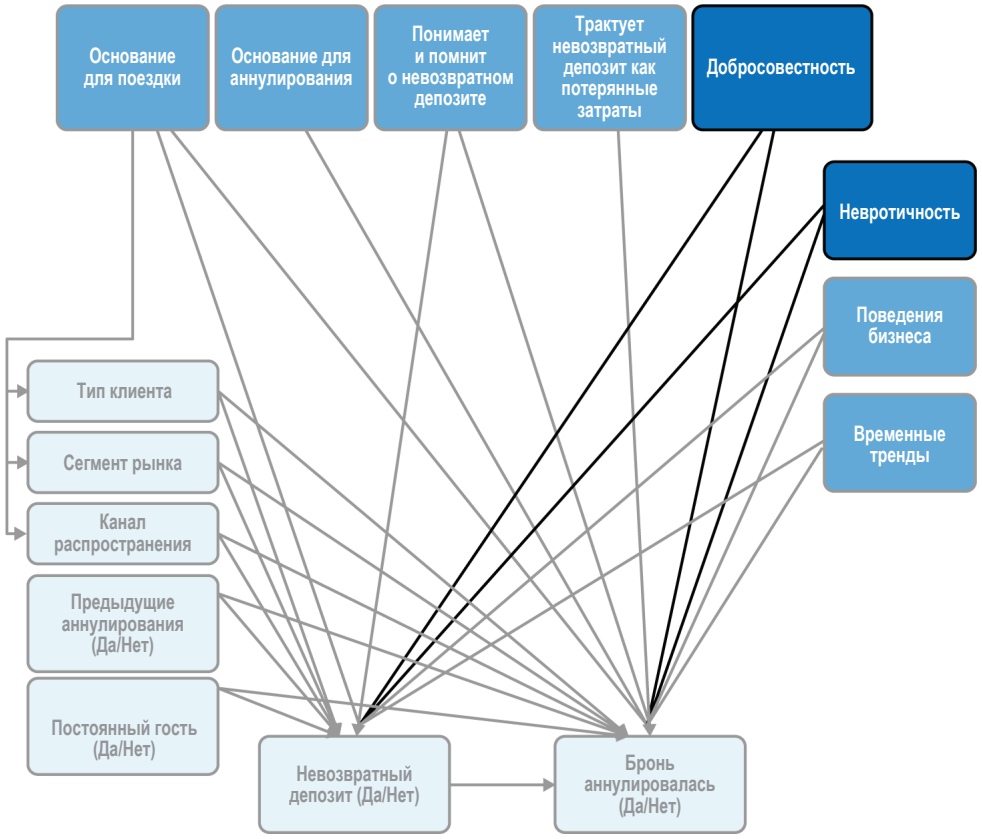

Рис. 4.9 ❖ Причинно-следственная диаграмма, дополненная личностными чертами

Демографические переменные

Ранее мы отмечали, что бронирование в гостиницах у нас делают корпоративные и некорпоративные клиенты. Помимо основания для поездки и для аннулирования, этот факт также воздействует на некоторые другие личностные характеристики, такие как ценовая эластичность, обе из которых воздействовали бы на две интересующие нас переменные. Давайте сгруппируем их под заголовком «финансовые характеристики». Вероятно, они в некоторой степени связаны с тремя переменными, которые мы встречали ранее: *ТипКлиента*, *СегментРынка* и *КаналРаспределения*, а также с несколькими другими переменными в наших данных, такими как *Дети*, *СреднесуточнаяСтоимостьНомера* (т. е. цена за ночь, от англ. Average Daily Rate, аббр. ADR) и *Страна* (рис. 4.10).

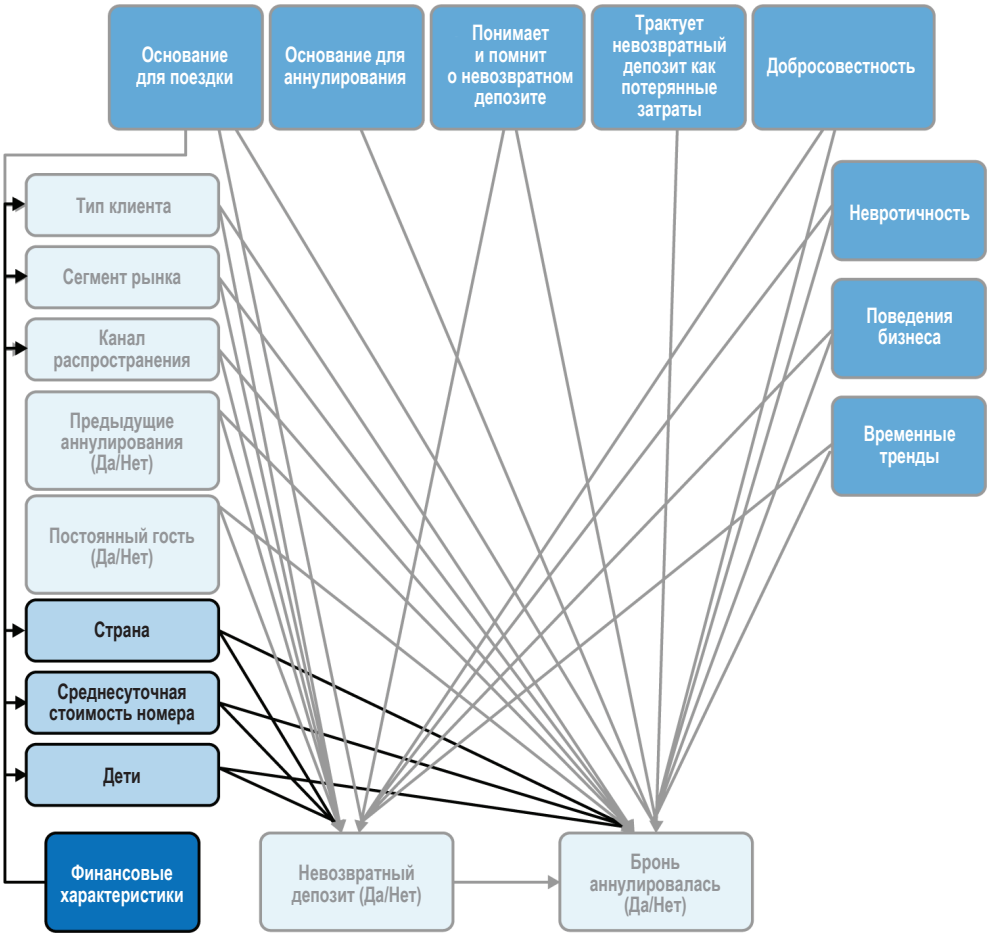

Рис. 4.10 ❖ Причинно-следственная диаграмма, обновленная демографическими переменными

Поведения бизнеса

Поведения бизнеса нередко играют большую роль в расследуемых нами взаимосвязях, но их легко проглядеть и трудно интегрировать.

В данном примере деловые правила, очевидно, играют важную роль, поскольку они определяют клиентов, которые должны будут уплачивать невозвратный депозит. В данном смысле они влияют на все стрелки на причинно-следственной диаграмме, входящие в *НевозвратныйДепозит*. Мы можем объяснить это влияние несколькими способами, в зависимости от того, какие формы оно принимает.

Деловое правило может связывать две наблюдаемые переменные в явной форме (возможно, включая интересующие нас переменные). Здесь, например, мы можем себе представить деловое правило, в котором говорится, что все клиенты, которые ранее аннулировали бронь, теперь должны предоставлять невозвратный депозит. Перечисляя все такие правила, мы можем подтверждать или отклонять все стрелки из наблюдаемой переменной, входящие непосредственно в *НевозвратныйДепозит*. Это также, возможно, проявит переменные, которые учавствуют в деловых правилах, но пока что еще не включены в наши данные: например, мы могли бы вообразить ситуацию, в которой клиенты, не предъявившие удостоверение личности во время бронирования, должны уплачивать невозвратный депозит. Я сказал «пока что еще нет в наших данных», потому что по определению любой критерий, являющийся частью делового правила, является наблюдаемым, даже если он не зафиксирован в базе данных[1].

В качестве альтернативы деловое правило лучше всего можно представлять в качестве дополнительной промежуточной переменной. Например, если все бронирования во время рождественских праздников должны подкрепляться невозвратным депозитом, то мы могли бы создать двоичную переменную *РождественскиеПраздники* со стрелкой в *НевозвратныйДепозит*. Эта переменная затем будет опосредовать эффект других переменных, таких как *ТипКлиента* или *Дети*, на *НевозвратныйДепозит*.

Мы не знаем, какие деловые правила применяются в двух гостиницах из нашего примера, поэтому нам придется оставить этот подраздел как что-то, что мы хотели бы разведать в ходе последующих собеседований.

Временные тренды

Наконец, в наших данных могут иметься некие глобально-временные тренды, такие как постепенное увеличение числа бронирований, требующих невозвратного депозита, параллельно с поступательным, но не связанным с этим увеличением частоты аннулирования. В дополнение к этому, учитывая сезонность гостиничной индустрии, вероятно, мы хотели бы охватить некие циклические аспекты (рис. 4.11).

В данном случае переменные *Год* и *Квартал* улавливают только тренды и циклы. Иногда также имеет смысл включать двоичные переменные для объяснения конкретных событий, которые выделяют определенный год или отмечают постоянное изменение. Очевидным примером является COVID-19, который, когда пыль осядет, в некоторых секторах окажется временным всплеском, а в других – началом серьезных изменений.

С этим последним дополнением на рис. 4.11 теперь у нас имеется лавина кандидатных переменных, некоторые из которых можно наблюдать, а некоторые нет. В следующем далее разделе мы взглянем на то, как подтверждать, какие из наблюдаемых переменных следует оставлять.

---

[1] Подумайте о следующем: как бы это правило было имплементировано в противном случае?

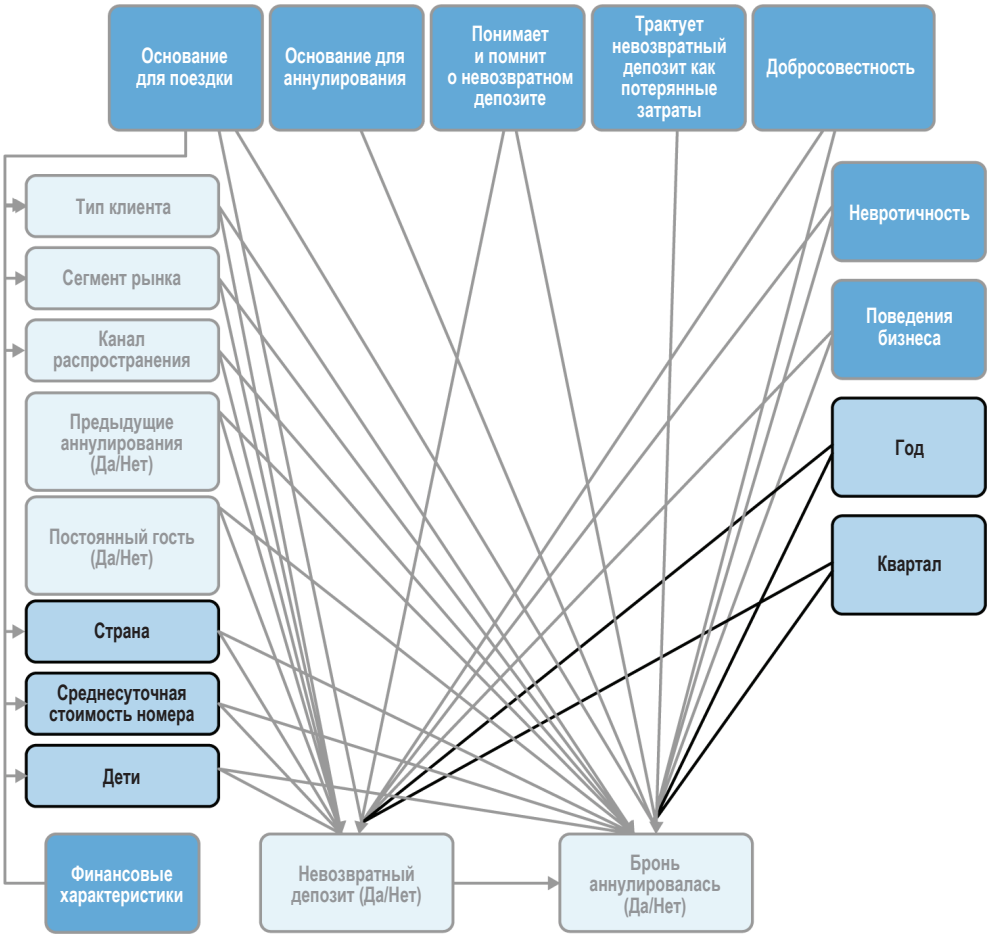

Рис. 4.11 ❖ Обновленная причинно-следственная диаграмма в конце шага временных трендов

Подтверждение наблюдаемых переменных для включения на основе данных

Давайте посмотрим на наблюдаемые переменные, которые мы имеем в качестве кандидатов в конце фазы выявления (рис. 4.12).

В данном конкретном примере все эти наблюдаемые переменные предварительно соединены с обеими интересующими нас переменными. Эта ситуация принята по умолчанию, но в некоторых случаях у вас может иметься очень сильная априорная аргументация для соединения предсказательной переменной только с одной из интересующих вас переменных (так было, например, с некоторыми ненаблюдаемыми переменными). Находясь в сомнении, я поступил благоразумно и включил оба соединения.

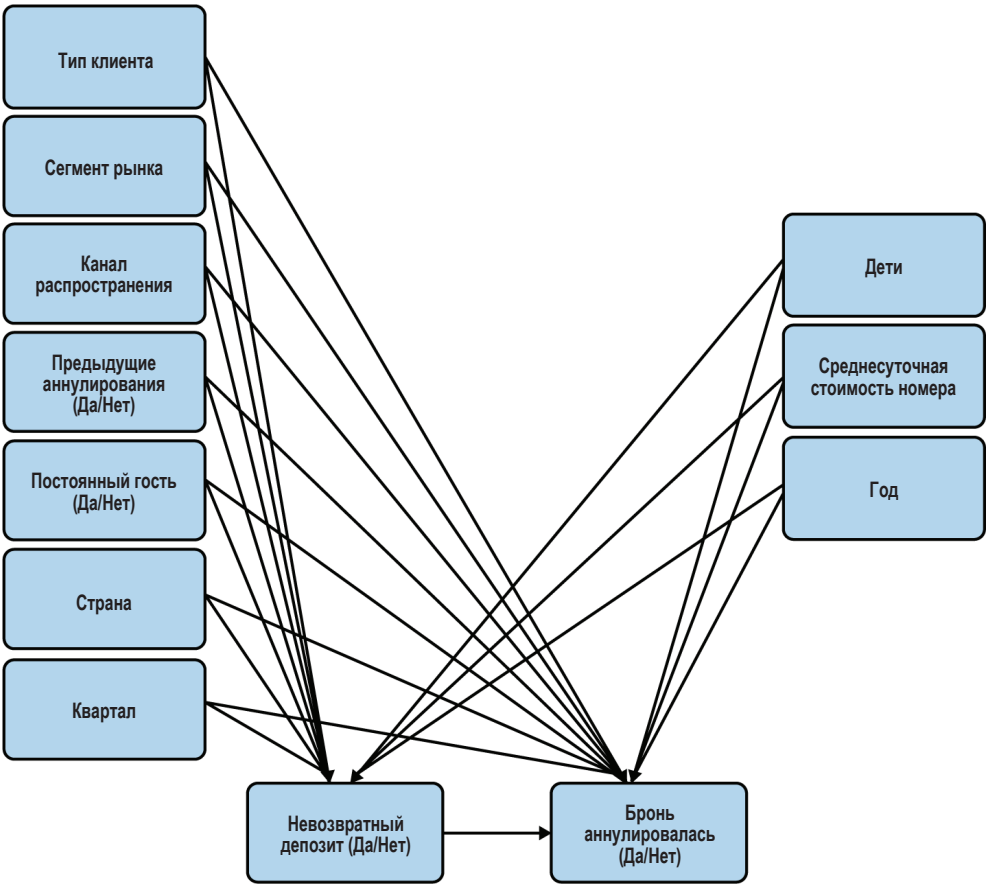

Рис. 4.12 ❖ Наблюдаемые переменные на причинно-следственной диаграмме в разрезе на категориальные (слева) и числовые (справа)

На рис. 4.12 наблюдаемые переменные разрезаны на категориальные (слева от причинно-следственной диаграммы) и числовые (справа от причинно-следственной диаграммы). Эти два типа данных требуют разных квантитативных инструментов, поэтому мы рассмотрим их по очереди.

Взаимосвязи между числовыми переменными

Нашим первым шагом будет просмотр матрицы корреляций для всех числовых переменных в наших данных. Полезный, но грязный трюк состоит в конвертировании двоичных переменных в нули/единицы (если они еще не находятся в этом формате), чтобы иметь возможность их трактовать как числовые. Это позволит вам почувствовать корреляцию между переменными; только не говорите об этом своим друзьям-статистикам!

Просматривая строки в отношении двух интересующих вас переменных, вы сможете увидеть степень их соотнесенности со всеми числовыми пере-

менными в вашем наборе данных. При беглом просмотре это также покажет вам любую большую корреляцию между этими другими переменными. Сила корреляции с интересующими нас причиной и следствием затем поможет нам определить, что делать с данной конкретной переменной.

Насколько сильна эта сила? Это зависит от ситуации. Помните, что наша цель состоит в том, чтобы правильно измерить причинно-следственный эффект интересующей нас причины на интересующее нас следствие; в качестве эмпирического правила: «сильной» можно считать любую корреляцию, имеющую тот же порядок величины (т. е. одинаковое число нулей между точкой и первой ненулевой цифрой), как корреляцию между интересующей вас причиной и интересующим вас следствием.

Как видно на рис. 4.13, коэффициент корреляции между двумя интересующими нас переменными составляет 0.16. В первом столбце указаны корреляции с переменной *НевозвратныйДепозит*, а во втором столбце – корреляции с переменной *БроньАннулировалась*. Переменная *ПредыдущееАннулирование* имеет коэффициенты корреляции с интересующими нас переменными, которые имеют одинаковый порядок величины (соответственно 0.15 и 0.13). Схожим образом переменная *СреднесуточнаяСтоимостьНомера* имеет коэффициент корреляции с переменной *БроньАннулировалась*, который является значимым по этому критерию (0.13).

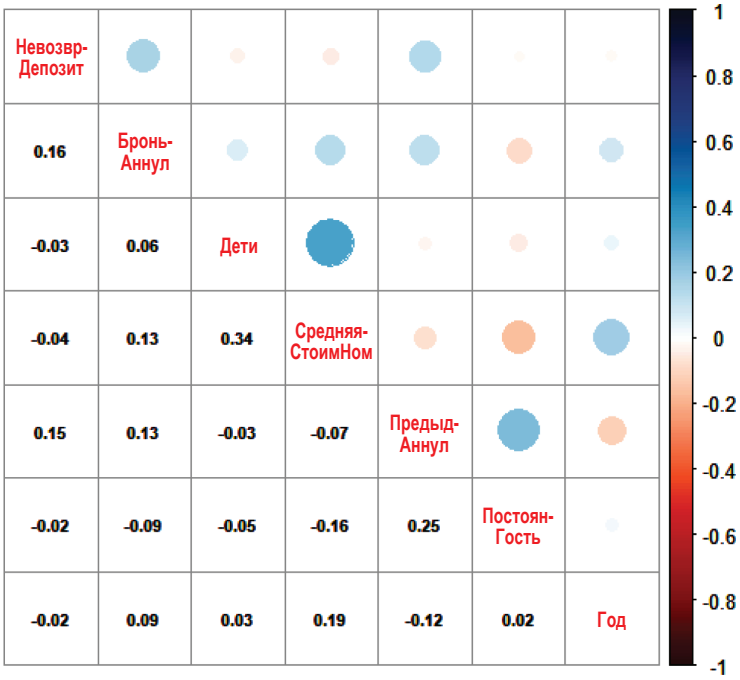

Рис. 4.13 ❖ Матрица корреляций для числовых и двоичных переменных

Порог «порядка величины» для включения ни в коей мере не является научным, и его можно ужесточать либо ослаблять в зависимости от числа

переменных у вас под рукой. Если несколько переменных этот порог преодолевают, а несколько других переменных находятся близко к нему, то их можно включить.

Можно возразить, что некая переменная может иметь низкую корреляцию с любой интересующей нас переменной, но все равно быть спутывающим фактором, который необходимо объяснить. Это верно, и вы можете включать переменную, даже если она слабо коррелирует с интересующими вас переменными, основываясь на сильной теоретической аргументации. Однако для практических целей вам обычно следует сосредоточиваться на переменных, имеющих по меньшей мере умеренный уровень корреляции с интересующими нас переменными.

Если мы включим все корреляции, которые на рис. 4.13 по абсолютному значению равны 0.1 или выше, и исключим остальные, то наша причинно-следственная диаграмма теперь будет такой, как на рис. 4.14.

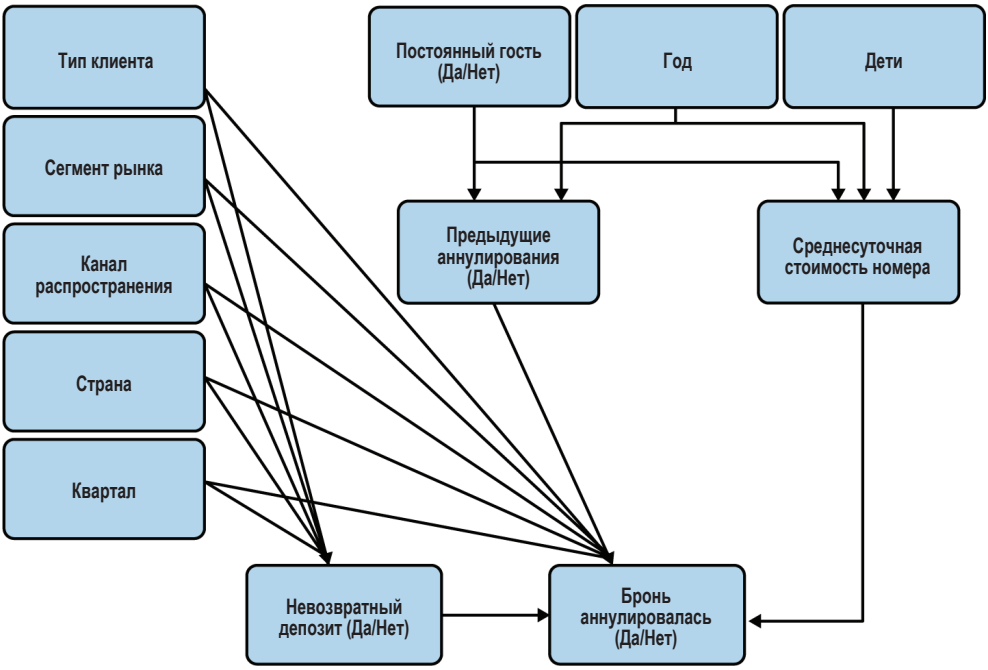

Рис. 4.14 ❖ Причинно-следственная диаграмма с обновленными стрелками для числовых и двоичных наблюдаемых переменных

В то время как корреляционная матрица дает нам только симметричные коэффициенты, которые могут иметь форму стрелок в любом направлении, я применил немного здравого смысла и деловые знания, чтобы допустить направленность стрелок. Гостиничная компания не владеет временем, поэтому мы можем допустить, что *Годы* являются причиной, а не следствием переменных, с которыми она коррелирует, хотя этот эффект может проходить через промежуточные переменные, такие как тренды в обществе с течением

времени. *ПостоянныйГость* является обязательным условием для *ПредыдущегоАннулирования*, и поскольку эта переменная относится к прошлому событию, она также должна быть причиной *СреднесуточнойСтоимостиНомера*, либо они обе делят общую причину между собой.

Не забывайте, что эта причинно-следственная диаграмма является предварительной:

- некоторые из этих корреляций, вероятно, являются ложноположительными (ложно утверждают о наличии корреляции, т. е. коэффициент кажется сильнее, чем он есть на самом деле, из-за чистой случайности), и, наоборот, некоторые из меньших корреляций могут быть ложноотрицательными (ложно отрицать корреляцию);
- на данном этапе мы трактуем корреляции предварительно как подтверждение причинно-следственной связи. Некоторые из стрелок, которые мы начертили на рис. 4.14, сами же могут отражать взаимосвязи, которые являются спутывающими. После адекватного измерения взаимосвязи между *НевозвратнымДепозитом* и *БроньАннулировалась* мы, возможно, захотим, либо нам потребуется, сделать то же самое для других взаимосвязей (например, между *ПостояннымГостем* и *СреднесуточнойСтоимостьюНомера*).

Взаимосвязи между категориальными переменными

Та же логика применима и к категориальным переменным, с единственным осложнением в том, что мы не сможем использовать коэффициент корреляции Пирсона. Однако его вариант, коэффициент V Крамера[1,2], был разработан для категориальных переменных. На языке R он имплементирован в пакете `rcompanion`:

```
## R
> with(dat, rcompanion::cramerV(NRDeposit, IsCanceled))
Cramer V
   0.165
```

Хорошо видно, что в случае двоичных переменных он дает результат, довольно близкий к прямому применению коэффициента корреляции Пирсона. К сожалению, он не имплементирован на Python, но я предоставил функцию, которая его вычисляет:

```
## Python
def CramerV(var1, var2):
    ...
    return V

V = CramerV(dat_df['NRDeposit'], dat_df['IsCanceled'])
print(V)
0.16483946381640308
```

---

[1] Коэффициент V Крамера – это мера тесноты связи и вычисляется делением статистики хи-квадрат на объем выборки и взятием корня квадратного из результата. – *Прим. перев.*

[2] См. https://oreil.ly/KAoIa.

На рис. 4.15 показана соответствующая матрица корреляций после переименования переменных для удобства чтения (в переводе на графике указаны русские сокращения, которые можно подставить в приведенный ниже исходный код).

```
## R
dat <- dat %>%
  rename(CustTyp= CustomerType) %>%
  rename(DistCh = DistributionChannel) %>%
  rename(RepGst = IsRepeatedGuest) %>%
  rename(MktSgmt = MarketSegment) %>%
  rename(IsCanc = IsCanceled) %>%
  rename(PrevCan = PreviousCancellations) %>%
  rename(NRDep = NRDeposit)

## Python
dat_df.rename(columns=
              {"CustomerType": "CustTyp",
               "DistributionChannel": "DistCh",
               "IsRepeatedGuest": "RepGst",
               "MarketSegment": "MktSgmt",
               "IsCanceled": "IsCanc",
               "PreviousCancellations": "PrevCan",
               "NRDeposit": "NRDep"},
              inplace=True)
```

Эта корреляция порождает множество идей. Глядя на нижнюю строку, мы видим, что *Квартал* не является содержательно коррелированным ни с чем другим. Это намекает на то, что сезонность не является релевантным фактором для нашего анализа. И наоборот, вполне возможно, что квартал является слишком грубой единицей времени и что нам нужно увеличить масштаб, наведя линзу на очень конкретные периоды времени, такие как *РождественскиеПразники*. Мы можем удалить переменную *Квартал* из причинно-следственной диаграммы и заменить ее ненаблюдаемой переменной *Сезонность* в качестве подсказки для будущих исследований.

Наши три переменные для клиентских сегментов, *ТипКлиента*, *СегментРынка* и *КаналРаспределения*, показывают смешанный шаблон, с некоторыми очень сильными и некоторыми слабыми корреляциями между ними. Аналогичная ситуация – и с их корреляциями с другими переменными, которые встречаются повсюду: например, все три из них имеют корреляции со *Страной* в цифрах 0.1X, но две из них имеют высокие корреляции с *ПостояннымГостем* (0.35 и 0.4), тогда как у третьей корреляция составляет всего 0.11. Все это говорит о том, что указанные переменные не просто взаимозаменяемы, но и отражают некоторые аспекты одних и тех же поведений. Это требует дальнейшего исследования и, скорее всего, создания новых переменных.

Применив эти идеи и тот же критерий включения корреляций, только выше 0.1, наша причинно-следственная диаграмма теперь примет вид как на рис. 4.16.

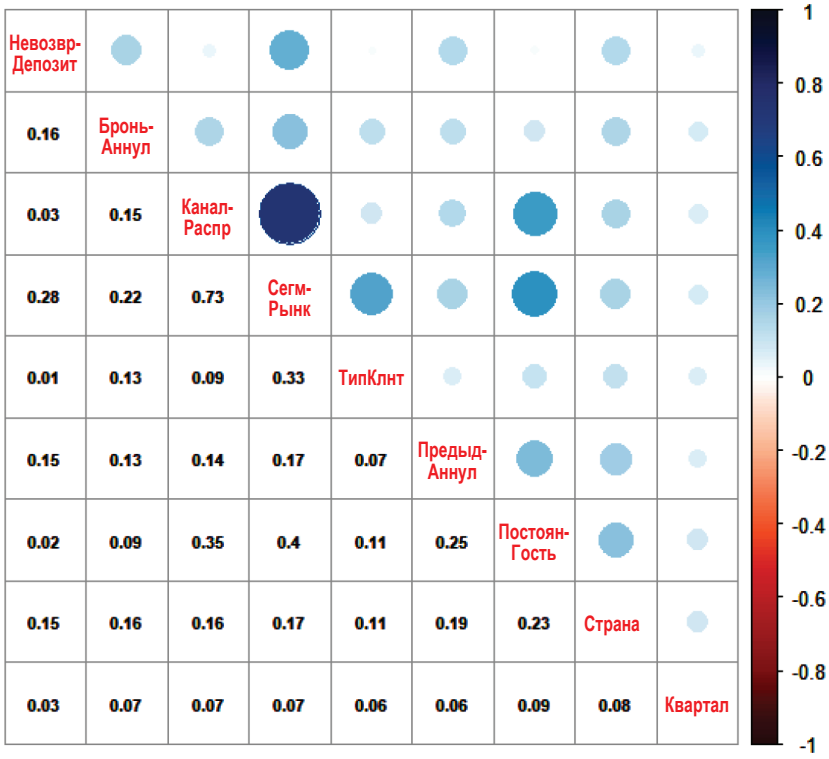

Рис. 4.15 ❖ Матрица корреляций
для категориальных и двоичных переменных

Наша причинно-следственная диаграмма начинает становиться умеренно сложной, но по большей части ее можно свести к нескольким поведенческим аргументам:

- четыре переменные слева отражают личностные характеристики, и они значительно коррелируют друг с другом. Я решил отразить эти корреляции с помощью двуглавых стрелок, потому что пытаться определить направление стрелок было бы бессмысленно: *ТипКлиента* является причиной *СегментаРынка* не больше, чем наоборот. На самом деле после проведения необходимых собеседований мы должны создать новые переменные, которые отражают более глубокие личностные характеристики, участвующие в игре;
- личностные характеристики, по-видимому, влияют на интересующие нас переменные, потенциально являясь причиной некоторого спутывания;
- личностные характеристики, по-видимому, повлияли на прошлые поведения *ПостоянногоГостя* и *ПредыдущегоАннулирования*. (Опять же, я принимаю допущение о направленности эффектов, основываясь на знании бизнеса. На первый взгляд кажется маловероятным, что факт аннулирования предыдущей брони побудит кого-то изменить *Страну*

или *СегментРынка*.) Выяснив природу более глубоких личностных характеристик, мы, возможно, решим свернуть эти прошлые поведения под зонтиком некоторых переменных личностных характеристик, неявно создав поведенческих персонажей, например «регулярно посещающего делового клиента (Да/Нет)».)

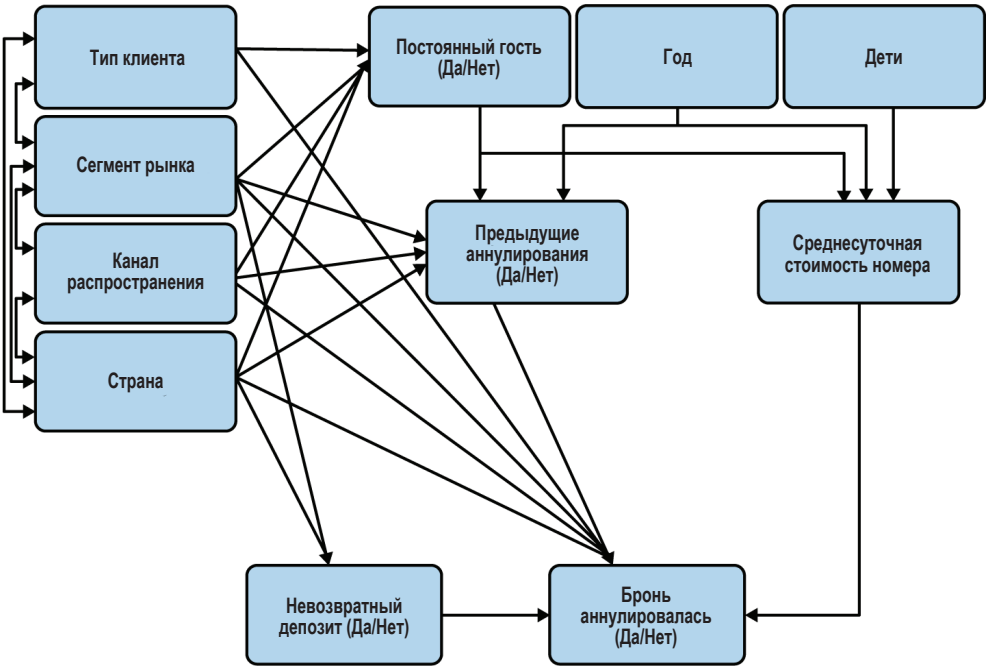

Рис. 4.16 ❖ Причинно-следственная диаграмма с обновленными стрелками для категориальных и двоичных наблюдаемых переменных

Взаимосвязи между числовыми и категориальными переменными

Процесс измерения корреляций между числовыми и категориальными переменными является более громоздким, чем процесс измерения корреляций внутри однородной категории.

Высказывание о том, что между числовой и категориальной переменными существует корреляция, эквивалентно высказыванию о том, что значения числовой переменной в среднем различаются по категориям категориальной переменной. Мы можем проверить истинность этого высказывания, сравнив среднее значение числовой переменной по всем категориям категориальной переменной. Например, мы ожидаем, что финансовые характеристики клиента, возможно, воздействуют на среднесуточную стоимость бронируемого номера. Лучше всего было бы разведать эту взаимосвязь после построения

более релевантных переменных для сегментации клиентов, но ради аргументации мы можем задействовать *ТипКлиента*:

```
## R (результат не показан)
> dat %>% group_by(CustTyp) %>% summarize(ADR = mean(ADR))

## Python
dat_df.groupby('CustTyp').agg(ADR = ('ADR', np.mean))
Out[10]:
                          ADR
CustTyp
Contract           92.753036
Group              84.361949
Transient         110.062373
Transient-Party    87.675056
```

Мы видим, что среднесуточная стоимость номера существенно варьируется в зависимости от типов клиентов.

Если вы не уверены в том, что вариации действительно имеют существенную величину либо что они отражают только ошибки случайного отбора данных, вы можете построить для них интервалы уверенности, используя для этого бутстрап, как будет объяснено далее в главе 7.

В нашем примере есть две числовые переменные, корреляцию которых с категориальными переменными мы, возможно, захотим проверить: *СреднесуточнаяСтоимостьНомера* и *Год*. Мы обнаружили, что *СреднесуточнаяСтоимостьНомера* существенно варьируется в зависимости от типов клиентов, но те достаточно стабильны во временной динамике, что подводит нас к окончательной причинно-следственной диаграмме для наблюдаемых переменных (рис. 4.17).

В этом месте я хотел бы подчеркнуть и расширить свое предыдущее предупреждение: в процессе проверки наблюдаемых переменных я неявно допустил, что корреляция была причинно-следственной. Но, возможно, эти взаимосвязи сами спутаны: корреляция между переменными личностных характеристик и *ПредыдущимАннулированием* могла полностью обуславливаться взаимосвязью между переменными личностных характеристик и *ПостояннымГостем*.

Давайте вообразим, например, что деловые клиенты с большей вероятностью будут постоянными гостями. В этом случае, возможно, также будет казаться, что у них более высокая частота предыдущих аннулирований, чем у праздных клиентов, даже несмотря на то, что среди постоянных гостей деловые и праздные клиенты имеют точно такую же частоту предыдущих аннулирований.

Эти допущения о причинно-следственной связи можно рассматривать как безобидную ложь: она не соответствует действительности, но она вполне нормальна, потому что мы не пытаемся строить истинную, полную причинно-следственную диаграмму, а пытаемся устранить спутывание из взаимосвязи между невозвратным депозитом и частотой аннулирования брони.

С этой точки зрения гораздо важнее правильно уяснить направление стрелок, чем иметь не очищенные от спутываний взаимосвязи между переменными, находящимися за пределами интересующих нас переменных. Если вы все еще настроены скептически, то в одном из упражнений в следующей далее главе этот вопрос будет разведан подробнее.

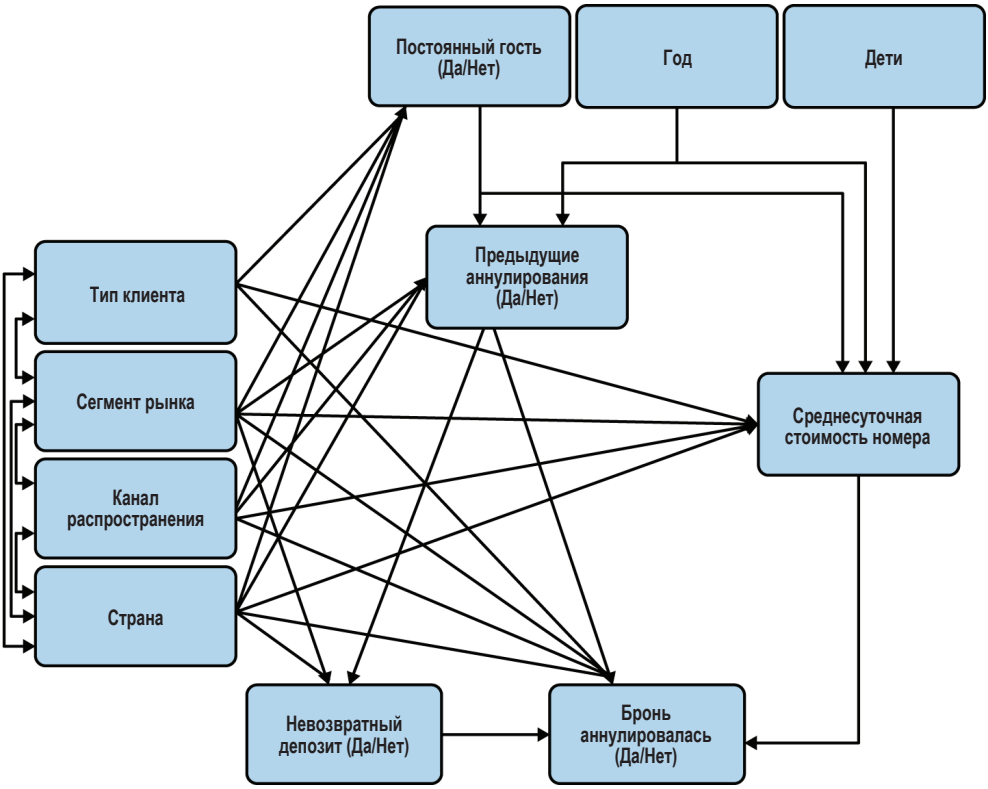

Рис. 4.17 ❖ Окончательная причинно-следственная диаграмма для наблюдаемых переменных

Итеративное расширение причинно-следственной диаграммы

После подтверждения или отклонения взаимосвязей между наблюдаемыми переменными на основе данных у нас будет на руках предварительно законченная причинно-следственная диаграмма (рис. 4.18).

Отталкиваясь от нее, мы будем расширять нашу причинно-следственную диаграмму итеративно, выявляя косвенные индикаторы для ненаблюдаемых переменных и выявляя дальнейшие причины текущих переменных.

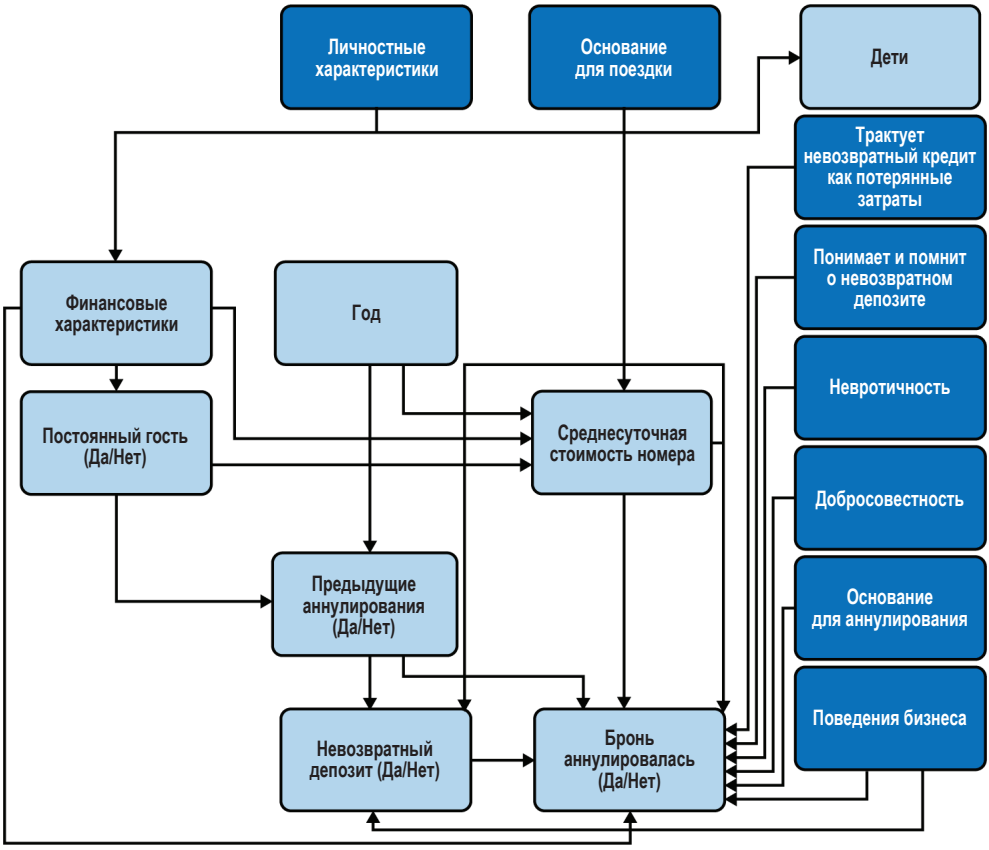

Рис. 4.18 ❖ Предварительно законченная причинно-следственная диаграмма с наблюдаемыми и ненаблюдаемыми переменными, группирующая переменные личностных характеристик под одним заголовком для удобства чтения

Выявление косвенных индикаторов для ненаблюдаемых переменных

Ненаблюдаемые переменные представляют серьезную трудность, поскольку, даже если они подтверждаются посредством собеседований или исследований пользовательского опыта (UX), их невозможно объяснять непосредственно в регрессионном анализе.

Мы все же можем попытаться несколько смягчить их, выявив потенциальные косвенные индикаторы посредством собесебований и исследований. Например, мы, возможно, обнаружим, что добросовестность действительно коррелирует с более низкой частотой аннулирования, но также и с запросом подтверждения по электронной почте (рис. 4.19).

Разумеется, запрос на подтверждение по электронной почте не обуславливается только добросовестностью – он также, возможно, отражает серьез-

ность намерения, недостаточное удобство в работе с цифровыми каналами и т. д. И наоборот, он, возможно, будет сам собой сокращать частоту аннулирования, предоставляя легкодоступную информацию о бронировании. Независимо от этого, если мы обнаружим, что такое поведение отрицательно коррелирует с частотой аннулирования, то мы можем задействовать это понимание, например, отправляя SMS-напоминание клиентам, которые решили не получать подтверждение по электронной почте.

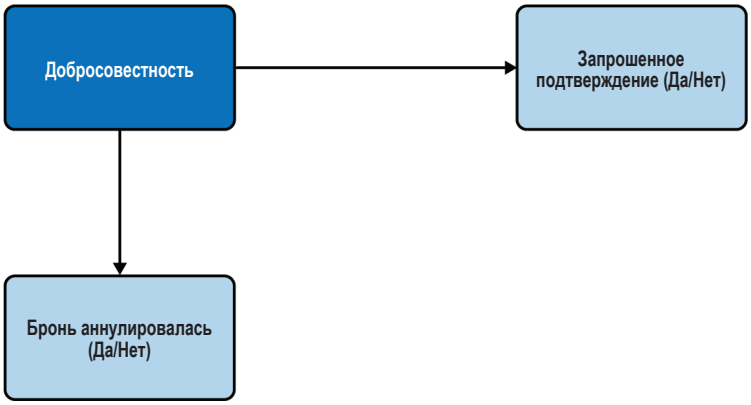

Рис. 4.19 ❖ Выявление косвенных индикаторов для ненаблюдаемых переменных

Путем проведения мозгового штурма и подтверждения посредством исследования потенциальных косвенных индикаторов для ненаблюдаемых переменных мы обеспечиваем содержательные взаимосвязи между наблюдаемыми переменными. Знание того, что *ЗапрошенноеПодтверждение* соединено с переменной *БроньАннулировалась* посредством Добросовестности, обеспечивает поведенческую аргументацию того, что в противном случае было бы сырой статистической регулярностью.

Выявление дальнейших причин

Мы также расширим нашу причинно-следственную диаграмму, выявив причины «внешних» переменных в нашей причинно-следственной диаграмме, т. е. переменных, у которых в настоящее время в ней нет ни одного родителя. В частности, когда у нас есть переменная A, которая влияет на интересующую нас причину (возможно, косвенно), но не на интересующее нас следствие, и еще одна переменная B, которая, наоборот, влияет на интересующее нас следствие, но не на интересующую нас причину, любая общая причина A и B вносит в причинно-следственную диаграмму спутывание, потому что эта общая причина также является общей причиной двух интересующих нас переменных.

В нашем примере единственной наблюдаемой переменной без какого-либо родителя (наблюдаемого или нет) является *Год*, и, очевидно, у нее его

не может быть (кроме, возможно, законов физики?), и поэтому данный шаг неприменим.

Итеративный повтор

Вводя новые переменные, вы создаете новые возможности для косвенных индикаторов и дальнейших причин. Например, наше только что введенное *ЗапрошенноеПодтверждение*, возможно, будет находиться под воздействием *Добросовестности*, но и *ОснованияПоездки* тоже. Из этого вытекает, что вы должны продолжать расширять свою причинно-следственную диаграмму до тех пор, пока она не будет объяснять все релевантные переменные, о которых вы можете подумать, и их взаимосоединения.

Однако в этом процессе наблюдается значительное снижение отдачи: по мере того как вы расширяете свою причинно-следственную диаграмму «наружу», вновь добавляемые переменные будут тяготеть к тому, чтобы иметь все меньше и меньше корреляций с интересующими вас переменными из-за всего того шума, который накапливается на этом пути. Это означает, что их объяснение будет распутывать интересующую вас связь во все меньших и меньших количествах.

Упрощения причинно-следственной диаграммы

Заключительный шаг после того, как вы решили прекратить итеративное расширение причинно-следственной диаграммы, состоит в ее упрощении. И действительно, теперь у вас есть диаграмма, которая, будем надеяться, является точной и готовой для практических целей, но она, возможно, будет структурирована не в том ключе, как это было бы полезнее всего для удовлетворения потребностей бизнеса. Поэтому я бы рекомендовал следующие шаги по упрощению:

- сворачивать цепочки, когда промежуточные переменные не представляют интереса или не наблюдаются;
- развертывать цепочки, когда вам нужно найти наблюдаемые переменные или если вы хотите отследить путь, которым еще одна переменная связана с диаграммой;
- разрезать переменные, когда вы думаете, что отдельные переменные будут содержать интересную информацию (например, корреляция с одной из интересующих вас переменных на самом деле обуславливается только одним конкретным срезом);
- комбинировать переменные для ясности при чтении диаграммы или когда вариация между типами не имеет значения;
- прерывать циклы, где бы вы их ни обнаруживали, вводя промежуточные шаги или выявляя важный аспект взаимосвязи.

В нашем примере мы можем решить, что *ПостоянныйГость*, *Дети* и *Год* не добавляют ценности сверх того, что было зафиксировано посредством *ПредыдущегоАннулирования* и *СреднесуточнойСтоимостиНомера*. И действительно, мы можем обойтись без этих трех переменных, потому что они не смогут спутывать интересующую нас взаимосвязь (рис. 4.20).

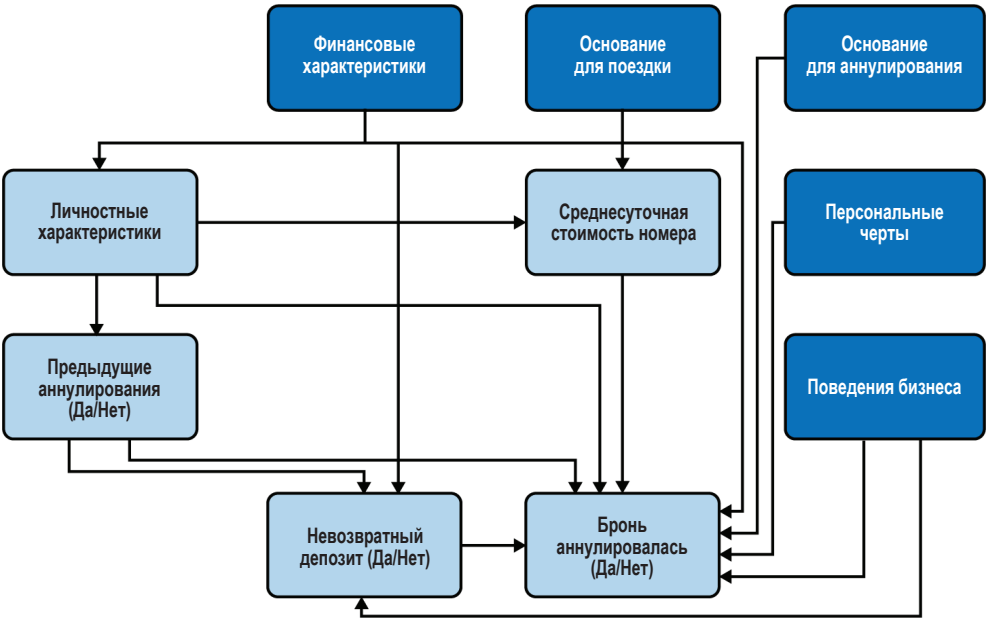

Рис. 4.20 ❖ Окончательная причинно-следственная диаграмма после упрощения

У вас должна остаться чистая и (несколько!) удобочитаемая диаграмма, хотя она, возможно, будет несколько больше, чем те, которые мы встречали до сих пор.

Если этот процесс выглядит долгим и несколько утомительным, то это потому, что он таким и является. Очень хорошее понимание вашего бизнеса – это цена, которую нужно заплатить, чтобы иметь возможность делать причинно-следственные выводы о поведении клиентов (или сотрудников), по меньшей мере в какой-то степени валидные, когда вы не в состоянии проводить эксперименты.

К счастью, этот процесс является чрезвычайно кумулятивным и переносимым. Проделав все это для некоторого анализа, ваши знания о причинно-следственных взаимосвязях, важных для вашего бизнеса, можно использовать повторно для очередного анализа. Даже если в первый раз вы не очень глубоко погружаетесь в этот процесс, вы можете просто сосредоточиться на одной категории спутывающих факторов и причин; в следующий раз, когда вы будете проводить этот или аналогичный анализ, вы сможете продолжить с того места, где вы остановились, и погрузиться глубже в еще одну категорию, возможно, проведя собеседование с клиентами о другом аспекте их

опыта. Схожим образом, после того как кто-то из сотрудников прошел через этот процесс, новый член коллектива или сотрудник сможет очень легко и быстро приобретать соответствующие знания и продолжать с того места, на котором тот остановился, просмотрев полученную причинно-следственную диаграмму или даже просто список релевантных переменных, которые следует иметь в виду.

Выводы

Как говорится в избитом высказывании, строительство причинно-следственной диаграммы – это искусство и наука. Я сделал все возможное, чтобы предоставить для этого как можно более четкий рецепт.

1. Начать со взаимосвязи, которую вы пытаетесь измерить.
2. Выявить кандидатные переменные для включения в диаграмму. А именно использовать свои знания в области бихевиористики и деловой опыт для выявления переменных, которые, вероятно, влияют на любую из интересующих вас переменных.
3. Подтвердить наблюдаемые переменные, которые следует включить, основываясь на их корреляциях в данных.
4. Итеративно расширить свою причинно-следственную диаграмму, добавляя косвенные индикаторы для ненаблюдаемых переменных, где это возможно, и внося дополнительные причины переменных, включенных до этого.
5. Наконец, упростить свою причинно-следственную диаграмму, удалив ненужные взаимосвязи и переменные.

При этом всегда следует иметь в виду свою конечную цель: измерение причинно-следственного воздействия интересующей вас причины на интересующее вас следствие. В следующей главе мы увидим, как использовать причинно-следственную диаграмму, чтобы устранять спутывания в вашем анализе и получать несмещенную оценку этого воздействия. Таким образом, наилучшая причинно-следственная диаграмма – это такая, которая позволяет вам использовать имеющиеся у вас в настоящее время данные наилучшим образом и способствует плодотворному дальнейшему исследованию.

Глава 5

Использование причинно-следственных диаграмм для распутывания аналитических расчетов

Причинно-следственные связи настолько важны для нашего понимания мира, что даже воспитанник детского сада интуитивно их понимает. Однако эта интуиция – и наши аналитические расчеты на данных – могут быть сбиты с пути *спутыванием*, как мы видели в главе 1. Если мы не объясним общую причину двух интересующих нас переменных, то мы неправильно истолкуем происходящее, и коэффициент регрессии для интересующей нас причины будет систематически смещен. Однако мы также увидели риски, связанные с принятием в расчет неправильных переменных. В силу этого вопрос выявления того, какие переменные включать или не включать, становится одним из наиболее важных вопросов в распутывании аналитических расчетов, проводимых в отношении данных и, в более широком смысле, в причинно-следственном мышлении.

Увы, это сложный вопрос, поскольку различные авторы предлагают различные правила, которые более или менее обширны. На более широком конце спектра у вас есть правила, которые склоняются к осторожности и простоте – их можно трактовать как разумные подходы в стиле «все, что есть, и кухонная раковина в придачу». На другом конце спектра у вас есть правила, которые пытаются сфокусироваться на точных требуемых переменных и ни на чем больше, но ценой более высокой сложности и концептуальных требований.

Интересно, что для ответа на этот вопрос не требуется никаких данных. То есть вы, возможно, захотите или вам понадобятся данные для строительства правильной причинно-следственной диаграммы, но как только у вас будет правильная причинно-следственная диаграмма, вам не нужно будет обращаться к каким-либо данным, чтобы выявлять спутывание. Это ставит нас прямо на ребро нашего каркаса «от причинно-следственной диаграммы к поведениям» (рис. 5.1), и, как следствие, в этой главе мы не будем использовать какие-либо данные.

Вместо этого я покажу вам два правила распутывания с разными плюсами и минусами, «критерий дизъюнктивной причины» и «критерий боковой двери», чтобы иметь возможность выбирать, какое из них использовать в зависимости от вашей ситуации. Я изложу нашу деловую задачу в следующем далее разделе, прежде чем, в свою очередь, обратиться к вопросу о том, как применять эти два критерия.

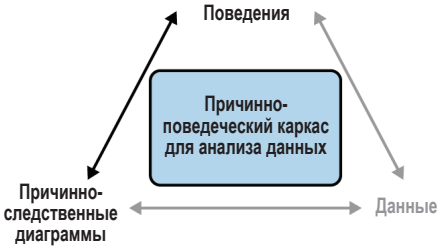

Рис. 5.1 ❖ Эта глава посвящена взаимосвязи между поведениями и причинно-следственными диаграммами

Деловая задача: продажи мороженого и бутилированной воды

Нашей отправной точкой в этой главе является опубликование отделом маркетинга C-Mart внутреннего отчета под названием «Здоровый покупатель», в котором прослеживается долгосрочный тренд к более здоровым продуктам. Основываясь на этом отчете, C-Mart запустила маркетинговую кампанию под названием «Хотите родниковую воду в придачу?» для своих концессионных киосков по продаже продуктов быстрого питания и мороженого. Наша аналитическая цель состоит в получении несмещенной оценки влияния продаж мороженого на продажи бутилированной воды.

Используя существующие данные и специальные опросы, коллектив маркетинговой аналитики сформулировал следующую ниже причинно-следственную диаграмму, в которой интересующая нас связь выделена более толстыми рамками (рис. 5.2).

Приведенная выше причинно-следственная диаграмма умеренно сложна, и не сразу очевидно, где могут скрываться спутывания, поэтому давайте разберем ее на более управляемые части (рис. 5.3).

Рис. 5.2 ❖ Причинно-следственная диаграмма для нашей деловой ситуации

Два приведенных выше концептуальных блока являются всего лишь педагогическим инструментом для понимания: они не являются эксклюзивными (интересующая нас связь является частью обоих) и не являются исчерпывающими (некоторые стрелки не указаны ни в одном из них).

Наш первый блок, в верхнем левом углу причинно-следственной диаграммы, показывает соединения между продажами мороженого и продажами гамбургеров и картофеля фри через число покупателей. В магазинах с большой проходимостью и в более загруженные дни совокупные продажи, как правило, выше, что побуждает различные переменные двигаться согласованно. В дополнение к этому сотрудникам магазина было поручено добавлять реплику «Хотите родниковую воду в придачу?» как для продаж мороженого, так и для продаж картофеля фри, поверх вопроса «Хотите картофель фри в придачу?» для продаж гамбургеров.

Наш второй блок, в правом нижнем углу причинно-следственной диаграммы, показывает влияние двух факторов, которые были выявлены в ходе опросов, но недоступны на уровне отдельных продаж: средний возраст покупателей (более молодые покупатели и покупатели с детьми чаще покупают сладкие продукты) и ориентированность покупателей на здоровье (покупатели, ориентированные на здоровье, чаще покупают воду и реже покупают газированные напитки, при прочих равных условиях).

В реально существующих условиях не стесняйтесь разбирать крупную или сложную причинно-следственную диаграмму на блоки так, как вы считаете нужным для ваших аналитических расчетов. Совершенно нормально, если в конце вы проведете некоторые служебно-хозяйственные мероприятия и проверите пути из интересующей вас причины к интересующему вас следствию, которые не являются частью какого-либо блока: вы должны обеспечивать, чтобы они не генерировали спутывания, потому что они не спутывают вообще либо потому что их спутывание было улажено при анализировании концептуальных блоков.

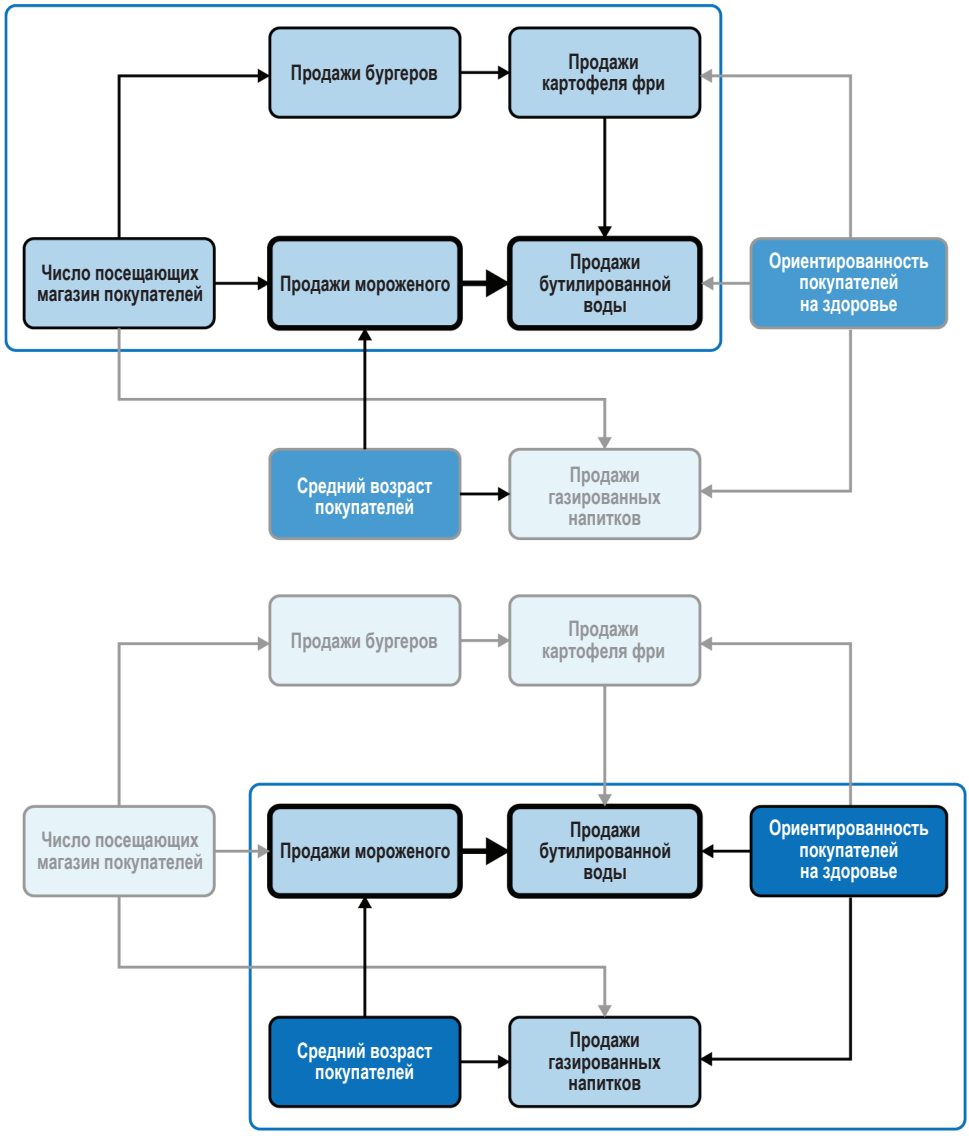

Рис. 5.3 ❖ Разложение нашей причинно-следственной диаграммы на концептуальные блоки

В этой ситуации не сразу ясно, является или нет интересующая нас связь между *ПродажамиМороженого* и *ПродажамиБутилированнойВоды* предметом спутывания, и если да, то как его улаживать. Технически говоря, на причинно-следственной диаграмме у нас нет никакой общей причины для обоих. Давайте обратимся к нашим правилам принятия решений.

Критерий дизъюнктивной причины

Критерий дизъюнктивной причины является нашим первым правилом принятия решений для распутывания. Как и чрезмерно заботливый родитель, оно выходит за рамки того, что строго необходимо для распутывания, что делает его проще для понимания и применения.

Определение

Критерий дизъюнктивной причины (disjunctive cause criterion, аббр. DCC) гласит, что:

> Добавление в нашу регрессию всех переменных, которые являются прямой причиной обеих либо любой из интересующих нас переменных, за исключением посредников между ними, устраняет любое спутывание в интересующей нас связи.

Первый блок

Давайте начнем с разложения этого определения, основываясь на первом блоке в нашем примере с мороженым.

1. Все переменные, которые являются прямой причиной обеих либо любой из интересующих нас переменных.
   Это означает, что мы должны включать любую переменную, которая является прямой причиной только *ПродажМороженого*, такую как *ЧислоПокупателей*. Мы также должны включать любую переменную, которая является причиной только *ПродажБутилированнойВоды*, например *ПродажиКартофеляФри*. И наконец, мы должны включать любую причину и того, и другого, но в данной ситуации у нас ее нет.

2. За исключением посредников между ними.
   Посредники – это переменные, которые «передают» воздействие интересующей нас причины на интересующее нас следствие. То есть они являются дочерним элементом интересующей нас причины и родительским элементом интересующего нас следствия. Мы рассмотрим посредников подробнее в главе 12, поэтому сейчас я просто отмечу, что нам нужно исключить их из нашего списка контрольных переменных, потому что их включение отменит некоторые причинно-следственные связи, которые мы пытаемся уловить. У нас нет посредников между *ПродажамиМороженого* и *ПродажамиБутилированнойВоды* (т. е. переменной, которая была бы дочерней по отношению к первой и родительской по отношению ко второй), поэтому на данном фронте у нас все в порядке.

3. Устраняет любое спутывание в интересующей нас связи.
   Если мы включим переменные, описанные в пункте 1, но не те, которые описаны в пункте 2, то наш регрессионный коэффициент эффекта *Про-*

дажМороженого на *ПродажиБутилированнойВоды* будет распутан по отношению к переменным в нашем первом блоке.

Важно отметить, что критерий дизъюнктивной причины является достаточным, но не необходимым правилом: его применения достаточно, чтобы устранить спутывание, но у вас нет необходимости это делать. Например, если у нас есть переменная, которая является причиной только одной из интересующих нас переменных, и мы уверены, что она абсолютно никак не связана с какой-либо другой переменной, то она не может быть спутывающим фактором, и нам не нужно включать ее, чтобы устранить спутывание.

Но когда у вас нет такой уверенности, критерий дизъюнктивной причины избавляет вас от мучительных размышлений о том, какая переменная является причиной чего и какая является или не является спутывающим фактором. Возможно, вам не будет хватать некоторых взаимосвязей между переменными либо вы будете считать, что взаимосвязи существуют там, где их нет; вы, возможно, будете полагать, что переменная является спутывающим фактором, когда это не так, или наоборот. До тех пор, пока вы правильно определяете, что переменная имеет прямую причинно-следственную связь с одной из двух интересующих вас переменных, вы будете принимать правильное решение о ее включении или невключении.

Например, давайте вглянем на цепочку из *ЧислаПокупателей* в *ПродажиБутилированнойВоды* через *ПродажиБургеров* и *ПродажиКартофеляФри*. В главе 2 мы убедились, что цепочка – это причинно-следственная диаграмма, которая соединяет переменные стрелками по прямой линии (рис. 5.4).

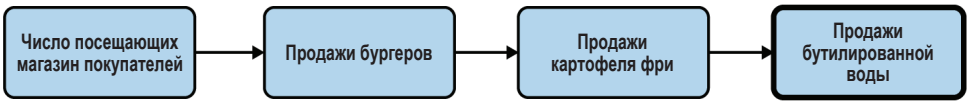

Рис. 5.4 ❖ Расширенная цепочка в нашем первом блоке

Конечно же, мы могли бы представить эту цепочку стрелками, идущими вверх, вниз или справа налево; ключевым моментом является то, что все они движутся в одном направлении, что позволяет нам свернуть цепочку и трактовать *ЧислоПокупателей* как прямую причину *ПродажБутилированнойВоды*. Но тогда *ЧислоПокупателей* действительно является общей прямой причиной *ПродажМороженого* и *ПродажБутилированнойВоды* и выступает спутывающим фактором их взаимосвязи (рис. 5.5).

По определению критерия дизъюнктивной причины его применение к этому первому блоку означает включение в нашу регрессию как *ЧислаПокупателей*, так и *ПродажКартофеляФри* в качестве контрольных переменных. Взглянув на рис. 5.5, мы видим, что это эффективно нейтрализует спутывающий эффект верхней цепочки. В более общем случае, поскольку цепочки можно расширять или сворачивать по желанию, переменная, которая в конечном счете является причиной обеих интересующих нас переменных (и, следовательно, спутывающим фактором), возможно, будет скрыта за последовательностью промежуточных переменных на причинно-следственной диаграмме.

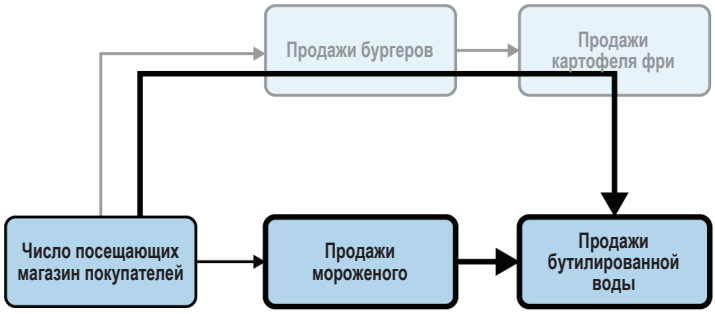

Рис. 5.5 ❖ Сворачивание верхней цепочки делает *ЧислоПокупателей* прямой причиной *ПродажБутилированнойВоды*

Прелесть критерия дизъюнктивной причины заключается в том, что даже если бы маркетинговый коллектив пропустил верхнюю цепочку из *Числа-Покупателей* в *ПродажиБутилированнойВоды* и она не была бы включена в причинно-следственную диаграмму, требование о включении как *Числа-Покупателей*, так и *ПродажКартофеляФри* позаботилось бы о спутывании. С другой стороны, основываясь на рис. 5.5, мы видим, что было бы достаточно включить только *ЧислоПокупателей*, а включение *ПродажКартофеляФри* является избыточным. Это один из компромиссов, о которых я упоминал во введении в главу: критерий дизъюнктивной причины является расширяющим правилом, которое будет устранять спутывание даже при наличии ошибок на вашей причинно-следственной диаграмме, но ценой избыточности и необходимости дополнительных данных. Давайте теперь перейдем ко второму блоку нашей причинно-следственной диаграммы.

Второй блок

Второй блок имеет более сложные взаимосвязи между переменными (рис. 5.6).

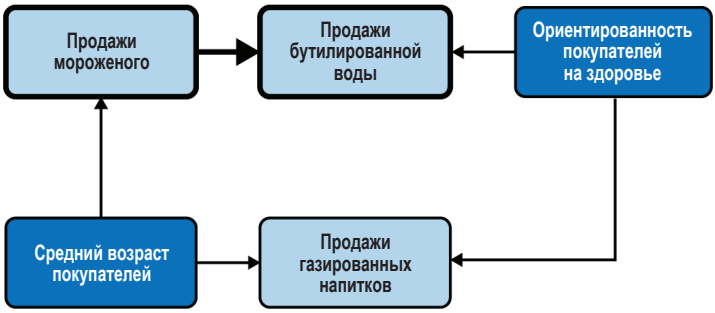

Рис. 5.6 ❖ Второй блок

Здесь единственная переменная, по которой у нас есть данные, выходящие за рамки интересующих нас переменных, – это *ПродажиГазированных-*

Напитков. Она не является причиной *ПродажМороженого* или *ПродажБутилированнойВоды*, поэтому критерий дизъюнктивной причины не будет требовать ее включения в регрессию. Однако он потребовал бы включения как *СреднегоВозрастаПокупателей*, так и *ОриентированностиПокупателей-НаЗдоровье*, для которых у нас нет данных. Это не обязательно означает, что происходит спутывание, но мы не можем быть уверены, что это не так. Самое большое ограничение критерия дизъюнктивной причины состоит в следующем: если у вас нет данных о некоторых причинах интересующих вас переменных, то он вам не поможет. Давайте теперь обратимся к критерию боковой двери.

Критерий боковой двери

Критерий боковой двери (backdoor criterion, аббр. BC) представляет собой альтернативное правило для контроля за спутывающими факторами. Он предлагает совсем другие компромиссы по сравнению с критерием дизъюнктивной причины: он сложнее для понимания и требует наличия полностью точной причинно-следственной диаграммы, но фокусируется на фактических спутывающих факторах и не требует включения какой-либо избыточной переменной в нашу регрессию. С формальной точки зрения, для устранения спутывания необходимо и достаточно отслеживать переменные, определенные этим правилом.

Определения

Критерий боковой двери гласит, что:

> Причинно-следственная связь между двумя переменными спутана, если между ними существует хотя бы один неблокированный непричинно-следственный путь, начинающийся со стрелки, ведущей к интересующей нас причине.
>
> И наоборот, в целях устранения всего спутывания нам нужно заблокировать все непричинно-следственные пути между ними, начиная со стрелки, ведущей к интересующей нас причине.

В целях понимания этого определения нам нужно будет ввести или вспомнить множество вторичных определений в контексте причинно-следственной диаграммы нашего примера, приводимого повторно ниже (рис. 5.7).

Прежде всего давайте вспомним определение пути: мы говорим, что между двумя переменными существует путь, если между ними есть стрелки, независимо от направления стрелок, и если ни одна переменная не появляется на этом пути дважды. Цепочка является путем по трем или более различным направлениям, но такими же являются и развилки и сталкиватели. В этом смысле любые две переменные на причинно-следственной диаграмме соединены по меньшей мере одним путем и, как правило, несколькими.

Например, на нашей причинно-следственной диаграмме существует семь четко различимых путей между *ЧисломПокупателей* и *ПродажамиБутилированнойВоды*:

1) *ЧислоПокупателей* → *ПродажиМороженого* → *ПродажиБутилированнойВоды*;
2) *ЧислоПокупателей* → *ПродажиБургеров* → *ПродажиКартофеляФри* → *ПродажиБутилированнойВоды*;
3) *ЧислоПокупателей* → *ПродажиБургеров* → *ПродажиКартофеляФри* ← *ОриентированностьПокупателейНаЗдоровье* → *ПродажиБутилированнойВоды*;
4) *ЧислоПокупателей* → *ПродажиБургеров* → *ПродажиКартофеляФри* ← *ОриентированностьПокупателейНаЗдоровье* → *ПродажиГазированныхНапитков* ← *СреднийВозрастПокупателей* → *ПродажиМороженого* → *ПродажиБутилированнойВоды*;
5) *ЧислоПокупателей* → *ПродажиГазированныхНапитков* ← *ОриентированностьПокупателейНаЗдоровье* → *ПродажиБутилированнойВоды*;
6) *ЧислоПокупателей* → *ПродажиГазированныхНапитков* ← *ОриентированностьПокупателейНаЗдоровье* → *ПродажиКартофеляФри* → *ПродажиБутилированнойВоды*;
7) *ЧислоПокупателей* → *ПродажиГазированныхНапитков* ← *СреднийВозрастПокупателей* → *ПродажиМороженого* → *ПродажиБутилированнойВоды*.

Рис. 5.7 ❖ Причинно-следственная диаграмма нашей деловой ситуации

Обратите внимание, что последовательность *ЧислоПокупателей → ПродажиБургеров → ПродажиКартофеляФри ← ОриентированностьПокупателейНаЗдоровье → ПродажиГазированныхНапитков ← ЧислоПокупателей → ПродажиМороженого → ПродажиБутилированнойВоды* не является путем, потому что переменная *ЧислоПокупателей* появляется в ней дважды, что запрещено.

Путь является причинно-следственным, если он представляет собой цепочку, то есть все стрелки в нем идут в одном направлении. Ярлык «причинно-следственный» относится к тому факту, что путь между двумя переменными является причинно-следственным, если одна из двух переменных является причиной другой на этом пути.

Пути 1 и 2 в приведенном выше списке являются причинно-следственными: они являются цепочками и представляют каналы, через которые *ЧислоПокупателей* воздействует на *ПродажиБутилированнойВоды*. Другие пути являются непричинно-следственными, потому что каждый из них включает в свой состав по меньшей мере один сталкиватель или развилку. Напомним, что сталкиватель – это ситуация, когда две переменные являются причинами одной и той же, тогда как развилка – это ситуация, когда две переменные обуславливаются одной и той же. Например, пути 3 и 4 имеют сталкиватель вокруг *ПродажКартофеляФри*, а путь 4 тоже включает в себя сталкиватель вокруг *ПродажГазированныхНапитков* и имеет две развилки вокруг *ОриентированностиПокупателейНаЗдоровье* и *СреднегоВозрастаПокупателей*.

Наконец, мы будем говорить, что путь между двумя переменными на нашей причинно-следственной диаграмме блокирован, если либо:

- одна из промежуточных переменных на этом пути включена в регрессию, и это не сталкиватель, либо
- на этом пути находится сталкиватель, центральная переменная которого не включена в нашу регрессию.

В противном случае этот путь является неблокированным.

Понятие «блокированный» или «неблокированный» трудно усвоить, потому что оно на самом деле содержит в себе два разных вопроса: является ли путь сам по себе спутывающим или нет, и контролируется ли он в нашей регрессии. Ярлык «неблокированный» можно трактовать как {спутанный *и* неконтролируемый}, а «блокированный» – как {неспутанный *или* контролируемый}.

Окончательной коренной причиной спутывания всегда является общая причина (рис. 5.8, левая панель). Однако, поскольку мы можем сворачивать или расширять цепочки по желанию, этот спутывающий фактор, возможно, будет «скрыт» за рядом промежуточных переменных (рис. 5.8, средняя панель). Однако мы не можем свернуть сталкивателя в середине цепочки, потому что он нарушает направление стрелок (рис. 5.8, правая панель). Следовательно, сталкиватель блокирует спутывание, если только не включить его в нашу регрессию, которая его нейтрализует.

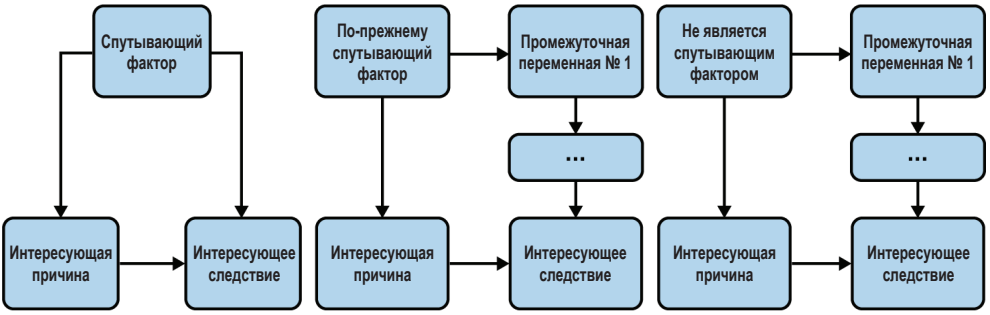

Рис. 5.8 ❖ Спутывающий фактор – это общая причина (левая панель), но он может быть скрыт за промежуточными переменными (средняя панель), тогда как сталкиватель не дает сворачивать цепочки и, следовательно, устраняет спутывание (правая панель)

Первый блок

Теперь, когда мы рассмотрели определение критерия боковой двери, давайте посмотрим, как он применяется к переменным в первом блоке причинно-следственной диаграммы. Вспомните, что критерий дизъюнктивной причины требовал, чтобы мы включили в нашу регрессию как *ЧислоПокупателей*, так и *ПродажиКартофеляФри* в качестве контрольных переменных.

Мы можем начать применять критерий боковой двери с условия «начиная со стрелки, указывающей на интересующую нас причину», что просто означает все причины интересующей нас причины, то есть в данном случае *ПродажиМороженого*. В первом блоке есть только одна из них, а именно *ЧислоПокупателей*.

Для каждого пути через *ЧислоПокупателей* давайте применим другие условия критерия боковой двери. Первый путь из *ПродажМороженого* в *ПродажиБутилированнойВоды* через *ЧислоПокупателей* в первом блоке – это *ПродажиМороженого* ← *ЧислоПокупателей* → *ПродажиБургеров* → *ПродажиКартофеляФри* → *ПродажиБутилированнойВоды*. Это путь, который критерий дизъюнктивной причины уловил и отслеживал путем включения в нашу регрессию *ЧислаПокупателей* и *ПродажКартофеляФри*. Давайте проверим эти условия:

- является ли этот путь непричинно-следственным? Да, из-за развилки вокруг *ЧислаПокупателей*;
- блокирован ли этот путь по умолчанию? Нет, так как на этом пути нет сталкивателя, и мы еще не включили какую-либо переменную в качестве контрольной.

Следовательно, этот путь спутывает интересующую нас связь, и нам нужно этот путь отслеживать путем включения в нашу регрессию любой из его переменных-несталкивателей. То есть критерий боковой двери говорит о том, что для отслеживания этого пути будет достаточно включения любой переменной (из *ЧислаПокупателей*, *ПродажБургеров*, *ПродажКартофеляФри*). Однако у меня есть личная рекомендация относительно того, какую переменную выбирать: всякий раз, когда вы можете включить первую переменную на этом

пути, т. е. интересующую вас причину, вы должны это делать. В нашем примере это будет *ЧислоПокупателей*. Основанием для этого выбора является то, что это также будет автоматически контролировать любой другой спутывающий путь, начинающийся с этой переменной, из чего следует, что нам даже не нужно проверять любой другой путь, начинающийся с этой переменной.

Хорошо видно, что критерий боковой двери экономичнее, чем критерий дизъюнктивной причины, эффективно задействуя допущение о том, что у нас есть полная и правильная причинно-следственная диаграмма для переменных в этом блоке: в отличие от критерия дизъюнктивной причины, который просил нас включить как *ЧислоПокупателей*, так и *ПродажиКартофеляФри*, критерий боковой двери требует включения только *ЧислаПокупателей*, и мы можем оставить первый блок без проверки любого другого пути.

Второй блок

Вспомните, что критерий дизъюнктивной причины (DCC) умалчивал о переменных во втором блоке: мы не могли включить переменные *СреднийВозрастПокупателей* и *ОриентированностьПокупателейНаЗдоровье*, потому что у нас нет соответствующих данных, и, следовательно, мы не были уверены в том, что там было неконтролируемое спутывание. Критерий боковой двери (BC) позволит нам быть гораздо более уверенными и точными.

СреднийВозрастПокупателей является причиной интересующей нас причины, *ПродажМороженого*, поэтому давайте проинспектируем путь *ПродажиМороженого* ← *СреднийВозрастПокупателей* → *ПродажиГазированныхНапитков* ← *ОриентированностьПокупателейНаЗдоровье* → *ПродажиБутилированнойВоды*:

- этот путь является непричинно-следственным (т. е. не является цепочкой): он имеет развилку вокруг *СреднегоВозрастаПокупателей*, сталкиватель вокруг *ПродажГазированныхНапитков*, а затем еще одну развилку вокруг *ОриентированностиПокупателейНаЗдоровье*;
- блокирован ли он по умолчанию? Да, из-за сталкивателя вокруг *ПродажГазированныхНапитков*.

Другими словами, этот путь не спутывает, и нам не нужно его отслеживать в нашей регрессии. Более того, включение *ПродажГазированныхНапитков* в нашу регрессию на самом деле создало бы спутывание посредством неблокирования этого пути!

Эта конфигурация двух развилок вокруг сталкивателя настолько своеобразна, что она имеет название: М-образный шаблон, который можно увидеть, перекомпоновав нашу причинно-следственную диаграмму (рис. 5.9). Следует признать, что этот пример может показаться излишне надуманным. Но на всякий случай, если вы почувствовали, что он искусствен и нереален, то обратите внимание на то, что он был адаптирован из примера в книге «Книга вопросов почему» о фактическом табачном судебном разбирательстве в 2006 году, когда включение контроля за использованием ремней безопасности повлияло на оценку влияния курения на рак легких.

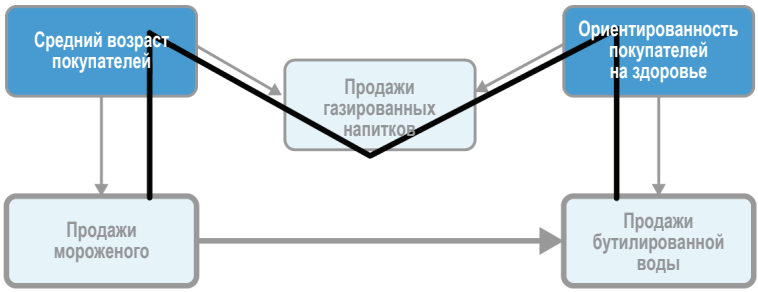

Рис. 5.9 ❖ Визуализация М-образного шаблона на нашей причинно-следственной диаграмме

Помимо этого, поскольку все пути из *ПродажМороженого* в *ПродажиБутилированнойВоды* через *СреднийВозрастПокупателей* также проходят через *ПродажиГазированныхНапитков*, все они будут блокированы до тех пор, пока мы не включим *ПродажиГазированныхНапитков* в нашу регрессию.

Отыскание спутываний на причинно-следственной диаграмме – это наука: применяйте правила, и к вам придет знание. Но оно также имеет свои закулисные пути: выявив две причины *ПродажМороженого*, из-за которых существовала возможность спутывания, и обеспечив, чтобы любое спутывание было блокировано, нам не нужно проверять каждый путь к *ПродажамМороженого*, проходящий через эти две причины. По мере того как вы будете строить и манипулировать все бóльшим числом причинно-следственных диаграмм, вы научитесь развивать в себе интуицию. И если у вас когда-нибудь возникнут сомнения, то вы всегда сможете применить правила по каждому возможному пути, чтобы убедиться, что вы правы.

Критерий боковой двери более прецизионен, чем критерий дизъюнктивной причины; критерий боковой двери менее устойчив к ошибкам, но он также менее устойчив к ошибкам на нашей причинно-следственной диаграмме. Давайте ради аргументации вообразим, что при строительстве причинно-следственной диаграммы маркетинговый коллектив допустил ошибку и ошибочно заключил, что *ПродажаГазированныхНапитков* является причиной *ОриентированностиПокупателейНаЗдоровье*, а не наоборот (в данном конкретном случае это не имеет большого смысла с поведенческой точки зрения, но наберитесь терпения), что приводит к связям, представленным на рис. 5.10.

В этом случае критерий боковой двери приведет нас к ошибочному мнению, что в игре присутствует спутывание, и включит *ПродажиГазированныхНапитков* в нашу регрессию, введя спутывание туда, где его не было.

Подытоживая: критерий боковой двери выявил два потенциальных маршрута спутывания через две прямые причины *ПродажМороженого*. Включив *ЧислоПокупателей* в нашу регрессию, мы позаботимся обо всех возможных спутываниях в ней. С другой стороны, не включив *ПродажиГазированныхНапитков*, мы оставляем в покое сталкивателя, который заботится о любом спутывании через *СреднийВозрастПокупателей*.

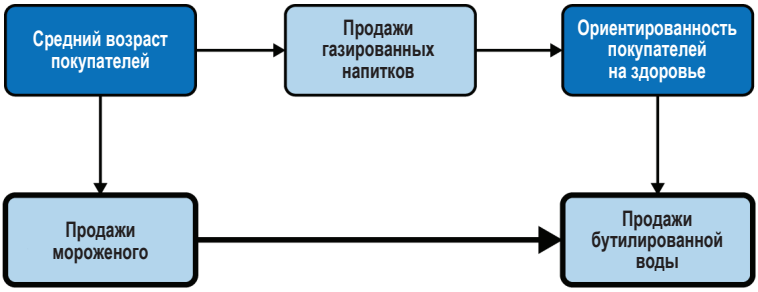

Рис. 5.10 ❖ Вот как будет выглядеть второй блок с ошибкой

Выводы

Распутывание причинно-следственных связей является одной из стержневых трудностей анализа поведенческих данных и решением топкой трясины «корреляция не есть каузация». В этой главе мы рассмотрели два правила распутывания: критерий дизъюнктивной причины и критерий боковой двери. Первое из них придерживается позиции включения всех прямых причин интересующих нас переменных (за исключением посредников). Второе – более хирургично в своем применении и добирается до самого механизма спутывания, но менее интуитивно и менее устойчиво к ошибкам на причинно-следственной диаграмме.

Часть III

УСТОЙЧИВЫЙ АНАЛИЗ ДАННЫХ

Идеальные данные являются крупными, полными и имеют регулярную форму (например, нормально распределены в случае числовых переменных). Это те данные, которые вы видите на вводных курсах статистики. Реально существующие данные часто менее удобны, в особенности когда речь идет о поведенческих данных.

В главе 6 мы рассмотрим принципы работы с пропущенными данными. В то время как пропущенные данные являются обычным явлением в анализе данных, поведенческие данные добавляют уровень сложности: то, какие значения пропущены, часто коррелирует с личностными характеристиками и поведениями, и это вносит систематическое смещение в наши аналитические расчеты. К счастью, использование причинно-следственной диаграммы позволит нам выявлять и разрешать такие ситуации как можно лучше.

В главе 7 мы поговорим о типе компьютерной симуляции, именуемом бутстрапом. Это очень универсальный инструмент, который особенно хорошо подходит для анализа поведенческих данных: он позволяет нам надлежаще измерять неопределенность вокруг наших оценок во время работы с малыми данными или данными странной формы. Более того, при работе с дизайном и анализом экспериментов он предлагает альтернативу *p*-значениям, которая значительно упростит нашу жизнь.

Глава 6

Работа с пропущенными данными

Пропущенные данные в анализе данных являются обычным явлением. В век больших данных многие авторы и еще больше практиков относятся к ним как к незначительной досаде, о которой мало кто думает: просто отфильтруйте строки с пропущенными данными – если вы перейдете от 12 миллионов строк к 11 миллионам, то в чем проблема-то? В результате вы все равно будете оставаться с большим объемом данных, необходимых для проведения своих аналитических расчетов.

К сожалению, отфильтровка строк, в которых данные пропущены, может приводить к значительным ошибкам в вашем анализе. Предположим, что у пожилых клиентов, по всей вероятности, будут пропущены данные, например потому, что они с меньшей вероятностью будут настраивать автоматические платежи; отфильтровывая этих клиентов, вы будете систематически смещать свой анализ в сторону более молодых клиентов, которые будут чрезмерно представлены в ваших отфильтрованных данных. Другие распространенные методы манипулирования пропущенными данными, такие как их замена средним значением этой переменной, тоже вносят свои собственные систематические смещения.

Статистики и методологи разработали методы, которые имеют гораздо меньшее или даже нулевое систематическое смещение. Указанные методы еще не были широко приняты на вооружение практикующими специалистами, но, будем надеяться, эта глава поможет вам оказаться на шаг впереди!

Теория пропущенных значений коренится в статистике и легко может принимать очень математический вид. Для того чтобы сделать наше путешествие в этой главе поконкретнее, мы пройдемся по набору симулированных данных AirCnC. Деловой контекст заключается в том, что отдел маркетинга, стремясь понять характеристики и мотивы клиентов глубже, разослал опрос по электронной почте выборке из 2000 клиентов в трех штатах и собрал следующую ниже информацию:

- демографические характеристики:
  - возраст,
  - пол,

  - штат (отобрав клиентов только из трех штатов, которые для удобства мы будем называть A, B и C);
- личностные черты:
  - открытость,
  - экстраверсия,
  - невротичность;
- сумма брони.

В целях упрощения мы допустим, что все демографические переменные являются причинами суммы брони и не связаны друг с другом (рис. 6.1).

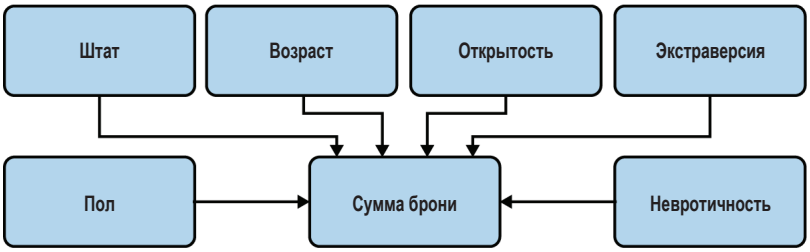

Рис. 6.1 ❖ Демографические переменные являются причиной суммы брони

Как мы обсуждали в главе 2, когда я говорю, что демографические переменные, такие как *Пол* и *Экстраверсия*, являются причинами *СуммыБрони*, я имею в виду две вещи: во-первых, что они являются экзогенными переменными (т. е. переменными, которые для наших целей являются первичными причинами), и, во-вторых, что они являются предсказателями *СуммыБрони* из-за причинно-следственных эффектов, которые в значительной степени опосредуются, а также модерируются социальными явлениями.

Например, эффект *Пола*, вероятно, опосредуется доходом, родом занятий и семейным положением человека, а также многими другими факторами. В этом смысле было бы точнее сказать, что *Пол* является причиной причин *СуммыБрони*. Однако важно отметить, что этот эффект не спутан, и в силу этого он по-настоящему причинно-следственный.

Изложение в этой главе будет следовать шагам, которые вы будете предпринимать, сталкиваясь с новым набором данных: сперва мы будем визуализировать наши пропущенные данные, чтобы получить приблизительное представление о том, что происходит. Затем мы научимся диагностировать пропущенные данные и познакомимся с классификацией, разработанной статистиком Дональдом Рубином, которая является эталоном в этом вопросе. Последние три раздела покажут, как обращаться с каждой категорией из этой классификации.

К сожалению для пользователей Python, отличные пакеты R, которые мы будем использовать, не имеют прямых аналогов на Python. Я сделаю все возможное, чтобы показать вам альтернативы и обходные пути на Python, но исходный код будет значительно длиннее и менее элегантным. Так что заранее прошу прощения!

Данные и пакеты

Одним из роскошеств от использования симулированных данных является то, что мы знаем истинные значения пропущенных данных. Папка этой главы в репозитории на GitHub[1] содержит три набора данных (табл. 6.1):

- полные данные по нашим четырем переменным;
- «располагаемые» данные, в которых для некоторых переменных пропущены некоторые значения;
- вторичный набор данных вспомогательных переменных, которые мы будем использовать в дополнение к нашему анализу.

Таблица 6.1. Переменные в наших данных

	Описание переменной	chap6-complete_data.csv	chap6-available_data.csv	chap6-available_data_supp.csv
Age (Возраст)	Возраст клиента	Полный	Полный	
Open (Открытый)	Психологическая черта открытости, 0–10	Полный	Полный	
Extra (Экстра)	Психологическая черта экстраверсии, 0–10	Полный	Частичный	
Neuro (Невро)	Психологическая черта невротичности, 0–10	Полный	Частичный	
Gender (Пол)	Категориальная переменная пола клиента, Ж/М	Полный	Полный	
State (Штат)	Категориальная переменная штата проживания клиента, A/B/C	Полный	Частичный	
Bkg_amt (СумБрн)	Сумма, израсходованная клиентом на бронь	Полный	Частичный	
Insurance (Страховка)	Сумма приобретенной клиентом туристической страховки			Полный
Active (Активность)	Числовая мера степени активности броней клиента			Полный

В этой главе мы будем использовать следующие ниже пакеты в дополнение к обычным:

```
## R
library(mice)    # Для множественного вменения
library(reshape) # Для функции melt()
library(psych)   # Для функции logistic()

## Python
from statsmodels.imputation import mice  # Для множественного вменения
import statsmodels.api as sm             # Для вызова ОНК в Mice
```

[1] См. https://oreil.ly/BehavioralDataAnalysisCh6.

Визуализация пропущенных данных

По определению, пропущенные данные трудно визуализировать. Одномерные методы (т. е. по одной переменной за раз) помогут нам только до сих пор, поэтому мы будем в основном полагаться на двумерные методы, выводя на графике две переменные в сопоставлении друг с другом, чтобы вычленять некоторые знания. Используемые в сочетании с причинно-следственными диаграммами двумерные графики позволят нам визуализировать взаимосвязи, которые в противном случае были бы очень сложными для понимания.

Наш первый шаг состоит в том, чтобы понять, «как» наши данные пропущены. Пакет mice в R имеет очень удобную функцию `md.pattern()` для визуализации пропущенных данных:

```
## R
> md.pattern(available_data)

    age open gender bkg_amt state extra neuro
368   1    1      1       1     1     1     1    0
358   1    1      1       1     1     1     0    1
249   1    1      1       1     1     0     1    1
228   1    1      1       1     1     0     0    2
163   1    1      1       1     0     1     1    1
214   1    1      1       1     0     1     0    2
125   1    1      1       1     0     0     1    2
120   1    1      1       1     0     0     0    3
33    1    1      1       0     1     1     1    1
23    1    1      1       0     1     1     0    2
15    1    1      1       0     1     0     1    2
15    1    1      1       0     1     0     0    3
24    1    1      1       0     0     1     1    2
24    1    1      1       0     0     1     0    3
23    1    1      1       0     0     0     1    3
18    1    1      1       0     0     0     0    4
      0    0      0     175   711   793  1000 2679
```

Функция `md.pattern()` возвращает таблицу, в которой каждая строка представляет шаблон доступности данных. Первая строка имеет «1» для каждой переменной, поэтому она представляет полные записи. Число слева от таблицы обозначает число строк с этим шаблоном, а число справа обозначает число полей, пропущенных в этом шаблоне. В наших данных 368 полных строк. Вторая строка имеет «0» только для *Невротичности*, поэтому она представляет записи, в которых пропущена только *Невротичность*; у нас 358 таких строк. Цифры в нижней части таблицы обозначают число пропущенных значений соответствующих переменных, а переменные упорядочены по увеличивающемуся числу пропущенных значений. Невротичность – это последняя переменная справа, и, значит, у нее наибольшее число пропущенных значений, 1000. Эта функция также удобно возвращает визуальную демонстрацию таблицы (рис. 6.2).

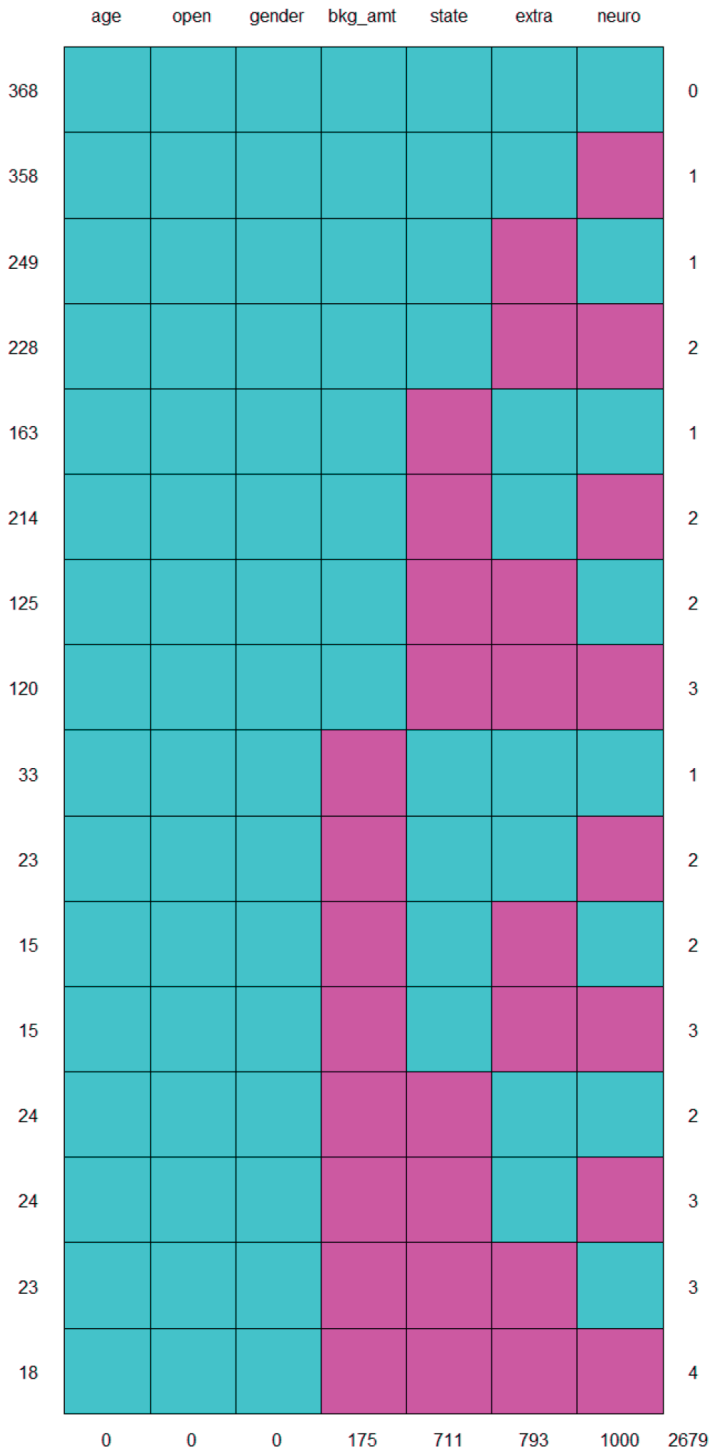

Рис. 6.2 ❖ Шаблоны пропущенных данных

Как мы видим на рис. 6.2, переменные *Возраст*, *Открытость* и *Пол* не имеют пропущенных данных, но у всех остальных переменных они есть. Мы можем получить те же результаты на Python с помощью специальной написанной мной функции, хотя и в менее удобном для чтения формате:

```
## Python
def md_pattern_fun(dat_df):
   # Получение имен всех столбцов
   all_cols = dat_df.columns.tolist()
   # Получение имен столбцов с пропущенными значениями
   miss_cols = [col for col in all_cols if dat_df[col].isnull().sum()]
   if miss_cols == all_cols: dat_df['index'] = dat_df.index
   # Устранение столбцов без пропущенных значений
   dat_df = dat_df.loc[:,miss_cols]
   # Вывод на экран суммарного числа
   # пропущенных значений по каждой переменной
   print(dat_df.isnull().sum())
   # Добавление значения численности
   dat_df['count'] = 1
   # Вывод на экран численности комбинаций пропущенности
   print(dat_df.isnull().groupby(miss_cols).count())
md_pattern_fun(available_data_df)
```

```
extra       793
neuro      1000
state       711
bkg_amt     175
dtype: int64
                           count
extra neuro state bkg_amt
False False False False      368
                  True        33
              True  False     163
                  True        24
      True  False False      358
                  True        23
              True  False     214
                  True        24
True  False False False      249
                  True        15
              True  False     125
                  True        23
      True  False False      228
                  True        15
              True  False     120
                  True        18
```

Результат состоит из двух таблиц:

- в первой таблице указано суммарное число пропущенных значений по каждой переменной в наших данных, как показано внизу рис. 6.2. Экстраверсия имеет 793 пропущенных значения и т. д.;

- во второй таблице представлены сведения о каждом шаблоне пропущенных данных. Переменные выше логических значений слева (т. е. *Экстраверсия*, *Невротичность*, *Штат*, *СуммаБрони*) – это те, у которых в данных пропущено несколько значений. Каждая строка таблицы указывает число строк данных с некоторым шаблоном пропущенных данных. Первая строка состоит из четырех `False`, т. е. шаблона, в котором ни одна из переменных не имеет пропущенных данных, и в наших данных таких строк 368, как вы видели ранее в первой строке на рис. 6.2. Вторая строка изменяет только последнее `False` на `True`, при этом первые три значения `False` опущены для удобства чтения (т. е. любое пустое логическое значение должно читаться). Шаблон `False/False/False/True` возникает, когда только *СуммаБрони* имеет пропущенное значение, что встречается в 33 строках наших данных, и т. д.

Даже с таким небольшим набором данных приведенная выше визуализация очень насыщенна, и в ней бывает трудно понять, что искать. Мы разведаем два аспекта:

объем пропущенных данных.
: Сколько наших данных пропущено? По каким переменным у нас самый высокий процент пропущенных данных? Можно ли просто отбросить строки с пропущенными данными?

корреляция пропущенности.
: Пропущены ли данные на индивидуальном уровне или же на уровне переменных?

Объем пропущенных данных

Первым делом необходимо определить, какая часть наших данных пропущена и какие переменные имеют наибольший процент пропущенных данных. Найти необходимые значения можно в нижней части рис. 6.2, с численностью пропущенных значений в расчете на каждую переменную, в возрастающем порядке пропущенности (missingness), или в нижней части результата Python. Если объем пропущенных данных очень лимитирован, например у вас есть набор данных из 10 миллионов строк, в котором ни одна переменная не содержит более 10 пропущенных значений, то надлежащая их обработка посредством многократного вменения будет излишней, как мы увидим позже. Надо просто удалить все строки с пропущенными данными, и все дела. Аргументация здесь в том, что даже если пропущенные значения крайне смещены, их слишком мало, чтобы каким-либо образом существенно повлиять на результаты вашего анализа.

В нашем примере переменная с наибольшим числом пропущенных значений – это Невротичность с 1000 пропущенных значений. Это много? Где же предел? 10, 100, 1000 строк или больше? Все зависит от контекста. Можно применить кондовую стратегию, которая заключается в следующем.

1. Взять переменную с наибольшим числом пропущенных значений и создать два новых набора данных: один, в котором все пропущенные значения заменяются минимумом этой переменной, и один, в котором они заменяются максимумом этой переменной.
2. Выполнить регрессию для наиболее важной взаимосвязи этой переменной с каждым из трех наборов данных, которые у вас теперь есть. Например, если эта переменная является предсказателем интересующего вас эффекта, то выполнить эту регрессию.
3. Если коэффициент регрессии не отличается в трех регрессиях существенно, т. е. вы бы сделали одинаковые выводы для бизнеса или предприняли одинаковые действия на основе разных значений, то вы находитесь ниже предела и можете удалить эти пропущенные данные. Проще говоря: будут ли эти цифры означать то же самое для ваших деловых партнеров? Если будут, то вы можете удалить эти пропущенные данные.

Указанное эмпирическое правило легко применяется к числовым переменным, но как быть с двоичными или категориальными переменными?

В случае двоичных переменных минимум будет равен 0, а максимум – 1, и это нормально. Два создаваемых вами набора данных транслируются в наилучший и наихудший сценарии.

В случае категориальных переменных правило минимума и максимума должно быть слегка скорректировано: заменить все пропущенные значения наименее частой либо наиболее частой категорией.

Давайте сделаем это ниже, например, для *Невротичности*. *Невротичность* является предсказателем интересующего нас эффекта, *СуммыБрони*, поэтому мы будем использовать эту взаимосвязь, как указано ранее:

```
## R (результат не показан)
min_data <- available_data %>%
  mutate(neuro = ifelse(!is.na(neuro), neuro, min(neuro, na.rm = TRUE)))
max_data <- available_data %>%
  mutate(neuro = ifelse(!is.na(neuro), neuro, max(neuro, na.rm = TRUE)))
summary(lm(bkg_amt~neuro, data=available_data))
summary(lm(bkg_amt~neuro, data=min_data))
summary(lm(bkg_amt~neuro, data=max_data))

## Python (результат не показан)
min_data_df = available_data_df.copy()
min_data_df.neuro = np.where(min_data_df.neuro.isna(), min_data_df.neuro.min(),
                             min_data_df.neuro)
max_data_df = available_data_df.copy()
max_data_df.neuro = np.where(max_data_df.neuro.isna(), max_data_df.neuro.max(),
                             max_data_df.neuro)

print(ols("bkg_amt~neuro", data=available_data_df).fit().summary())
print(ols("bkg_amt~neuro", data=min_data_df).fit().summary())
print(ols("bkg_amt~neuro", data=max_data_df).fit().summary())
```

Результаты таковы:

- коэффициент, основанный на располагаемых данных, составляет –5.9;
- коэффициент, основанный на замене пропущенных значений минимумом *Невротичности*, составляет –8.0;
- коэффициент, основанный на замене пропущенных значений максимумом *Невротичности*, составляет 2.7.

Эти значения очень отличаются друг от друга, вплоть до того, что имеют разные знаки, поэтому мы определенно находимся выше порога материальной значимости. Мы не можем просто отбросить строки, в которых пропущены данные по *Невротичности*. Применение того же подхода к другим переменным тоже показало бы, что мы не можем игнорировать их пропущенные значения и должны обрабатывать их адекватным образом.

Корреляция пропущенности

После того как мы определили переменные, с которыми нам нужно будет поработать, мы захотим узнать, насколько коррелирует их пропущенность. Если у вас есть переменные, пропущенность которых сильно коррелирует, то это говорит о том, что пропущенность одной является причиной пропущенности других (например, если кто-то перестанет отвечать на опрос на полпути, то все ответы после некоторого момента будут пропущены). С другой стороны, их пропущенность может иметь общую причину (например, если некоторые испытуемые неохотнее раскрывают информацию о себе). В обоих этих случаях выявление корреляции пропущенности поможет вам создать более точную причинно-следственную диаграмму, что сэкономит ваше время и сделает ваши аналитические расчеты эффективнее.

Давайте посмотрим на это посредством простой иллюстрации: вообразите, что у нас есть данные собеседований для двух офисов: в Тампе и в Такоме. В обоих офисах кандидаты должны пройти одинаковые обязательные три раздела собеседования, но в Тампе первый проводящий собеседование интервьюер отвечает за регистрацию всех баллов кандидата, тогда как в Такоме каждый интервьюер записывает балл своего раздела. Интервьюеры – это люди, и они иногда забывают передать данные в отдел кадров. В Тампе, когда интервьюер забывает предоставить данные, у нас нет никаких данных о кандидате, кроме его ИД в системе (на рис. 6.3 показаны данные только для Тампы).

Характерным признаком сильной корреляции пропущенности является строка с крупным числом заштрихованных квадратов (здесь 3), которые представляют большое число случаев (здесь 400 из 2000 строк). В дополнение к этому на рисунке нет строки только с одним или двумя светлыми квадратами.

В такой ситуации не имело бы смысла анализировать наши пропущенные данные по одной переменной за раз. Если мы обнаружим, что данные для первого раздела, скорее всего, пропущены, когда первым интервьюером является Мерфи, то это будет справедливо и для других разделов. (У тебя какая-никакая, но была работа, Мерфи!)

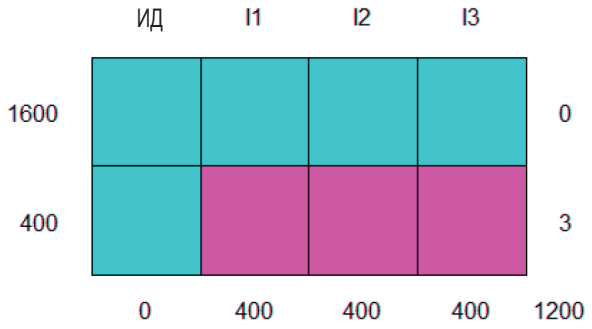

Рис. 6.3 ❖ Сильно коррелированная пропущенность в данных Тампы

С другой стороны, в Такоме пропущенность в разных разделах совершенно не коррелирует (рис. 6.4).

Там шаблон прямо противоположен шаблону Тампы:

- у нас есть большое число строк с малым числом пропущенных переменных (см. все единицы и двойки справа от рисунка);
- эти строки представляют основную часть наших данных (слева мы видим, что только 17 строк имеют 3 пропущенные переменные);
- строки с большим числом светлых квадратов внизу рисунка представляют очень мало случаев (те же 17 индивидуумов), потому что они являются результатом независимой случайности.

Аргументацию в пользу последнего пункта маркерного списка можно продолжить, посмотрев шире на то, что можно было бы назвать последовательностями увеличения пропущенности в стиле «Русской матрешки», в которых каждый шаблон добавляет пропущенную переменную в предыдущий шаблон, например (I3) → (I3, I2) → (I3, I2, I1). Соответствующие числа случаев составляют 262 → 55 → 17. Эти числа образуют убывающую последовательность, что логично, ибо если пропущенность переменных полностью некоррелирована, то мы имеем:

$$Prob(I3\ пропущено\ \&\ I2\ пропущено) = Prob(I3\ пропущено) * Prob(I2\ пропущено);$$

$$Prob(I3\ пропущено\ \&\ I2\ пропущено\ \&\ I1\ пропущено) = Prob(I3\ пропущено) * Prob(I2\ пропущено) * Prob(I1\ пропущено).$$

При малой выборке и/или очень высоких уровнях пропущенности эти уравнения, возможно, не будут строго истинными в наших данных, но если пропущено менее 50 % какой-либо переменной, то мы, как правило, должны иметь:

$$Prob(I3\ пропущено\ \&\ I2\ пропущено\ \&\ I1\ пропущено) < Prob(I3\ пропущено\ \&\ I2\ пропущено) < Prob(I3\ пропущено).$$

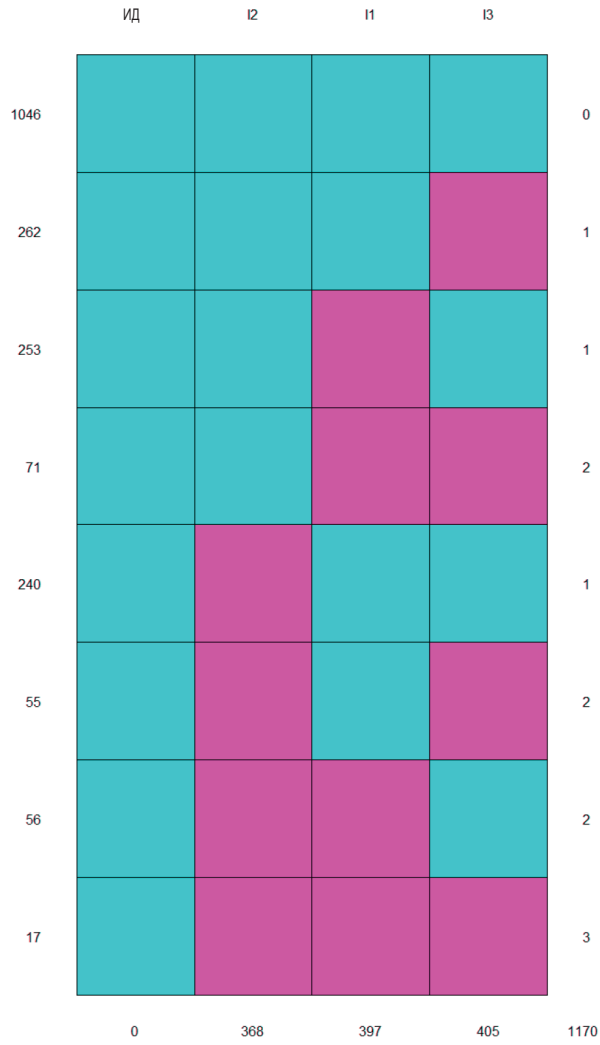

Рис. 6.4 ❖ Некоррелированная пропущенность в данных Такомы

В реально-практических ситуациях было бы довольно громоздко самостоятельно проверять все эти неравенства, хотя вы могли бы написать функцию, чтобы делать это в масштабе. Вместо этого я бы порекомендовал обратиться к визуализации на предмет каких-либо существенных выбросов (т. е. значений нескольких переменных, которые намного больше, чем значения для некоторых из тех же переменных).

В более широком смысле эта визуализация проста в использовании только с несколькими переменными. Как только у вас появится большое число переменных, вам придется строить и визуализировать матрицу корреляций для пропущенности:

```
## R (результат не показан)
# Строительство матриц корреляций
tampa_miss <- tampa %>%
  select(-ID) %>%
 mutate(across(everything(),is.na))
tampa_cor <- cor(tampa_miss) %>%
  melt()

tacoma_miss <- tacoma %>%
  select(-ID) %>%
  mutate(across(everything(),is.na))
tacoma_cor <- cor(tacoma_miss) %>%
  melt()

## Python (результат не показан)
# Строительство матриц корреляций
tampa_miss_df = tampa_df.copy().drop(['ID'], axis=1).isna()
tacoma_miss_df = tacoma_df.copy().drop(['ID'], axis=1).isna()

tampa_cor = tampa_miss_df.corr()
tacoma_cor = tacoma_miss_df.corr()
```

На рис. 6.5 показаны результирующие матрицы корреляций. В матрице слева, например, все значения равны 1: если пропущена одна переменная, то и две другие тоже. В матрице корреляций справа для Такомы значения равны 1 по главной диагонали, но в остальных случаях равны 0: знание того, что пропущена одна переменная, ничего не говорит о том, что пропущены другие.

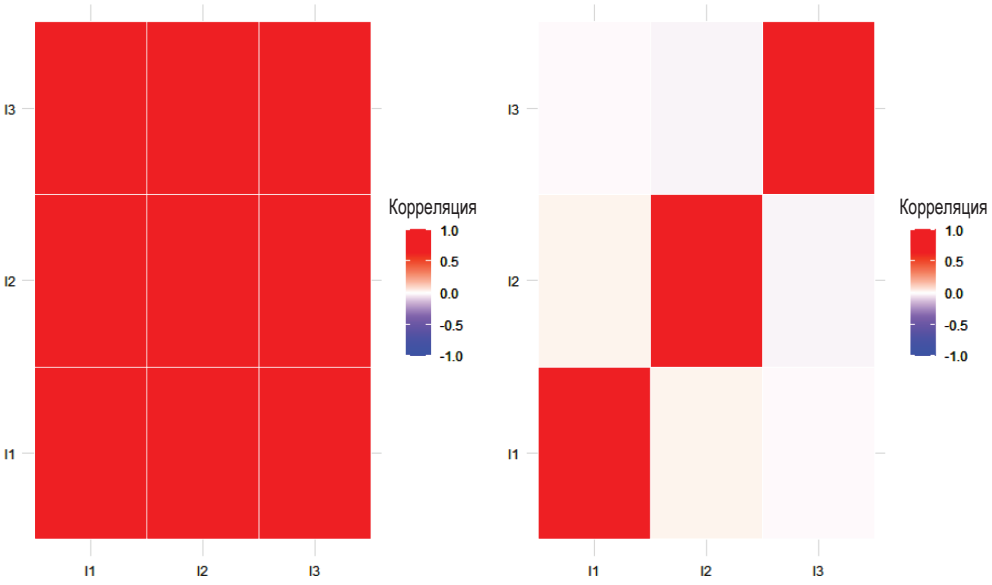

Рис. 6.5 ❖ Матрицы корреляций для полностью коррелированной пропущенности (слева) и полностью некоррелированной пропущенности (справа)

Давайте вернемся к нашему набору данных AirCnC и посмотрим, где он находится между двумя крайностями, описанными в примере теоретического собеседования. Рисунок 6.6 повторяет рис. 6.2 для удобства доступа.

Рисунок 6.6 находится где-то посередине: все возможные шаблоны пропущенности достаточно хорошо представлены, что говорит о том, что у нас нет сильно кластеризованных источников пропущенности.

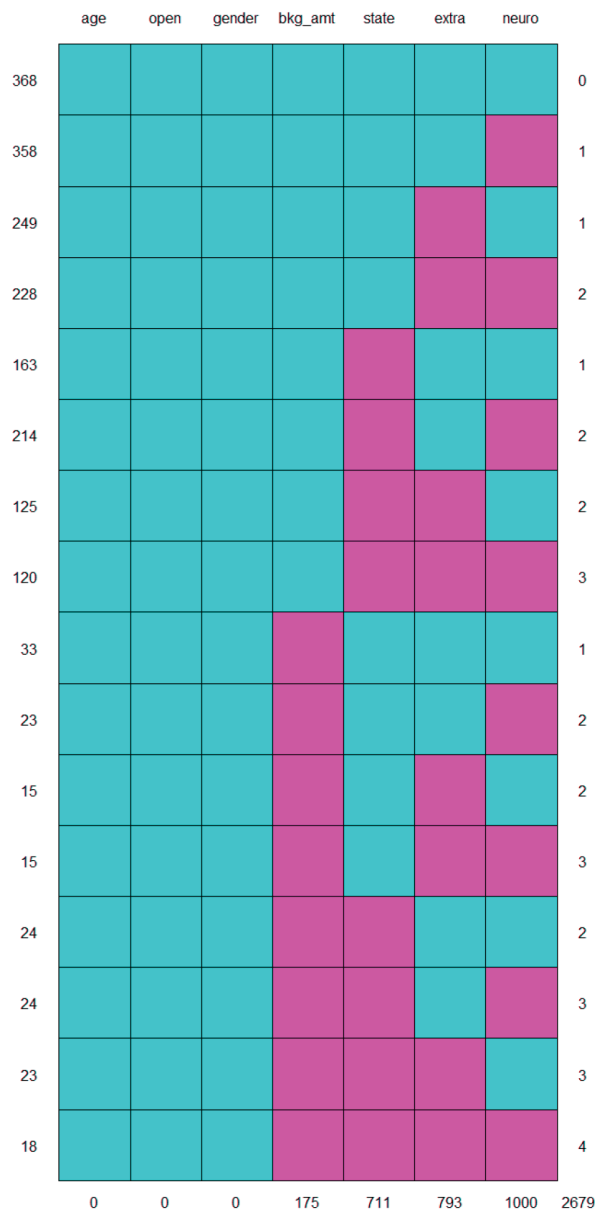

Рис. 6.6 ❖ Шаблоны пропущенных данных (повторяет рис. 6.2)

На рис. 6.7 показана корреляционная матрица пропущенности для данных AirCnC. Хорошо видно, что пропущенность наших переменных почти полностью некоррелирована, что находится в диапазоне случайных колебаний. Если вы хотите ознакомиться с корреляционными шаблонами пропущенности подробнее, то одно из упражнений этой главы в репозитории на GitHub вас попросит выявить некоторые из них. Напомним, что собственно обращаться к шаблонам корреляции совсем нет необходимости, но это часто бывает поучительно и экономит ваше время.

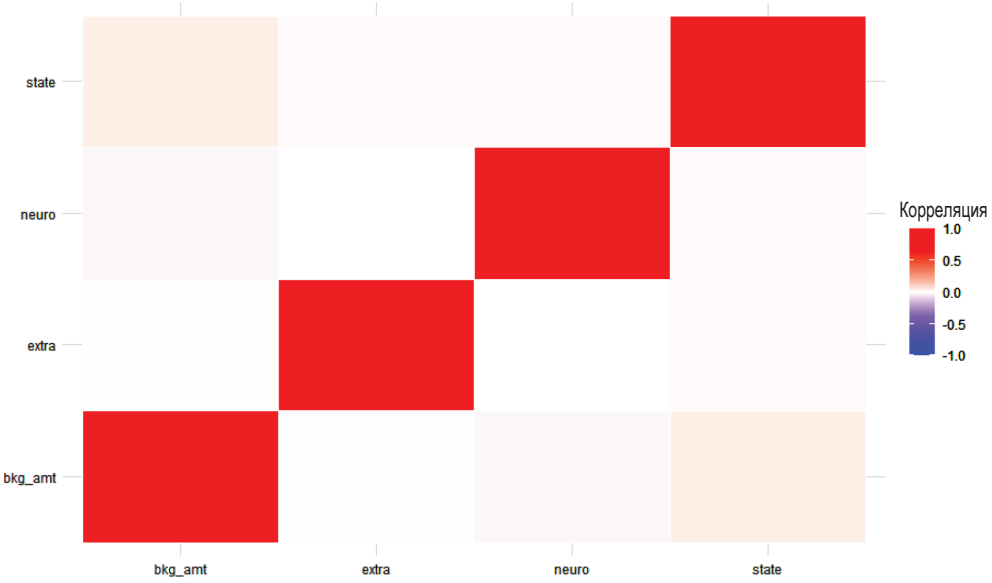

Рис. 6.7 ❖ Корреляционная матрица пропущенности в наших данных AirCnC

Диагностика пропущенных данных

Теперь, когда мы визуализировали наши пропущенные данные, самое время понять, что является их причиной. Здесь на помощь приходят причинно-следственные диаграммы, поскольку мы будем использовать их для представления причинно-следственных механизмов пропущенных данных.

Давайте начнем с очень простого примера из главы 1. Вводя причинно-следственные диаграммы, я упомянул, что ненаблюдаемые переменные, такие как пристрастие покупателя к ванильному мороженому, представлены в более темном прямоугольнике (рис. 6.8).

Ненаблюдаемые переменные, которые в некоторых дисциплинах иногда называют «латентными» переменными, относятся к информации, которой у нас нет на практике, даже если она может иметься или не иметься теоретически. В данном случае, допустим, мы заставляем наших клиентов раскрывать свое предпочтение ванильного вкуса перед совершением по-

купки. В результате в наших системах были бы созданы соответствующие данные, в отношении которых мы затем проводили бы аналитические расчеты (рис. 6.9).

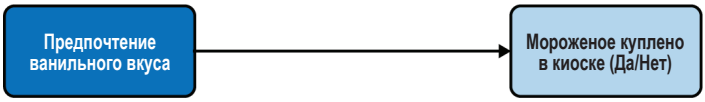

Рис. 6.8 ❖ Ненаблюдаемые переменные представлены в прямоугольнике с более темным оттенком

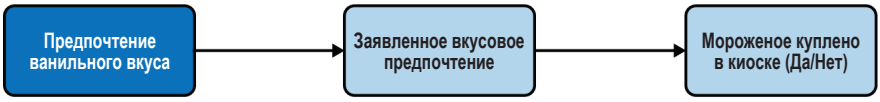

Рис. 6.9 ❖ Сбор ранее не наблюдавшейся информации

Однако, как правило, плохая деловая практика состоит в том, чтобы пытаться заставлять покупателей раскрывать информацию, которую они не хотят раскрывать, и она зачастую остается необязательной. В более общем случае большой объем данных собирается о некоторых клиентах, но не о других. Мы будем представлять эту ситуацию, нарисовав на причинно-следственной диаграмме соответствующий прямоугольник, обрамленный тире (рис. 6.10).

Рис. 6.10 ❖ Представление частично наблюдаемых переменных пунктирным прямоугольником

Например, с тремя покупателями мы могли бы получить следующие ниже данные, при этом один покупатель отказался раскрывать свое вкусовое пристрастие к ванильному мороженому (табл. 6.2).

Таблица 6.2. Данные в основе нашей причинно-следственной диаграммы

Имя покупателя	Предпочтение ванильного вкуса	Заявленное вкусовое предпочтение	Мороженое куплено в киоске (Да/Нет)
Энн	Низкое	Низкое	Нет
Боб	Высокое	Высокое	Да
Кэролин	Высокое	Отсутствует	Да

В этой главе нам интересно понять, что является причиной пропущенности переменной, а не только то, что является причиной значений переменной. Поэтому мы создадим переменную в целях отслеживания ситуации, когда переменная заявленного вкусового пристрастия пропущена (табл. 6.3).

От пропущенных значений к неправильным или ложным значениям

В этой книге мы примем упрощающее допущение, что значения либо правильны, либо пропущены. То есть люди никогда не лгут, не ошибаются при воспоминании и не допускают ошибок при вводе. Разумеется, они делают это в реальной жизни, и это то, что вам придется учитывать, используя идеи из данной главы и главы 2: допустим, что существует скрытая переменная с истинными значениями и наблюдаемая переменная, которая отражает скрытую переменную с некоторым «шумом». Тогда мы определим, с какого рода шумом мы имеем дело: чисто случайным, зависящим от значения еще одной переменной либо зависящим от скрытой переменной, аналогично классификации Рубина (обсуждаемой позже в этой главе). Например, люди, купившие продукт, в использовании которого им, возможно, будет неловко признаваться (парик?), с некоторой вероятностью будут притворяться, что это не так, но обратное крайне маловероятно. В терминологии Рубина это сделало бы *ПокупкиПариков* «ложными неслучайным образом».

Таблица 6.3. Добавление переменной пропущенности

Имя покупателя	Предпочтение ванильного вкуса	Заявленное предпочтение ванильного вкуса	Заявленное вкусовое предпочтение пропущено (Да/Нет)	Мороженое куплено в киоске (Да/Нет)
Энн	Низкое	Низкое	Нет	Нет
Боб	Высокое	Высокое	Нет	Да
Кэролин	Высокое	Отсутствует	Да	Да

Давайте добавим эту переменную на нашу причинно-следственную диаграмму (рис. 6.11).

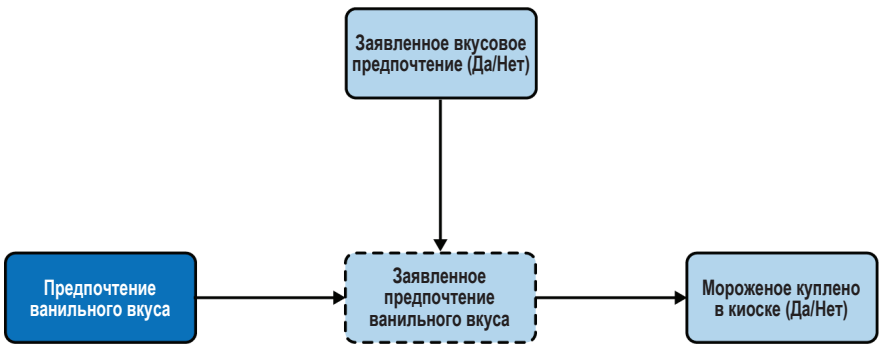

Рис. 6.11 ❖ Добавление пропущенности на причинно-следственную диаграмму

Мы условно делаем пропущенность причиной соответствующей частично наблюдаемой переменной. Интуиция подсказывает, что информация существует полностью в ненаблюдаемой переменной, а частично наблюдаемая переменная равна ненаблюдаемой переменной, если только информация не «скрыта» переменной пропущенности. Эта условность значительно облегчит нашу жизнь, потому что она позволит нам выражать и обсуждать причины

пропущенности на причинно-следственной диаграмме, которая представляет интересующие нас связи, вместо того чтобы вынужденно рассматривать пропущенность отдельно.

Теперь, когда пропущенность является частью нашей причинно-следственной диаграммы, естественным следующим шагом будет спросить себя: «Что является ее причиной?»

Причины пропущенности: классификация Рубина

Существуют три базовые и взаимоисключающие возможности для причины пропущенности переменной, которые были категоризированы статистиком Дональдом Рубином.

Прежде всего если пропущенность переменной зависит только от переменных за пределами наших данных, таких как чисто случайные факторы, то принято говорить, что эта переменная пропущена совершенно случайным образом (missing completely at random, аббр. MCAR) (рис. 6.12).

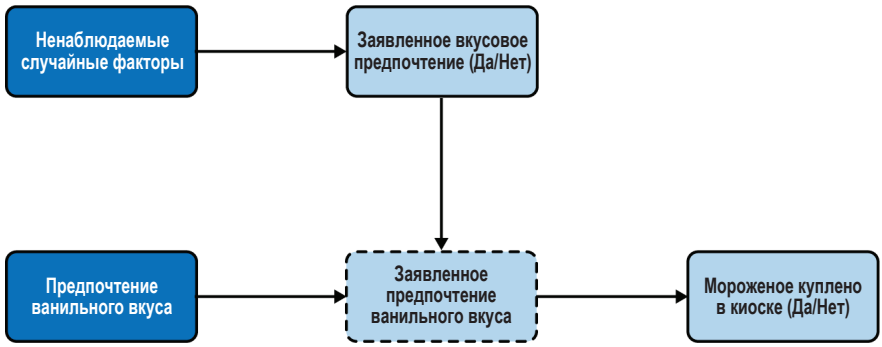

Рис. 6.12 ❖ Заявленное вкусовое предпочтение пропущено совершенно случайным образом

Затем переменная переходит из MCAR к пропущенным случайным образом (missing at random, аббр. MAR), если хотя бы одна переменная в наших данных влияет на ее пропущенность. Переменные, не зависящие от наших данных, и случайные факторы, возможно, тоже будут играть свою роль, но значение рассматриваемой переменной, вероятно, не будет влиять на ее собственную пропущенность. Это имело бы место, например, если бы переменная *Куплено* была бы причиной пропущенности переменной *ЗаявленноеВкусовоеПредпочтение*, например потому, что мы опрашиваем не любых прохожих, а только тех покупателей, которые сделали покупку (рис. 6.13).

Наконец, любая переменная, значение которой влияет на ее собственную пропущенность, пропущена не случайным образом (missing not at random, аббр. MNAR), даже если другие переменные внутри или за пределами данных тоже влияют на пропущенность. Другие переменные внутри или за пределами наших данных тоже могут играть свою роль, но переменная переходит из

MCAR или MAR в MNAR, как только переменная влияет на свою собственную пропущенность. В нашем примере это означало бы, что *ПредпочтениеВанильногоВкуса* является причиной пропущенности *ЗаявленногоПредпочтенияВанильногоВкуса* (рис. 6.14).

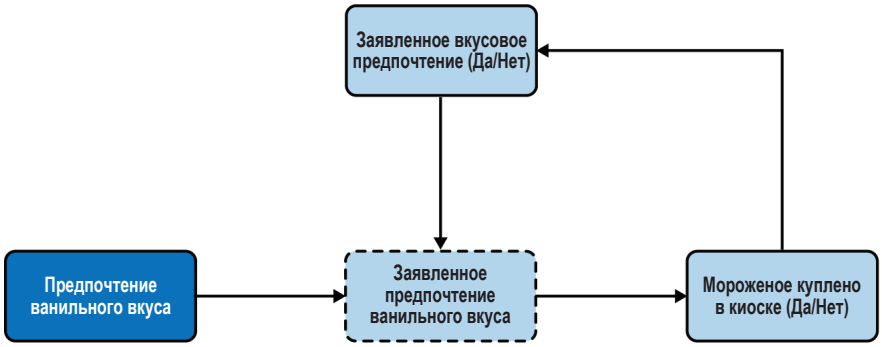

Рис. 6.13 ❖ Заявленное вкусовое предпочтение пропущено случайным образом

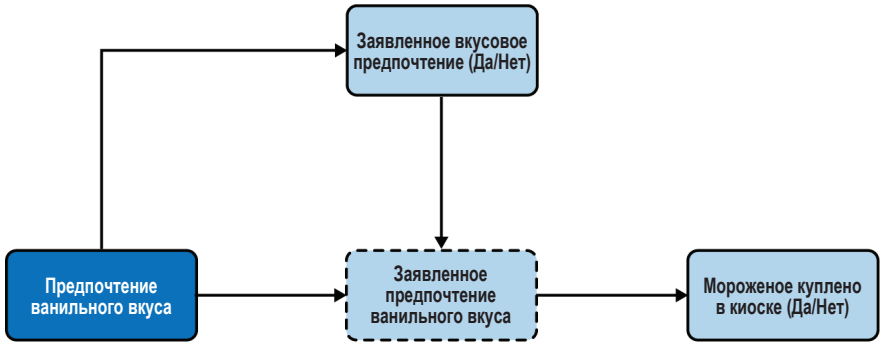

Рис. 6.14 ❖ Заявленное вкусовое предпочтение пропущено неслучайным образом

Мы представляем идею о том, что значения переменной влияют на ее пропущенность, рисуя стрелку не из частично наблюдаемой переменной, а из ненаблюдаемой переменной. Благодаря этому мы можем делать содержательные высказывания, такие как «все значения, которые в действительности ниже некоторого порога, пропущены в наших данных». Если бы стрелка указывала на частично наблюдаемую переменную, то мы бы застряли с неинформативными высказываниями, типа «пропущенные значения являются своей собственной причиной пропущенности».

В идеальном мире остальная часть этого раздела состояла бы из рецептов, позволяющих выявлять каждую категорию пропущенности. К сожалению, анализ пропущенных данных все еще остается открытой областью исследований, которая пока что разведана не полностью. В частности, не совсем понятно, как взаимодействуют пропущенность и причинно-следственная связь. Поэтому работа с пропущенными данными остается скорее искусством, чем наукой. Попытка создать систематические рецепты потребует

решения непрослеживаемого числа исключений, а также введения циклических аргументов, таких как «шаблон X указывает на то, что переменная 1 является MAR, если только переменная 2 не является MNAR; шаблон Y указывает на то, что переменная 2 является MNAR, если только переменная 1 не является MAR». Я сделал все возможное, чтобы охватить как можно больше случаев в рамках ограниченного набора данных, но в реальном мире вы, возможно, столкнетесь с ситуациями, которые являются «немного тем и немного этим», и вам придется выносить суждение о том, как действовать дальше.

Однако есть и немного хороших новостей, а именно что, за некоторыми исключениями, которые я назову, осторожность занимает больше времени, но не привносит систематическое смещение. Если вы не уверены в том, чем является переменная: MCAR, MAR либо MNAR, – просто примите наихудший из возможных сценариев, и ваши аналитические расчеты будут настолько несмещенными, насколько это возможно.

Имея в виду это предостережение, давайте вернемся к нашим данным AirCnC и посмотрим, каким образом можно было бы выявить пропущенные данные в реалистичном наборе данных. В качестве быстрого напоминания наш набор данных содержит следующие переменные:

- демографические характеристики:
  - возраст,
  - пол,
  - штат (A, B и C);
- личностные черты:
  - открытость,
  - экстраверсия,
  - невротичность;
- сумма брони.

Диагностика переменных MCAR

Переменные MCAR являются самым простым случаем. Датчик вышел из строя, ошибка помешала передаче данных из клиентского мобильного приложения, либо клиент попросту пропустил поле ввода, чтобы указать свое пристрастие к ванильному мороженому. При любом раскладе пропущенность происходит в ключе, который является интуитивно «случайным». Мы диагностируем переменные MCAR по умолчанию: если переменная не выглядит как MAR, мы будем трактовать ее как MCAR. Другими словами, вы можете считать MCAR как нашу нулевую гипотезу в отсутствие подтверждений об обратном.

Главным инструментом, который мы будем использовать для диагностики пропущенности, является логистическая регрессия в отношении того, что переменная пропущена на всех других переменных в нашем наборе данных. Давайте рассмотрим, например, переменную *Экстраверсии*:

```
## Python (результат не показан)
available_data_df['md_extra'] = available_data_df['extra'].isnull().astype(float)
md_extra_mod =smf.logit('md_extra~age+open+neuro+gender+state+bkg_amt',
                        data=available_data_df)
```

```
md_extra_mod.fit().summary()

## R
> md_extra_mod <- glm(is.na(extra)~.,
                      family = binomial(link = "logit"),
                      data=available_data)
> summary(md_extra_mod)

...
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.7234738  0.7048598  -1.026    0.305
age         -0.0016082  0.0090084  -0.179    0.858
open         0.0557508  0.0425013   1.312    0.190
neuro        0.0501370  0.0705626   0.711    0.477
gender      -0.0236904  0.1659661  -0.143    0.886
stateB      -0.0780339  0.2000428  -0.390    0.696
stateC      -0.0556228  0.2048822  -0.271    0.786
bkg_amt     -0.0007701  0.0011301  -0.681    0.496
...
```

Ни одна из переменных не имеет большого и сильно статистически значимого коэффициента. В отсутствие каких-либо других подтверждений это говорит о том, что источник пропущенности для *Экстраверсии* является чисто случайным, и мы будем трактовать нашу переменную *Экстраверсии* как MCAR.

Данные MCAR можно рассматривать как бросание кубика или подбрасывание монеты. Оба этих действия являются «случайными» с нашей точки зрения, но они все равно подчиняются законам физики. Теоретически, если бы у нас было достаточно информации и вычислительных мощностей, то исход был бы совершенно предсказуемым. То же самое может произойти и здесь. Говоря, что *Экстраверсия* является MCAR, мы не говорим, что «пропущенность *Экстраверсии* является принципиально случайной и непредсказуемой», мы просто говорим, что «ни одна из переменных, включенных в настоящее время в наш анализ, не коррелирует с пропущенностью *Экстраверсии*». Но, возможно – и даже вероятно, – другие переменные (добросовестность? доверие? знакомство с технологией?) будут коррелированы. Наша цель состоит не в том, чтобы сделать философское заявление об *Экстраверсии*, а в том, чтобы определить, может ли ее пропущенность повлиять на наш анализ, имея располагаемые в настоящее время данные.

Статистическая значимость

В части IV я подробно объясню, что такое статистическая значимость и почему вам не следует придавать ей слишком большое значение. В частности, вы не должны использовать порог 0.05 в качестве какого-либо строгого источника истины. Здесь вы должны просто проявить свое наилучшее суждение о том, имеет коэффициент смысл или нет. Если вы сомневаетесь, то вы всегда можете относиться к коэффициенту так, как если бы он действительно был сильным и статистически значимым, – как мы увидим, это добавит вам лишней работы, но не повлияет на ваши аналитические расчеты.

Диагностика переменных MAR

Переменные MAR – это переменные, пропущенность которых зависит от значений других переменных в наших данных. Если другие переменные в нашем наборе данных предсказывают пропущенность переменной, то MAR становится нашей принятой по умолчанию гипотезой для этой переменной, если у нас нет достаточно веских подтверждений того, что она является MNAR. Давайте посмотрим, как это выглядит с переменной *Штата*:

```
## R (результат не показан)
md_state_mod <- glm(is.na(state)~.,
                    family = binomial(link = "logit"),
                    data=available_data)
summary(md_state_mod)

## Python
available_data_df['md_state'] = available_data_df['state'].isnull()\
   .astype(float)
md_state_mod =smf.logit('md_state~age+open+extra+neuro+gender+bkg_amt',
                        data=available_data_df)
md_state_mod.fit(disp=0).summary()
...
               coef    std err          z      P>|z|    [0.025   0.975]
Intercept   -0.2410    0.809     -0.298      0.766    -1.826   1.344
gender[T.F] -0.1742    0.192     -0.907      0.364    -0.551   0.202
age          0.0206    0.010      2.035      0.042     0.001   0.040
open         0.0362    0.050      0.727      0.467    -0.061   0.134
extra        0.0078    0.048      0.162      0.871    -0.087   0.102
neuro       -0.1462    0.087     -1.687      0.092    -0.316   0.024
bkg_amt     -0.0019    0.001     -1.445      0.149    -0.005   0.001
...
```

Возраст умеренно значим, с положительным коэффициентом. Другими словами, пожилые клиенты, по-видимому, предоставляют свой штат с меньшей вероятностью. Соответствующая причинно-следственная диаграмма показана на рис. 6.15.

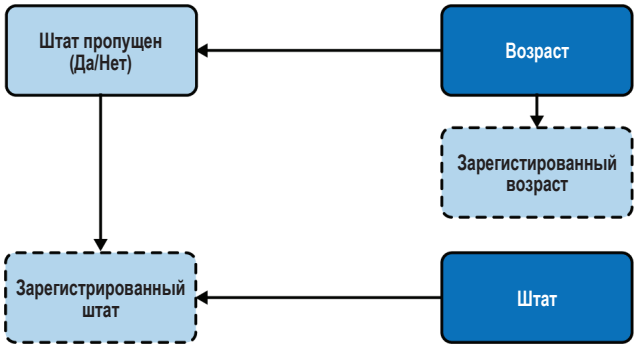

Рис. 6.15 ❖ Пол пропущен случайным образом

Мы можем подтвердить эту корреляцию, построив график плотности пропущенности *Штата* в разбивке по зарегистрированному *Возрасту* (рис. 6.16). *Штат* имеет больше наблюдаемых значений для молодых клиентов, чем для пожилых клиентов, или, наоборот, пропущенных значений больше для пожилых клиентов, чем для молодых клиентов.

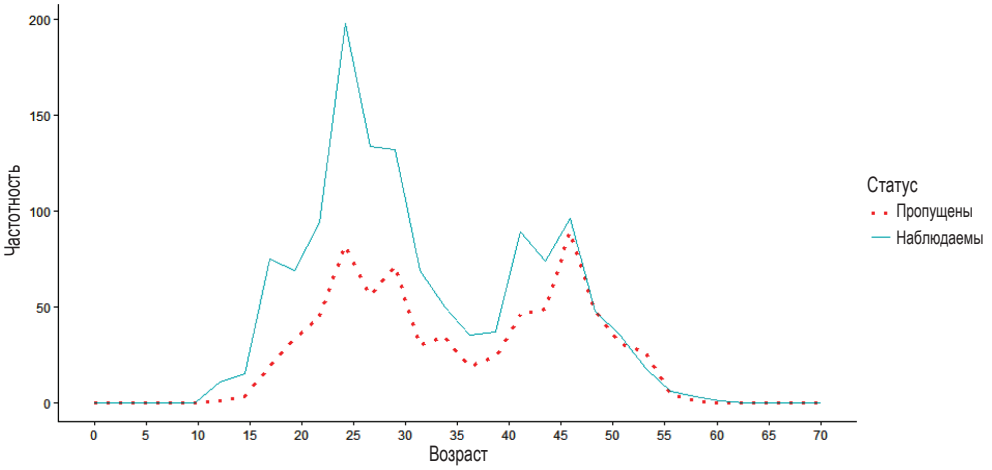

Рис. 6.16 ❖ Плотность пропущенных и наблюдаемых данных о *Штате* в разбивке по наблюдаемому *Возрасту*

Одним из ограничений этого графика плотности является то, что он не показывает строки, в которых также пропущена переменная *X* (здесь *Возраст*). Это может стать проблемой либо ввести в заблуждение, если эта переменная также имеет пропущенные значения. Возможный трюк состоит в том, чтобы заменить пропущенные в переменной *X* значения бессмысленным значением, таким как –10. *Возраст* не имеет пропущенного значения, и поэтому вместо него в качестве переменной *X* мы будем использовать *Экстраверсию*, которая имеет пропущенные значения. Давайте построим график плотности наблюдаемых и пропущенных данных о *Штате* в разбивке по значениям *Экстраверсии* (рис. 6.17).

На рис. 6.17 показано, что среди индивидуумов, не имеющих данных по *Экстраверсии*, непропорционально больше индивидуумов, для которых мы наблюдаем *Штат*, чем индивидуумов, для которых *Штат* пропущен. В целом мы видим убедительные подтверждения того, что *Штат* является не MCAR, а на самом деле MAR, поскольку его пропущенность, по-видимому, коррелирует с другими переменными, имеющимися в нашем наборе данных.

Возможно, вы заметили, что я применил слово «корреляция» ранее, когда говорил о взаимосвязи между *Возрастом* (или *Экстраверсией*) и пропущенностью *Штата*. И действительно, мы показывали корреляцию только до этих пор, и вполне возможно, что *Возраст* не является причиной пропущенности *Штата*, но что обе они обусловлены третьей ненаблюдаемой переменной. К счастью, когда мы говорим о пропущен-

ности, причинно-следственная природа корреляции (или ее отсутствие) не влияет на наши аналитические расчеты. Свободное приравнивание этих двух факторов не будет вносить систематического смещения, потому что мы никогда на самом деле не будем иметь дело с коэффициентом этой взаимосвязи.

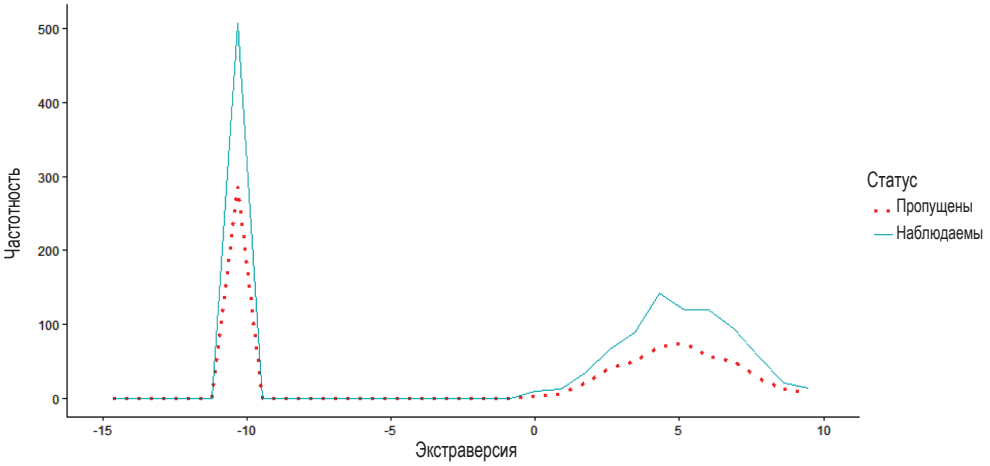

Рис. 6.17 ❖ Плотность пропущенных и наблюдаемых данных о *Штате* в разбивке по уровню *Экстраверсии*, включая пропущенную *Экстраверсию*

Диагностика переменных MNAR

Переменные MNAR – это переменные, пропущенность которых зависит от их собственных значений: более высокие значения с большей вероятностью будут пропущены, чем более низкие значения, или наоборот. Эта ситуация является как наиболее проблематичной для анализа данных, так и самой сложной для диагностики. Ее сложнее всего диагностировать, потому что, по определению, мы не знаем, какие значения пропущены. Поэтому нам нужно будет проделать немного больше сыскной работы.

Давайте посмотрим на переменную *Невротичности* и, как и прежде, начнем с регрессии ее пропущенности на других переменных в данных:

```
## Python (результат не показан)
available_data_df['md_neuro'] = available_data_df['neuro'].isnull()\
  .astype(float)
md_neuro_mod =smf.logit('md_neuro~age+open+extra+state+gender+bkg_amt',
                        data=available_data_df)
md_neuro_mod.fit(disp=0).summary()

## R
md_neuro_mod <- glm(is.na(neuro)~.,
                    family = binomial(link = "logit"),
                    data=available_data)
summary(md_neuro_mod)
...
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.162896   0.457919  -0.356  0.72204
age         -0.012610   0.008126  -1.552  0.12071
open         0.052419   0.038502   1.361  0.17337
extra       -0.084991   0.040617  -2.092  0.03639 *
genderF     -0.093537   0.151376  -0.618  0.53663
stateB       0.047106   0.181932   0.259  0.79570
stateC      -0.128346   0.187978  -0.683  0.49475
bkg_amt      0.003216   0.001065   3.020  0.00253 **
...
```

Мы видим, что *СуммаБрони* имеет сильно значимый коэффициент. На первый взгляд, это наводит на мысль о том, что *Невротичность* влияет на *СуммуБрони*. Однако, и это важный ключ к разгадке, *СуммаБрони* является дочерним элементом *Невротичности* на нашей причинно-следственной диаграмме. С поведенческой точки зрения также кажется более вероятным, что *Невротичность* является MNAR, а не MAR на *СуммеБрони* (т. е. пропущенность обусловлена личностной чертой, а не суммой, израсходованной клиентом).

Подтвердить наши подозрения можно путем выявления еще одного дочернего элемента переменной с пропущенными данными, в идеале как можно более коррелированными с ней и как можно менее коррелированными с первым дочерним элементом. В нашем вторичном наборе данных у нас есть данные об общей сумме туристической страховки, которую клиенты приобретали в компании за всю свою жизнь. Плата за поездку зависит от характеристик поездки, которые очень слабо коррелируют с суммой брони, поэтому на данном фронте у нас все нормально. Добавив страховку в наш набор данных, мы обнаруживаем, что она сильно предсказывает пропущенность *Невротичности* и что распределение суммы *Страховки* с наблюдаемой и пропущенной *Невротичностью* значительно отличается друг от друга (рис. 6.18).

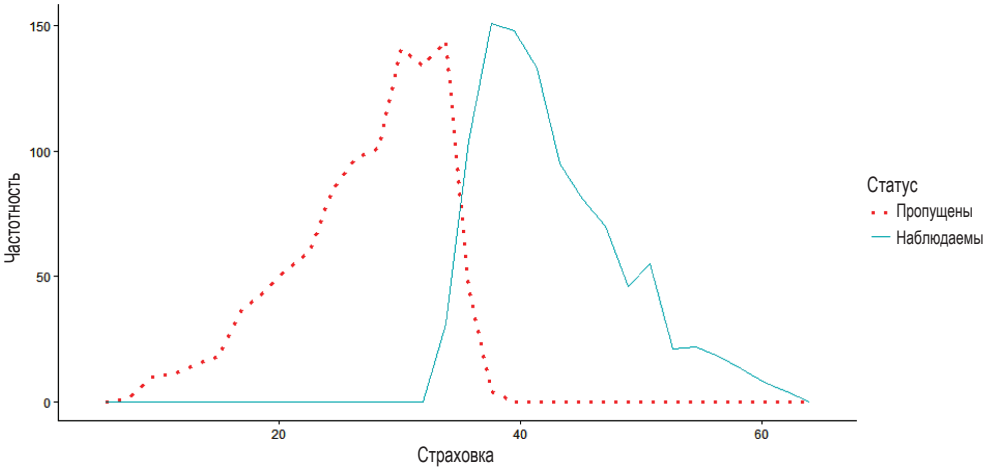

Рис. 6.18 ❖ Плотность пропущенных и наблюдаемых данных о *Невротичности* в разбивке по наблюдаемой сумме *Страховки*

Чем больше дочерних переменных мы находим коррелированными с пропущенностью Невротичности, тем сильнее мы убеждаемся в том, что эта переменная является MNAR. Как мы увидим позже, манипулирование переменной MNAR осуществляется путем добавления переменных в нашу модель вменения вспомогательных, и наши дочерние переменные являются идеальными для этого кандидатами, поэтому отыскание нескольких из них не только не будет пустой тратой времени, но и будет хорошим началом для следующего шага.

В техническом плане мы никогда не сможем полностью доказать, что переменная является MNAR, а не просто MAR, на нескольких ее дочерних элементах, но это не проблема: вспомогательные переменные не смещают вменение, если изначальная переменная действительно является MAR, а не MNAR.

Пропущенность как спектр

Классификация Рубина основана на двоичных тестах. Например, как только переменная с большей вероятностью будет пропускаться в случае более высоких значений, чем в случае более низких (или наоборот), она будет MNAR, независимо от любых других соображений. Однако форма этой взаимосвязи между значениями и пропущенностью имеет важность для практических целей: если все значения переменной пропущены выше или ниже некоторого порога, то нам нужно будет манипулировать этой переменной иначе, чем в подходе, принятом по умолчанию. Такая ситуация также может возникать с переменными MAR, поэтому стоит сделать шаг назад и подумать шире о формах пропущенности.

Пропущенность переменной можно трактовать как попадание в спектр от полностью вероятностного до полностью детерминированного. На «вероятностном» конце спектра переменная является MCAR и все значения, скорее всего, будут пропущены. На «детерминированном» конце спектра существует пороговое значение: значения пропущены для всех индивидуумов по одну сторону порога и имеются в наличии для всех индивидуумов по другую сторону порога. Это часто является результатом применения делового правила. Например, в контексте найма, если бы собеседование проходили кандидаты со *СреднимБалломАттестации* (GPA) исключительно выше 3.0, у вас не было бы балла собеседования для кандидатов ниже этого порога. Это позволило бы сделать MAR-переменную *БаллаСобеседования* более точной в отношении *СреднегоБаллаАттестации* (рис. 6.19).

Классификация Рубина на переменные MCAR/MAR/MNAR основана исключительно на выяснении причины пропущенности. Она не учитывает наличие или отсутствие случайности в этой причинно-следственной связи. Здесь, вопреки здравому смыслу, тот факт, что пропущенность *БаллаСобеседования* детерминированно основана на *СреднемБаллеАттестации*, делает *БаллСобеседования* переменной MAR на *СреднемБаллеАттестации*, даже если случайность не задействована.

Это также может происходить для переменных, которые являются MNAR, где регистрируются только те значения, которые выше или ниже некоторого

порога. Например, в файле могут сохраняться только те значения, которые выходят за пределы обычного диапазона, или будут регистрироваться только те люди, которые ниже или выше некоторого порога (например, для целей налогообложения).

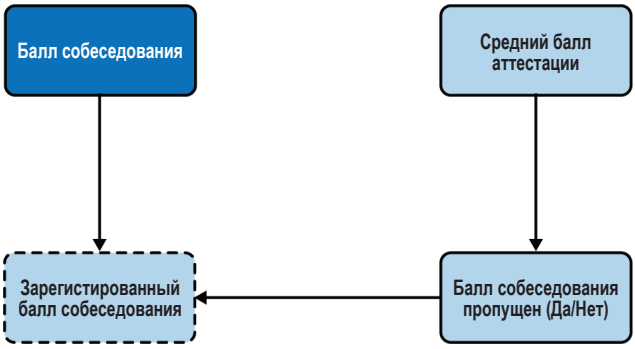

Рис. 6.19 ❖ Балл собеседования является MAR на среднем балле аттестации (GPA)

Между этими двумя крайностями полной случайности и полного детерминизма (либо типа MAR, либо типа MNAR) существуют ситуации, в которых вероятность пропущенности постоянно увеличивается или уменьшается в зависимости от значений причины пропущенности.

На рис. 6.20 показано, как это выглядит в простейшем случае двух переменных, *X* и *Y*, где *X* имеет пропущенные значения. Для удобства чтения имеющиеся значения показаны в виде полных квадратиков, в то время как «пропущенные» значения показаны в виде крестиков. В первой строке рис. 6.20 показаны диаграммы рассеяния для *Y* по отношению к *X*, а во второй строке для каждой из них показан линейный график взаимосвязи между *X* и вероятностью пропущенности:

- в крайнем левом столбце показано, что *X* является MCAR. Вероятность пропущенности является постоянной на уровне 0.5 и не зависит от *X*. Квадратики и крестики одинаково распределены по всему графику;
- центральные столбцы показывают, что *X* является вероятностно MNAR с увеличивающейся силой. Квадратики чаще встречаются слева от графика, а крестики чаще встречаются справа, но все равно слева есть крестики и справа – квадратики;
- крайний правый столбец показывает, что *X* является детерминированно MNAR. Все значения *X* ниже 5 имеются в распоряжении (квадратики), а все значения выше 5 «пропущены» (крестики).

Этот спектр пропущенности редко обсуждается в статистических трактовках пропущенных данных, поскольку его трудно подтвердить посредством чисто математических методов. Но данная книга посвящена поведенческой аналитике, поэтому мы можем и должны использовать здравый смысл и деловые знания. В примере со *СреднимБалломАттестации* пороговое значение

в данных является результатом применения делового правила, о котором вы должны знать. В большинстве ситуаций вы ожидаете, что переменная будет находиться в некотором интервале значений, и у вас должен быть нюх в отношении степени вероятности, что возможное значение не будет представлено в ваших данных.

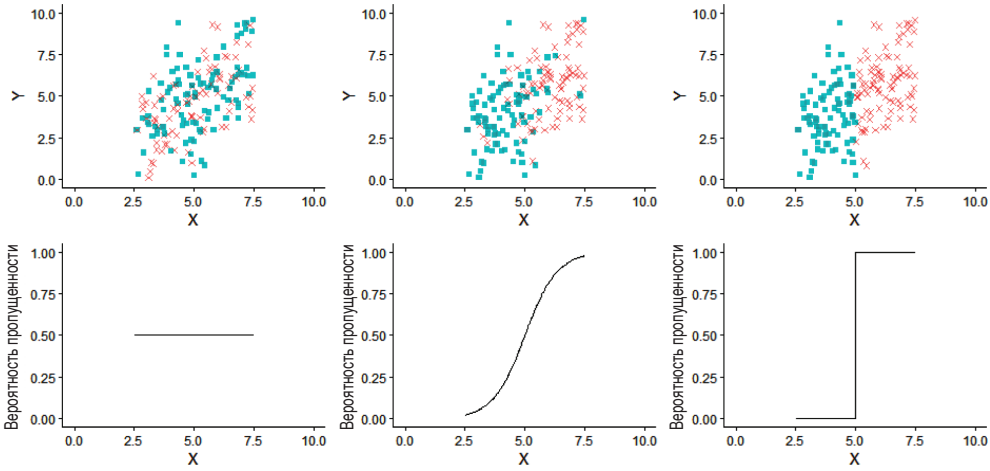

Рис. 6.20 ❖ Спектр пропущенности, от MCAR (крайний слева) до детерминированной MNAR (крайний справа) через вероятностную MNAR (в центре)

В наших данных опроса AirCnC у нас есть три личностные черты: *Открытость*, *Экстраверсия* и *Невротичность*. В реальной жизни эти переменные были бы результатом агрегирования ответов на несколько вопросов и имели бы колоколообразное распределение по известному интервалу (см. неплохое введение в психологию личности в работе Фандера (2016)). Давайте допустим, что релевантный интервал в наших данных составляет от 0 до 10, и посмотрим на распределение наших переменных (рис. 6.21).

Очевидно, что с невротичностью что-то происходит. Основываясь на том, как строятся личностные черты, мы ожидали бы одного и того же типа кривой для всех трех переменных, и мы определенно не ожидали, что у нас будет большое число клиентов со значением 5 и ни одного со значением 4. Это в подавляющем большинстве намекает о переменной детерминированно MNAR, которую нам придется обрабатывать соответствующим образом.

Теперь вы должны быть способны сформировать разумное мнение о шаблоне пропущенности в наборе данных. Сколько здесь пропущенных значений? Выглядит ли их пропущенность связанной со значениями самой переменной (MNAR), еще одной переменной (MAR) или ни то, ни другое (MCAR)? Являются ли эти связи пропущенности вероятностными или детерминированными?

На рис. 6.22 представлено дерево решений, подытоживающее нашу логику диагностики пропущенных данных.

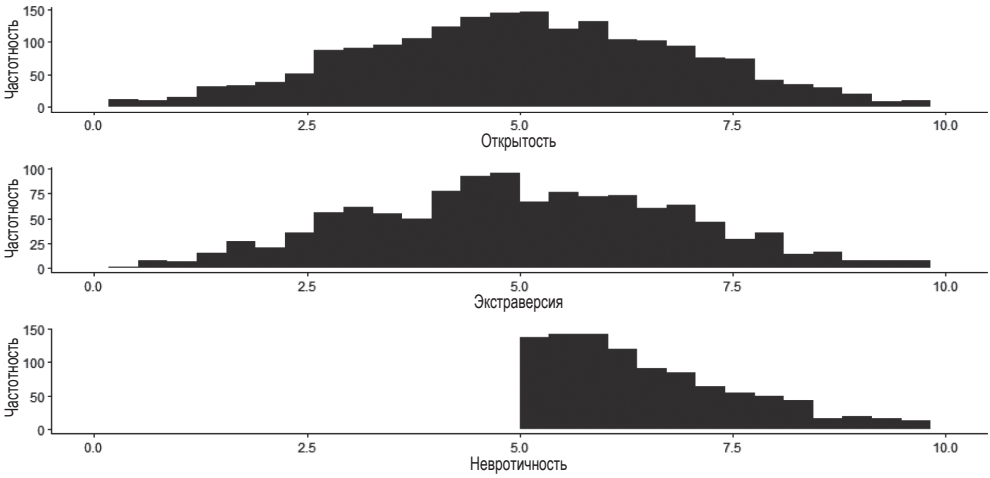

Рис. 6.21 ❖ Гистограммы личностных черт в наших данных

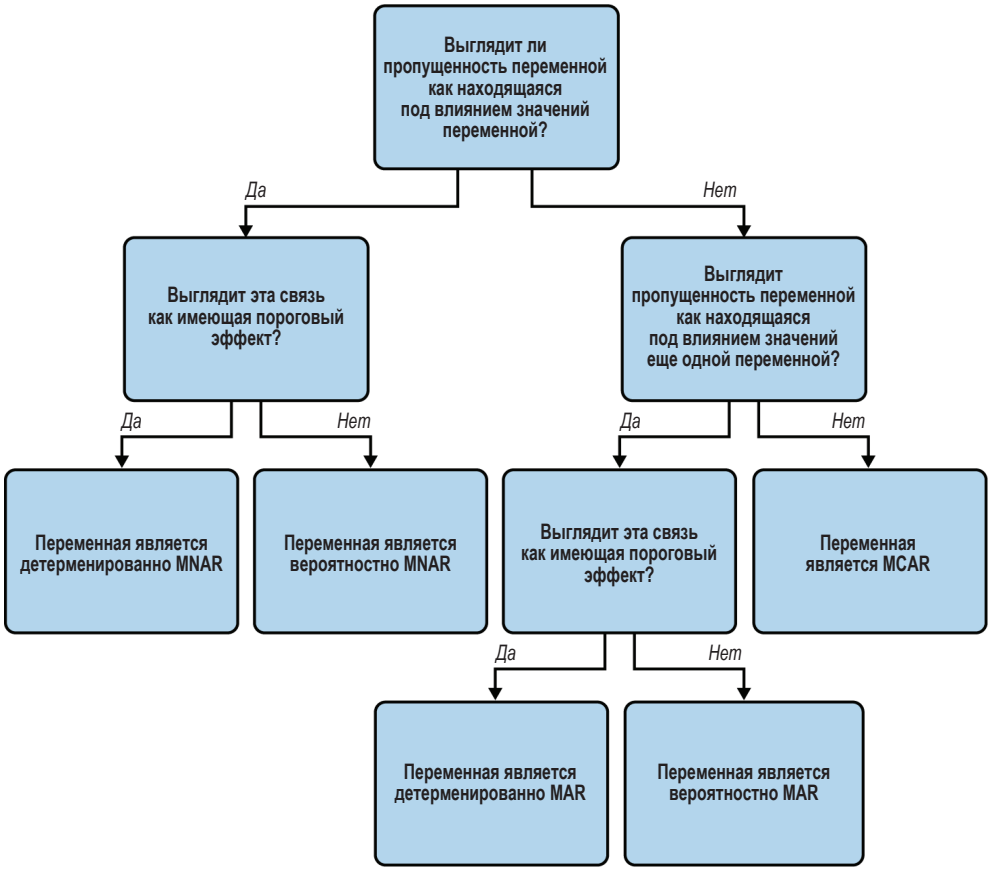

Рис. 6.22 ❖ Дерево решений для диагностики пропущенных данных

В следующем далее разделе мы увидим, как работать с пропущенными данными в каждом из этих случаев.

Работа с пропущенными данными

Первое, что следует иметь в виду, когда мы приступаем к разделу практических инструкций этой главы, – это то, что мы не пытаемся обрабатывать пропущенные данные ради них самих: наша цель состоит в том, чтобы получить несмещенные и точные оценки причинно-следственных связей в наших данных. Пропущенные данные создают проблемы только в той мере, в какой они мешают достижению указанной цели.

Это следует подчеркнуть, потому что ваш первый инстинкт, возможно, будет заключаться в том, что исходом успешного решения проблемы пропущенных данных является набор данных без пропущенных данных, а это не просто так. Метод, который мы будем использовать, множественное вменение (multiple imputation, аббр. MI), создает несколько копий ваших данных, каждая из которых имеет свои собственные вмененные значения. С точки зрения непрофессионала, мы никогда не будем говорить, что «правильной заменой пропущенного возраста Боба является 42», а скорее наоборот, мы будем говорить, что «Бобу может быть 42, 38, 44, 42 или 38 лет». Одна-единственная наилучшая догадка отсутствует, вместо нее существует распределение возможностей. Еще один наилучший из практически опробованных подходов, оценивание максимального правдоподобия, даже не предполагает никаких догадок об отдельных значениях и имеет дело только с коэффициентами более высокого порядка, такими как средние значения и ковариации.

В следующем далее подразделе я дам вам высокоуровневый обзор подхода на основе множественного вменения. После этого мы перейдем к более подробным алгоритмическим спецификациям для модели:

1) сначала алгоритм соотнесения с предсказательным средним значением;
2) затем нормальный алгоритм;
3) наконец, как добавлять в алгоритм вспомогательные переменные.

К сожалению, нет взаимно однозначной связи между типом пропущенности в классификациях Рубина и надлежащей алгоритмической спецификацией, поскольку объем имеющейся информации также имеет значение (табл. 6.4).

Таблица 6.4. Оптимальные параметры множественного вменения, основанные на типе пропущенности и имеющейся информации

Тип пропущенности	Нет информации	Распределение переменной является нормальным	Распределение пропущенности является детерминированным
MCAR	Соотнесение со средним значением	Нормальное	(Невозможно)
MAR	Соотнесение со средним значением	Нормальное	Нормальное + вспомогательные переменные
MNAR	Соотнесение со средним значением + вспомогательные переменные	Нормальное + вспомогательные переменные	Нормальное + вспомогательные переменные

Введение во множественное вменение (MI)

В целях понимания принципа работы множественного вменения полезно сравнить его с традиционными подходами к пропущенным данным. Помимо простого удаления всех строк с пропущенными значениями, все традиционные подходы основаны на замене пропущенных значений конкретным значением. Замещаемое значение может быть совокупным средним значением переменной или предсказанными значениями, основанными на других переменных, имеющихся для этого клиента. Независимо от правила, используемого для замещаемого значения, эти подходы в корне ошибочны, поскольку они игнорируют дополнительную неопределенность, вводимую наличием пропущенных данных, и они могут вносить систематическое смещение в наши аналитические расчеты.

Решение этой проблемы на основе множественного вменения, как следует из названия, состоит в том, чтобы строить несколько наборов данных, в которых пропущенные значения заменяются разными значениями, затем выполнять интересующий нас анализ с каждым из них и, наконец, агрегировать результирующие коэффициенты.

Как в R, так и в Python весь этот процесс управляется за кулисами, и если вы не хотите его усложнять, то вы можете просто указывать данные и аналитические расчеты, которые хотите выполнять.

Давайте сначала посмотрим на исходный код R:

```
## R
> MI_data <- mice(available_data, print = FALSE)
> MI_summ <- MI_data %>%
    with(lm(bkg_amt~age+open+extra+neuro+gender+state)) %>%
   pool() %>%
   summary()
> print(MI_summ)
         Term   estimate  std.error statistic        df      p.value
1 (Intercept) 240.990671 15.9971117 15.064636  22.51173 3.033129e-13
2         age  -1.051678  0.2267569 -4.637912  11.61047 6.238993e-04
3        open   3.131074  0.8811587  3.553360 140.26375 5.186727e-04
4       extra  11.621288  1.2787856  9.087753  10.58035 2.531137e-06
5       neuro  -6.799830  1.9339658 -3.516003  15.73106 2.929145e-03
6     genderF -11.409747  4.2044368 -2.713740  20.73345 1.310002e-02
7      stateB  -9.063281  4.0018260 -2.264786 432.54286 2.401986e-02
8      stateC  -5.334055  4.7478347 -1.123471  42.72826 2.675102e-01
```

Пакет mice (Множественное вменение с помощью цепных уравнений) содержит функцию `mice()`, которая генерирует несколько наборов данных. Затем мы применяем интересующую нас регрессию к каждому из них, используя ключевое слово `with()`. Наконец, функция `pool()` из mice агрегирует результаты в формате, который мы можем прочитать с помощью традиционной функции `summary()`.

Исходный код Python почти идентичен, потому что в нем имплементирован один и тот же подход:

```
## Python
MI_data_df = mice.MICEData(available_data_df)
fit = mice.MICE(model_formula='bkg_amt~age+open+extra+neuro+gender+state',
                model_class=sm.OLS, data=MI_data_df)
MI_summ = fit.fit().summary()
print(MI_summ)
```

```
                          Results: MICE
=================================================================
Method:                  MICE         Sample size:          2000
Model:                   OLS          Scale                 5017.30
Dependent variable:      bkg_amt      Num. imputations      20
-----------------------------------------------------------------
           Coef.   Std.Err.    t    P>|t|   [0.025 0.975]    FMI
-----------------------------------------------------------------
Intercept 120.3570  8.8662 13.5748 0.0000 102.9795 137.7344  0.4712
Age        -1.1318  0.1726 -6.5555 0.0000  -1.4702  -0.7934  0.2689
Open        3.1316  0.8923  3.5098 0.0004   1.3828   4.8804  0.1723
extra      11.1265  1.0238 10.8680 0.0000   9.1200  13.1331  0.3855
neuro      -4.5894  1.7968 -2.5542 0.0106  -8.1111  -1.0677  0.4219
gender_M   65.9603  4.8191 13.6873 0.0000  56.5151  75.4055  0.4397
gender_F   54.3966  4.6824 11.6171 0.0000  45.2192  63.5741  0.4154
state_A    40.9352  3.9080 10.4748 0.0000  33.2757  48.5946  0.3921
state_B    37.3490  4.0727  9.1706 0.0000  29.3666  45.3313  0.2904
state_C    42.0728  3.8643 10.8875 0.0000  34.4989  49.6468  0.2298
=================================================================
```

Здесь алгоритм mice импортируется из пакета statsmodels.imputation. Функция `MICEData()` генерирует несколько наборов данных. Затем мы указываем с помощью функции `MICE()` модельную формулу, тип регрессии (здесь обычные наименьшие квадраты, `statsmodels.OLS`) и данные, которые мы хотим использовать. Перед распечаткой результата мы выполняем подгонку нашей модели с помощью методов `.fit()` и `.summary()`.

Одна из сложностей имплементации функции `mice` на Python заключается в том, что в ней не учитываются категориальные переменные в качестве предсказателей. Тем не менее если вы действительно хотите использовать Python, то вам придется сначала конвертировать категориальные переменные в кодировку с одним активным состоянием[1]. Следующий ниже фрагмент исходного кода показывает, как это сделать для переменной Пола:

```
## Python
gender_dummies = pd.get_dummies(available_data_df.\
                                gender,
                                prefix='gender')
available_data_df = pd.concat([available_data_df,
                               gender_dummies],
                              axis=1)
```

[1] Словосочетание «кодирование с одним активным состоянием» (англ. one-hot encoding) пришло из терминологии цифровых интегральных микросхем, в которой оно описывает конфигурацию микросхемы, допускающую, чтобы только один бит был положительным (активным). – *Прим. перев.*

```
available_data_df.gender_F = \
    np.where(available_data_df.gender.isna(),
             float('NaN'), available_data_df.gender_F)
available_data_df.gender_M = \
    np.where(available_data_df.gender.isna(),
             float('NaN'), available_data_df.gender_M)
available_data_df = available_data_df.\
    drop(['gender'], axis=1)
```

Сначала мы используем функцию `get_dummies()` из pandas для создания переменных `gender_F` и `gender_M`. После добавления этих столбцов в наш кадр данных мы указываем место, где находятся пропущенне значения (по умолчанию функция конвертирования в кодировку с одним активным состоянием устанавливает значение всех двоичных переменных равным 0, если значение категориальной переменной пропущено). Наконец, мы отбрасываем нашу изначальную категориальную переменную из наших данных и выполняем подгонку нашей модели, в которую включены новые переменные.

Однако кодировка с одним активным состоянием нарушает некоторую внутреннюю структуру данных, удаляя логические взаимосвязи между переменными, поэтому ваш пробег может отличаться (например, вы, возможно, увидите, что из-за разных структур коэффициенты категориальных переменных различаются между R и Python), и если в ваших данных категориальные переменные играют важную роль, то вместо этого я бы порекомендовал вам использовать R.

Вуаля! Если бы вы прекратили читать эту главу прямо сейчас, то у вас было бы техническое решение по работе с пропущенными данными, которое было бы значительно лучше, чем традиционные подходы. Однако мы можем добиться еще большего успеха, потратив немного времени на то, чтобы поднять капот и разобраться в алгоритмах вменения получше.

Метод вменения по умолчанию: соотнесение с предсказательным средним значением

В предыдущем подразделе мы оставили метод вменения неспецифицированным и положились на принятые в функции `mice` значения по умолчанию. В Python единственный доступный метод вменения – это соотнесение с предсказательным средним значением, так что там делать нечего. Давайте проверим, какие методы вменения применяются в R по умолчанию, запросив сводку процесса вменения:

```
## R
> summary(MI_data)
Class: mids
Number of multiple imputations: 5
Imputation methods:
     age     open    extra    neuro   gender    state  bkg_amt
      ""       ""    "pmm"    "pmm"       "" "logreg"    "pmm"
PredictorMatrix:
       age open extra neuro gender state bkg_amt
age      0    1     1     1      1     1       1
open     1    0     1     1      1     1       1
```

```
extra    1    1     0     1     1     1     1
neuro    1    1     1     0     1     1     1
gender   1    1     1     1     0     1     1
state    1    1     1     1     1     0     1
```

Да, тут масса информации. Для начала давайте посмотрим только на строку `Imputation methods` (Методы вменения). Переменные, в которых нет пропущенных данных, имеют пустое поле `""`, что резонно, поскольку к ним вменение не применяется. Категориальные переменные имеют метод `logreg`, т. е. логистическую регрессию. Наконец, числовые переменные имеют метод `pmm`, который расшифровывается как соотнесение с предсказательным средним значением (predictive mean matching, аббр. PMM). Метод `pmm` работает путем отбора ближайших соседей индивидуума с пропущенным значением и замены пропущенного значения значением одного из соседей. Представьте себе, например, набор данных, содержащий только две переменные: *Возраст* и *ПочтовыйИндекс*. Если у вас есть клиент из почтового индекса 60612 с пропущенным возрастом, то этот алгоритм подберет возраст другого клиента случайным образом в том же почтовом индексе или максимально близком.

Из-за некоторой случайности, зашитой в этот процесс, каждый из наборов вмененных данных в итоге будет иметь слегка разные значения, как наглядно видно, если применить удобную функцию `densityplot()` из пакета mice в R:

```
## R
> densityplot(MI_data, thicker = 3, lty = c(1,rep(2,5)))
```

На рис. 6.23 показаны распределения числовых переменных в изначальных располагаемых данных (толстая линия) и в наборах вмененных данных (тонкие пунктирные линии). Хорошо видно, что распределения довольно близки к изначальным данным; исключением является *СуммаБрони*, которая в целом более сосредоточена вокруг среднего значения (т. е. «более высоких пиков») в наборе вмененных данных, чем в изначальных данных.

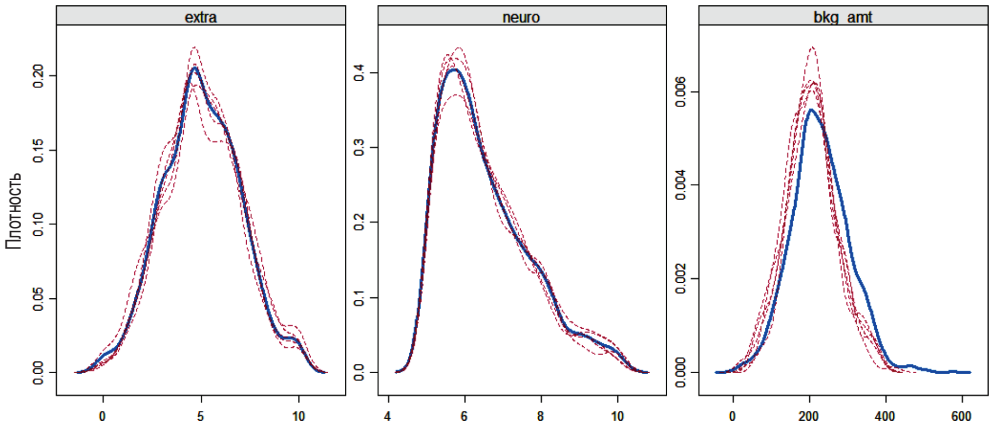

Рис. 6.23 ❖ Распределения вмененных значений числовых переменных в наших данных

Метод PMM обладает некоторыми важными свойствами, которые могут быть или не быть желательными, в зависимости от контекста. Наиболее важным свойством является то, что он в сущности является методом интерполяции. Следовательно, PMM можно трактовать как создание значений, которые находятся между существующими значениями. Поступая в таком ключе, он минимизирует риск создания бессмысленных ситуаций, таких как беременные отцы или отрицательные суммы. Этот подход также будет хорошо работать, когда переменная имеет странное распределение, такое как *Возраст* в наших данных, которое имеет два пика, потому что в нем не делается никаких допущений о форме совокупного распределения; он просто ловит соседа.

Однако у метода PMM есть несколько недостатков: он работает медленно и плохо масштабируется на крупные наборы данных, поскольку ему приходится постоянно пересчитывать расстояния между индивидуумами. В добавление к этому, когда у вас много переменных или много пропущенных значений, ближайшие соседи могут находиться «далеко», и качество вменения будет ухудшаться. Вот почему PMM не будет нашим предпочтительным вариантом, когда у нас будет информация о распределении, как мы увидим в следующем подразделе.

От PMM к нормальному вменению (только для R)

Хотя метод PMM является хорошей отправной точкой, у нас часто бывает информация о распределении числовых переменных, которую в R можно использовать для ускорения и улучшения наших моделей вменения. В частности, в бихевиористике и естественных науках нередко принято исходить из допущения, что числовые переменные подчиняются нормальному распределению, потому что оно очень распространено. В этом случае мы можем выполнить подгонку нормального распределения к переменной, а затем взять вменяемые значения из этого распределения вместо использования PMM. Это делается путем создания вектора методов вменения со значением `"norm.nob"` в переменных, в отношении которых мы будем исходить из нормальности, а затем передачи этого вектора методу `parameter` функции `mice()`:

```
## R
> imp_meth_dist <- c("pmm", rep("norm.nob",3), "", "logreg", "norm.nob")
> MI_data_dist <- mice(available_data, print = FALSE, method = imp_meth_dist)
```

Как можно видеть, синтаксис очень прост. Остается единственный вопрос – определить, для какой из числовых переменных мы хотим использовать нормальное вменение. Давайте взглянем на числовые переменные в располагаемых нами данных (рис. 6.24).

Возраст с его двумя пиками, очевидно, не является нормальным, но все остальные переменные имеют только один пик. *Открытость*, *Экстраверсия* и *СуммаБрони* внешне тоже выглядят достаточно симметричными (с технической точки зрения, они не спутаны). Статистические симуляции показывают, что до тех пор, пока переменная имеет один пик и не имеет «толстого

хвоста» только в одном направлении, допущение о нормальности не приводит к систематическому смещению. Поэтому мы можем допустить нормальность для *Открытости*, *Экстраверсии* и *СуммыБрони*.

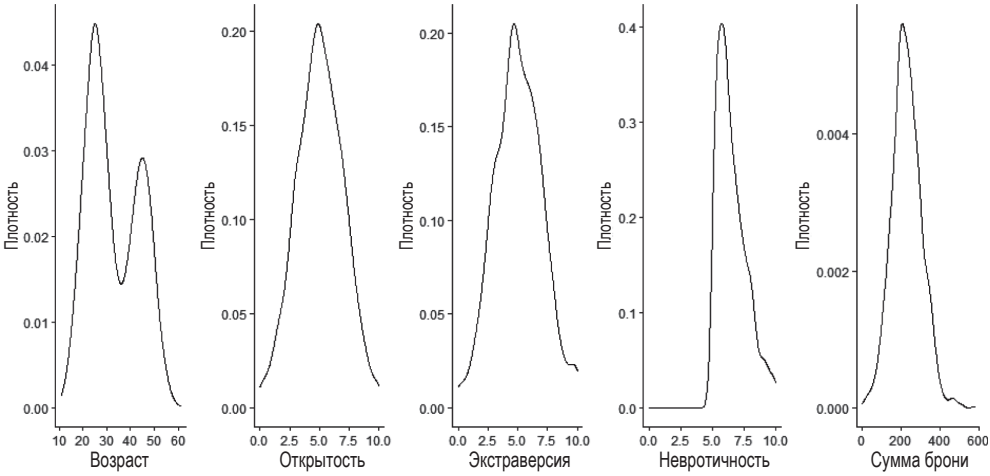

Рис. 6.24 ❖ Распределение числовых переменных в наших данных

Как мы видели в предыдущем разделе, *Невротичность* представляет собой необычный асимметричный шаблон: значения ограничены интервалом [5, 10], хотя используемая нами психологическая шкала колеблется от 0 до 10, намекая на то, что невротичность может быть «детерминированной», т. е. все значения *Невротичности* ниже некоторого порога будут пропущены. Использование метода PMM для вменения в такой ситуации выглядит проблематичным: для вменения в значительном диапазоне значений имеется всего несколько соседей либо их нет. В крайнем случае всем пропущенным значениям X будет вменено 5, значение порога. Это как раз та ситуация, в которой нормальный метод будет способен восстанавливать истинные пропущенные значения гораздо лучше. Мы видим это, сравнивая значения, вменяемые этими двумя методами. На рис. 6.25 показаны располагаемые значения невротичности (квадратики) и значения, рассчитанные методом PMM (крестики, верхняя панель) и нормальным методом (крестики, нижняя панель).

Как хорошо видно, метод PMM не вменяет никакого значения *Невротичности* ниже 5, тогда как нормальный метод это делает. В дополнение к этому метод PMM вменяет слишком много значений, близких к 10, тогда как нормальный метод адекватнее улавливает совокупную форму распределения. Тем не менее нормальный метод далек от восстановления истинного распределения (которое доходит до самого нуля). Это распространенная проблема в случае переменных, которые являются детерминированно MAR или MNAR. Добиться еще большего улучшения на базе обычного вменения можно, используя вспомогательные переменные, как мы увидим в следующем подразделе.

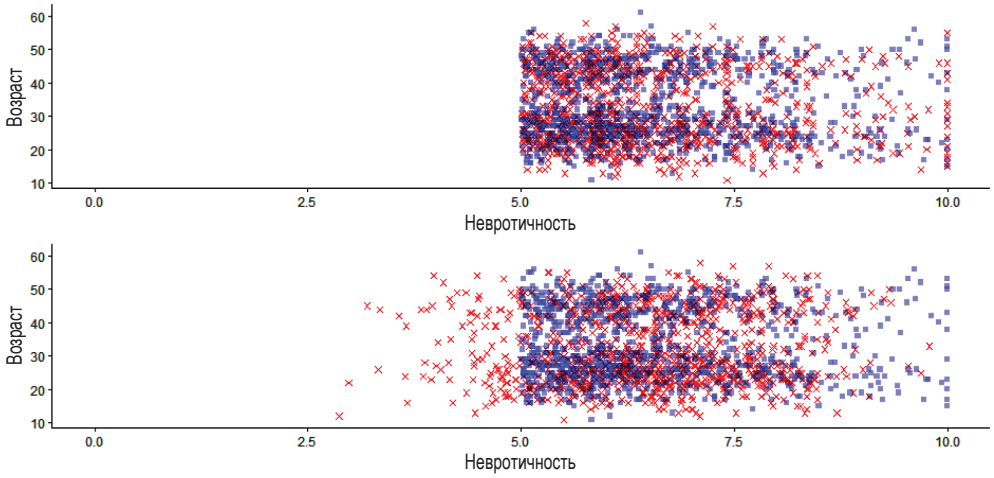

Рис. 6.25 ❖ Вменение методом PMM (вверху) и нормальное вменение (внизу) с переменной детерминированно MNAR

Устойчивое вменение

Если у вас есть однопиковые переменные, но они не являются нормальными (например, по меньшей мере один хвост распределения необычно толстый или тонкий по сравнению с нормальным распределением), и вы заботитесь о дополнительной точности, то пакет ImputeRobust языка R, опираясь на пакет mice, уточняет вашу модель вменения. Обратитесь к виньетке CRAN[1] для получения более подробной информации.

Добавление вспомогательных переменных

Довольно часто у нас будут переменные, коррелированные с одной из наших переменных, в которой пропущены данные (например, причинами или следствиями этой переменной), но они не входят в нашу регрессионную модель. Это как раз та ситуация, в которой алгоритм `mice` особенно хорош, потому что мы можем добавить эти переменные в нашу модель вменения, чтобы повысить ее точность. После этого на них принято ссылаться как на «вспомогательные переменные» нашей модели вменения.

В нашем примере AirCnC дополнительный располагаемый набор данных содержит две переменные, *Страховка* и *Активность*. Первая обозначает сумму приобретенной клиентом туристической страховки и сильно коррелирует с *Невротичностью*, тогда как последняя измеряет степень, с которой клиент выбирал активные отпуска (например, скалолазание), и сильно коррелирует с *Экстраверсией*. Мы будем использовать их в помощь для вменения двух личностных переменных.

[1] См. https://oreil.ly/5quMo.

Добавлять вспомогательные переменные в модель вменения чрезвычайно просто: нужно просто добавить их в набор данных перед фазой вменения:

```
## R
augmented_data <- cbind(available_data, available_data_supp)
MI_data_aux <- mice(augmented_data, print = FALSE)

## Python
augmented_data_df = pd.concat([available_data_df, available_data_supp_df],
                              axis=1)
MI_data_aux_df = mice.MICEData(augmented_data_df)
```

Затем мы можем выполнить все наши аналитические расчеты, как и раньше. При добавлении вспомогательных переменных, как правило, имеет смысл использовать нормальный метод для переменных, коррелированных со вспомогательными переменными (здесь *Невротичность* и *Экстраверсия*), в особенности когда эти переменные усечены или являются MNAR.

Помимо ограничений на вычисления, нет никаких лимитов на число вспомогательных переменных, которые можно включать. Однако потенциальный риск заключается в том, что некоторые вспомогательные переменные, возможно, будут ошибочно выглядеть коррелированными с переменной в нашем изначальном наборе данных просто из-за чистой случайности, например *Страховка* может коррелировать с *Экстраверсией*, хотя это не так. Такая «ложноположительная» корреляция была бы затем неоправданно подкреплена моделью вменения.

Решение этой потенциальной проблемы состоит в ограничении использования вспомогательных переменных лишь с целью вменения некоторых переменных. К сожалению, это техническое решение доступно только в R. Именно здесь на сцену выходит предсказательная матрица функции `mice()`. Указанная матрица появляется при распечатке сводки фазы вменения, а также может извлекаться непосредственно из нашего объекта MIDS:

```
## R
> pred_mat <- MI_data_aux$predictorMatrix
> pred_mat
          age open extra neuro gender state bkg_amt insurance active
age         0    1     1     1      1     1       1         1      1
open        1    0     1     1      1     1       1         1      1
extra       1    1     0     1      1     1       1         1      1
neuro       1    1     1     0      1     1       1         1      1
gender      1    1     1     1      0     1       1         1      1
state       1    1     1     1      1     0       1         1      1
bkg_amt     1    1     1     1      1     1       0         1      1
insurance   1    1     1     1      1     1       1         0      1
active      1    1     1     1      1     1       1         1      0
```

Эта матрица указывает на то, какая переменная используется для вменения какой переменной. По умолчанию все переменные используются для вменения всех переменных, кроме самих себя. «1» в матрице указывает на то, что «столбцовая» переменная используется для вменения «строчной» пере-

менной. Поэтому мы хотим изменить последние два столбца, для *Страховки* и *Активности*, чтобы они использовались только для обозначения соответственно *Невротичности* и *Экстраверсии*:

```
## R
> pred_mat[,"insurance"] <- 0
> pred_mat[,"active"] <- 0
> pred_mat["neuro","insurance"] <- 1
> pred_mat["extra","active"] <- 1
> pred_mat
          age open extra neuro gender state bkg_amt insurance active
age         0    1     1     1      1     1       1         0      0
open        1    0     1     1      1     1       1         0      0
extra       1    1     0     1      1     1       1         0      1
neuro       1    1     1     0      1     1       1         1      0
gender      1    1     1     1      0     1       1         0      0
state       1    1     1     1      1     0       1         0      0
bkg_amt     1    1     1     1      1     1       0         0      0
insurance   1    1     1     1      1     1       1         0      0
active      1    1     1     1      1     1       1         0      0
```

С помощью этой модификации мы снизим риск непреднамеренного включения в нашу модель вменения чисто случайных корреляций.

Вертикальное масштабирование числа наборов вмененных данных

По умолчанию число наборов вмененных данных, создаваемых алгоритмом mice, равно 5 в R и 10 в Python. Эти принятые по умолчанию величины прекрасно подходят для исследовательских аналитических расчетов.

Для вашего окончательного прогона вам следует использовать 20 (передав `m=20` в функцию `mice()` в качестве параметрического значения), если вас интересуют только оценочные значения коэффициентов регрессии. Если вам нужна более прецизионная информация, такая как взаимодействия между переменными или интервалы уверенности, то вы, возможно, захотите нацелиться на 50–100. Тогда главными ограничениями становятся скорость и память компьютера – если ваш набор данных «весит» 100 Мб или даже 1 Гб, то хватит ли у вас RAM для создания ста его копий? – а также вашего терпения.

Синтаксис для изменения числа наборов вмененных данных прост. В R оно передается в функцию `mice()` в качестве параметра, тогда как в Python оно передается в качестве параметра метода `.fit()` объекта `MICE`:

```
## R
MI_data <- mice(available_data, print = FALSE, m=20)

## Python
fit = mice.MICE(model_formula='bkg_amt~age+open+extra+neuro+gender+state',
                model_class=sm.OLS, data=MI_data_df)
MI_summ = fit.fit(n_imputations=20).summary()
```

Выводы

Пропущенные данные могут представлять реальную проблему в анализе поведенческих данных, но это не обязательно должно быть так. Как минимум, использование пакета `mice` в R или Python со своими принятыми по умолчанию параметрами превзойдет удаление всех строк с пропущенными значениями. Правильно определяя пропущенность, основываясь на классификации Рубина и задействуя всю имеющуюся информацию, вы, как правило, сможете добиваться большего. В целях краткого подытоживания правил принятия решений в одном месте на рис. 6.26 показано дерево решений для диагностики пропущенных данных, а в табл. 6.5 приведены оптимальные параметры множественного вменения, основанные на типе пропущенных данных и располагаемой информации.

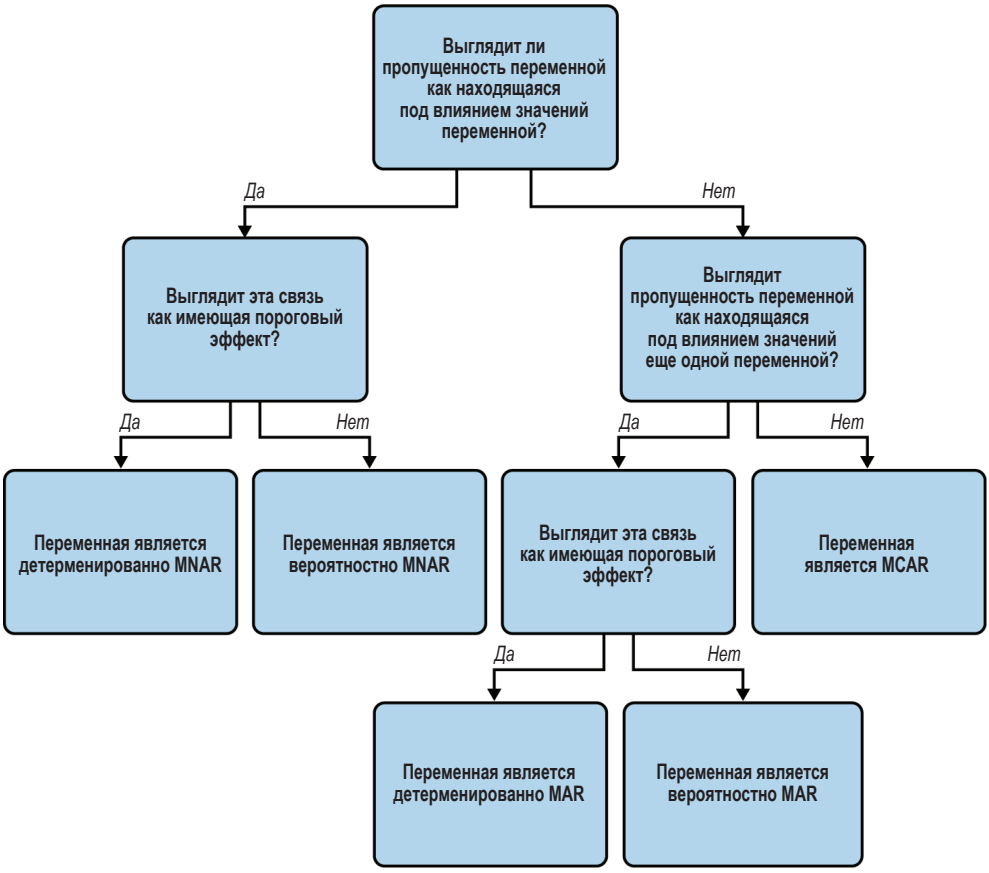

Рис. 6.26 ❖ Дерево решений для диагностики пропущенных данных

Таблица 6.5. Оптимальные параметры множественного вменения, основанные на типе пропущенности и располагаемой информации

Тип пропущенности	Нет информации	Распределение переменной является нормальным	Распределение пропущенности является детерминированным
MCAR	Соотнесение со средним значением (PMM)	`norm.nob`	
MAR	Соотнесение со средним значением (PMM)	`norm.nob`	`norm.nob` + вспомогательные переменные
MNAR	Соотнесение со средним значением (PMM) + вспомогательные переменные	`norm.nob` + вспомогательные переменные	`norm.nob` + вспомогательные переменные

Глава 7

Измерение неопределенности с помощью бутстрапа

Имея идеальные данные, вы теперь можете делать устойчивые выводы на основе поведенческих данных и измерять причинно-следственное воздействие изменений на поведения людей в деловой/окружающей среде. Но что делать, если у вас субоптимальные данные? В академических исследованиях, столкнувшись с недоказательными данными, всегда можно вернуться к нулевой гипотезе и отказаться выносить суждения. Но в прикладных исследованиях нет никаких нулевых гипотез, а есть только альтернативные варианты действий на выбор.

Малые размеры выборок, переменные странной формы или ситуации, требующие передовых аналитических инструментов (например, иерархического моделирования, которое мы увидим в книге позже), могут приводить к шатким выводам. Разумеется, линейно-регрессионный алгоритм будет выплевывать коэффициент во всех случаях, кроме самых экстремальных, но стоит ли ему доверять? Сможете ли вы с уверенностью посоветовать своему боссу поставить на него миллионы долларов?

В этой главе я познакомлю вас с чрезвычайно мощным и общим инструментом симулирования – бутстрапом, который позволит нам делать устойчивые выводы из любых данных, какими бы малыми или странными они ни были. Он работает путем создания и анализирования слегка отличающихся версий ваших данных, основываясь на случайных числах. Великолепная особенность бутстрапа заключается в том, что, применяя его, вы буквально ни разу не ошибетесь: в ситуациях, которые являются наилучшим сценарием для традиционных статистических методов (например, выполнение базовой линейной регрессии на крупном и хорошо отработанном наборе данных), бутстрап выполняется медленнее и менее точно, однако он все еще находится на игровом поле. Но как только вы отходите от подобных сценариев, бутстрап быстро превосходит традиционные статистические методы, нередко

с большим отрывом[1]. Поэтому мы будем широко на него опираться на протяжении всей остальной части книги. В частности, мы будем использовать его во время дизайна и анализирования экспериментов в части IV, чтобы строить симулированные эквиваленты *p*-значений, которые являются более интуитивными, чем традиционные статистические.

В первом разделе мы сосредоточимся на разведывательном/описательном анализе данных и увидим, что бутстрап бывает полезен уже на этой стадии. Во втором разделе мы будем использовать бутстрап в контексте регрессии. Затем мы расширим нашу перспективу, чтобы обсудить вопросы о том, когда следует использовать бутстрап и какие инструменты можно задействовать, чтобы облегчить свою жизнь с его помощью.

Введение в бутстрап: «опрашивание» самого себя

Хотя нашей конечной целью является использование бутстрапа для регрессии, мы можем начать с более простого примера описательной статистики: получения среднего значения набора выборочных данных.

Пакеты

В этой главе мы будем использовать следующие пакеты в дополнение к обычным:

```
## Python
import statsmodels.api as sm                          # Для QQ-графика
import statsmodels.stats.outliers_influence as st_inf # Для расстояния Кука
```

Деловая задача: малые данные с выбросом

Руководство C-Mart заинтересовано в том, чтобы понять, сколько времени требуется пекарям для приготовления пирогов на заказ, с целью возможного пересмотра структуры ценообразования. С этой целью они попросили инженера-технолога C-Mart провести исследование времени. Как следует из названия, исследование времени (т. н. изучение временных затрат и трудовых движений – time-and-motion study) – это прямое наблюдение за производственным процессом с целью измерения продолжительности выполняемых задач. Учитывая, что указанный процесс занимает много времени (каламбур), инженер отобрал десять разных магазинов, которые в некоторой степени репрезентабельны в отношении бизнеса C-Mart. В каждом магазине они

[1] См. работу Wilcox (2010), в которой показана опасность принятия нормальности как само собой разумеющейся.

наблюдали, как один пекарь готовит один пирог. Они также регистрировали практику каждого пекаря, измеренную в месяцах работы.

У инженера имеется выборка всего в объеме 10 наблюдений, что для начала не очень-то много. Даже если бы все данные очень согласованно соответствовали четкой взаимосвязи, уже один размер выборки обычно подсказывает, что необходимо использовать бутстрап. Однако, занимаясь разведкой своих данных, инженер заметил наличие выброса (рис. 7.1).

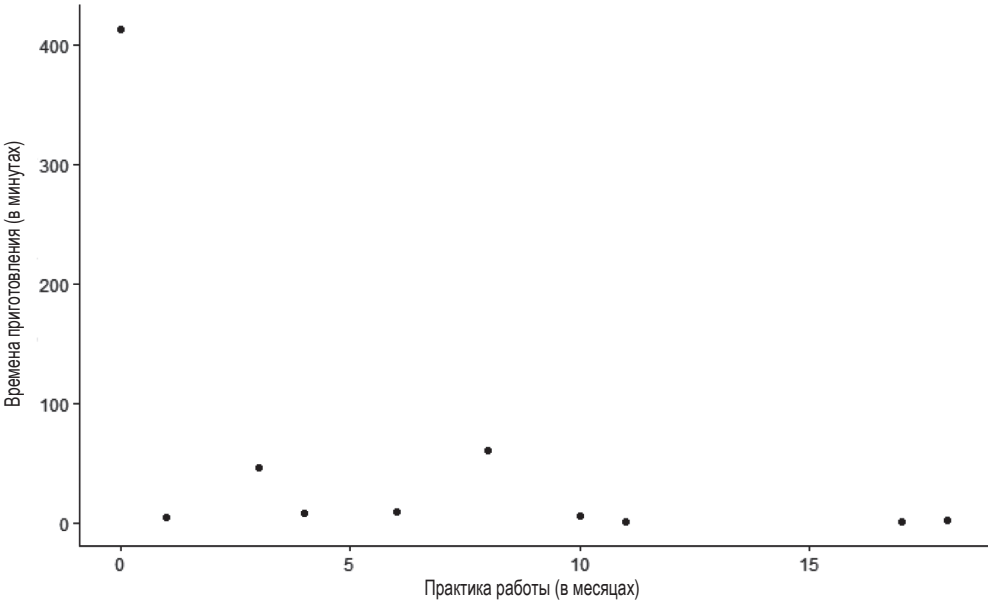

Рис. 7.1 ❖ Практика работы пекаря и время приготовления им изделий

У нас есть одна экстремальная точка в левом верхнем углу, соответствующая новому сотруднику, который провел большую часть дня, работая над сложным пирогом для корпоративного мероприятия. Каким образом инженеру следует отчитаться о данных своего исследования? У него, возможно, возникнет искушение отнестись к самому крупному наблюдению как к выбросу, что является вежливым способом сказать: «отбрось и притворись, что его не было». Но это наблюдение, хотя и необычное, не является аберрацией в чистом виде. Ошибки в измерениях не было, и такие обстоятельства, вероятно, время от времени возникают. Альтернативой было бы сообщить только совокупную среднюю продолжительность, 56 минут, но это также вводило бы в заблуждение, поскольку не передавало бы вариативность и неопределенность данных. Традиционной рекомендацией в этой ситуации было бы использовать интервал уверенности вокруг среднего значения. Давайте рассчитаем нормальный 95%-ный интервал уверенности посредством регрессии. (Использование регрессии в этом случае будет излишним – есть гораздо более простые способы вычисления среднего значения, – но это послужит осторожным введением в процесс, который мы будем использовать позже в данной главе.)

Сначала мы выполняем регрессию `times~1`, т. е. только с пересечением (коэффициентом сдвига). Затем мы извлекаем результирующую оценку коэффициента пересечения, которая, если вы незнакомы с этим расчетом, равна среднему значению нашей зависимой переменной. Мы также извлекаем стандартную ошибку этого коэффициента. Как известно из любого курса статистики, нижний предел нормального 95%-го интервала уверенности равен среднему значению минус 1.96, помноженному на стандартную ошибку, а верхний предел равен среднему значению плюс 1.96, помноженному на стандартную ошибку:

```
## R (результат не показан)
lin_mod_summ <- summary(lm(times~1, data=dat))
est <- lin_mod_summ$coefficients[1,1]
se <- lin_mod_summ$coefficients[1,2]
LL <- est-1.96*se
UL <- est+1.96*se

## Python
lin_mod = ols("times~1", data=data_df).fit()
est = lin_mod.params['Intercept']
se = lin_mod.bse['Intercept']
LL = est-1.96*se #Lower limit
UL = est+1.96*se #Upper limit
print("LL = ", LL)
print("UL = ",UL)

LL = -23.040199740431333
UL = 134.64019974043134
```

К сожалению, 95%-й интервал уверенности в этом случае равен [–23; 135], что, очевидно, является бессмыслицей, поскольку сроки продолжительностей не могут быть отрицательными. Это произошло потому, что традиционные интервалы уверенности исходят из допущения, что рассматриваемая переменная подчиняется нормальному распределению вокруг своего среднего значения, что в данном случае неверно. Можно себе представить реакцию аудитории инженера, которая отнесется к отрицательным продолжительностям, мягко говоря, не слишком благосклонно, но это одна из проблем, которую способен решить бутстрап.

Бутстраповский интервал уверенности для выборочного среднего

Бутстрап позволяет нам использовать имеющиеся у нас данные в полной мере и делать разумные выводы независимо от размера выборки или трудностей с формой данных. Это достигается путем создания нескольких воображаемых наборов данных на основе располагаемых данных. Сравнение этих наборов данных друг с другом позволяет нам преодолевать шум и точнее представлять важность выбросных значений. Оно также может обеспечивать

более жесткие интервалы уверенности, поскольку устраняет некоторую неопределенность, создаваемую шумом.

Это с самого начала отличается от простого выбора более узкого диапазона (например, выбора 80%-го интервала уверенности вместо 95%-го), поскольку генерируемые бутстрапом наборы данных отражают истинное распределение вероятностей с учетом располагаемых данных. Набор данных с отрицательной продолжительностью генерироваться не будет, поскольку данные не отражают эту возможность, однако будут наборы данных, которые отражают очень долгую продолжительность, т. к. изначальные данные и впрямь включают ее как возможность. Таким образом, можно ожидать, что генерируемый бутстрапом интервал уверенности будет удалять больше с отрицательной стороны диапазона, но он, возможно, не будет удалять столь много с положительной стороны диапазона (или даже добавлять к ней).

Бутстрап и смысл статистики как науки

Почему мы берем выборки с возвратом? Для того чтобы по-настоящему разобраться в том, что происходит с бутстрапом, стоит сделать шаг назад и вспомнить суть статистики как науки: у нас есть популяция, которую мы не можем проинспектировать полностью, поэтому мы пытаемся делать об этой популяции выводы, основываясь на лимитированной выборке из нее. Для этого мы создаем «воображаемую» популяцию посредством статистических допущений либо посредством бутстрапа. В случае статистических допущений мы говорим: «вообразите, что эта выборка взята из популяции с нормальным распределением», а затем, основываясь на этом, делаем выводы. В случае бутстрапа мы говорим: «вообразите, что популяция имеет точно такое же распределение вероятностей, как и выборка», или, что эквивалентно, «вообразите, что выборка взята из популяции, состоящей из большого (или бесконечного) числа копий этой выборки». Тогда взятие из этой выборки с возвратом эквивалентно взятию из этой воображаемой популяции безвозвратно. Как скажут статистики, «бутстраповская выборка – для выборки то же самое, что выборка для популяции».

Процесс строительства бутстраповского интервала уверенности концептуально прост.

1. Мы симулируем новые выборки того же размера, беря их из нашей наблюдаемой выборки, с возвратом взятого назад.
2. Затем для каждой симулированной выборки мы рассчитываем интересующую нас статистику (здесь среднее значение, то есть именно то, что наш инженер-технолог хочет измерить).
3. Наконец, мы строим наш интервал уверенности, глядя на процентили значений, полученных на шаге 2.

Взятие с возвратом означает, что каждое значение каждый раз имеет одинаковую вероятность быть взятым, независимо от того, было оно уже взято или нет.

Например, взятие с возвратом из (A, B, C) с равной вероятностью приведет к (B, C, C), либо (A, C, B), либо (B, B, B) и т. д. Поскольку для каждой из трех позиций существует три возможности, есть $3 \times 3 \times 3 = 27$ возможных симу-

лированных выборок. Если бы мы брали безвозвратно, то это означало бы, что значение нельзя брать более одного раза, и единственными возможными комбинациями были бы перестановки изначальной выборки, такие как (A, C, B) или (B, A, C). Это было бы просто равносильно перетасовке значений, что было бы бессмысленно, потому что среднее значение (или любая другая представляющая интерес статистика) останется точно таким же.

Взятие с возвратом выполняется очень просто как на R, так и на Python:

```
## R
boot_dat <- slice_sample(dat, n=nrow(dat), replace = TRUE)

## Python
boot_df = data_df.sample(len(data_df), replace = True)
```

Прелесть генерирования новых выборок путем взятия только из нашей наблюдаемой выборки заключается в том, что это позволяет избегать каких-либо допущений о распределении данных за пределами наблюдаемой нами выборки. В целях понимания смысла этого преимущества давайте просимулируем B = 2000 бутстраповских выборок (во избежание путаницы я всегда буду использовать B для числа бутстраповских выборок и N для размера выборки) и рассчитаем среднее значение для каждой. Наш исходный код выглядит следующим образом (номера выносок относятся как к R, так и к Python):

```
## R
mean_lst <- list() ❶
B <- 2000
N <- nrow(dat)
for(i in 1:B){ ❷
   boot_dat <- slice_sample(dat, n=N, replace = TRUE)
   M <- mean(boot_dat$times)
   mean_lst[[i]] <- M}
mean_summ <- tibble(means = unlist(mean_lst)) ❸

## Python
res_boot_sim = [] ❶
B = 2000
N = len(data_df)
for i in range(B): ❷
   boot_df = data_df.sample(N, replace = True)
   M = np.mean(boot_df.times)
   res_boot_sim.append(M)
```

❶ **Сначала я инициализирую пустой список результатов, а также B и N.**

❷ **Затем я использую цикл for для генерирования бутстраповских выборок путем взятия с возвратом из изначальных данных, всякий раз вычисляя среднее значение и добавляя его в список результатов.**

❸ **Наконец, в R я переформатирую список в тиббл[1] для удобства использования вместе с ggplot2.**

На рис. 7.2 показано распределение средних значений.

[1] Тибблы (tibbles) – это кадры данных в R, которые адаптируют старые линии поведения, чтобы немного облегчить жизнь во время работы с унаследованным исходным кодом. R – это старый язык, и некоторые вещи, которые были полезны 10 или 20 лет назад, теперь мешают, и тибблы в этом помогают. – *Прим. перев.*

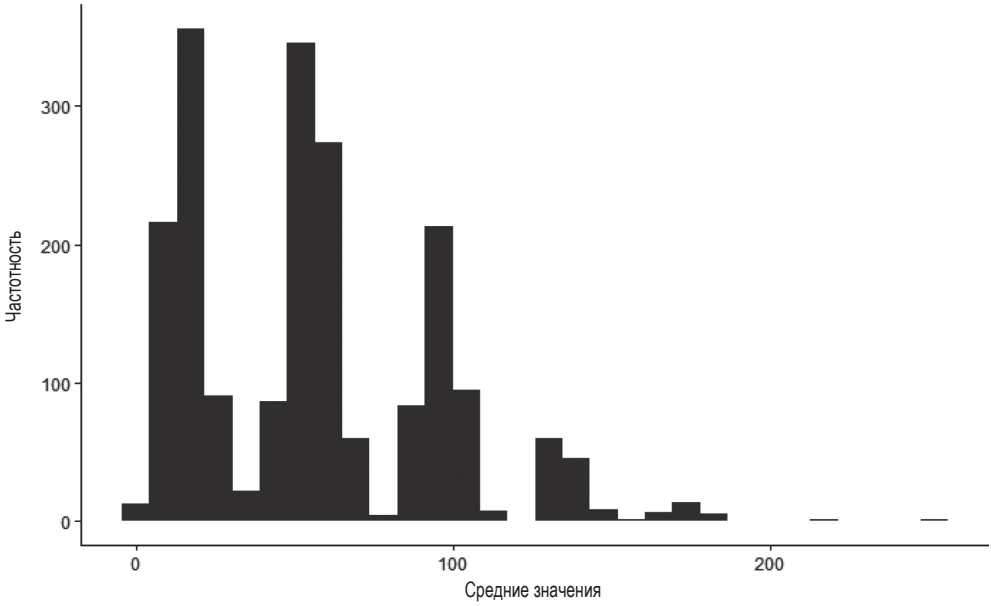

Рис. 7.2 ❖ Распределение средних значений 2000 выборок

Хорошо видно, что гистограмма очень нерегулярна: есть большой пик, близкий к среднему значению нашего изначального набора данных, а также меньшие пики, соответствующие некоторым шаблонам. Учитывая степень экстремальности нашего выброса, каждый из семи пиков соответствует числу повторений в бутстраповской выборке, от нуля до шести. Другими словами, он вообще не появляется в выборках, средние значения которых находятся в первом (крайнем левом) пике, он появляется ровно один раз в выборках, средние значения которых находятся во втором пике, и т. д. Стоит отметить, что даже если бы мы увеличили число бутстраповских выборок, нерегулярность гистограммы не исчезла бы (т. е. «долины» между пиками не были бы заполнены), потому что это отражает грубость наших данных, а не пределы нашего случайного процесса. Диапазон значений внутри наших данных настолько широк, что максимально возможные средние значения при исключении выброса по-прежнему редко бывают достаточно высокими, чтобы соответствовать минимально возможным средним значениям при включении выброса. Если бы значение выброса было уменьшено вдвое и, значит, было бы ближе к остальной популяции, то гистограмма, по-видимому, значительно сгладилась бы, поскольку края пиков численности выбросов накладывались бы друг на друга.

Число бутстраповских выборок тем не менее имеет важность, но по другой причине: чем больше это число, тем больше вы сможете увидеть очень маловероятных выборок и, следовательно, экстремальных значений. Здесь, если бы мы брали выброс 10 раз, абсолютное максимально возможное значение для среднего значения выборки составило бы 413. Оно имеет вероятность $(0.1)^{10}$ (одна десятая в степени 10), из чего следует, что это будет происходить примерно один раз на 10 миллиардов выборок. С нашими всего лишь

2000 выборок мы едва видим значения вокруг 200. Но совокупное среднее или медиана наших выборок будут оставаться прежними плюс или минус незначительные вариации взятия выборок.

Вот некоторые общие рекомендации по числу выборок:

- от 100 до 200 выборок для получения точной центральной оценки (например, коэффициента в регрессии; она называется «центральной», потому что, грубо говоря, находится в центре интервала уверенности, в отличие от границ или пределов интервала уверенности);
- от 1000 до 2000 выборок для получения точных 90%-ных оценок интервала уверенности;
- 5000 выборок для получения точных 99%-ных оценок интервала уверенности.

В общем случае начинайте с малого, а если сомневаетесь, то увеличивайте число и повторяйте попытку. Этот процесс принципиально отличается, например, от многократных аналитических расчетов на ваших данных до тех пор, пока вы не получите нужные вам цифры (т. н. «взлом *p*-значения», или «взлом *p*»); он больше похож на изменение разрешающей способности экрана, глядя на рисунок. Он не влечет за собой никакого риска для ваших аналитических расчетов; он просто занимает больше или меньше вашего времени, в зависимости от размера данных и вычислительной мощности машины.

Учитывая имеющиеся у нас данные, единственный способ увеличить плавность гистограммы состоит в увеличении размера выборки. Однако нам пришлось бы увеличивать размер изначальной выборки из реального мира, а не размер бутстраповских выборок. Почему мы не можем увеличить размер бутстраповских выборок (например, брать 100 значений из нашей выборки, состоящей из 10 значений, с возвратом взятого назад)? Потому что наша цель состоит не в том, чтобы создавать новые выборки, а в том, чтобы определять, насколько далека может быть наша оценка среднего значения, когда мы делаем допущение о том, что популяция пропорционально идентична нашей изначальной выборке. Для этого нам нужно использовать всю информацию изначальной выборки – не меньше и не больше. Создавать более крупные выборки из наших 10 изначальных значений означало бы «притворяться», что у нас больше информации, чем есть на самом деле.

Инженер-технолог готов использовать бутстрап для определения границ интервала уверенности для продолжительности приготовления пирога. Указанные границы определяются на основе эмпирического распределения предыдущих средних. Это означает, что вместо того, чтобы пытаться укладываться в статистическое распределение (например, нормальное), он может просто упорядочивать значения от наименьшего до наибольшего, а затем смотреть на 2.5%-й и 97.5%-й квантили, чтобы найти двухвостый 95%-й интервал уверенности. При 2000 выборок 2.5%-й квантиль равен значению 50-го наименьшего среднего (потому что 2000 * 0.025 = 50), а 97.5%-й квантиль равен значению 1950-го среднего от меньшего до большего или 50-го самого крупного среднего (потому что оба хвоста имеют одинаковое число значений). К счастью, нам не нужно вычислять их вручную:

```
## R (результат не показан)
LL_b <- as.numeric(quantile(mean_summ$means, c(0.025)))
UL_b <- as.numeric(quantile(mean_summ$means, c(0.975)))

## Python
LL_b = np.quantile(mean_lst, 0.025)
UL_b = np.quantile(mean_lst, 0.975)
print("LL_b = ", LL_b)
print("UL_b = ",UL_b)

LL_b = 7.4975000000000005
UL_b = 140.80249999999998
```

Бутстраповский 95%-й интервал уверенности равен [7.50; 140.80] (плюс или минус некоторая разница в процессе взятия выборок), что гораздо реалистичнее. На рис. 7.3 показана та же гистограмма, что и на рис. 7.2, но добавляется среднее значение средних, нормальные границы интервала уверенности и бутстраповские границы интервала уверенности.

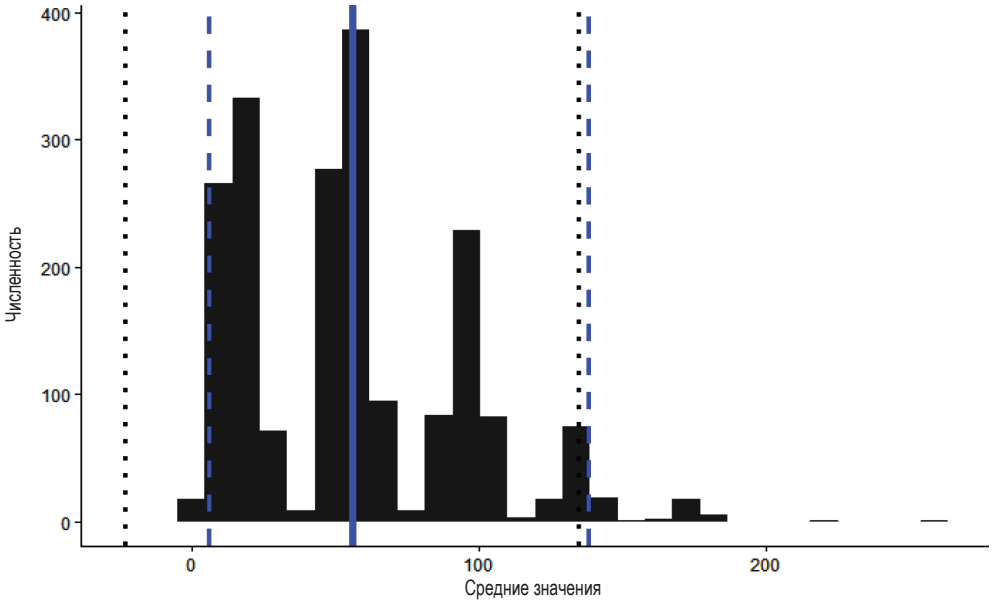

Рис. 7.3 ❖ Распределение средних значений 2000 выборок со средним значением средних (толстая линия), нормальными границами 95%-го интервала уверенности (пунктирные линии) и бутстраповскими границами интервала уверенности (пунктирные линии)

В дополнение к тому, что нижняя граница бутстрапа выше нуля, мы также можем отметить, что верхняя граница бутстрапа немного выше нормальной верхней границы, что лучше отражает асимметрию (скошенность) распределения вправо.

Бутстраповские интервалы уверенности для нерегламентированной статистики

Использование бутстрапа позволило нам создать разумный интервал уверенности, когда традиционный статистический подход отказал. Мы также можем использовать его для создания интервалов уверенности в ситуациях, когда другого способа сделать это нет. Давайте вообразим, например, что руководство C-Mart подумывает о введении обещания на основе времени – «пирог за три часа или скидка 50 %» – и хочет знать, насколько часто пирог в настоящее время приготавливается в течение более трех часов. Нашей оценкой будет процент выборки: это происходит в 1 из 10 наблюдаемых случаев, или в 10 %. Но мы не можем оставить все как есть, потому что вокруг этой оценки существует значительная неопределенность, которую мы должны донести. 10 % из 10 наблюдений являются гораздо более неопределенными, чем 10 % из 100 или 1000 наблюдений.

Тогда как построить интервал уверенности вокруг этого 10%-го значения? С помощью бутстрапа, разумеется. Процесс будет точно таким же, как и ранее, за исключением того, что вместо того, чтобы брать среднее значение для каждой симулированной выборки, мы будем измерять процент значений в выборке, элементы которой превышают 180 минут:

```
## R
promise_lst <- list()
N <- nrow(dat)
B <- 2000
for(i in 1:B){
    boot_dat <- slice_sample(dat, n=N, replace = TRUE)
    above180 <- sum(boot_dat$times >= 180)/N
    promise_lst[[i]] <- above180}
promise_summ <- tibble(above180 = unlist(promise_lst))
LL_b <- as.numeric(quantile(promise_summ$above180, c(0.025)))
UL_b <- as.numeric(quantile(promise_summ$above180, c(0.975)))

## Python
promise_lst = []
B = 2000
N = len(data_df)
for i in range(B):
    boot_df = data_df.sample(N, replace = True)
    above180 = len(boot_df[boot_df.times >= 180]) / N
    promise_lst.append(above180)
LL_b = np.quantile(promise_lst, 0.025)
UL_b = np.quantile(promise_lst, 0.975)
```

Гистограмма результатов показана на рис. 7.4. Между столбцами снова есть «пробел», потому что у нас всего 10 точек данных, поэтому проценты кратны 10 %. Этого не будет в случае с бóльшим числом точек данных; в общем случае процентные значения будут кратными $1/N$, где N – это размер выборки (например, при 20 точках процентные значения будут кратны 5 %).

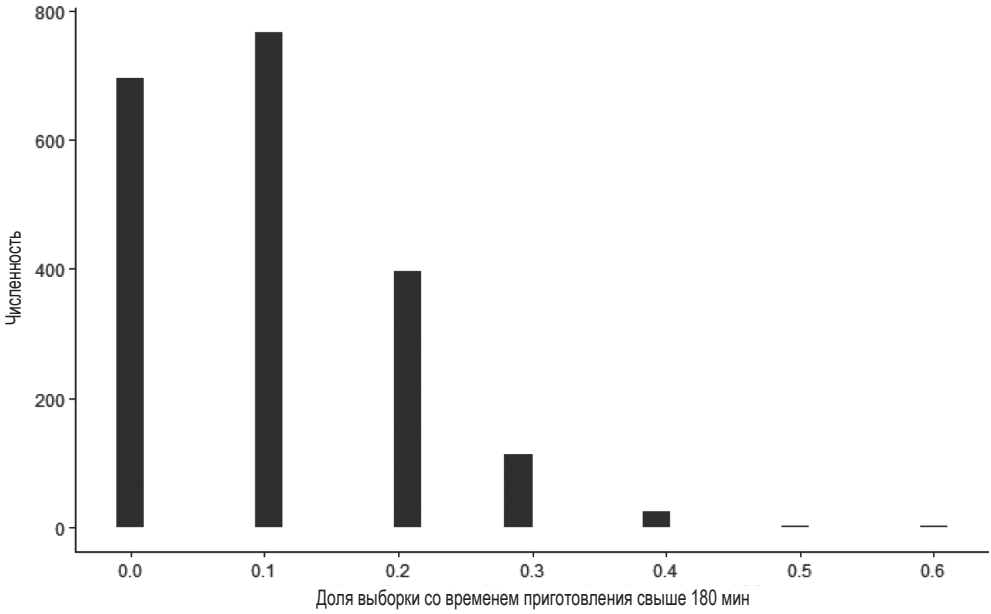

Рис. 7.4 ❖ Гистограмма численности выборок с заданной долей продолжительностей свыше 180 минут

Примерно в 700 из 2000 симулированных выборок не было пирога со временем приготовления свыше 180 минут. Примерно в 750-й был ровно один такой пирог и т. д. Соответствующий 95%-й интервал уверенности равен [0; 0.3]: 50-е наименьшее значение равно 0, а 50-е наибольшее значение равно 0.3.

Другими словами, даже с такими лимитированными данными мы можем с полной уверенностью сказать, что очень маловероятно (хотя и не невозможно), что приготовление более 30 % пирогов занимает более трех часов. Указанный интервал уверенности все еще довольно большой, но не слишком большой для всего 10 наблюдений и такого рода уникальной статистики!

Если вам трудно это охватить, то приведенную выше задачу можно переформулировать, рассчитав интервал уверенности для биномиального распределения с одним успехом в 10 наблюдениях. Для вычисления интервалов уверенности в этом случае в R и Python имеются методы аппроксимации. Они, как правило, консервативнее (т. е. шире), чем наш бутстраповский интервал уверенности, но не сильно шире.

С помощью бутстрапа инженер может оттачивать аналитические расчеты, которые он хотел бы регулярно выполнять со своими данными. Он в состоянии использовать лимитированные данные, чтобы отвечать на разнообразные вопросы с разумной степенью уверенности (и, соответственно, терпимой неопределенностью).

Бутстрап для регрессионного анализа

Хотя строительство интервала уверенности вокруг среднего значения бывает и полезным, регрессия – вот действительно то, чему посвящена эта книга, поэтому давайте посмотрим, как применять бутстрап для этой цели. Наш инженер-технолог из C-Mart хочет определить эффект практики работы на время приготовления, используя те же данные о приготовлении пирога. Соответствующая причинно-следственная диаграмма очень проста (рис. 7.5).

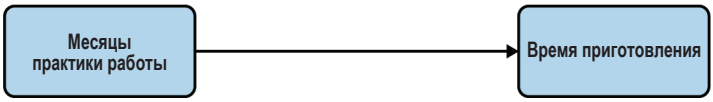

Рис. 7.5 ❖ Причинно-следственная диаграмма для интересующей нас взаимосвязи

Выполнить регрессию на наших данных довольно просто, учитывая, что причинно-следственная диаграмма не выявила никаких спутывающих факторов. Однако результирующий коэффициент не является значимым:

```
## Python (результат не показан)
print(ols("times~experience", data=data_df).fit().summary())

## R
mod <- lm(times~experience, data=dat)
mod_summ <- summary(mod)
mod_summ
...

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  132.389     61.750   2.144   0.0644
experience    -9.819      6.302  -1.558   0.1578
...
```

Наш оценочный коэффициент составляет –9.8, из чего следует, что каждый дополнительный месяц практики, как ожидается, будет сокращать 9.8 минуты времени на приготовление. Однако традиционный интервал уверенности, основанный на стандартной ошибке регрессии, будет равен [–22.2; 2.5]. С традиционной точки зрения это было бы окончанием игры: интервал уверенности включает ноль, из чего вытекает, что месяцы практики могли бы иметь положительное, отрицательное или нулевой эффект на время приготовления, поэтому мы отказались бы делать какие-либо существенные выводы. Давайте вместо этого посмотрим, что говорит нам бутстрап. Процесс точно такой же, как и раньше: мы симулируем выборки из 10 точек данных, беря из нашей изначальной выборки большое число раз с возвратом взятого назад, затем сохраняем коэффициент регрессии. В прошлый раз мы использовали $B = 2000$ выборок. На этот раз давайте использовать $B = 4000$, так как это вынуждает соответствующую гистограмму выглядеть более гладко (рис. 7.6):

```
## R (результат не показан)
reg_fun <- function(dat, B){
   N <- nrow(dat)
   reg_lst <- list()
   for(i in 1:B){
      boot_dat <- slice_sample(dat, n=N, replace = TRUE)
      summ <- summary(lm(times~experience, data=boot_dat))
      coeff <- summ$coefficients['experience','Estimate']
      reg_lst[[i]] <- coeff}
   reg_summ <- tibble(coeff = unlist(reg_lst))
   return(reg_summ)}
reg_summ <- reg_fun(dat, B=4000)

## Python (результат не показан)
reg_lst = []
B = 4000
N = len(data_df)
for i in range(B):
   boot_df = data_df.sample(N, replace = True)
   lin_mod = ols("times~experience", data=boot_df).fit()
   coeff = lin_mod.params['experience']
   reg_lst.append(coeff)
LL_b = np.quantile(reg_lst, 0.025)
UL_b = np.quantile(reg_lst, 0.975)
```

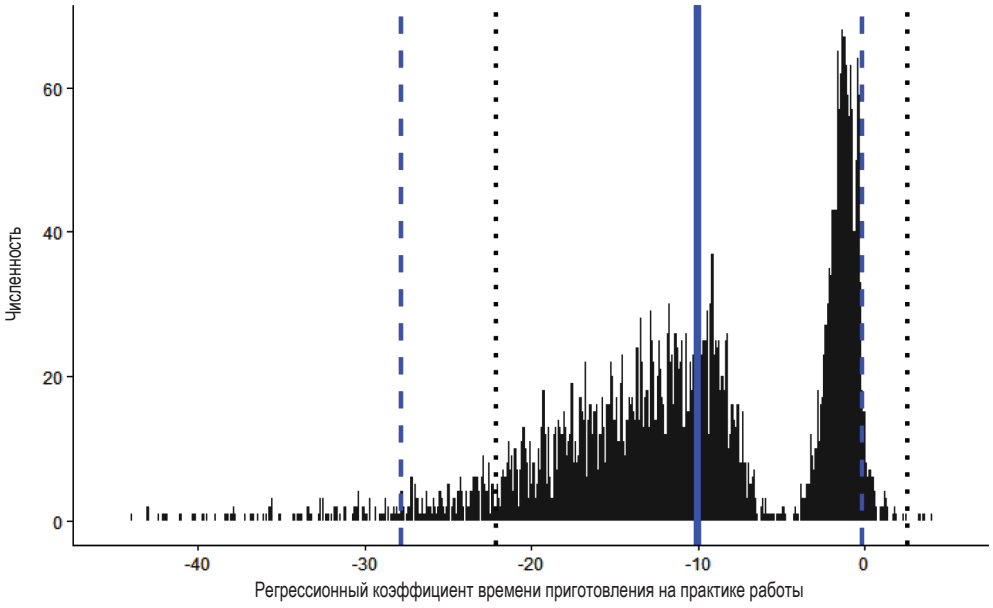

Рис. 7.6 ❖ Распределение регрессионных коэффициентов времени приготовления на практике работы, с их средним значением (толстая линия), бутстраповскими границами интервала уверенности (толстые пунктирные линии) и нормальными бутстраповскими границами интервала уверенности (тонкие пунктирные линии) (B = 4000 бутстраповских выборок)

Бутстраповский интервал уверенности равен [–28; –0.2]. Как видно на рис. 7.6, он снова асимметричен по сравнению с симметричными нормальными границами, имея длинный хвост слева от среднего значения. Крайне нерегулярная форма распределения отражает существование двух соперничающих гипотез:

- высокий и узкий пик возле нуля состоит из выборок, которые не содержат выброс, и в силу этого он соответствует мнению о том, что выброс – это нелепый случай, который не повторится. Это тот интервал уверенности, который вы получили бы, если бы устранили выброс;
- широкий и плоский выступ влево состоит из выборок, которые содержат выброс один или несколько раз. Он отражает гипотезу о том, что выброс действительно представляет наши данные и что его истинная частота, возможно, даже выше, чем в нашей малой выборке.

Это можно трактовать как задаваемый данными сценарный анализ. Что делать, если бы этого шаблона не существовало? Что делать, если бы он доминировал над нашими данными? Вместо того чтобы выбирать между устранением выброса либо предоставлением ему возможности управлять нашими результатами, бутстрап позволяет нам рассматривать все возможности сразу.

В дополнение к составлению интервала уверенности бутстрап можно использовать для определения эквивалента *p*-значения. Если вы посмотрите на результаты нашей регрессии в начале этого раздела, то увидите значение 0.16 для практики работы в столбце для *p*-значений (т. е. в столбце с меткой Pr(>|*t*|)). Вам, вероятно, уже говорили, что коэффициент статистически значим (т. е. статистически значимо отличается от нуля), если его *p*-значение меньше 0.05 или 0.01 в более строгих случаях. Математически говоря, *p*-значение таково, что интервал уверенности от (1 минус *p*-значение) имеет ноль в качестве одной из своих границ. В случае нормальной регрессии ноль является верхней границей 84%-го интервала уверенности. Поскольку 84 % меньше 95 % или 99 %, коэффициент для практики работы не будет считаться статистически значимым. Точно такая же логика может использоваться с бутстрапом; нам просто нужно вычислять долю бутстраповской выборки, коэффициент которой выше нуля, и умножать ее на 2, потому что это двухвостый тест[1]:

```
## Python (результат не показан)
pval = 2 * sum(1 for x in reg_lst if x > 0) / B

## R
reg_summ %>% summarise(pval = 2 * sum(coeff > 0)/n())
```

[1] Для того чтобы увидеть причину, обратите внимание, что если у вас 90%-й интервал уверенности (ИУ), то у вас останется по 5 % значений с каждой стороны, потому что (1 – 0.9)/2 = 0.05. И наоборот, если вы видите, что у вас 5 % значений с одной стороны ИУ, то это 90%-й ИУ, потому что (1 – 2 * 0.05) = 0.9.

```
# Тиббл: 1 x 1
    pval
  <dbl>
1 0.04
```

Из этого следует, что наше эмпирическое бутстраповское *p*-значение[1] равно примерно 0.04, в отличие от традиционного *p*-значения, равного 0.16, основанного на статистических допущениях. Это полезно тем, что люди часто знакомы со статистическими *p*-значениями, и вместо них можно использовать бутстраповские *p*-значения. С точки зрения бизнеса, теперь мы можем быть уверены в том, что коэффициент регрессии находится между нулем и сильно отрицательным значением. В дополнение к этому мы могли бы легко вычислить эквивалент *p*-значения для любого другого порога (например, если бы в качестве порога мы захотели использовать –1 вместо нуля) или для любого интервала, такого как [–1; +1], если бы мы захотели.

Когда следует использовать бутстрап

Будем надеяться, что к настоящему времени вы убедились в добродетелях бутстрапа для малых и странно сформированных наборов данных. Но что делать с крупными или равномерно сформированными наборами данных? Следует ли вам всегда использовать бутстрап? Короткий ответ будет таким: его использование никогда не бывает неправильным, но оно бывает непрактичным или излишним. В случае экспериментальных данных мы будем широко опираться на бутстрап, как мы увидим в части IV книги. В случае анализа наблюдательных данных, которому посвящена эта глава, все обстоит сложнее. На рис. 7.7 представлено дерево решений, которое мы будем использовать. Оно, возможно, покажется немного пугающим, но его можно условно разбить на три блока:

- если вы хотите только центральную оценку (например, регрессионный коэффициент) и соблюдены условия достаточности традиционной оценки, то вы можете использовать ее;
- если вы хотите интервал уверенности и соблюдены условия достаточности традиционного интервала уверенности, то вы можете использовать его;
- в любом другом случае либо в случае сомнений используйте бутстраповский интервал уверенности.

Давайте проведем обзор этих блоков по очереди.

---

[1] Для того чтобы быть совершенно точным, наше бутстраповское *p*-значение лучше было бы назвать бутстраповским достигнутым уровнем значимости (achieved significance level, аббр. ASL).

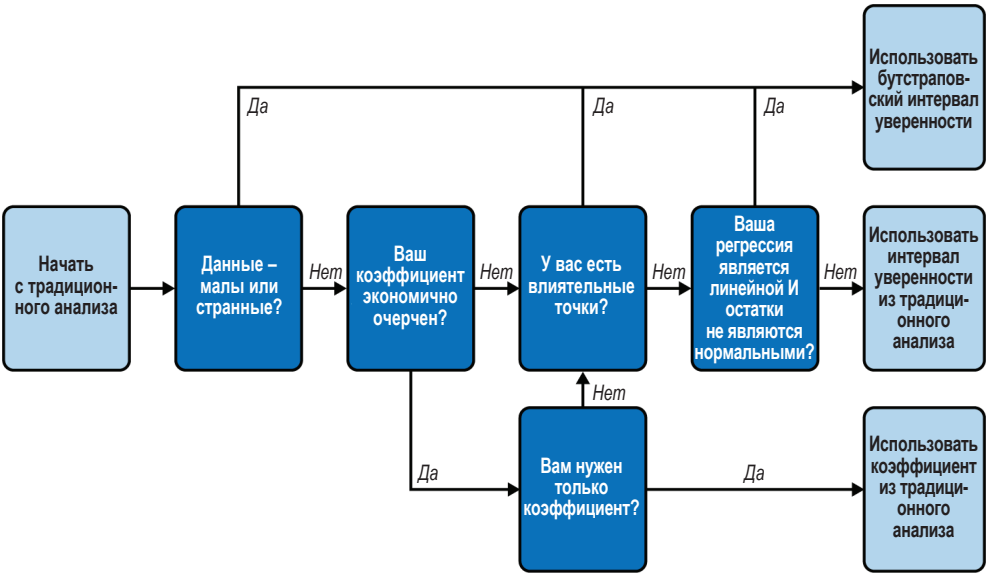

Рис. 7.7 ❖ Дерево решений по использованию бутстрапа

Условия достаточности традиционной центральной оценки

Прежде всего следует иметь в виду то, что бутстрап порождает центральную оценку или коэффициент, очень близкий к тому, который получается традиционными методами (т. е. какими угодно, если бы вы не знали о бутстрапе). Поэтому никогда не имеет смысла начинать непосредственно с бутстрапа, когда традиционная оценка находится на расстоянии одной строки исходного кода.

Однако если ваши данные невелики (обычно менее 100 строк) или в каком-либо отношении странны (например, они имеют несколько пиков или асимметричны), то эта центральная оценка может оказаться неверной. В таком случае вам действительно следует использовать бутстрап для вычисления интервала уверенности, в идеале выводя на экран его результаты в виде гистограммы, как мы это сделали на рис. 7.6.

Схожим образом, если коэффициент близок к границе или порогу и, следовательно, не является экономично четко очерченным, то вам потребуется использовать интервал уверенности, а центральной оценки будет недостаточно.

Даже когда все чисто и четко очерчено настолько, насколько это возможно, вы все равно, возможно, захотите иметь интервал уверенности, например потому, что этого требовал ваш босс или ваш деловой партнер.

Условия достаточности традиционного интервала уверенности

Если вы хотите иметь интервал уверенности, но ваши данные не настолько малы или странны, чтобы требовался бутстраповский интервал уверенности, то возникает вопрос, будет ли традиционный интервал уверенности надежным и достаточным для вашей цели. В этой ситуации вам необходимо выполнить два теста:

- проверить наличие или отсутствие влиятельных точек;
- проверить нормальность остатков регрессии (только если регрессия является линейной).

Традиционный интервал уверенности можно использовать, только если ваши данные лишены влиятельных точек и вы не видите никаких проблем с остатками.

Влиятельные точки – это точки, удаление которых существенно изменило бы регрессию, и существует статистическая величина, расстояние Кука, которая измеряет именно это. Здесь, для наших целей, достаточно знать, что точка данных считается влиятельной, если расстояние до нее больше единицы. В R и Python есть однострочники[1], которые вычисляют расстояние Кука для точек относительно регрессионной модели:

```
## Python (результат не показан)
CD = st_inf.OLSInfluence(lin_mod).summary_frame()['cooks_d']
CD[CD > 1]

## R
> CD <- cooks.distance(mod)
> CD[CD > 1]
     10
1.45656
```

По определению, влиятельная точка не подчиняется тому же шаблону, что и другие точки (в противном случае ее удаление не изменило бы результаты нашей регрессии кардинально). Это означает, что влиятельная точка всегда является выбросом, но выброс не всегда является влиятельной точкой: выброс находится далеко от облака, которое образуется другими точками, но он все равно может быть близок к линии регрессии, рассчитанной без нее, и иметь малое расстояние Кука. В нашем примере с приготовлением пирога выбросная точка также является влиятельной точкой.

Если в ваших данных есть какая-либо влиятельная точка, то это говорит о том, что стандартные допущения о распределении не соблюдаются и, возможно, будет разумнее использовать бутстрап.

---

[1] То есть функции или выражения размером в одну строку. – *Прим. перев.*

Если в ваших данных нет влиятельных точек, то в случае линейной регрессии вам необходимо выполнить вторую проверку: вам нужно убедиться, что остатки регрессии приближенно нормальны. Это не относится к логистической регрессии, поскольку ее остатки подчиняются распределению Бернулли, а не нормальному распределению. Эта проверка отвечает на оба вопроса: «Насколько ненормальное ненормально?» и «Насколько большое велико?», потому что они взаимосвязаны: крупные данные гасят незначительные отклонения от нормальности, поэтому степень ненормальности, которая была бы проблематичной при ста точках, может быть приемлемой при ста тысячах.

Давайте извлечем остатки регрессии и оценим их нормальность визуально. В R мы получаем остатки, применяя функцию `resid()` к нашей линейно-регрессионной модели:

```
## R
res_dat <- tibble(res = resid(mod))
p1 <- ggplot(res_dat, aes(res)) + geom_density() + xlab("regression residuals")
p2 <- ggplot(res_dat, aes(sample=res)) + geom_qq() + geom_qq_line() +
  coord_flip()
ggarrange(p1, p2, ncol=2, nrow=1)
```

Синтаксис в Python также прост: сначала мы получаем остатки из модели, затем чертим график плотности из пакета Seaborn и чертим QQ-график с помощью пакета statsmodels:

```
## Python
res_df = lin_mod.resid
sns.kdeplot(res_df)
fig = sm.qqplot(res_df, line='s')
plt.show()
```

На рис. 7.8 показаны два графика, которые мы создали на языке R, график плотности и QQ-график.

Давайте сначала посмотрим на график плотности слева. Для нормальной плотности мы ожидали бы увидеть кривую с единственным пиком, центрированным на нуле, и плавно уменьшающимися симметричными левым и правым хвостами. Здесь явно не так из-за наличия выброса с большим остатком, поэтому мы приходим к выводу, что остатки распределены ненормально.

График справа представляет собой QQ-график (или квантильно-квантильный график), строящийся с помощью `geom_qq()` или `qqplot()`, который показывает значения наших остатков по оси *x* и теоретическое нормальное распределение по оси *y*. В случае нормальной плотности мы ожидали бы, что все точки будут находиться на линии или очень близко к ней, что здесь опять же не так из-за выброса.

Всякий раз, когда остатки линейной регрессии ненормально распределены, бутстрап будет давать вам более качественные результаты для интервалов уверенности и *p*-значений, чем при традиционном подходе.

Резюмируем: строить бутстраповские интервалы уверенности никогда не бывает неправильным, и вы всегда можете вернуться к ним. Но когда вам

нужна только центральная оценка и вы можете безопасно опираться на нее, или когда вы можете безопасно опираться на традиционный интервал уверенности, то переход к бутстрапу может оказаться излишним.

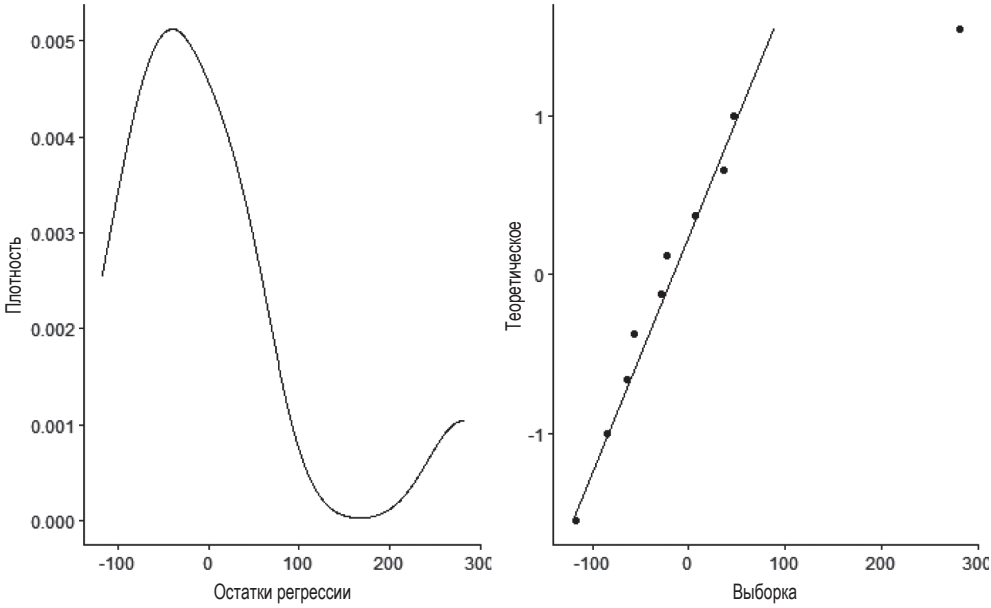

Рис. 7.8 ❖ График плотности (слева) и QQ-график (справа) остатков регрессии

Наконец, давайте остановимся подробнее на том, как определять число используемых бутстраповских выборок.

Определение числа бутстраповских выборок

После того как вы решили использовать бутстрап, вам необходимо определить число выборок для использования в симуляции. Если вы хотите получить просто широкое представление о вариабельности оценки, то согласно Эфрону, «изобретателю» бутстрапа, B = от 25 до 200 дает достаточно устойчивые результаты для главных оценок. Думайте об этом как о 75%-м интервале уверенности. Вы бы не стали ставить на него все свое имущество, но он говорит вам больше, чем просто среднее значение.

С другой стороны, допустим, что вам требуется прецизионное p-значение или 95%-й интервал уверенности, потому что существует неопределенность в отношении наличия или отсутствия в нем критического порога, обычно равного нулю. Тогда вам понадобится гораздо большее B, потому что мы обычно смотрим на 2.5 % наименьших или 2.5 % наибольших значений бутстраповского распределения. При B = 200 нижняя граница 95%-го двухвостового интервала уверенности равна 200 * 2.5 %, или пятому наимень-

шему значению, и аналогичным образом верхняя граница равна пятому наибольшему значению. Пять – это довольно малое число. Вам довольно легко может не повезти, и вы получите пять чисел, которые меньше или больше, чем ожидалось, и отбросите свою границу интервала уверенности. Давайте визуализируем это, повторив бутстраповскую регрессию из предыдущего раздела, только с 200 выборками. Как видно на рис. 7.9, форма распределения в целом аналогична рис. 7.5, но теперь верхняя граница нашего интервала уверенности находится выше нуля.

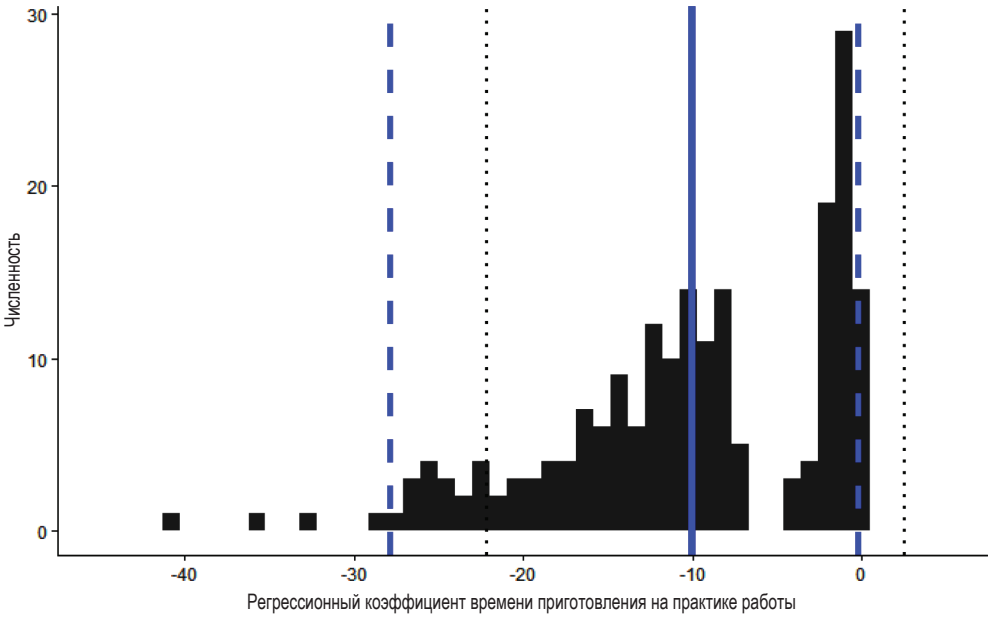

Рис. 7.9 ❖ Распределение коэффициентов регрессии времени приготовления на практике работы с их средним значением (толстая линия), границами бутстраповского интервала уверенности (толстые пунктирные линии) и нормальными границами интервала уверенности (тонкие пунктирные линии) (*B* = 200 бутстраповских выборок)

Следовательно, если деловое решение зависит от того, где эта граница находится по отношению к нулю, то вам нужно будет убедиться, что вы точно ее оценили, увеличив *B*. В таких обстоятельствах наличие 1000 или даже 2000 выборок является общепринятым направляющим принципом. При *B* = 2000 2.5%-й квантиль равен 50-му значению, так что шансы гораздо больше в вашу пользу. В дополнение к этому при очень малых наборах данных, таких как тот, который используется в данной главе, даже симулирование 4000 выборок занимает не более нескольких секунд, поэтому я использовал такое большое *B*.

Давайте подытожим ситуации, когда следует использовать бутстрап с наблюдательными данными, сведя воедино тестовые условия и число выборок:

- всегда начинайте с вашей традиционной регрессионной модели, чтобы получить главную оценку;
- если в ваших данных меньше 100 точек, то всегда используйте бутстрап с B от 25 до 200, чтобы квалифицировать неопределенность вокруг этой оценки;
- при $N > 100$ проверьте свои данные на наличие признаков влиятельных точек (с помощью расстояния Кука) или ненормальности (с помощью графика плотности и QQ-графика остатков). Если что-то выглядит подозрительно, то используйте бутстрап, опять же с B от 25 до 200 для главных оценок;
- независимо от N, если вам нужен прецизионный интервал уверенности или достигнутый уровень значимости (т. н. p-значение), то выполните еще одну бутстраповскую симуляцию с B от 1000 до 2000;
- как только вы поймете, сколько времени требуется на выполнение бутстраповской симуляции на ваших данных с малым или средним B и как выглядит соответствующая гистограмма или интервал уверенности, никогда не стесняйтесь нажимать на кнопку B. Не стесняйтесь выполнять ночную симуляцию с B = 10 000, чтобы к утру получать складно помятый график и взыскательно прецизионную границу интервала уверенности.

Оптимизирование бутстрапа на R и Python

Я показал вам, как применять алгоритм бутстрапа «вручную», чтобы вы могли понять принцип его работы, однако есть пакеты, которые будут выполнять это в меньшем числе строк исходного кода и будут работать быстрее. Они также позволят вам использовать улучшенные версии бутстрапа, которые было бы непрактичным кодировать вручную.

R: пакет boot

Пакет boot и его функция `boot()` обеспечивают мастерскую по принципу одного окна для бутстраповских аналитических расчетов. При всей своей простоте способ генерирования бутстраповских выборок не является интуитивно понятным, поэтому сначала стоит рассмотреть эту функциональность отдельно.

Вспомните, что в предыдущем разделе, посвященном бутстрапу для регрессионного анализа, я генерировал бутстраповские выборки с помощью функции `slice_sample()` перед выполнением на них интересующей нас регрессии:

```
## R
(...)
for(i in 1:B){
    boot_dat <- slice_sample(dat, n=N, replace = TRUE)
    summ <- summary(lm(times~experience, data=boot_dat))
(...)
```

Альтернативный подход к генерированию бутстраповских выборок состоит в том, чтобы брать список индексов, брать из него выборки с возвратом, а затем брать подмножество из наших данных, основываясь на этом списке:

```
## R
> I <- c(1:10)
> I
 [1] 1 2 3 4 5 6 7 8 9 10
> J <- sample(I, 10, replace = TRUE)
> J
 [1] 10 3 1 1 6 1 9 3
> boot_dat <- dat[J,]
```

Это тот подход, который используется в функции `boot()`. Мы должны создать функцию, берущую в качестве аргументов изначальные данные и список индекса *J* и возвращающую интересующую нас переменную (здесь регрессионную оценку практики работы). Функция `boot()` будет заботиться о генерировании этого списка для каждой итерации; нам нужно только извлекать из наших данных подмножество с помощью нее внутри нашей функции:

```
## R
boot_fun <- function(dat, J){
    Boot_dat <- dat[J,]
    summ <- summary(lm(times~experience, data=boot_dat))
    coeff <- summ$coefficients['experience','Estimate']
    return(coeff)
}
```

После создания этой функции мы передаем ее в функцию `boot()` в качестве аргумента `statistic`, а также изначальные данные в качестве data и число бутстраповских выборок в качестве *R* (репликация). Функция `boot()` возвращает объект, который мы затем передаем в функцию `boot.ci()` для получения интервала уверенности:

```
## R
> boot.out <- boot(data = dat, statistic = boot_fun, R = 2000)
> boot.ci(boot.out, conf = 0.95, type = c('norm', 'perc', 'bca'))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 2000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.out, conf = 0.95, type = c("norm", "perc",
    "bca"))

Intervals :
Level      Normal            Percentile           BCa
95 %   (-25.740, 6.567 )   (-28.784, -0.168 )   (-38.144, -0.383 )
Calculations and Intervals on Original Scale
```

Функция `boot.ci()` может возвращать самые разные интервалы уверенности, в соответствии с заданным параметром `type`. norm – это традиционный интервал уверенности, основанный на нормальном распределении.

perc – это процентильный, или квантильный, бутстрап, который мы ранее рассчитывали вручную. bca – это скорректированный на систематическую смещенность и ускоренный процентильный бутстрап (bias-corrected and accelerated percentile Bootstrap, аббр. BC_a). Бутстрап BC_a уточняет процентильный бутстрап, задействуя несколько его статистических свойств; здесь они выходят за рамки нашей книги. Вы можете узнать о них больше в любом из источников, перечисленных в качестве справочных материалов; достаточно сказать, что бутстрап BC_a считается наилучшим практическим приемом при использовании бутстраповских симуляций. Однако он бывает довольно сложным с точки зрения вычислений, поэтому я бы рекомендовал сначала использовать процентильный бутстрап, и как только у вас будет сравнительно окончательная версия исходного кода, попробуйте переключиться на бутстрап BC_a.

В нашем случае, как и ожидалось, нормальный и процентильный интервалы уверенности очень близки к тому, который мы рассчитали вручную. Интервал уверенности BC_a сдвигается влево, укрепляя наши изначальные выводы о том, что коэффициент, скорее всего, является сильно отрицательным.

Теперь, когда вы интуитивно понимаете идею, лежащую в основе использования пакета boot, давайте создадим функцию, допускающую ее многоразовое использование:

```
## R
boot_CI_fun <- function(dat, metric_fun){
    # Задание числа бутстраповских выборок
    B <- 100

    boot_metric_fun <- function(dat, J){
        boot_dat <- dat[J,]
        return(metric_fun(boot_dat))
    }
    boot.out <- boot(data=dat, statistic=boot_metric_fun, R=B)
    confint <- boot.ci(boot.out, conf = 0.90, type = c('perc'))
    CI <- confint$percent[c(4,5)]

    return(CI)
}
```

Функция `boot_CI_fun()` берет в качестве аргументов набор данных и метрическую функцию и возвращает 90%-й интервал уверенности для этой метрической функции на этом наборе данных, основываясь на 100 бутстраповских выборках и процентильном подходе.

Высокопроизводительный бутстрап

Использование пакета boot в типичной ситуации может сделать ваш исходный код в два–пять раз быстрее, но иногда этого просто недостаточно. Если вам потребуется больше вычислительной крутизны, то пакет Rfast предлагает простую имплементацию регрессии, которая даст вам дополнительное улучшение на порядок (например, в двадцать–пятьдесят раз быстрее, чем наш изначальный исходный код).

Оптимизация на Python

Python предлагает специалисту-аналитику совсем другие компромиссы по сравнению с R: с одной стороны, в нем меньше статистических пакетов, и нет эквивалента пакета boot, который бы напрямую имплементировал алгоритм бутстрапа, как в языке R. С другой стороны, я рассматриваю это как более снисходительное отношение к новичкам с точки зрения производительности. Это особенно верно для бутстрапирования, потому что циклы for, к частому применению которых тяготеют новички, сравнительно намного дешевле. Поэтому я ожидаю, что пользователи Python получат гораздо больше пользы от наивной имплементации, с которой мы начали.

Тем не менее если вам потребуется добавить в вашу имплементацию бутстрапа на Python вычислительную крутизну, то вы можете сделать это, перейдя на «полный NumPy»:

```
## Python
# Создание уникального массива numpy для взятия выборок
data_ar = data_df.to_numpy() ❶
rng = np.random.default_rng() ❷

np_lst = []
for i in range(B):
    # Извлечение релевантных столбцов из массива
    boot_ar = rng.choice(data_ar, size=N, replace=True) ❸
    X = boot_ar[:,1] ❹
    X = np.c_[X, np.ones(N)]
    Y = boot_ar[:,0] ❺

    ### Имплементация LSTQ
    np_lst.append(np.linalg.lstsq(X, Y, rcond=-1)[0][0]) ❻
```

❶ **Мы конвертируем наш изначальный кадр данных pandas в массив NumPy.**

❷ **Мы инициализируем генератор случайных чисел NumPy только один раз, вне цикла.**

❸ **Мы создаем наш бутстрапированный набор данных с помощью генератора случайных чисел NumPy, который значительно быстрее, чем метод .sample() пакета pandas для кадров данных.**

❹ **Мы извлекаем предсказательные столбцы из нашего массива и в следующей строке вручную добавляем столбец констант для коэффициента пересечения (тогда как statsmodel ранее обрабатывал это за нас под капотом).**

❺ **Мы извлекаем столбец зависимой переменной из нашего массива.**

❻ **Читая вызовы функций справа налево: мы выполняем подгонку линейно-регрессионной модели с помощью функции np.linalg.lstsq() к данным предсказателя и зависимой переменной. Параметр rcond=–1 удаляет неважное предупреждение. В этой конкретной модели нужное нам значение находится в ячейке [0][0]; нужную вам конкретную ячейку можно найти, выполнив np.linalg.lstsq(X, Y, rcond=–1) один раз и проверив ее результат на выходе. Наконец, мы добавляем значение в наш список результатов.**

Использование полного NumPy значительно повышает вашу производительность, примерно в пятьдесят раз быстрее или около того для крупных наборов данных. Тем не менее наш изначальный исходный код отлично подошел для нашего малого набора данных, он был более удобочитаемым и менее подверженным ошибкам. Более того, если вы выйдете за рамки простых линейных или логистических регрессий, то вам придется поискать простую

имплементацию нужного вам алгоритма в интернете. Но если вам понадобится повысить производительность вашего бутстраповского исходного кода на Python, то теперь вы знаете, как это сделать.

Выводы

Проводя аналитические расчеты на поведенческих данных, часто приходится иметь дело с малыми или странными данными. К счастью, появление компьютерных симуляций дало нам в лице бустрапа отличный инструмент для урегулирования таких ситуаций. Бутстраповские интервалы уверенности позволяют нам правильно квалифицировать неопределенность в наших оценках, не полагаясь на потенциально ошибочные статистические допущения о распределении наших данных. В случае наблюдательных данных бутстрап наиболее полезен, когда наши данные демонстрируют признаки влиятельных точек или ненормальности; в противном случае он нередко бывает излишним. Однако в случае экспериментальных данных сильная опора на *p*-значения при принятии решений означает, что мы будем широко его использовать, как увидим в следующей части книги.

Часть IV

ДИЗАЙН И АНАЛИЗ ЭКСПЕРИМЕНТОВ

Проведение экспериментов – это хлеб с маслом для бихевиористов и исследователей причинно-следственных связей в бизнесе. И действительно, рандомизация распределения испытуемых между экспериментальными группами позволяет нам сводить на нет любое потенциальное спутывание без необходимости даже его выявлять.

Книг об A/B-тестировании предостаточно. Чем отличается изложение в этой? Я бы аргументировал тем, что несколько аспектов его подхода делают его одновременно проще и мощнее.

Во-первых, реорганизация экспериментов внутри причинно-поведенческого каркаса позволит вам создавать более качественные и более эффективные эксперименты и глубже понимать спектр от анализа наблюдательных данных до анализа экспериментальных данных, вместо того чтобы рассуждать о них по отдельности.

Во-вторых, большинство книг по A/B-тестированию основаны на статистических тестах, таких как T-тест средних или тест пропорций. Вместо этого я буду опираться на наших известных рабочих лошадок, линейную и логистическую регрессии, которые сделают наши эксперименты проще и мощнее.

Наконец, традиционные подходы к экспериментированию принимают решения о необходимости имплементировать протестированное вмешательство, основываясь на его *p*-значении, что не приводит к наилучшим деловым решениям. Вместо этого я буду опираться на бутстрап и его интервалы уверенности, которые поступательно получают признание как самая лучшая практика.

Поэтому в главе 8 будет показано, как выглядит «простой» A/B-тест при использовании регрессии и бутстрапа. То есть для каждого клиента мы подбрасываем метафорическую монету. Орел – и они видят версию A, решка – и они видят версию B. Это часто является единственным решением для A/B-тестирования веб-сайта.

Однако если вы заранее знаете людей, из которых вы будете черпать своих экспериментальных испытуемых, то вы можете создавать более сбалансиро-

ванные экспериментальные группы с помощью стратификации. За счет этого эффективность ваших экспериментов может значительно увеличиваться, как я покажу в главе 9.

Наконец, довольно часто случается, что вы не можете рандомизировать на желаемом уровне. Например, вас интересует влияние изменений на клиентов, но вы должны проводить рандомизацию на уровне представителей кол-центра. Это требует кластерной рандомизации и иерархического моделирования, которые мы рассмотрим в главе 10.

Глава 8

Экспериментальный дизайн: основы

Давайте начнем наше исследование экспериментирования с очень простого эксперимента: под влиянием ведущего онлайнового магазина руководство AirCnC решило, что кнопка «бронирование в 1 клик» – это именно то, что необходимо для подстегивания частоты бронирования в AirCnC. Как я уже говорил ранее, мы будем размещать клиентов в наши экспериментальные группы по одному по мере их подсоединения к веб-сайту. Это самый простой из возможных типов экспериментов, и многие компании предлагают интерфейсы, которые позволяют создавать и запускать подобного рода A/B-тесты за считанные минуты.

Указанный простейший эксперимент даст возможность пройти весь процесс, не увязая в технических соображениях.

1. Первым шагом является планирование эксперимента. Здесь на первый план выходит причинно-поведенческая перспектива, которая помогает обеспечивать, чтобы у вас были четко определенные критерии успеха и чтобы вы понимали, что именно вы тестируете и как, по вашему мнению, это повлияет на вашу целевую метрику.
2. Затем, после обзора данных и пакетов, которые мы будем использовать в остальной части главы, я покажу вам, как выполнять случайное размещение и определять размер выборки для вашего эксперимента.
3. Наконец, мы проанализируем результаты эксперимента, который в таком простом случае будет очень быстрым.

Словарь экспериментального дизайна[1] во многом обязан своим статистическим и научным корням. Я буду говорить о «контрольной» (control) и «процедурной» (treatment) группах, а также о «вмешательствах» (intervention), которые, возможно, будут звучать зловеще или казаться излишними, когда мы в общем-то обсуждаем позицию кнопки на странице веб-сайта или размер скидки. Говоря об экспериментах в целом, я мало что могу сделать в части использования более простого словаря; но когда вы говорите со своими деловыми партнерами о конкретном эксперименте, я бы посоветовал вам придерживаться конкретных терминов, относящихся к этому эксперименту (например, «старые и новые креативы», «группа с меньшей скидкой и группа с большей скидкой» и т. д.).

Случай из жизни: однажды, когда я предложил «вмешательство», деловой партнер подумал, что я имел в виду, что он плохо выполняет свою работу и мне нужно было вмешаться. Не лучшее начало для плодотворных и доверительных отношений. Встречайте людей такими, какие они есть, и старайтесь говорить на их языке, вместо того чтобы ожидать, что они будут знать ваш.

Планирование эксперимента: теория изменения

Планирование – важнейшая стадия экспериментального дизайна. Эксперименты могут завершаться безуспешно по целому ряду причин, многие из которых вы не можете контролировать, например если имплементация идет наперекосяк; но слабое планирование является частой причиной неуспешности и той, которую можно контролировать. У любого, кто зарабатывает на жизнь экспериментами, есть ужасные истории об экспериментах, которые, возможно, были или не были технически безупречными, но были совершенно бессмысленными, потому что у людей не было ясности в том, что тестировалось.

В конце концов, никакой процесс вас не спасет, если вы просто проходите по списку телодвижений, не проявляя деловой хватки и здравого смысла, но будем надяться, что формула, которую я собираюсь очертить, поможет

---

[1] В качестве дополнительной справки по языку экспериментального дизайна: *экспериментальный дизайн* (experimental design) – это надлежащая организация эксперимента в целях обеспечения нужного типа данных и их достаточного объема для максимально четкого и эффективного ответа на интересующие вопросы. *Процедуры* (treatments) в экспериментах – это меры, которые исследователи назначают экспериментальным группам. Эксперимент намеренно навязывает процедуру группе объектов или субъектов (испытуемых) в интересах наблюдения за реакцией или результатом. В эксперименте возникает проблема, когда экспериментатор пренебрегает контролем (слежением) за эффектом различий в рассматриваемом процессе. Это приводит к экспериментальному систематическому смещению (bias), т. е. предпочтению одних результатов над другими. Использование рандомизации с целью устранения систематического смещения в экспериментах является обычной практикой. Помимо полностью рандомизированного дизайна, также применяется рандомизированный блочный дизайн, в котором испытуемые сначала делятся на однородные блоки, после чего их случайным образом размещают в процедурную группу. – *Прим. перев.*

вам обеспечивать, чтобы вы отрабатывали все свои базы. Мы позаимствуем понятие из некоммерческого и государственного планирования, а именно теорию изменения (theory of change, аббр. ToC). Если выразить ее в одном предложении, то ваша теория изменения должна связывать то, что вы делаете, с вашей конечной деловой целью и целевой метрикой посредством поведенческого изменения:

> Имплементирование [ВМЕШАТЕЛЬСТВА] поможет нам достичь [ДЕЛОВОЙ ЦЕЛИ], измеряемой [ЦЕЛЕВОЙ МЕТРИКОЙ], за счет [ПОВЕДЕНЧЕСКОЙ ЛОГИКИ].

Я подробно остановлюсь на каждом из четырех компонентов по очереди, но чтобы дать вам представление о том, где мы приземлимся, вот как будет выглядеть наша окончательная теория изменения:

> Имплементирование [кнопки бронирования в 1 клик] поможет нам достичь [более высокой выручки], измеряемой [вероятностью бронирования], за счет [сокращения продолжительности процесса бронирования].

Это можно представить в формате причинно-следственной диаграммы, как показано на рис. 8.1.

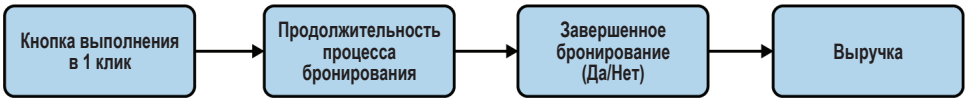

Рис. 8.1 ❖ Теория изменения для нашего эксперимента

Давайте сначала проведем обзор нашей деловой цели и целевой метрики, затем нашего вмешательства и, наконец, нашей поведенческой логики.

Деловая цель и целевая метрика

Вы, возможно, будете удивлены тем, что я начинаю с деловой цели и целевой метрики, а не с определения термина вмешательства. В конце концов, разве мы не должны знать, что мы вообще тестируем? К сожалению, распространенной причиной неуспешности является решение тестировать что-то (часто очередную прихоть руководства или что-то, о чем читал начальник вашего босса) без четкого понимания того, чего мы пытаемся достичь.

Деловая цель

Первым шагом является определение деловой цели эксперимента. Компании обычно пытаются увеличивать свою прибыль, однако просто ставить «более высокую прибыль» в качестве своей деловой цели было бы не очень полезно. Вместо этого я бы рекомендовал углубляться на один уровень ниже и использовать такие переменные, как выручка, затраты, удержание клиентов

и т. д., которые не только являются более важными, но и приносят очевидную пользу компании. Этот шаг, возможно, покажется тривиальным, но на самом деле он может вывести на поверхность разногласия по поводу цели эксперимента (например, снизить затраты или увеличить выручку?). Здесь деловой целью эксперимента с кнопкой выполнения в 1 клик является более высокая выручка (рис. 8.2).

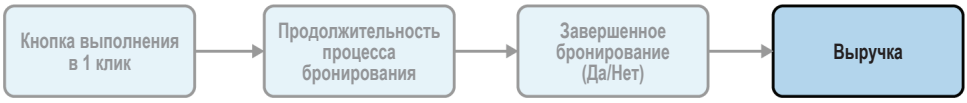

Рис. 8.2 ❖ Нашей деловой целью является выручка

Целевая метрика

Второй шаг состоит в принятии решения о том, как вы будете измерять успех в конце эксперимента, то есть о вашей целевой метрике. Здесь есть компромисс: с одной стороны, вы хотите использовать метрику, максимально приближенную к прибыли, такую как дополнительная выручка в долларах или сниженные затраты; с другой стороны, вы хотите выбрать метрику, максимально приближенную к вашему вмешательству, чтобы сократить посторонний шум.

Компромисс, на который вы часто будете вынуждены идти в этом месте, будет состоять в использовании «опережающих индикаторов» – в сущности, причин переменных, которые вас в конечном счете интересуют. Например, вы, возможно, в конечном счете будете заботиться о пожизненной ценности ваших клиентов (общей сумме, которую они израсходуют на вашу компанию, – lifetime value, аббр. LTV), но вам придется остановиться на сумме бронирования за три месяца. Аналогичным образом регистрация на веб-сайте может использоваться в качестве опережающего индикатора для используемости, используемости для суммы покупок и т. д. Это позволит вам сообщать о результатах намного раньше, чем если бы вы использовали долгосрочные деловые метрики, при этом сохраняя четкое соединение с вашей деловой целью.

Однако если ваша целевая метрика является не финансовой, а операционной, то все может запахнуть жареным. Если кнопка сокращает время, необходимое кому-либо для бронирования поездки, но в остальном никак не увеличивает число бронирований, то является это успехом или нет? А что делать с удовлетворенностью опытом бронирования и чистым баллом промоутера? Их, возможно, не получится перевести непосредственно в долларовые цифры, но в то же время не будет неразумным допустить, что их улучшение окажет положительное влияние на ваш бизнес. Иногда это делается неофициально, выбирая операционные метрики в качестве целей для экспериментирования и допуская, при некотором размахивании руками, что они в итоге принесут пользу компании. Разумеется, вооружившись этой книгой, мы сможем добиться большего. Мы сможем подтвердить и измерить причинно-следственную связь между краткосрочными операционными метриками

и долгосрочными деловыми результатами за счет изучения наблюдательных данных или проведения выделенного эксперимента, как мы увидим позже.

Подводные камни слабых целевых метрик

Здесь цель опять-таки состоит в том, чтобы принимать правильные деловые решения. Я не хочу быть фанатиком долларовых цифр, потому что это было бы неоправданно ограничительным и исключало бы широкий спектр совершенствований в бизнесе. В то же время вы хотите обеспечить, чтобы у вас была измеримая целевая метрика, которую вы можете отслеживать. Здесь есть несколько потенциальных подводных камней, которых вам следует избегать.

Первый подводный камень состоит в выборе того, что вы не в состоянии надежно измерить. Что бы это значило, говоря, что «кнопка в 1 клик облегчает опыт бронирования»? Как бы вы это измерили? Просьба к продуктовому менеджеру или владельцу продукта вынести решение постфактум в отношении того, произошло улучшение или нет, не является измерением. Метрика наподобие «Наши клиенты оценивают веб-сайт как более удобный в навигации, если судить по двухвопросному опросу в конце посещения», возможно, не будет несовершенным косвенным индикатором, но, по меньшей мере, ее можно измерить. Вот почему имеет смысл выражать деловую цель и целевую метрику отдельно: первая выражает ваше истинное намерение, даже если она не поддается измерению, тогда как вторая четко показывает то, что вы намерены измерять. Это позволяет избегать недоразумений и перенесения целевых столбов.

Второй подводный камень заключается в ведении подробного списка метрик, таких как «успех будет заключаться в повышении частоты бронирования, суммы бронирования, удовлетворенности клиентов или чистого балла промоутер», или, что еще хуже, ждать до тех пор, пока вы не увидите результаты, чтобы определить метрики для успеха, например вы изначально думали, что эксперимент улучшит клиентский опыт, но когда приходят результаты, клиентский опыт остается неизменным, а средняя продолжительность сеанса на веб-сайте улучшилась. Проблема с таким подходом заключается в том, что он увеличивает риск ложноположительных результатов (называя результат успешным, когда на самом деле это была чистая случайность)[1]. Однако вполне нормально иметь до двух-трех целевых метрик, которые четко определены до начала эксперимента, при условии что вы принимаете их во внимание при анализе результатов (подробнее об этом позже). Некоторые люди выступают за использование композита из многочисленных метрик (например, средневзвешенного значения), который называется критерием

[1] Вот почему фармацевтические испытания, а также все большее число экспериментов в области социальных наук регистрируются предварительно. Вы не можете задаваться целью тестировать препарат от болезни сердца, а затем принимать решение после того, как он станет эффективным препаратом против выпадения волос, потому что пациенты в процедурной группе засвидетельствовали, как их волосы пошли в рост; вам нужно будет провести второй эксперимент, сам предварительно зарегистрированный, с целью проверки нового гипотетического эффекта.

совокупного оценивания (Overall Evaluation Criterion, аббр. OEC), но лично я чувствую, что это нередко затуманивает вещи больше, чем помогает. Я бы предпочел, чтобы вы четко формулировали свою теорию изменения и характер связанности разных метрик между собой, например ожидаете ли вы, что кнопка выполнения в 1 клик улучшит частоту бронирования и опыт клиентов, или частоту бронирования через клиентский опыт?

В заключение для эксперимента с кнопкой выполнения в 1 клик мы могли бы использовать выручку от бронирования в качестве нашей целевой метрики напрямую, но мы не ожидаем, что наше вмешательство изменит среднюю сумму бронирования, поэтому больше смысла имеет использование вероятности того, что клиент завершит бронирование (рис. 8.3).

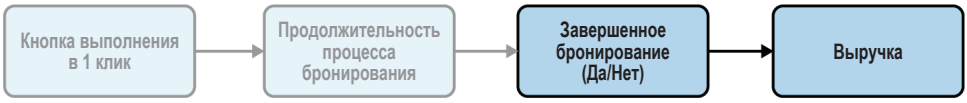

Рис. 8.3 ❖ Добавление целевой метрики, вероятности завершения бронирования

Вмешательство

Имея на руках деловую цель и целевую метрику, мы сможем приступить к работе над определением нашего вмешательства. Идея вмешательства для «кнопки выполнения в 1 клик» здесь исходит от руководства компании (рис. 8.4), но она также могла исходить из исследования пользовательского опыта или поведений: выявление трудностей и возможностей для совершенствования в процессах, продуктах и услугах компании, по правде говоря, является одной из главных задач исследователей в бизнесе, но этот вопрос выходит за рамки данной книги.

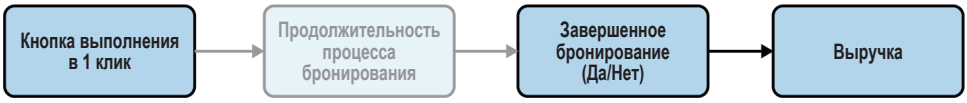

Рис. 8.4 ❖ Добавление вмешательства, кнопка выполнения в 1 клик

На первый взгляд, что может быть проще, чем кнопка бронирования в 1 клик? Большинство из нас видели, как это имплементируется на вебсайте онлайнового розничного продавца или где-то еще, и эта идея кажется совершенно простой. Но между деловой идеей, используемой в таком ключе и знакомой настолько, что каждый сразу чувствует, что знает, что это такое, и конкретной имплементацией может существовать большой разрыв. Если подумать о мельчайших деталях того, как это будет имплементировано, то на самом деле вам предстоит ответить на массу вопросов, на каждый из которых есть несколько возможных ответов:

- в какой момент процесса кнопка становится доступной?
- где находится кнопка?

- как выглядит кнопка? Она того же цвета, что и другие кнопки на странице, или выделяется ярким и насыщенным цветом?
- что написано на кнопке?
- какая информация нам нужна о клиенте, чтобы сделать бронирование в 1 клик доступным, и каким образом мы можем убедиться, что она у нас есть?
- что происходит после того, как клиент нажимает на кнопку? На какую страницу он переходит, и какие действия, если таковые имеются, ему еще предстоит выполнить?
- и т. д.

Допустим, например, что цветовой темой веб-сайта являются пастельно-зеленый и синий, чтобы намекнуть на природу и путешествия, а новая кнопка – блестящая красная. Если эта кнопка увеличивает бронирование, то это, возможно, будет связано с привлекательностью бронирования в 1 клик, но может оказаться и так, что в иных случаях клиенты будут мучиться найти обычную кнопку бронирования и откажутся продолжать. В этом случае причиной увеличения бронирования на самом деле является «более заметная кнопка бронирования», а не «кнопка бронирования в 1 клик», но вы не можете отличить одно от другого, потому что кнопка выполнения в 1 клик тоже была более заметной. К сожалению, вы никогда не сможете тестировать только одну идею, так как вы всегда тестируете также многие аспекты того, как она имплементирована.

Уроком здесь является то, что A/B-тестирование представляет собой мощный, но узкий инструмент. Вам следует быть осторожным в том, чтобы не делать – или не позволять делать другим – громкие заявления о том, что говорит конкретный опыт. Это определенно легче сказать, чем сделать, потому что деловые партнеры зачастую хотят получать ответы, которые являются широкими, четко очерченными и без мелкого шрифта. С учетом сказанного, даже простое заявление в вашей презентации о том, что вы тестируете конкретную имплементацию, а не общую идею, бывает полезным, например: «этот эксперимент будет тестировать влияние кнопки выполнения в 1 клик при таких-то и таких-то условиях, и его результаты не следует интерпретировать как применяющиеся к кнопкам бронирования в более широком смысле».

В более общем плане я бы рекомендовал вам тестировать минимально возможное вмешательство, которое вы в состоянии сделать. В данном случае вы можете попробовать изменить цвет или позицию кнопки бронирования, прежде чем вносить более масштабное изменение, то есть кнопку выполнения в 1 клик. Вы, возможно, получите отпор от своих деловых партнеров, которые нередко желают проводить «омнибус»-тест с многочисленными изменениями сразу; в этом случае дайте понять, что на самом деле вы просто тестируете, что ни одно из изменений не нарушает опыт, в отличие от фактического измерения влияния. Более качественной альтернативой было бы протестировать разные имплементации одной и той же концепции в эксперименте. Если четыре слегка разные экспериментальные процедуры кнопки выполнения в 1 клик оказывают одинаковое влияние, то вы можете увереннее делать выводы об общем влиянии бронирования в 1 клик; с другой

стороны, если они оказывают очень разное влияние, то это говорит о том, что имплементация имеет большое значение и что вам нужно быть очень осторожным с выводами о том, как конкретная имплементация будет обобщаться.

Поведенческая логика

Узнав наши деловые цели и целевые метрики и определив наше вмешательство, последний шаг будет состоять в соединении обоих с помощью поведенческой логики вашей теории изменения: почему и как наше вмешательство будет влиять на нашу целевую метрику?

Это еще один удивительно распространенный источник неуспеха для экспериментов: была выявлена проблема, и кто-то решает имплементировать очередную прихоть руководства, о которой там думали в последнее время, хотя не ясно, почему это поможет решить эту конкретную проблему. Или кто-то решает, что более привлекательный и простой пользовательский интерфейс (UI) увеличит суммы покупок. Для того чтобы быть уверенным в нашем эксперименте, вам нужно уметь формулировать разумную поведенческую историю[1]. В случае бронирования в 1 клик вы, возможно, выдвинете гипотезу о том, что клиенты идентифицируют привлекательное бронирование, но отказываются, не доводя свое бронирование до конца, поскольку процесс бронирования является громоздким; кнопка выполнения в 1 клик будет влиять на вероятность бронирования путем сокращения и упрощения процесса бронирования. В типичной ситуации здесь ваша теория изменения собирается вместе на причинно-следственной диаграмме, в данном случае на той, которую я вам показал в начале главы (рис. 8.5).

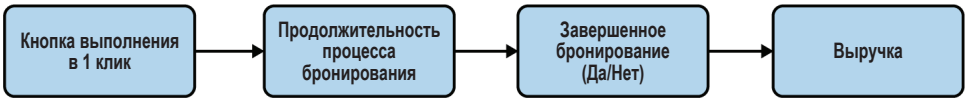

Рис. 8.5 ❖ Полная причинно-следственная диаграмма с нашей теорией изменения

В целом формулирование вашей поведенческой логики имеет две выгоды. Первая состоит в том, что поведенческая логика часто поддается проверке сама по себе. Действительно ли большое число клиентов отказываются продолжать между началом бронирования и его завершением? Если это так, то гипотетическая логика имеет смысл с точки зрения поведенческих данных. Но если, например, большинство клиентов покидают, не начав процесс бронирования с некоторого продукта, например, потому, что они не смогли найти что-то, что им понравилось, либо они были перегружены числом вариантов, то AirCnC пытается решить неправильную задачу; маловероятно, что предложение бронирования в 1 клик улучшит их числа.

---

[1] Работа Венделя (2020) является отличным ресурсом для понимания препятствий и движущих сил поведения и строительства сильной поведенческой логики.

Спросите себя: что подтвердило бы или опровергло бы нашу логику? Как бы выглядели данные, если бы они были истинными или ложными? Если у вас нет под рукой необходимых данных, то, возможно, стоит провести предварительный тест, например привлечь 10 человек для тестирования пользовательских выпечатлений, допустим понаблюдать за тем, как они пытаются использовать ваш веб-сайт, при этом выражая свои мысли вслух. Это, возможно, не полностью подтвердит или опровергнет вашу логику, но, вероятно, даст вам некоторое представление за мизерную стоимость разработки рассматриваемого технического решения. Как гласит часто цитируемая цитата Альберта Эйнштейна: «Если бы у меня был час на решение задачи, то я бы потратил 55 минут на обдумывание задачи и 5 минут на обдумывание решений».

Вторая выгода от формулирования поведенческой логики вашего вмешательства заключается в том, что она, как правило, дает вам представление о потенциальной положительной стороне. Как выглядели бы цифры в наилучшем сценарии, если бы задача была решена? Если допустить, что все клиенты, которые отказываются во время процесса бронирования, завершат его бронированием в 1 клик, то насколько увеличится сумма бронирования? Это является наилучшим сценарием с точки зрения эксперимента, потому что мы исходим из того, что оно решит задачу полностью, что на самом деле маловероятно. Если увеличение в частоте бронирования по этому сценарию не окупит имплементацию бронирования в 1 клик, то даже не утруждайте себя его тестированием.

Как только вы подтвердите, что ваш наихудший сценарий будет прибыльным, вы можете начать думать о своем наиболее вероятном сценарии. Насколько, по нашим ожиданиям, бронирование в 1 клик улучшит частоту бронирования? Несомненно, здесь много субъективности и неопределенности, но, сформулировав поведенческий механизм, вы, как правило, сможете делать разумные догадки. Действительно ли вы ожидаете, что 75 % людей отказываются от процесса бронирования, потому что это занимает слишком много времени? В дополнение к этому бывает полезно поупражняться в четком формулировании допущений и внутренних ощущений людей. Если продуктовый менеджер и исследователь UX сильно расходятся во мнениях относительно процента клиентов, которые отказываются из-за того, что процесс занимает слишком много времени, то вам нужно сначала устранить этот разрыв. Что знает один из них такого, чего не знает другой? Используйте свое деловое чутье и понимание процессов. Если большинство клиентов отказываются на одном и том же шаге процесса, например при оплате, то, скорее всего, с этим конкретным шагом что-то не так – не у всех людей заканчивается терпение в одно и то же время. Затем вы можете сравнить ожидаемые выгоды со стоимостью имплементирования решения. Стоит ли это того?

Вы также можете подойти к этому вопросу с другой стороны: начните с определения точки безубыточности решения, т. е. улучшения целевой метрики, которое сделало бы имплементирование решения прибыльным, а затем подумайте о том, является ли это улучшение реалистичным с поведенческой точки зрения. В психологической перспективе лучше начинать с ожидаемого результата, чем с точки безубыточности: если вы начнете с точ-

ки безубыточности, то у вас будет больше шансов закрепиться на ней и найти причины, чтобы оправдать ее достижимость. Однако во многих случаях вы сначала узнаете точку безубыточности, например если она была рассчитана в ходе предварительного анализа затрат и выгод; ваша компания или деловые партнеры также, возможно, запросят ее и откажутся сначала думать об ожидаемом результате. Не беспокойтесь об этом слишком сильно. Независимо от того, работаете ли вы с ожидаемым результатом или результатом наилучшего случая, она нам понадобится для определения минимального обнаруживаемого эффекта в рамках нашего эксперимента.

Важно, чтобы ваша поведенческая логика соединяла предлагаемое вами решение с вашей целевой метрикой. Не оставляйте это оправданию, дескать, «это улучшит клиентский опыт». Как вы это узнаете? Если ваша логика убедительна, то вы должны быть в состоянии выразить ее в терминах причинно-следственной диаграммы, в которой по меньшей мере некоторые эффекты будут наблюдаемы.

Полезное эмпирическое правило, которое поможет вам формулировать свою логику, состоит в том, чтобы подразделять вашу деловую метрику на компоненты. Например, выручка (или большинство ее вариаций) может быть подразделена на число покупателей, вероятность/частоту покупок, количество приобретенных товаров и уплаченную цену. Определение того, какие компоненты, вероятно, будут затронуты, может позволить вам лучше формулировать деловое предложение. Если ваши деловые партнеры обеспокоены уменьшением числа клиентов и предлагаемое вмешательство, по всей видимости, увеличит только количество приобретенных товаров, то вам необходимо прояснить им, что они все равно будут называть это выигрышем. Указанный подход также может снизить уровень шума в вашем эксперименте; если предлагаемое вмешательство, скорее всего, только увеличит количество покупаемых товаров, то вы можете сосредоточиться на этой метрике и не обращать внимания на отчасти случайные колебания уплаченной цены.

Данные и пакеты

Папка этой главы в репозитории на GitHub[1] содержит два CSV-файла с переменными, перечисленными в табл. 8.1. В таблице галочка (✓) обозначает переменные, присутствующие в этом файле, в то время как крестик (✗) обозначает пропущенные переменные.

В этой главе мы будем использовать следующие ниже пакеты в дополнение к стандартным, указанным в предисловии:

```
## R
library(pwr) # Для традиционного анализа мощности

## Python
import statsmodels.stats.proportion as ssprop # Для размера стандартизированного эффекта
import statsmodels.stats.power as ssp         # Для традиционного анализа мощности
```

[1] См. https://oreil.ly/BehavioralDataAnalysisCh8.

Таблица 8.1. Переменные в наших данных

	Описание переменной	chap8-historical_data.csv	chap8-experimental_data.csv
Gender (Пол)	Категориальная, мужской/женский	✓	✓
Period (Период)	Индекс месяца, 1–32 в исторических данных, 33 в экспериментальных данных	✓	✓
Seasonality (Сезонность)	Годовая сезонность, между 0 и 1	✓	✓
Month (Месяц)	Месяц года, 1–12	✓	✓
Booked (Забронировано)	Двоичная 0/1, целевая переменная	✓	✓
OneClick (Один клик)	Двоичная 0/1, экспериментальная процедура	×	✓

Определение случайного размещения и размера/мощности выборки

После того как вы построите и подтвердите теорию изменения в своем эксперименте, следующим шагом будет определение того, как вы будете выполнять случайное размещение и какой размер выборки вам понадобится.

По моему опыту, ситуация, когда вы впервые проводите эксперимент в определенной среде, нередко становится большим шагом. Серьезный взгляд на ваши исторические данные часто дает удивительные идеи, которые могут придать эксперименту другую форму. В дополнение к этому бывает, что в зависимости от шума в ваших данных и ожидаемого размера воздействия (небольшого, если вы правильно выполнили свою работу по определению узкой области вашего эксперимента) обнаружение того, насколько крупная выборка вам понадобится, усмиряет спесь. Я до сих пор помню, как в первый раз пришли цифры и мне сказали, что нам нужно будет проводить эксперимент в течение почти года.

Случайное размещение

Теория случайного размещения не бывает проще: всякий раз, когда клиент заходит на соответствующую страницу, ему следует показывать текущую версию страницы (на жаргоне экспериментирования именуемую контрольной) с некоторой вероятностью и версию с кнопкой бронирования в 1 клик (именуемую экспериментальной процедурой) с противоположной вероятностью.

Самый простой вариант состоит в размещении с разбивкой 50/50 % до тех пор, пока вы не достигнете целевого размера выборки, но вы можете использовать другую разбивку, если у вас очень большой объем транзакций. Давайте вообразим, например, что вы управляете веб-сайтом со 100 миллионами посещений в день, и вы определили, что необходимый размер выборки

составляет 2 миллиона. Вы могли бы просто успокоиться на разбивке 50/50 % и завершить свой эксперимент примерно за 30 минут. Однако если с вашей экспериментальной процедурой что-то пойдет не так (например, ошибка приведет к аварийному сбою веб-сайта, что по общему признанию является экстремальным случаем), то у вас на руках будет 1 миллион недовольных клиентов, прежде чем вы это узнаете. В дополнение к этому, возможно, клиенты в течение этих 30 минут не являются репрезентативными для вашей полной клиентской базы (например, Китай в это время спит, и вы получаете в основном американских посетителей, или наоборот). В такой ситуации было бы лучше получить 1 миллион посещений, которые вы хотите иметь в своей процедурной группе, в течение более репрезентативного периода, например недели или месяца (вам не нужно беспокоиться о том, что контрольная группа превышает 1 миллион). Для веб-сайта со 100 миллионами посещений в день это приведет к разбивке соответственно 99.86/0.14 % (потому что 1/(7 * 100) = 0.14 %) и 99.97/0,03 % (потому что 1/(30 * 100) = 0.03 %). Для простоты я допущу, что в остальной части главы будет разбивка 50/50 %.

Имплементация исходного кода

С точки зрения программирования, исходя из того, что вы не используете программно-информационное обеспечение, которое заботится об этом за вас, это может быть легко имплементировано на R или Python.

1. Всякий раз, когда новый клиент заходит на соответствующую страницу, мы генерируем случайное число от 0 до 1.
2. Затем мы назначаем клиенту группу, т. е. размещаем его в ней, на основе этого случайного числа: если K – это число групп, которые мы хотим (включая контрольную группу), то все люди со случайным числом менее $1/K$ назначаются в первую группу; все люди со случайным числом от $1/K$ до $2/K$ назначаются во вторую группу и т. д.

Здесь K равно 2, что переводится в очень простую формулу:

```
## R
> K <- 2
> assgnt = runif(1,0,1)
> group = ifelse(assgnt <= 1/K, "control", "treatment")

## Python
K = 2
assgnt = np.random.uniform(0,1,1)
group = "control" if assgnt <= 1/K else "treatment"
```

Подводные камни случайного размещения

Однако есть некоторое число тонкостей, которые способны сбить с толку начинающих экспериментаторов. В этой главе я охвачу два вопроса: выбор времени и уровень размещения.

ВЫБОР ВРЕМЕНИ СЛУЧАЙНОГО РАЗМЕЩЕНИЯ Первый – это определение правильной точки в процессе случайного размещения. Предположим,

что всякий раз, когда клиент попадает на первую страницу веб-сайта, вы размещаете его либо в контрольную, либо в процедурную группу. Многие из этих клиентов никогда не достигнут точки бронирования и, следовательно, не увидят ваш интерфейс бронирования. Это значительно снизило бы эффективность вашего эксперимента, потому что вы практически экспериментировали бы только с малой долей вашей выборки.

При определении клиентов, которые должны участвовать в эксперименте, и того, когда их следует размещать в процедурной группе, вам следует отрефлексировать вопрос о том, как экспериментальная процедура будет имплементироваться, если эксперимент пройдет успешно. Дизайн вашего эксперимента должен включать тех же людей, которые будут видеть экспериментальную процедуру, если бы она была имплементирована в бизнесе в обычном порядке, и только их. Например, посетители, которые покидают веб-сайт до бронирования, так и так не увидят кнопку, тогда как любая будущая промоакция или изменение страницы бронирования будет строиться дополнительно, так сказать «поверх» кнопки, которая всегда будет присутствовать. Следовательно, небронирующие посетители должны быть исключены, но клиенты с промоакцией должны быть включены.

УРОВЕНЬ СЛУЧАЙНОГО РАЗМЕЩЕНИЯ Вторая трудность состоит в обеспечении того, чтобы случайное размещение происходило на «правильном» поведенческом уровне. Я объясню, что это значит, на примере. Допустим, что посетитель заходит на веб-сайт AirCnC и начинает бронирование, но затем по какой-то причине покидает веб-сайт (он отсоединился, настало время ужина и т. д.) и возвращается к нему позже. Должен ли он видеть одну и ту же страницу бронирования? Если ему было предложено пройти бронирование в 1 клик в первый раз, то следует ли ему предлагать его во второй раз?

Проблема здесь в том, что на самом деле существует несколько уровней, которые потенциально могут иметь смысл. Вы могли бы относить к контрольной либо процедурной группе на уровне одного посещения веб-сайта, бронирования, независимо от числа посещений, либо на уровне учетной записи клиента (который может быть или не быть одним и тем же лицом, если несколько человек в домохозяйстве используют одну и ту же учетную запись). К сожалению, здесь нет жестких правил, и правильный подход должен определяться в каждом конкретном случае, думая о выводах, которые вы хотите сделать, и о том, как будет выглядеть постоянная имплементация.

Во многих случаях имеет смысл выполнить размещение на самом близком уровне из возможных для человека: учетная запись клиента, если вы не способны отличить людей в домохозяйстве, либо отдельный клиент, если у каждого из них есть субсчет, как в случае, например, с Netflix. У людей есть устойчивые воспоминания, и чередующиеся варианты для одного и того же человека могут сбить с толку. Здесь это означало бы, что наш клиент AirCnC должен видеть кнопку выполнения в 1 клик на протяжении всего эксперимента, независимо от того, сколько посещений и бронирований он совершит за это время. К сожалению, из этого следует, что вы не можете просто метафорически бросать кости всякий раз, когда кто-то начинает бронирование на веб-сайте, чтобы определить его размещение; вам нужно отслеживать, был

ли он размещен в прошлом, и если да, то в какую группу. Для эксперимента на веб-сайте это можно сделать с помощью файлов cookie (если допустить, что клиент их разрешает!).

Уровень, на котором вы выполняете случайное размещение, также должен быть уровнем, на котором вы рассчитываете размер выборки. Если вы выполняете размещения на уровне клиента, а клиенты совершают в среднем три посещения в месяц, то вам для эксперимента потребуется в три раза больше времени, чем если бы вы выполняли размещения на уровне посещения. Но уровень, который вы выбираете для случайного размещения, должен определять величину вашей выборки, а не наоборот!

Какой бы уровень вы ни выбрали, вам придется отслеживать размещение(я), чтобы позже связать его(их) с результатами бизнеса. Вот почему лучший в своем классе подход заключается в использовании централизованных систем, которые регистрируют все размещения и соединяют их с идентификаторами клиентов в базе данных, чтобы та могла со временем предоставлять клиентам согласованный опыт.

В более широком смысле эти две трудности указывают на то, что имплементация делового эксперимента почти всегда является сложным техническим мероприятием. В настоящее время множество поставщиков предлагают решения отчасти в стиле «подключи и играй», которые скрывают сложность под капотом, в особенности для экспериментов на веб-сайтах. Независимо от того, полагаетесь ли вы на них либо на своих внутренних технических специалистов, вам нужно будет понять, как они выполняют случайное размещение, чтобы убедиться в том, что вы получаете желаемый эксперимент.

Проверять правильность работы системы неплохо, начиная с A/A-теста, в котором есть случайное размещение, но две группы видят одну и ту же версию страницы. Это позволит вам проверить, что в двух группах действительно находится одинаковое число людей и что они ничем существенным не отличаются.

Размер выборки и анализ мощности

После того как мы узнаем, что и как мы собираемся тестировать, нам нужно определить размер выборки. В некоторых случаях, как в нашем эксперименте с бронированием в 1 клик, мы можем выбрать размер выборки сами: мы можем просто принять решение о желаемой нами продолжительности проведения эксперимента. В других случаях размер выборки может быть определен за нас или, по меньшей мере, его максимальный размер. Если мы собираемся проводить тест по всей популяции наших клиентов или сотрудников, то мы не можем увеличивать эту популяцию только ради экспериментирования!

Независимо от ситуации, в которой мы находимся, мы будем смотреть на размер выборки не в вакууме, а в отношении с другими экспериментальными переменными, такими как статистическая значимость. Понимание того, как эти переменные связаны, имеет решающее значение для обеспечения того, чтобы наши экспериментальные результаты были пригодны для ис-

пользования и чтобы мы делали из них правильные выводы. К сожалению, это очень сложные и нюансированные статистические концепции, и они не были разработаны для принятия деловых решений.

В соответствии с духом этой книги я сделаю все возможное, чтобы объяснить эти статистические концепции и условности в контексте принятия деловых решений. Затем я поделюсь своими оговорками по поводу традиционных условностей и предложу свои два цента относительно того, как их можно отрегулировать, оставаясь в традиционных рамках. И наконец, я опишу альтернативный подход, который медленно набирает обороты и который, я думаю, является превосходным, а именно использование компьютерных симуляций.

Чуть-чуть теории статистики без математики

При проведении эксперимента, такого как кнопка бронирования в 1 клик, наша цель состоит в том, чтобы принять правильное решение: следует нам ее имплементировать или нет? К сожалению, даже проведя эксперимент (или сотню), мы никогда не сможем быть на 100 % уверены в том, что принимаем правильное решение, потому что у нас есть только частичная информация. Определенно, если бы мы проводили эксперимент годами подряд, то мы могли бы достичь точки, когда есть только один шанс из миллиона, что мы ошибаемся, но никогда не равный строго нулю. Более того, мы обычно не хотим проводить эксперимент годами подряд, когда вместо этого мы могли бы проводить другие эксперименты! Следовательно, существует компромисс между размером выборки эксперимента и нашей степенью определенности.

Поскольку мы никогда не можем знать заранее, будет или нет конкретное решение правильным, наш подход будет заключаться в том, чтобы попытаться отобрать хороший размер выборки и хорошее правило принятия решения до начала эксперимента. Что здесь означает «хороший»? Так вот, самый лучший возможный размер выборки и правило – это те, которые максимизируют нашу ожидаемую прибыль с течением времени. Соответствующие расчеты выполнимы, но требуют передовых методов, которые выходят за рамки этой книги[1]. Вместо этого мы будем полагаться на следующие ниже меры:

- если допустить, что наша кнопка выполнения в 1 клик действительно увеличивает частоту бронирования, то какова вероятность того, что наша имплементация этой кнопки будет правильной? Это называется вероятностью «истинно положительных результатов». С другой стороны, если есть положительный эффект и мы ошибочно заключаем, что его нет, то это называется «ложноотрицательным результатом» (т. н. ложным отрицанием);
- если допустить, что наша кнопка выполнения в 1 клик не оказывает заметного эффекта (или, не дай бог, отрицательного эффекта!) на частоту

[1] В случае если вам интересно, то вам нужно будет использовать байесовы методы. Может быть, я дойду до этой темы в следующем издании этой книги! Между тем книга «Думай как Байес» Аллена Дауни (*Think Bayes*, Allen Downey, O'Reilly) является одним из самых доступных введений, которые я знаю по данной теме.

бронирования, то какова вероятность того, что наша имплементация кнопки будет неверной? Эта вероятность называется вероятностью «ложноположительных результатов». С другой стороны, если эффекта нет и мы правильно делаем вывод, что эффекта нет, то это называется «истинно отрицательным результатом» (т. н. истинным отрицанием).

Эти различные конфигурации резюмированы в табл. 8.2.

Таблица 8.2. Принятие правильного решения в правильной ситуации[1]

		Имплементируем ли мы кнопку бронирования в 1 клик?	
		ДА	**НЕТ**
Увеличивает ли кнопка бронирования в 1 клик частоту бронирования?	**ДА**	Истинно положительный результат	Ложноотрицательный результат
	НЕТ	Ложноположительный результат	Истинно отрицательный результат

Мы хотели бы, чтобы наши частоты истинно положительных и истинно отрицательных результатов были как можно выше, а наши частоты ложноположительных и ложноотрицательных результатов были как можно ниже. Однако простота этой таблицы обманчива, и на самом деле она охватывает бесконечное число ситуаций: когда мы говорим, что кнопка увеличивает частоту бронирования, это может означать, что увеличение составляет 1 %, 2 % и т. д. С другой стороны, когда мы говорим, что кнопка не увеличивает частоту бронирования, это может означать, что она имеет строго нулевой эффект или что она снижает частоту бронирования на 1 %, 2 % и т. д. Все эти размеры эффекта должны быть учтены для расчета совокупных частот истинно положительных и истинно отрицательных результатов, что было бы слишком сложно. Вместо этого мы будем опираться на два пороговых значения.

Первый – это влияние строго нулевое для всех испытуемых, также именуемое «резко нулевой гипотезой»[2] (нерезко нулевая гипотеза будет средненулевым эффектом для всех субъектов). Частота ложноположительных результатов для этого значения называется статистической значимостью нашего эксперимента. Поскольку отрицательное воздействие было бы легче уловить, чем нулевой эффект, частота ложноположительных результатов для любого отрицательного значения будет по меньшей мере такой же, как статистическая значимость, а более крупные положительные эффекты будут иметь более высокие частоты ложноположительных результатов. Наиболее

---

[1] Для справки: частота ложноположительных результатов – это вероятность того, что будет поднята ложная тревога, т. е. что будет получен положительный результат, когда истинное значение отрицательно. Частота ложноотрицательных результатов, также именуемая частотой промахов, – это вероятность того, что тест пропустит истинно положительный результат. Частота истинно положительных результатов (также именуемая чувствительностью) – это вероятность того, что в результате теста фактический положительный результат будет положительным. Частота истинно отрицательных результатов (также именуемая специфичностью) – это вероятность того, что в результате теста фактический отрицательный результат будет отрицательным. – *Прим. перев.*

[2] Англ. sharp null hypothesis.

распространенным соглашением в академических исследованиях является установление статистической значимости на уровне 5 %, хотя в некоторых областях, таких как физика элементарных частиц, иногда она может достигать 0.00005 %.

Второе пороговое значение устанавливается при некотором положительном эффекте, в измерении которого мы заинтересованы. Например, мы могли бы сказать, что хотим подобрать размер выборки, чтобы быть «достаточно уверенными» в том, что мы уловим увеличение частоты бронирования на 1 %, но мы не против отсутствия меньших эффектов, чем этот. Указанное значение часто называют «альтернативной гипотезой», а частота истинно положительных результатов для этого значения называется статистической мощностью нашего эксперимента. Поскольку более крупные эффекты было бы легче улавливать, частота истинно положительных результатов для любого большего значения будет по меньшей мере такой же, как и статистическая мощность, а более крупные положительные эффекты будут иметь более высокие частоты истинно положительных результатов. Традиционно считается, что «разумная уверенность» имеет 80 %. Стоит подчеркнуть, что это не означает, что ваш эксперимент «имеет мощность 80 %», и это словосочетание на самом деле бессмысленно само по себе: эксперимент также имеет мощность 90 % для некоторого большего размера эффекта и мощность 70 % для некоторого меньшего размера эффекта и т. д.

Поэтому наша таблица, обновленная в соответствии с традиционным соглашением, будет выглядеть как табл. 8.3.

Таблица 8.3. Пороговые значения, используемые в традиционном статистическом подходе

	Имлементируем ли мы кнопку бронирования в 1 клик?	
	ДА	**НЕТ**
Кнопка бронирования в 1 клик увеличивает частоту бронирования более чем на 1 %	> 80 % (крупнее для более крупных размеров эффекта)	< 20 % (меньше для более крупных размеров эффекта)
Кнопка бронирования в 1 клик увеличивает частоту бронирования ровно на 1 %	**80 % (статистическая мощность)**	**20 % (1 минус статистическая мощность)**
Кнопка бронирования в 1 клик увеличивает частоту бронирования менее чем на 1 %	< 80 % (крупнее для более крупных размеров эффекта)	> 20 % (меньше для более крупных размеров эффекта)
Кнопка бронирования в 1 клик не имеет совершенно никакого воздействия на частоту бронирования	**5 % (статистическая значимость)**	**95 % (1 минус статистическая значимость)**
Кнопка бронирования в 1 клик строго увеличивает частоту бронирования	< 5 % (меньше для более отрицательных размеров эффекта)	> 95 % (крупнее для более отрицательных размеров эффекта)

Я не большой фанат использования произвольного числа исключительно потому, что это условно, и вы должны, не стесняясь, свободно корректировать условие «80 % мощности» в соответствии с вашими потребностями. Использование мощности 80 % для релевантного порогового размера эффекта

означало бы, что если бы вмешательство имело в среднем именно такой размер эффекта, то у вас была бы 20%-ная вероятность того, что вмешательство не будет имплементировано, потому что вы ошибочно получили отрицательный результат. Для больших и дорогостоящих вмешательств, которые трудно проверить, я считаю, что она слишком низкая, и я лично нацелился бы на мощность 90 %. С другой стороны, чем выше мощность, которую вы хотите, тем больше должен быть размер вашей выборки. Возможно, вам не захочется тратить полгода на то, чтобы получить абсолютную определенность в отношении ценности кнопки выполнения в 1 клик, если за это время ваш конкурент дважды полностью обновит свой веб-сайт и будет есть ваш обед.

По моему личному опыту, одним из ключевых, но часто игнорируемых соображений для анализа мощности и определения размера выборки в реальном мире является скорость организационного тестирования: сколько экспериментов вы можете провести за год? Во многих компаниях это число ограничено чьим-то временем (аналитика или делового партнера), принятым в компании циклом планирования, бюджетными лимитами и т. д., но не числом располагаемых клиентов. Если вы можете реально надеяться спланировать, протестировать и имплементировать только одно вмешательство в год, то действительно ли вы хотите провести трехмесячный эксперимент, а затем ничего не предпринимать до конца года? С другой стороны, если вы можете проводить один эксперимент в неделю, то действительно ли вы хотите потратить три месяца на то, чтобы получить определенность в отношении положительного, но посредственного воздействия, вместо того чтобы 12-кратно рискнуть в большом? Поэтому после выполнения математических расчетов вы всегда должны делать проверку исправности относительно продолжительности вашего эксперимента, основываясь на вашей скорости тестирования, и надлежаще ее корректировать.

Что касается статистической значимости, то традиционный подход вводит асимметрию между контрольной версией и экспериментальной процедурой с порогом статистической значимости 95 %. Планка подтверждающих данных, которую экспериментальная процедура должна пройти, чтобы быть имплементированой, намного выше, чем для контрольной, которая имплементирована по умолчанию. Допустим, вы готовите новую маркетинговую кампанию по электронной почте, и у вас есть два варианта тестирования. Почему одной версии должно быть предоставлено преимущество сомнения перед другой? С другой стороны, если у вас есть рекламная кампания, которая работает уже много лет и для которой вы провели сотни тестов, то текущая версия, вероятно, будет чрезвычайно хорошей, и 5%-ная вероятность неправильного отказа от нее, возможно, будет слишком высокой; правильный порог здесь может составлять 99 % вместо 95 %. В более широком смысле полагаться на общепринятое значение, одинаковое для всех экспериментов, кажется мне упущенной возможностью поразмыслить о соответствующих издержках ложноположительных и ложноотрицательных результатов. В случае кнопки выполнения в 1 клик, которая легко обратима и имеет минимальные затраты на имплементацию, я бы тоже установил порог статистической значимости равным 90 %.

Резюмируя со статистической точки зрения, наш эксперимент можно подытожить четырьмя значениями:

- статистическая значимость, нередко представленная греческой буквой бета (β);
- размер эффекта, выбранный для альтернативной гипотезы, т. н. минимальный обнаруживаемый эффект;
- статистическая мощность, нередко представленная в виде 1 – α, где α – это частота ложноотрицательных результатов для выбранного альтернативного размера эффекта;
- размер выборки нашего эксперимента, представленный N.

Эти четыре переменные называются B.E.A.N. (бета, размер эффекта, альфа, размер выборки N), и их определение для эксперимента называется «анализом мощности»[1]. Для нашего эксперимента с кнопкой выполнения в 1 клик мы выбрали первые три из них, и нам нужно только определить размер выборки. Далее мы увидим, как это сделать с помощью традиционных статистических формул, а затем посмотрим, как это сделать с помощью компьютерных симуляций.

Традиционный анализ мощности

Статистики разработали формулы определения требуемого размера выборки для некоторых статистических тестов. Учитывая, что вместо тестов мы будем опираться на регрессию, вы можете задаться вопросом, почему мы хотели бы использовать эти формулы. По моему опыту, это даст вам значения того же порядка, что и «истинный» требуемый размер выборки. Это быстрый и простой способ получить разумные начальные значения для ваших симуляций, если вы понятия не имеете, должен ли размер вашей выборки быть 100 или 100 000 (в данном конкретном примере в конце наших симуляций мы получим почти точно такой же размер выборки!).

Тест пропорций является стандартным тестом, и формула для расчета соответствующего размера выборки легко доступна на R и Python. Давайте сначала посмотрим на формулу R.

При средней частоте бронирования 18.25 % в наших исторических данных выбранный размер эффекта в 1 % будет транслироваться в ожидаемую частоту бронирования для нашей процедурной группы, равной 19.25 %. Для стандартных значений параметров – статистическая значимость = 0.05 и мощность = 0.8 – соответствующая формула на R будет такова:

```
## R
> effect_size <- ES.h(0.1925,0.1825)
> pwr.2p.test(h = effect_size, n = NULL, sig.level = 0.05, power = 0.8,
                      alternative = "greater")

     Difference of proportion power calculation for binomial distribution
                      (arcsine transformation)
```

[1] См. работу Андерсона (2019).

```
             h = 0.02562255
             n = 18834.47
     sig.level = 0.05
         power = 0.8
   alternative = greater

NOTE: same sample sizes
```

Синтаксис всех функций для анализа мощности в пакете `pwr` одинаков, за исключением обозначения размера эффекта, которое меняется от одной формулы к другой:

- `h` – размер эффекта, основанный на увеличении вероятности, которую мы хотим иметь возможность наблюдать по сравнению с базовой вероятностью;
- `n` – размер выборки для каждой группы;
- `sig.level` – статистическая значимость;
- `power` – статистическая мощность, равная 1 – α.

При вводе формулы вы должны предоставить значения для трех из этих переменных и установить для оставшейся значение NULL. В приведенной выше формуле мы вычисляем размер выборки, поэтому устанавливаем `n` = NULL.

Обратите внимание, что для теста двух пропорций величина эффекта для статистических целей зависит от базовоуровневой частоты; увеличение на 5 % по сравнению с базовым уровнем 10 % или 90 % «важнее», чем по сравнению с базовым уровнем 50 %. К счастью, пакет pwr предоставляет функцию `ES.h()`, которая транслирует ожидаемую вероятность и базовоуровневую вероятность в правильный размер эффекта для этой формулы.

Обратите также внимание на параметр в конце формулы: `alternative` указывает, какой тест вы хотите выполнить – односторонний (`greater` или `less`) либо двухсторонний (`two.sided`). До тех пор, пока наша экспериментальная процедура не увеличивает частоту бронирования, нам на самом деле все равно, будет ли у нее такая же частота бронирования или более низкая частота бронирования по сравнению с нашей контрольной версией; в любом случае, мы не будем ее имплементировать. Это означает, что мы можем выполнить односторонний тест вместо двухсторонего, установив `alternative = 'greater'`.

Исходный код для Python аналогичен, используя функцию `proportion_effectsize()` из пакета `statsmodels.stats.proportion`:

```
## Python
effect_size = ssprop.proportion_effectsize(0.194, 0.184)
ssp.tt_ind_solve_power(effect_size = effect_size,
                       alpha = 0.05,
                       nobs1 = None,
                       alternative = 'larger',
                       power=0.8)
Out[1]: 18950.818821558503
```

Размер выборки, возвращаемый этой формулой, составляет 18 800 в расчете на группу (плюс или минус некоторые незначительные различия между

R и Python), т. е. всего 37 600, а значит, мы можем достичь необходимого размера выборки чуть менее чем за четыре месяца. Это было легко! Использование статистической значимости 0.1 и мощности 0.9 позволило бы получить размер выборки, равный 20 000, в расчете на группу, чуть-чуть больше.

Что означает общий размер выборки 40 000 для статистической значимости 0.1 и мощности 0.9 с точки зрения модели принятия решения, которую я изложил в предыдущем разделе? Представьте себе следующее:

- вы проводите очень большое число экспериментов с общим размером выборки, равным 40 000, как описано выше;
- ваше правило принятия решения в каждом конкретном случае заключается в том, что вы будете использовать кнопку выполнения в 1 клик, если статистика из теста пропорций имеет *p*-значение ниже 0.1;
- во всех этих экспериментах истинный размер эффекта составляет 1 %.

Тогда вы будете находить значительный положительный результат и имплементировать кнопку выполнения в 1 клик в 90 % (т. е. 0.9) этих экспериментов; в оставшихся 10 % этих экспериментов вы получите нулевой результат и ошибочно откажетесь от имплементирования кнопки выполнения в 1 клик.

Существует несколько эквивалентных формул регрессии, но только для простейших случаев, и я нахожу, что даже в этих ситуациях их сложность значительно перевешивает их полезность. Тем не менее в качестве концептуального шага к нашему симуляционному подходу давайте проведем обзор того, как будет выглядеть традиционный статистический подход с точки зрения модели принятия решения с регрессией. Давайте выполним логистическую регрессию на неких имитационных данных:

```
## R (результат не показан)
exp_null_data <- hist_data %>%
    slice_sample(n=20000) %>%
    mutate(oneclick = ifelse(runif(20000)>0.5,1,0)) %>%
    mutate(oneclick = factor(oneclick, levels=c(0,1)))
    summary(glm(booked ~ oneclick + age + gender,
                         data = exp_null_data, family = binomial(link = "logit")))

## Python
exp_null_data_df = hist_data_df.copy().sample(2000)
exp_null_data_df['oneclick'] = np.where(np.random.uniform(0,1,2000)>0.5, 1, 0)
mod = smf.logit('booked ~ oneclick + age + gender', data = exp_null_data_df)
mod.fit(disp=0).summary()
...
                coef     std err    z      P>|z|    [0.025    0.975]
Intercept      9.5764    0.621   15.412    0.000    8.359   10.794
gender[T.male] 0.1589    0.136    1.167    0.243   -0.108    0.426
oneclick       0.0496    0.136    0.365    0.715   -0.217    0.316
age           -0.3017    0.017  -17.434    0.000   -0.336   -0.268
...
```

Традиционное правило принятия решения состояло бы в том, чтобы считать воздействие кнопки выполнения в 1 клик значимым и имплементиро-

вать ее, если соответствующий коэффициент (здесь приближенно 0.0475) имел *p*-значение меньше 0.1. Поскольку с этими имитационными данными он составляет примерно 0.28, мы бы сочли эффект незначимым и отказались от имплементирования кнопки (фактические цифры для вас будут случайным образом варьироваться в зависимости от вашей симуляции).

Определение размера выборки для нашего анализа, основываясь на этом подходе, повлечет за собой определение размера выборки такого, что в 90 % большого числа экспериментов, где истинный эффект составляет 1 %, мы будем получать *p*-значение для коэффициента регрессии менее 0.1. Но, как я описал в главе 7, это неявно принимает статистические допущения о нормальном распределении наших данных, что может быть проблематичным, и потому, как мы сейчас увидим, вместо этого мы будем использовать бутстраповские симуляции.

Использовать формулу размера выборки для теста пропорций по-прежнему бывает полезно в качестве быстрого и скорого первого шага, поскольку его результат должен иметь тот же порядок величины, что и окончательный размер выборки, который вам понадобится. Общий размер выборки, равный 40 000, в случае теста пропорций означает, что если только ваши другие предсказатели не обладают сумасшедшей высокой предсказательной способностью, порядок величины для вашего требуемого размера выборки составит 10 000, а не 1000 или 100 000 (т. е. ваш размер выборки будет состоять из пяти цифр). Мы начнем наши симуляции с предварительного размера выборки, равного 20 000, и, основываясь на том, какую эффективную мощность это нам дает, мы будем корректировать это число в сторону увеличения или уменьшения.

Анализ мощности без статистики: бутстраповские симуляции

Традиционный статистический анализ имел прекрасный смысл, когда данные были лимитированы, а расчеты выполнялись кропотливо вручную. Я твердо убежден, что сейчас он себя изжил: бутстраповские симуляции предлагают альтернативу, которая лучше отражает реалии и потребности прикладного анализа данных. Степень, с которой эксперимент может оказаться неправильным (например, говоря, что экспериментальная процедура на 1 % лучше, чем контрольная версия, когда на самом деле она на 10 % хуже), нередко вызывает у деловых партнеров бóльшую озабоченность, чем вероятность того, что разница равна нулю[1].

СОЕДИНЕНИЕ СИМУЛЯЦИЙ И СТАТИСТИЧЕСКОЙ ТЕОРИИ При использовании бутстраповских симуляций наше правило принятия решения не зависит от *p*-значений. Вместо этого мы имплементируем экспериментальную процедуру, если бутстраповский интервал уверенности для интересующего коэффициента превышает некоторый порог, обычно равный нулю. Если допущения статистического анализа мощности подтверждены, то бутстра-

[1] Работа Хаббарда (2010) является неплохим ресурсом, если вы хотите больше подумать о том, как конструировать полезные измерения в бизнесе, даже при очень лимитированной информации.

повские симуляции дают результаты, которые очень похожи и интуитивно связаны:

- в соответствии с резко нулевой гипотезой об отсутствии эффекта мы ожидаем, что 90%-й интервал уверенности будет включать ноль в 90 % случаев, 80%-й интервал уверенности будет включать ноль в 80 % случаев и т. д. Из этого свойства, именуемого охватом интервалов уверенности, вытекает, что процент, который мы используем для определения нашего интервала уверенности, эквивалентен статистической значимости, т. е. 90%-й интервал уверенности будет иметь примерно 5%-ную частоту ложноположительных результатов в каждом направлении. В 5 % случаев мы будем наблюдать интервал уверенности, который является строго отрицательным, и в 5 % случаев – интервал уверенности, который является строго положительным;
- учитывая альтернативную гипотезу, т. е. размер целевого эффекта, мы можем определить нашу мощность как процент симуляций, которые дают истинно положительный результат. Например, если мы установим эффект кнопки выполнения в 1 клик равным 1 %, просимулируем большое число экспериментов и заметим, что 75 % наших бутстраповских интервалов уверенности являются строго положительными, то наша мощность составит 75 %.

Как мы увидим в следующей главе, если допущения традиционного статистического анализа мощности не подтверждаются, то охват бутстраповского интервала уверенности может отличаться. То есть 90%-й интервал уверенности может включать ноль более или менее чем в 90 % случаев. Такой эффективный охват представляет реальный риск ложноположительных результатов, который необходимо установить на желаемый уровень значимости. Это просто заблаговременное предупреждение; мы рассмотрим указанный вопрос подробнее в главе 9.

Симуляции предлагают очень разнообразный, но прозрачный способ определения необходимого размера выборки для любого эксперимента, какими бы странными ни были данные или сложными ни были деловые решения. Эти преимущества происходят из того, что вы четко констатируете то, как будете анализировать свои данные и писать соответствующий исходный код до фактического выполнения эксперимента, что обеспечивает дополнительную проверку исправности и возможность вносить коррективы. Противоположностью этим преимуществам является то, что нам придется больше кодировать самим, вместо того чтобы опираться на готовые формулы. Я попытаюсь лимитировать сложность исходного кода, разбив его на интуитивно понятные функции.

НАПИСАНИЕ ИСХОДНОГО КОДА НАШЕГО АНАЛИЗА Давайте сначала создадим функцию, которая будет распечатывать интересующую нас метрику, а именно коэффициент для одного клика в нашей логистической регрессии:

```
## R
# Метрическая функция
log_reg_fun <- function(dat){
```

```
    # Выполнение логистической регрессии
    log_mod_exp <- glm(booked ~ oneclick + age + gender,
                       data = dat, family = binomial(link = "logit"))
    summ <- summary(log_mod_exp)
    metric <- summ$coefficients['oneclick1', 'Estimate']
    return(metric)}

## Python
def log_reg_fun(dat_df):
    model = smf.logit('booked ~ oneclick + age + gender', data = dat_df)
    res = model.fit(disp=0)
    coeff = res.params['oneclick']
    return coeff
```

Это всего лишь функциональная обертка для приведенного выше анализа, и применение данной функции к нашему набору имитационных данных вернет тот же коэффициент, приближенно равный 0.0475.

Затем давайте рассчитаем бутстраповские интервалы уверенности для этой метрики, повторно используя функцию из главы 7:

```
## R
boot_CI_fun <- function(dat, metric_fun){
    # Установка числа бутстраповских выборок
    B <- 100

    boot_metric_fun <- function(dat, J){
        boot_dat <- dat[J,]
        return(metric_fun(boot_dat))}
    boot.out <- boot(data=dat, statistic=boot_metric_fun, R=B)
    confint <- boot.ci(boot.out, conf = 0.90, type = c('perc'))
    CI <- confint$percent[c(4,5)]
    return(CI)}

## Python
def boot_CI_fun(dat_df, metric_fun, B = 100, conf_level = 0.9):
    # Установка числа выборок
    N = len(dat_df)
    conf_level = conf_level
    coeffs = []

    for i in range(B):
        sim_data_df = dat_df.sample(n=N, replace = True)
        coeff = metric_fun(sim_data_df)
        coeffs.append(coeff)

    coeffs.sort()
    start_idx = round(B * (1 - conf_level) / 2)
    end_idx = - round(B * (1 - conf_level) / 2)
    confint = [coeffs[start_idx], coeffs[end_idx]]
    return(confint)
```

Схожим образом мы примем в качестве нашего правила принятия решения, что мы будем имплементировать кнопку тогда и только тогда, когда

бутстраповский 90%-й интервал уверенности будет строго положительным (т. е. он не включает ноль):

```
## R
decision_fun <- function(dat){
    boot_CI <- boot_CI_fun(dat, metric_fun)
    decision <- ifelse(boot_CI[1]>0,1,0)
    return(decision)}

## Python
def decision_fun(dat_df, metric_fun, B = 100, conf_level = 0.9):
    boot_CI = boot_CI_fun(dat_df, metric_fun, B = B, conf_level = conf_level)
    decision = 1 if boot_CI[0] > 0 else 0
    return decision
```

Это эквивалентно правилу принятия решения об имплементировании кнопки тогда и только тогда, когда *p*-значение находится ниже порогового значения 0.10. Вы можете убедиться сами, что применение этой функции к нашему набору имитационных данных возвращает 0, как и должно быть.

Определение мощности нашего эксперимента для заданного размера эффекта и заданного размера выборки остается неизменным: это процент от большого числа таких экспериментов, для которых мы бы имплементировали эту кнопку. Давайте теперь перейдем к симулированию этого большого числа экспериментов!

СИМУЛЯЦИЯ МОЩНОСТИ Далее мы напишем нашу функцию для выполнения одной симуляции. Исходный код работает следующим образом (номера выносок относятся как к R, так и к Python):

```
## R
> single_sim_fun <- function(dat, metric_fun, Nexp, eff_size, B = 100,
                             conf.level = 0.9){

    # Добавить предсказанную вероятность бронирования ❶
    hist_mod <- glm(booked ~ age + gender + period,
                    family = binomial(link = "logit"), data = dat)
    sim_data <- dat %>%
      mutate(pred_prob_bkg = hist_mod$fitted.values) %>%
      # Отфильтровать до желаемого размера выборки ❷
      slice_sample(n = Nexp) %>%
      # Случайно разместить экспериментальные группы ❸
      mutate(oneclick = ifelse(runif(Nexp,0,1) <= 1/2, 0, 1)) %>%
      mutate(oneclick = factor(oneclick, levels=c(0,1))) %>%
      # Добавить эффект в процедурную группу ❹
      mutate(pred_prob_bkg = ifelse(oneclick == 1,
                                    pred_prob_bkg + eff_size,
                                    pred_prob_bkg)) %>%
      mutate(booked = ifelse(pred_prob_bkg >= runif(Nexp,0,1),1, 0))

    # Вычислить решение (мы хотим, чтобы оно было равно 1) ❺
    decision <- decision_fun(sim_data, metric_fun, B = B,
                             conf.level = conf.level)
```

```
    return(decision)}

## Python
def single_sim_fun(Nexp, dat_df, metric_fun, eff_size, B = 100,
                   conf_level = 0.9):

    # Добавить предсказанную вероятность бронирования ❶
    hist_model = smf.logit('booked ~ age + gender + period', data = dat_df)
    res = hist_model.fit(disp=0)
    sim_data_df = dat_df.copy()
    sim_data_df['pred_prob_bkg'] = res.predict()
    # Отфильтровать до желаемого размера выборки ❷
    sim_data_df = sim_data_df.sample(Nexp)
    # Случайно разместить экспериментальные группы ❸
    sim_data_df['oneclick'] = np.where(np.random.uniform(size=Nexp) <= 0.5, 0, 1)
    # Добавить эффект в процедурную группу ❹
    sim_data_df['pred_prob_bkg'] = np.where(sim_data_df.oneclick == 1,
                                            sim_data_df.pred_prob_bkg + eff_size,
                                            sim_data_df.pred_prob_bkg)
    sim_data_df['booked'] = np.where(sim_data_df.pred_prob_bkg >= \
                                     np.random.uniform(size=Nexp), 1, 0)

    # Вычислить решение (мы хотим, чтобы оно было равно 1) ❺
    decision = decision_fun(sim_data_df, metric_fun = metric_fun, B = B,
                            conf_level = conf_level)
    return decision
```

❶ Добавить в данные предсказанную вероятность бронирования.
❷ Отфильтровать до нужного размера выборки.
❸ Разместить экспериментальные группы.
❹ Добавить эффект в процедурную группу.
❺ Применить функцию принятия решения и вернуть ее результат.

Затем мы можем написать нашу функцию мощности для определенного размера эффекта и размера выборки. Эта функция многократно генерирует наборы экспериментальных данных, а затем применяет к ним нашу функцию принятия решения; она возвращает ту их часть, для которой мы бы имплементировали кнопку:

```
## R
power_sim_fun <- function(dat, metric_fun, Nexp, eff_size, Nsim,
                          B = 100, conf.level = 0.9){
    power_list <- vector(mode = "list", length = Nsim)
    for(i in 1:Nsim){
        power_list[[i]] <- single_sim_fun(dat, metric_fun, Nexp, eff_size,
                                          B = B, conf.level = conf.level)}
    power <- mean(unlist(power_list))
    return(power)}

## Python
def power_sim_fun(dat_df, metric_fun, Nexp, eff_size, Nsim, B = 100,
                  conf_level = 0.9):
    power_lst = []
```

```
    for i in range(Nsim):
        print("номер стартовой симуляции", i, "\n")
        power_lst.append(single_sim_fun(Nexp = Nexp, dat_df = dat_df,
                                        metric_fun = metric_fun,
                                        eff_size = eff_size, B = B,
                                        conf_level = conf_level))
    power = np.mean(power_lst)
    return(power)
```

Сколько наборов данных вы должны просимулировать? Неплохой отправной точкой будет число двадцать; это даст вам шумную оценку, но если вы получите мощность 0 или 1, то будете знать, что вам нужно скорректировать размер выборки:

```
## Python (результат не показан)
power_sim_fun(dat_df=hist_data_df, metric_fun = log_reg_fun, Nexp = int(4e4),
              eff_size=0.01, Nsim=20)

## R
> set.seed(1234)
> power_sim_fun(dat=hist_data, effect_size=0.01, Nexp=4e4, Nsim=20)
[1] 0.9
```

Эта первая оценка составляет 90%-ю мощность; как я уже сказал, традиционные формулы дают вам разумную основу для начала симулирований. Затем я выполняю функцию симуляции мощности с 30 000 и 50 000 строк для 100 симуляций каждая и, наконец, 35 000 и 45 000 строк для 200 симуляций каждая. В сущности, по мере того как вы получаете все более и более узкий интервал размеров выборки, вы захотите повысить точность путем увеличения числа симуляций. На рис. 8.6 показаны результаты моих поочередных итераций.

Как было объявлено ранее, мы получаем мощность 0.9 при примерно 40 000. Мы могли бы продолжить симуляции, если бы нам нужно было получить более прецизионную информацию (например, следует ли нам получать размер выборки, равный 38 000? 41 000?), но для данного примера этого достаточно.

Теперь, когда мы определили наш размер выборки, последнее, что мне нравится делать, – это строить график кривой мощности для нескольких размеров эффектов при этом размере выборки. Это дает нам более четкое представление о том, насколько велика вероятность того, что в целом мы получим положительный результат, если допустить, что фактический размер эффекта является положительным. Это также позволяет вам лучше донести до ваших деловых партнеров, что сила вашего эксперимента определяется не только одним размером эффекта. Здесь мы видим, как мощность эксперимента увеличивается с эффекта 0.5 % до эффекта 2 % (рис. 8.7).

При 200 симуляциях в расчете на размер эффекта расчетные значения мощности должны быть довольно точными, хотя все еще не идеальными, как показано отсутствием плавности кривой. Другими словами, тот факт, что мы видим мощность 1 для размера эффекта 2 %, не означает, что мы имеем буквально 100 % мощности для этого размера эффекта, но очень близко к нему.

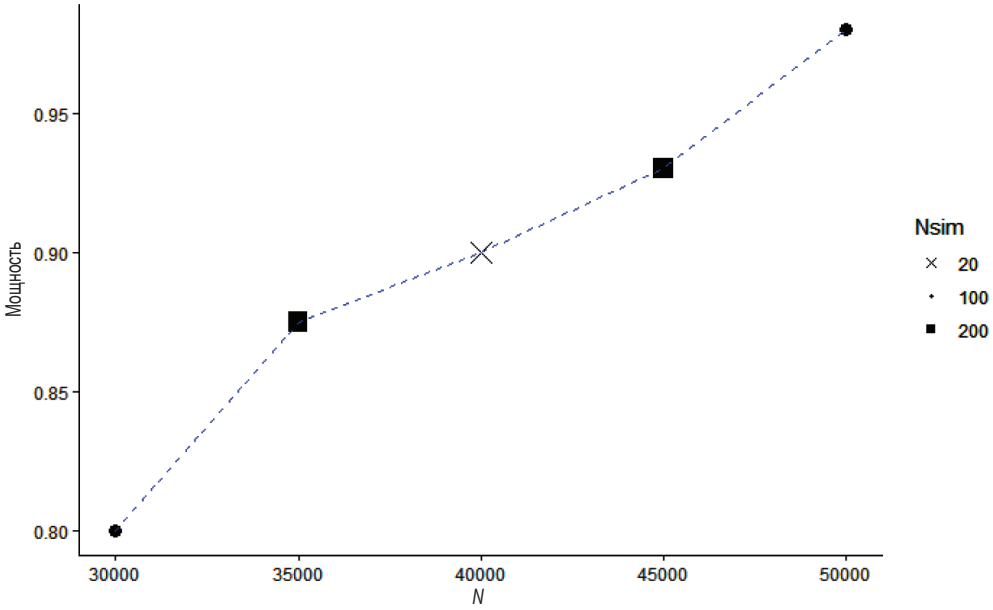

Рис. 8.6 ❖ Симуляции мощности для различных размеров выборки

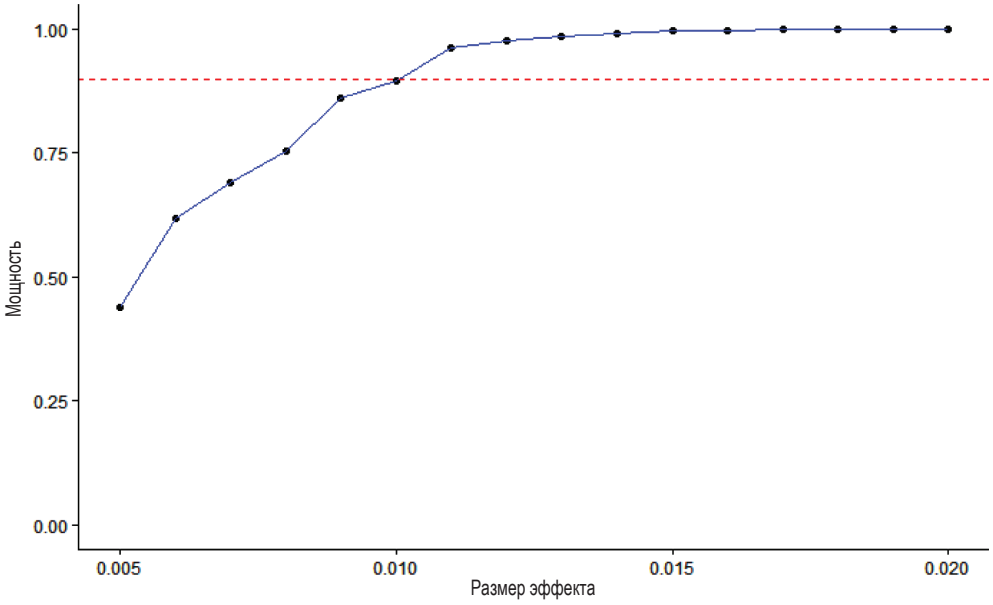

Рис. 8.7 ❖ Симуляции мощности для различных размеров эффекта при *N* = 40 000, с 200 симуляциями в расчете на размер эффекта, пунктирная линия при мощности = 0.9

Напоминание: симулированная статистическая значимость бутстраповского интервала уверенности должна быть довольно близка к нормальному интервалу уверенности, если ваши переменные распределены относительно плавно и нормально. Для

более странных данных (несколько пиков, толстые хвосты и т. д.) это, возможно, больше не будет соблюдаться, и вам определенно следует проверить, что ваша симулированная статистическая значимость не уходит широко в сторону.

Если вы хотите, вы также можете выполнить симуляцию с размером эффекта, равным нулю. Она даст вам эмпирическую статистическую значимость вашего анализа. Поскольку мы используем бутстраповские 90%-ные интервалы уверенности, около 5 % этих симуляций в итоге должны приводить к решению (ошибочно) имплементировать кнопку выполнения в 1 клик, и это то, что мы наблюдаем здесь.

Это ни в коем случае не симуляции больших данных, но они занимают достаточно много времени (самая продолжительная из них заняла около получаса на моем ноутбуке), чтобы вы захотели внести усовершенствования в производительность своих функций, оставить свой исходный код работать, пока вы делаете что-то еще, или и то, и другое. Исходный код в репозитории на GitHub[1] содержит функции, которые я оптимизировал с помощью пакетов `Rfast` и `doParallel` в R, а также пакетов `joblib` и `psutil` в Python.

Анализирование и интерпретирование экспериментальных результатов

Проведя эксперимент и собрав соответствующие данные, вы сможете их проанализировать. После всех просимулированных аналитических расчетов, которые вы провели для оценивания мощности, сами окончательные аналитические расчеты должны быть прогулкой в парке. Мы выполняем логистическую регрессию и определяем соответствующий бутстраповский 90%-ный интервал уверенности. Из-за случайного размещения мы знаем, что наш коэффициент для кнопки выполнения в 1 клик не спутан – нам не нужно контролировать наличие какого-либо спутывающего фактора. Однако, добавив другие переменные, которые также являются причинами вероятности бронирования, мы можем сократить шум и значительно улучшить прецизионность нашего оценивания:

```
## Python code (результат не показан)
import statsmodels.formula.api as smf
model = smf.logit('booked ~ age + gender + oneclick', data = exp_data_df)
res = model.fit()
res.summary()

## R
> log_mod_exp <- glm(booked ~ oneclick + age + gender,
                     data = exp_data, family = binomial(link = "logit"))
> summary(log_mod_exp)

...
Coefficients:
```

[1] См. https://oreil.ly/BehavioralDataAnalysis.

```
             Estimate Std. Error z value    Pr(>|z|)
(Intercept)  11.94701    0.22601  52.861     < 2e-16 ***
oneclick1     0.15784    0.04702   3.357    0.000789 ***
age          -0.39406    0.00643 -61.282     < 2e-16 ***
genderfemale -0.25420    0.04905  -5.182 0.000000219 ***
...
```

Коэффициент для кнопки выполнения в 1 клик равен 0.15784, а бутстраповский 90%-ный интервал уверенности для нее составляет приближенно [0.073; 0.250]. Основываясь на нашем правиле принятия решения, мы бы пошли вперед и имплементировали кнопку выполнения в 1 клик.

Коэффициенты из логистической регрессии не поддаются прямой интерпретации, и я нахожу, что рекомендуемое решение с использованием соотношения шансов помогает лишь незначительно (в частности, когда у вас есть модерационные эффекты). Мое предпочтительное эмпирическое правило состоит в генерировании двух копий экспериментальных данных, в одной из которых переменная для кнопки выполнения в 1 клик для всех установлена равной 1, а в другой – равной 0. Сравнивая вероятность бронирования, предсказываемую нашей логистической моделью для этих двух наборов данных, мы можем рассчитать «средний» эффект, очень близкий к эффекту, который вы наблюдали бы при имплементировании эксперимантальной процедуры для всех. Это ненаучно, но полезно:

```
## R (результат не показан)
> diff_prob_fun <- function(dat, reg_model = log_mod_exp){
    no_button <- dat %>% ❶
        mutate(oneclick = 0) %>%
        mutate(oneclick = factor(oneclick, levels=c(0, 1))) %>%
        select(age, gender, oneclick)
    button <- dat %>% ❷
        mutate(oneclick = 1) %>%
        mutate(oneclick = factor(oneclick, levels=c(0, 1))) %>%
         select(age, gender, oneclick)
    # Добавление предсказаний в модель
    no_button <- no_button %>%
        mutate(pred_mod = predict(object=reg_model, newdata = no_button,
                                  type="response"))
    button <- button %>% ❸
        mutate(pred_mod = predict(object=reg_model, newdata = button,
                                  type="response"))
    # Вычисление средней разницы в вероятностях
    diff <- button$pred_mod - no_button$pred_mod ❹
    return(mean(diff))}
> diff_prob_fun(exp_data, reg_model = log_mod_exp)

## Python
def diff_prob_fun(dat_df, reg_model = log_mod_exp):

    # Создание новых копий данных
    no_button_df = dat_df.loc[:, 'age':'gender'] ❶
    no_button_df.loc[:, 'oneclick'] = 0
    button_df = dat_df.loc[:,'age':'gender'] ❷
```

```
    button_df.loc[:, 'oneclick'] = 1

    # Добавление предсказаний в модель
    no_button_df.loc[:, 'pred_bkg_rate'] = res.predict(no_button_df) ❸
    button_df.loc[:, 'pred_bkg_rate'] = res.predict(button_df)

    diff = button_df.loc[:,'pred_bkg_rate'] \  ❹
        - no_button_df.loc[:,'pred_bkg_rate']
    return diff.mean()
diff_prob_fun(exp_data_df, reg_model = log_mod_exp)
0.007129714313551981
```

❶ **Мы создаем набор данных под названием no_button, для которого устанавливаем переменную oneclick равной нулю для всех строк (и конвертируем ее в коэффициент, чтобы предсказательная функция смогла позже работать).**

❷ **Мы создаем набор данных под названием button, для которого устанавливаем переменную oneclick равной единице для всех строк.**

❸ **Мы рассчитываем предсказанную вероятность бронирования в каждом конкретном случае, используя функцию predict() с нашей моделью log_mod_exp.**

❹ **Мы вычисляем разницу между предсказанными вероятностями.**

Мы видим, что наш средний эффект в экспериментальной популяции составляет около 0.712 п. п., положительный, но ниже нашей цели в 1 п. п. Как обычно, давайте построим бутстраповский 90%-ный интервал уверенности, который приближенно составляет [0.705 п. п.; 0.721 п. п.]. Этот интервал очень узок и не пересекает ноль. Поэтому мы можем трактовать наш результат как эмпирически статистически значимый на уровне 5 %. В этом случае мы можем даже получить гораздо большую уверенность: 99.8%-ный интервал уверенности приближенно составляет [0.697 п. п.; 0.728 п. п.], все еще далеко от нуля, поэтому мы можем трактовать наш результат как значительный на уровне (1 – 0.998)/2 = 0.1 %.

В целях охвата всех случаев давайте подытожим наше правило принятия решений (табл. 8.4). Благодаря этому вы увидите, что делать в зависимости от статистической значимости или незначимости наблюдаемого оценочного эффекта и экономической значимости (здесь подразумевается увеличение на 1 п. п.) или незначимости. В данном случае я бы имплементировал кнопку, притом что она имеет строго положительный эффект, а стоимость имплементации невелика.

Последним пунктом следует отметить, что средний эффект по всей нашей экспериментальной популяции, 0.712 п. п., довольно далек от прямой разницы между нашей контрольной группой и процедурной группой, которая составляет около 0.337 п. п. Это связано со случайными разницами между нашими двумя экспериментальными группами. Средний возраст в нашей контрольной группе составляет 40.63 года против 40.78 в процедурной группе. Доля мужчин также немного выше в процедурной группе. При очень малых размерах эффекта этих незначительных разниц достаточно, чтобы запутать прямое сравнение двух групп: наш размер выборки достаточно велик, чтобы наши две группы были идентичны примерно на 0.3 п. п., что довольно близко в абсолютном выражении, но это примерно половина нашего экспериментального эффекта.

Таблица 8.4. Правило принятия решения для кнопки бронирования в 1 клик

		Наблюдаемый оценочный эффект		
		оценочный эффект <= 0	**0 < оценочный эффект < 1 п. п.**	**1 п. п. <= оценочный эффект**
Эмпирическая статистическая значимость наблюдаемых результатов	**«Высокая» (бутстраповский интервал уверенности для 90 % или выше не пересекает 0)**	Не имплементировать кнопку	Имплементировать кнопку или нет, все зависит от размера оценочного эффекта, затрат и аппетита к риску (здесь это наш случай)	Имплементировать кнопку
	«Низкая» (бутстраповский интервал уверенности для 90 % пересекает 0)	Не имплементировать кнопку	Не имплементировать кнопку	Имплементировать кнопку либо выполнить тест, в зависимости от интервала уверенности и аппетита к риску

К сожалению, в этом эксперименте, где клиенты размещаются случайно в разбивке по двум группам по мере их поступления, мы ничего не сможем с этим поделать. Но если мы знаем всю нашу экспериментальную выборку в начале эксперимента, то можем добиться значительно большего успеха, обеспечив, чтобы контрольная группа и процедурная группа были как можно более идентичными, задействовав для этого стратифицированную рандомизацию, как мы увидим в следующей далее главе.

Выводы

В этой главе мы увидели, как конструировать простейшую форму эксперимента – онлайновый A/B-тест с простой рандомизацией. Я подчеркнул, что хорошо продуманный эксперимент – это гораздо больше, чем просто раскидывание клиентам случайных разных версий веб-сайта или электронной почты. Вам нужно определить свою деловую цель и целевую метрику, а затем четко сформулировать принцип, по которому ваше вмешательство с ними связано, с помощью поведенческой логики. Взятые вместе, ваша деловая цель, целевая метрика, вмешательство и поведенческая логика составляют теорию изменения в вашем эксперименте.

Затем мы обратились к количественным аспектам экспериментального дизайна. В этой первой главе, посвященной экспериментам, случайное размещение было чрезвычайно простым, и я потратил больше времени на анализ мощности и вычисления размера выборки. Хотя и существуют статистические формулы, в качестве инструментов анализа я предпочитаю использовать регрессии, а не статистические тесты, а в качестве меры неопределенности вокруг нашего расчетного коэффициента использовать бутстраповские интервалы уверенности, а не *p*-значения, что приводит к использованию симуляций мощности вместо формул. В этом случае результаты двух из них почти идентичны, но в следующих двух главах мы перейдем к более сложным видам дизайна, в которых формулы отсутствуют.

Глава 9

Стратифицированная рандомизация

В предыдущей главе мы увидели простейшую форму рандомизации: появляется клиент, и мы бросаем метафорическую монету или кости. Орел – и он видит версию A, решка – и он видит версию B. Вероятности могут отличаться от разбивки 50/50, но они постоянны и не зависят от характеристик клиента. Нет ничего вроде «моя контрольная группа немного старше моей процедурной группы, давайте убедимся, чтобы все, кто появятся в следующем тысячелетии, вошли в контрольную группу». Как следствие ваши контрольная и процедурная группы «вероятностно эквивалентны», что является статистическим способом сказать, что если бы вы продолжили проводить свой эксперимент вечно, то ваши две группы имели бы точно такие же доли, как и ваша популяция в целом. Однако на практике ваши экспериментальные группы могут в итоге сильно отличаться друг от друга. Добавление объясняющих переменных в ваш окончательный анализ может несколько компенсировать эти дисбалансы, но, как мы сейчас увидим, мы можем добиться большего, если заранее узнаем, кто будет участвовать в нашем эксперименте.

В этой главе я познакомлю вас со стратифицированной рандомизацией, которая позволит нам обеспечивать максимально возможное сходство наших экспериментальных групп. Это резко увеличивает объяснительную силу эксперимента, что особенно полезно, когда у вас нет больших размеров выборки.

Стратифицированная рандомизация может применяться к любой ситуации, когда у нас есть заранее составленный список клиентов/сотрудников и т. д. для строительства наших экспериментальных групп. Учитывая, что A/B-тесты чаще всего обсуждаются в связи с незначительными изменениями в электронной почте или на веб-сайте, я бы мог взять пример рекламной кампании по электронной почте. Но я хотел продемонстрировать, что более крупные деловые инициативы, которые часто предпринимаются руководителями компаний на основе их «стратегического чутья», тоже могут быть проверены и подтверждены.

Деловой контекст здесь заключается в том, что AirCnC по умолчанию предоставляет собственникам не менее 24 часов на уборку своего объекта недвижимости между двумя бронированиями. На рынках с высоким спросом,

где объект недвижимости бронируется сразу же, как только он становится доступным, это представляет собой существенный лимитирующий фактор. Руководители бизнеса стремятся его сокращать и увеличить ежемесячную прибыль в расчете на объект недвижимости. Как это часто бывает, в компании существуют две школы мышления:

- финансовый отдел выступает за предоставление собственникам возможности устанавливать минимальную двухсуточную продолжительность в расчете на бронирование;
- отдел обслуживания клиентов считает, что минимальная продолжительность негативно скажется на удовлетворенности клиентов; вместо этого он выступает за бесплатное предоставление собственникам услуг профессиональной клининговой компании в обмен на сокращение времени уборки с 24 до 8 часов.

Подобные ситуации возникают в бизнесе часто. У обеих сторон есть несколько убедительных аргументов, которые говорят о разных аспектах проблемы или подчеркивают разные методы (здесь прибыль от брони в сравнении с клиентским опытом) и/или предоставляют разрозненные подтверждения из жизни в пользу их позиции («эта компания делает X, поэтому мы тоже должны это делать»). Распространенным исходом является «победа» любого, чья аргументация имеет наибольший вес, и его решение имплементируется, т. н. правило организационной политики.

В этом месте вы, вероятно, ожидаете, что я скажу что-то вроде «но опыт позволяет вам обходить всю политику окольными путями и находить наилучшее решение без какой-либо суеты». Хотел бы я, чтобы это было так просто! Правда же заключается в том, что эксперименты в таких ситуациях могут оказывать огромную помощь, но это не панацея по двум причинам.

Первая причина заключается в том, что если решение не окажется превосходящим другое по всем направлениям, то возникнет потребность в отыскании нескольких реальных компромиссов между соперничающими целями: на какое снижение удовлетворенности клиентов компания готова пойти ради увеличения прибыли? Этот вопрос по своей сути является политическим, потому что у разных интересантов в компании разные предпочтения по этому вопросу. Вам придется как можно четче излагать эти компромиссы своим руководителям и улаживать их как можно раньше, если вы хотите, чтобы ваш эксперимент был успешным.

Вторая причина заключается в том, что ваш эксперимент сделает счастливой не более одной стороны (он также может сделать несчастными обе стороны, если контрольная группа будет иметь наилучшие результаты!). Недовольные руководители бизнеса, как и любой другой несчастный человек, умеют очень хорошо находить объяснения постфактум: «Район залива Сан-Франциско отличается от других востребованных рынков», «Измеренная в ходе опроса удовлетворенность клиентов снизилась, но чистый балл промоутера вырос, а этот показатель является более качественной мерой “истинной” удовлетворенности клиентов» и т. д.

Эти две причины делают особенно важным правильное планирование и проведение эксперимента не только с точки зрения экспериментального дизайна, но и с точки зрения бизнеса.

Планирование эксперимента

Как мы видели в предыдущей главе, успешное планирование эксперимента требует от нас четкого изложения его теории изменения:

- каковы деловая цель и целевая метрика?
- каково определение нашего вмешательства?
- как они связаны нашей поведенческой логикой?

Будем надеяться, что вы уже хорошо познакомились с этим процессом, поэтому я не буду подробно останавливаться на том, как это делать, а просто пройдусь по нему пошагово, чтобы дать вам более глубокое понимание эксперимента. В частности, я воспользуюсь возможностью, чтобы изложить некоторые особенности эксперимента с поведенческой точки зрения.

Деловая цель и целевая метрика

Деловая цель этого эксперимента, или, что то же самое, деловая задача, которую мы пытаемся решить, состоит в том, чтобы повысить прибыльность за счет сокращения времени простоя на востребованных рынках. Поскольку стоимость предоставления бесплатных услуг по уборке для собственников значительна (финансисты оценивают их в размере 10 долларов в день), нам нужно будет включить ее в наш анализ.

Мы сделаем это, соответствующим образом модифицировав нашу целевую метрику. Нашей базовой метрикой является средняя прибыль от брони в день; вместо этого мы будем использовать среднюю прибыль от брони в день за вычетом дополнительных затрат. Это просто означает, что нам придется вычесть 10 долларов из базовой метрики для экспериментальной процедуры бесплатной уборки.

Однако существуют также некоторые опасения, что вмешательство в форме минимальной продолжительности негативно скажется на удовлетворенности клиентов. Как это учесть?

Решение, которое, как я видел, отстаивают другие авторы, заключается в использовании средневзвешенного значения метрик (иногда именуемого критерием совокупного оценивания [OEC]). В нашем настоящем примере это означало бы присвоение весов, например, по 50 % каждой, нашим двум переменным, а затем использование этой новой метрики в качестве нашей цели. Вы, безусловно, должны, не стесняясь, свободно использовать этот подход, если вы или ваши деловые партнеры хотите, но я бы рекомендовал вместо этого выбрать уникальную целевую метрику и, при необходимости, включать несколько других «ограждающих» метрик в список особого внимания по нескольким причинам.

Первая из них заключается в том, что компромиссы между деловыми целями в конечном счете являются стратегическими и политическими решениями. Нет объективного наилучшего ответа на вопрос о том, во сколько пунктов удовлетворенности клиента (CSAT) оценивается увеличение прибыли; это зависит от организации, ее контекста и ее текущих приоритетов. Использо-

вание средневзвешенного значения, определенного в один момент времени, придает процессу видимость технической объективности, но на самом деле это всего лишь окаменелая субъективность.

Вторая, критерий совокупного оценивания, делает эти компромиссы линейными. Если один пункт удовлетворенности покупателей эквивалентен 10 миллионам долларов прибыли с указанным выше критерием, то пять пунктов удовлетворенности эквивалентны 50 миллионам долларов прибыли. Но снижение на один пункт удовлетворенности может означать меньше восхищенных покупателей, тогда как снижение на пять пунктов может означать бурю в социальных сетях. Доллар всегда имеет ценность еще одного доллара, но практически для всего остального серия малых изменений обычно предпочтительнее одного большого. Слепое полагание на критерий совокупного оценивания может приводить к более рискованным решениям. Сторонники подхода на базе указанного критерия в этом месте могут возразить, что, очевидно, вам не следует полагаться на него слепо; но если вы все равно собираетесь обращаться к разнообразным компонентам и проводить открытое обсуждение, то я не совсем уверен, в чем этот критерий помогает.

Более того, критерий совокупного оценивания принимает деловые вмешательства как фиксированные. Придерживаясь эквивалента удовлетворенности покупателей 1 пункт = 10 миллионов долларов, следующие ниже два варианта будут иметь эквивалентный рейтинг, равный нулю.

1. Первое вмешательство увеличивает прибыль на 1 миллион долларов и снижает удовлетворенность на 0.1 пункта.
2. Второе вмешательство увеличивает прибыль на 50 миллионов долларов и снижает удовлетворенность на 5 пунктов.

Но с точки зрения поведения есть большая разница. Первый вариант – это в основном мертвая лошадь, и мало надежды вдохнуть в нее жизнь, в то время как второй вариант больше похож на привередливую чистокровку. Нацеливая его на конкретный сегмент покупателей, изменяя его условия или способ его представления, можно было бы получать по меньшей мере некоторые из его выгод без затрат. В этом смысле вмешательство, имеющее как большие положительные, так и отрицательные эффекты, требует итераций разведывательного анализа и дизайна, а не двоичных решений, поощряемых критерием совокупного оценивания.

Наконец, я считаю, что в некоторых случаях критерий совокупного оценивания используется в качестве укороченного пути. Предположим, что вмешательство увеличивает краткосрочную прибыль, но и увеличивает вероятность неуспеха. Это не подлинный стратегический компромисс: мы должны измерить влияние вероятности неуспеха на пожизненную ценность клиента, а затем определить чистый эффект на прибыльность. Высказываться о том, что ваш критерий совокупного оценивания будет на 90 % краткосрочной прибылью и на 10 % влиянием на уровень неуспеха, – это способ выдвигать догадку об обменном курсе, вместо того чтобы измерять истинный обменный курс. С помощью причинно-поведенческого каркаса этой книги мы можем добиваться большего, чем просто строить догадки.

Поэтому до конца этой главы мы будем использовать среднюю прибыль от брони в день в качестве нашей единственной целевой метрики и будем исходить из допущения, что удовлетворенность клиентов отслеживается в фоновом режиме на предмет любых тревожных изменений.

Все это хорошо и ладно, но когда мы говорим о слежении за удовлетворенностью клиентов, о каких клиентах мы говорим? Если клиенты хотят забронировать место всего на одни сутки и им предлагают место с минимумом продолжительности, они могут решить забронировать другое (в контрольной группе либо в группе бесплатной уборки) либо полностью отказаться от бронирования через AirCnC и вместо этого забронировать гостиницу. Поэтому мы не можем измерять клиентский опыт просто для четко определенной группы клиентов.

К сожалению, эта проблема не является редкостью: всякий раз, когда вы проводите эксперимент, в котором экспериментальная единица для случайного размещения не является клиентом, вы должны себя спрашивать, а как это отразится на стороне клиента. Мы можем лишь задействовать тот факт, что минимальная продолжительность будет влиять только на тех клиентов, которые ищут бронирование на одни сутки; мы также знаем, что клиенты вводят желаемую продолжительность до того, как им покажут доступные объекты. Следовательно, в нашем эксперименте мы могли бы отслеживать всех клиентов, у которых желаемая продолжительность составляет одни сутки, и проверять, бронируют ли они несколько суток в одной и той же гостинице или одни сутки в другой гостинице, или вообще не бронируют. Всякий раз, когда они выполняют бронирование, мы также отслеживаем их оценку своего пребывания. Это, конечно, далеко не идеально, потому что мы будем отслеживать разные метрики для разных подгрупп клиентов, но это лучшее, что можно сделать, и это еще один пример того, почему я гораздо больше верю в мониторинговые заградительные переменные, чем в их агрегирование в критерии совокупного оценивания.

Определение вмешательства

После определения критериев успеха нам нужно убедиться, что у нас есть ясность в том, что мы тестируем. Это особенно важно, когда ставки в организации высоки, например руководители бизнеса ломают копья на этом вопросе. Здесь важно отметить, что мы предлагаем собственникам возможность устанавливать минимальную продолжительность, что не то же самое, когда минимальная продолжительность делается обязательной. Аналогичным образом собственники могут воспользоваться или не воспользоваться предложением о бесплатной уборке. В дополнение к этому сами экспериментальные процедуры по-прежнему могут быть открыты для интерпретации. Насколько тщательной и дорогостоящей для компании является бесплатная уборка? Какова минимальная продолжительность, которую мы навязываем клиентам?

Поскольку оба наших вмешательства несколько сложны и опираются на понимание собственниками предложения и решение его принять, вероятно, будет неплохой идеей создать несколько разных вариантов дизайна и протестировать их качественно с помощью исследования пользовательского опыта. В дополнение к этому после проведения эксперимента было бы неплохо рассмотреть слегка иные вариации экспериментальной процедуры, какую бы мы ни решили имплементировать.

В конечном счете вы захотите убедиться, что все интересанты удовлетворены вариантами дизайна эксперимента и готовы его завизировать. Это сократит (но не устранит!) риск того, что когда появятся результаты, они будут спорить о том, что протестированное отражает предлагаемое ими решение неадекватно.

Поведенческая логика

Поведенческая логика отличается для двух экспериментальных процедур: подход на основе минимальной продолжительности будет увеличивать продолжительность и сумму в расчете на бронирование, но, возможно, уменьшать совокупное число бронирований; с другой стороны, подход на основе бесплатной уборки будет увеличивать число бронирований, но снижать прибыль в расчете на бронирование из-за дополнительных затрат. В дополнение к этому мы должны учитывать, что наши экспериментальные процедуры/вмешательства являются *предложениями*, которые собственники, возможно, примут либо не примут (рис. 9.1).

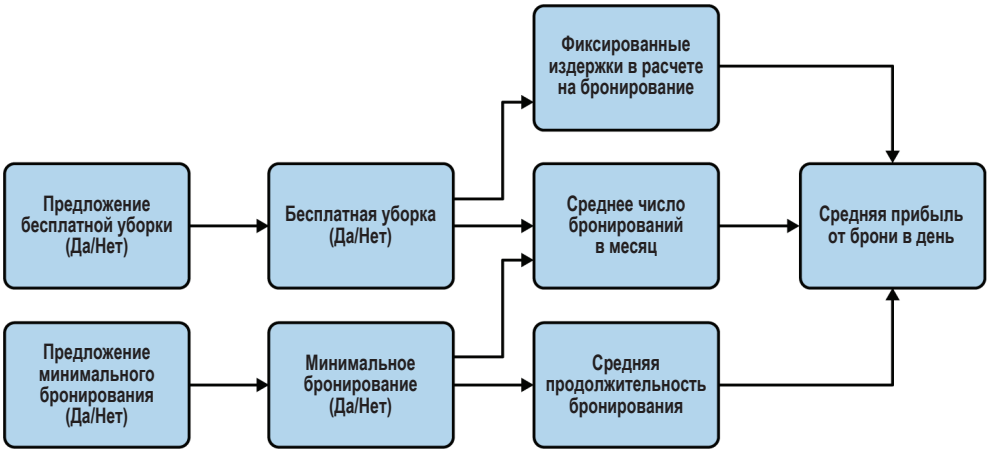

Рис. 9.1 ❖ Причинно-следственная диаграмма двух рассматриваемых экспериментальных процедур

Данные и пакеты

Папка этой главы в репозитории на GitHub[1] содержит два CSV-файла с переменными, перечисленными в табл. 9.1. Флажок (✓) обозначает переменные, присутствующие в этом файле, тогда как крестик (×) обозначает переменные, которые не присутствуют.

[1] См. https://oreil.ly/BehavioralDataAnalysisCh9.

Таблица 9.1. Переменные в наших данных

	Описание переменной	chap9-historical_data.csv	chap9-experimental_data.csv
ID (Идентификатор)	ИД недвижимости/собственника, 1–500	✓	✓
sq_ft (квадратные футы)	Площадь объекта недвижимости в кв. футах, 460–1120	✓	✓
tier (ярус)	Занимаемая недвижимостью площадь в кв. футах, от 1 до 3 в порядке уменьшения ярусов	✓	✓
avg_review (средний отзыв)	Средний отзыв о недвижимости, 0–10	✓	✓
BPday (Прибыль от брони в день)	Прибыль от брони в день, целевая переменная, 0–126	✓	✓
Period (Период)	Индекс месяца, 1–35 в исторических данных, 36 в явной форме в экспериментальных данных	✓	
Month (Месяц)	Месяц года, 1–12	✓	✓
group (группа)	Экспериментальное размещение, «ctrl» (контрольная группа), «treat1» (бесплатная уборка), «treat2» (минимальная продолжительность бронирования)	×	✓
compliant (соблюдение требований)	Двоичная переменная, обозначающая, что собственник подвергался экспериментальной процедуре в соответствии или не в соответствии с требованиями для группы, в которой он размещен	×	✓

В этой главе мы будем использовать следующие ниже пакеты в дополнение к обычным:

```
## R
library(blockTools) # Для функции block()
library(caret)      # Для функции кодирования с одним активным состоянием dummyVars()
library(scales)     # Для функции rescale()

## Python
import random       # Для функций sample() и shuffle()
# Для перешкалирования числовых переменных
from sklearn.preprocessing import MinMaxScaler
# Для кодирования с одним активным состоянием категориальных переменных
from sklearn.preprocessing import OneHotEncoder
```

Определение случайного размещения и размера/мощности выборки

В этом эксперименте мы будем размещать в экспериментальных группах всех сразу, основываясь на списке наших собственников на данный момент

времени. Это дает нам возможность значительно улучшить чисто случайное размещение, обеспечивая с самого начала, чтобы наши две группы были хорошо сбалансированы с помощью метода, именуемого стратификацией. Это позволит нам получать больше статистической мощности из любого имеющегося размера выборки.

Поэтому я сначала объясню метод случайного размещения, чтобы иметь возможность его использовать в симуляциях для нашего анализа мощности. Наконец, мы сравним результаты этих симуляций с традиционным статистическим анализом мощности.

Случайное размещение

Прежде чем перейти к стратификации, давайте посмотрим, как будет выглядеть стандартная рандомизация.

Уровень случайного размещения

Первым соображением в отношении случайного размещения является уровень, на котором мы будем его имплементировать, а затем измерять результаты эксперимента. В предыдущей главе я обсуждал вопрос о том, на каком уровне должно происходить случайное размещение: на уровне клиента или на уровне бронирования. В данном случае логистика экспериментальных процедур, в особенности бесплатной уборки, исключает имплементацию на уровне бронирования. Следовательно, мы будем выполнять случайное размещение на уровне собственника объекта недвижимости, по данным AirCnC на текущий момент таких объектов имеется в количестве 5000.

Стандартная рандомизация

Этот процесс аналогичен тому, который мы использовали в предыдущем эксперименте, но проще, потому что его можно выполнять в офлайновом режиме, а не в реальном времени: сначала мы назначаем каждому индивидууму в нашей экспериментальной популяции случайное число от 0 до 1. Затем на основе этого случайного числа мы назначаем группы: если K – это желаемое нами число групп (включая контрольную группу), то все индивидуумы со случайным числом менее $1/K$ входят в первую группу, все индивидуумы со случайным числом от $1/K$ до $2/K$ входят во вторую группу и т. д. Следующий ниже исходный код иллюстрирует указанный подход с тремя группами и размером выборки (для целей иллюстрации), равным 5000:

```
## R
no_strat_assgnt_fun <- function(dat, Nexp){
    K <- 3
    dat <- dat %>%
        distinct(ID) %>%
        slice_sample(n=Nexp) %>%
        mutate(assgnt = runif(Nexp,0,1)) %>%
        mutate(group = case_when(
            assgnt < = 1/K ~ "ctrl",
```

```
            assgnt > 1/K & assgnt < = 2/K ~ "treat1",
            assgnt > 2/K ~ "treat2")) %>%
        mutate(group = as.factor(group)) %>%
        select(-assgnt)
    return(dat)
}
no_strat_assgnt <- no_strat_assgnt_fun(hist_data, Nexp = 5000)

## Python
def no_strat_assgnt_fun(dat_df, Nexp, K):
    dat_df = pd.DataFrame({'ID': dat_df.ID.unique()})
    dat_df = dat_df.sample(Nexp)
    dat_df['assgnt'] = np.random.uniform(0,1,Nexp)
    dat_df['group'] = 'ctrl'
    dat_df.loc[dat_df['assgnt'].between(0, 1/K, inclusive=True),
               'group'] = 'treat1'
    dat_df.loc[dat_df['assgnt'].between(1/K, 2/K, inclusive=False),
               'group'] = 'treat2'
    del(dat_df['assgnt'])
    return dat_df
no_strat_assgnt = no_strat_assgnt_fun(hist_data_df, Nexp = 5000, K = 3)
```

Одним из приятных аспектов этого подхода является то, что его можно легко обобщать на сколь угодно большое число групп, создав простой цикл; затем контрольная группа помечается 0, первая процедурная группа 1 и т. д.:

```
## R
no_strat_assgnt_fun <- function(dat, Nexp, K){
    dat <- dat %>%
        distinct(ID) %>%
        slice_sample(n=Nexp) %>%
        mutate(assgnt = runif(Nexp,0,1)) %>%
        mutate(group = -1) # initializing the "group" variable
    for(i in seq(1,K)){
        dat$group = ifelse(dat$assgnt >= (i-1)/K & dat$assgnt < i/K,i-1,dat$group)}
    dat <- dat %>%
        mutate(group = as.factor(group)) %>%
        select(-assgnt)
    return(dat)
}
no_strat_assgnt <- no_strat_assgnt_fun(hist_data, Nexp = 5000, K = 4)

## Python
def no_strat_assgnt_fun(dat_df, Nexp, K):
    dat_df = pd.DataFrame({'ID': dat_df.ID.unique()})
    dat_df = dat_df.sample(Nexp)
    dat_df['assgnt'] = np.random.uniform(0,1,Nexp)
    dat_df['group'] = -1 # initializing the "group" variable
    for i in range(K):
        dat_df.loc[dat_df['assgnt'].between(i/K, (i+1)/K, inclusive=True),
                   'group'] = i
    del(dat_df['assgnt'])
    return dat_df
no_strat_assgnt = no_strat_assgnt_fun(hist_data_df, Nexp = 5000, K = 4)
```

Однако трудность с приведенным выше подходом состоит в том, что экспериментальные группы вряд ли будут идеально сбалансированы по всем характеристикам клиентов. В целях создания сбалансированных экспериментальных групп мы захотим использовать технический прием, именуемый стратификацией.

Стратифицированная рандомизация

Почему чисто случайное размещение не является нашим лучшим вариантом выбора? Давайте вообразим, что мы проводим эксперимент на 20 клиентах, 10 из которых – мужчины и 10 – женщины. Если мы случайным образом относим каждого клиента либо в контрольную группу, либо в процедурную группу с вероятностью 50 %, то мы ожидаем, что в среднем в каждой из этих двух групп будет 5 мужчин и 5 женщин. «В среднем» здесь означает, что если бы мы повторили это размещение большое число раз, то среднее число мужчин в контрольной группе по всем размещениям составило бы 5. Но, основываясь на гипергеометрическом распределении[1], в любом данном эксперименте вероятность того, что в каждой группе будет ровно 5 мужчин и 5 женщин, составляет всего 34.4 %, а вероятность того, что в одной группе будет 7 или более мужчин, составляет 8.9 %. Очевидно, что эта проблема становится менее выраженной при более крупных размерах выборки.

При наличии 100 мужчин и 100 женщин вероятность того, что в одной группе будет 70 мужчин или более, становится ничтожной. Но нас волнует не только пол: в идеале мы также хотели бы иметь хороший баланс возраста, места жительства, шаблон пользования и т. д. Это обеспечивало бы максимальную релевантность наших результатов для всей нашей клиентской базы, а не только для определенной ее подгруппы.

К счастью, когда мы можем позволить себе роскошь назначать экспериментальные группы, например всем индивидуумам сразу, мы можем добиваться значительно большего, чем просто скрещивать пальцы и надеяться на лучшее. Мы можем стратифицировать наши данные: мы создаем «слои» похожих клиентов, именуемые стратами[2], и разбиваем их между нашими экспериментальными группами. В случае наших 10 клиентов мужского и 10 клиентов женского пола мы создали бы слой мужчин, 5 из которых отправились бы в контрольную группу, а 5 – в процедурную группу, и аналогично для женщин. Из этого следует, что у каждого индивидуума все еще есть 50%-ная вероятность попасть в любую из групп, но наши контрольная и процедурная группы теперь идеально сбалансированы по полу.

Стратификация может применяться к любому числу переменных. Имея пол и состояние здоровья, мы создали бы группу всех женщин из Канзаса и разбили бы ее поровну между нашей контрольной группой и процедурной группой и т. д. С крупным числом переменных или с непрерывными переменными становится невозможным найти точные сочетания; в наших

---

[1] Спасибо Андреасу Кальтенбруннеру (Andreas Kaltenbrunner) за указание на то, что это гипергеометрическое, а не биномиальное распределение.

[2] Это латинское слово обозначает слой, отсюда и слово стратификация.

данных у нас, возможно, не будет двух женщин в Канзасе с одинаковым возрастом и чьи объекты недвижимости имеют одинаковую площадь. Решение состоит в создании пар индивидуумов, которые «максимально похожи», например 58-летняя женщина с недвижимостью площадью 900 квадратных футов и 56-летняя женщина с недвижимостью площадью 930 квадратных футов, а затем случайном отнесении одной из них в контрольную группу, а другой – в процедурную группу. Благодаря этому они по-прежнему имеют одинаковую вероятность индивидуально оказаться в любой экспериментальной группе. Когда у нас есть только два индивидуума в расчете на страту, это называется «сочетанием» также потому, что мы создаем сочетающиеся пары клиентов.

Как это часто бывает, интуиция достаточно ясна, но дьявол кроется в деталях имплементации. Здесь есть два шага:

1) придать математический смысл словосочетанию «как можно более похожий»;
2) эффективно просматривать наши данные, чтобы размещать каждого клиента в паре.

Математическая концепция, которую мы будем использовать для выражения фразы «как можно более похожий», представлена расстоянием. Расстояние может легко применяться к одной числовой переменной. Если одному собственнику 56 лет, а другому 58 лет, то расстояние между ними составляет 58 – 56 = 2 года. Аналогичным образом мы могли бы сказать, что расстояние между объектом недвижимости площадью 900 квадратных футов и объектом недвижимости площадью 930 квадратных футов составляет 30 квадратных футов.

Первое осложнение заключается в агрегировании нескольких числовых переменных. Мы могли бы просто сложить (или, что эквивалентно, взять среднее значение) два числа и сказать, что наши два собственника «дистанцированы» на 2 + 30 = 32 единицы расстояния. Проблема такого подхода заключается в том, что, как видно в нашем примере, числа квадратных футов намного больше, чем числа лет. Разница в 30 лет между двумя собственниками, вероятно, будет гораздо важнее с точки зрения поведения, чем разница в 30 квадратных футов между их объектами недвижимости. Это решается путем перешкалирования всех наших числовых переменных в таком ключе, чтобы их минимум был сброшен до 0, а максимум – до 1. Из этого следует, что «расстояние» между самыми молодыми и самыми пожилыми собственниками равняется 1, а расстояние между самым малым и самым большим объектами недвижимости также равняется 1. Это решение не идеально, в особенности когда у вас есть аутсайдеры, но оно быстрое и достаточно неплохое для большинства целей.

Второе осложнение связано с категориальными переменными. Каково «расстояние» между таунхаусом и квартирой? Или между наличием бассейна и его отсутствием? Распространенное решение состоит в том, чтобы сказать, что расстояние между двумя объектами недвижимости равно 0, если они относятся к одной и той же категории, и 1 в противном случае. Например, расстояние между таунхаусом и домом будет равно 1 для переменной типа недвижимости. Математически это делается с помощью кодирования

категориальных переменных, используя одно активное состояние: то есть мы создаем столько двоичных переменных 0/1, сколько у нас категорий. Например, мы бы преобразовали тип недвижимости = («дом», «таунхаус», «квартира») в три переменные: `type.house`, `type.townhouse` и `type.apartment`. Объект недвижимости, который является квартирой, будет иметь значение 1 для переменной `type.apartment` и 0 для двух других переменных. Это также имеет добавочное преимущество, заключающееся в том, что категориальные «расстояния» сочетаются с числовым расстоянием. По сути, мы говорим, что разница между таунхаусом и квартирой так же важна, как разница между самым малым и самым большим объектами недвижимости. Этот подход снова спорен с точки зрения поведения, но он является хорошей отправной точкой, а часто и хорошей остановочной точкой.

Я написал функции R и Python, которые подготавливают наши данные путем перешкалирования числовых переменных и кодирования категориальных переменных с использованием одного активного состояния. Это всего лишь стереотипный исходный код, поэтому просто пропустите этот фрагмент кода, если вас не волнуют детали имплементации:

```
## Python (результат не показан)
def strat_prep_fun(dat_df):
    # Извлекаем переменные уровня недвижимости
    dat_df = dat_df.groupby(['ID']).agg(
        tier = ('tier', 'mean'),
        avg_review = ('avg_review', 'mean'),
        sq_ft = ('sq_ft', 'mean'),
        BPday = ('BPday', 'mean')).reset_index()
    dat_df['tier'] = pd.Categorical(dat_df.tier, categories=[3,2,1],
                                    ordered = True)
    dat_df['ID'] = dat_df.ID.astype(str)
    num_df = dat_df.copy().loc[:,dat_df.dtypes=='float64'] #Numeric vars
    cat_df = dat_df.copy().loc[:,dat_df.dtypes=='category'] #Categorical vars

    # Нормализуем все числовые переменные в интервал [0,1]
    scaler = MinMaxScaler()
    scaler.fit(num_df)
    num_np = scaler.transform(num_df)

    # Конвертируем все категориальные переменные, используя
    # кодировку с одним активным состоянием
    enc = OneHotEncoder(handle_unknown='ignore')
    enc.fit(cat_df)
    cat_np = enc.transform(cat_df).toarray()

    # Привязываем массивы
    data_np = np.concatenate((num_np, cat_np), axis=1)
    del num_df, num_np, cat_df, cat_np, enc, scaler
    return data_np
prepped_data_np = strat_prep_fun(hist_data_df)

## R
> strat_prep_fun <- function(dat){
    # Извлекаем переменные уровня недвижимости
```

```
    dat <- dat %>%
        group_by(ID, tier) %>%
        summarise(sq_ft = mean(sq_ft),
                  avg_review = mean(avg_review),
                  BPday = mean(BPday)) %>%
        ungroup()

    # Изолируем разные компоненты наших данных
    ID <- dat$ID # Owner identifier
    dat <- dat %>% select(-ID)
    cat_vars <- dat %>%
        # Отбираем категориальные переменные
        select_if(is.factor)
    num_vars <- dat %>%
        # Отбираем числовые переменные
        select_if(function(x) is.numeric(x)|is.integer(x))

    # Конвертируем все категориальные переменные, используя
    # кодировку с одним активным состоянием
    cat_vars_out <- data.frame(predict(dummyVars(" ~.", data=cat_vars),
                                       newdata = cat_vars))

    # Нормализуем числовые переменные
    num_vars_out <- num_vars %>%
        mutate_all(rescale)

    # Кладем переменные снова вместе
    dat_out <- cbind(ID, num_vars_out, cat_vars_out) %>%
        mutate(ID = as.character(ID)) %>%
        mutate_if(is.numeric, function(x) round(x, 4)) # Округляем для читаемости

    return(dat_out)}
> prepped_data <- strat_prep_fun(hist_data)
`summarise()` regrouping output by 'ID' (override with `.groups` argument)
> head(prepped_data, 5)
    ID  sq_ft avg_review  BPday tier.3 tier.2 tier.1
1    1 0.3321     0.3514 0.2365      1      0      0
2   10 0.3802     0.7191 0.5231      1      0      0
3  100 0.8370     0.6105 0.6603      0      0      1
4 1000 0.4476     0.4882 0.3843      1      0      0
5 1001 0.3323     0.7276 0.4316      0      1      0
```

После подготовки наших данных вторым шагом будет создание пар. Эта трудоемкая вычислительная задача быстро становится непрослеживаемой с более крупными данными (по меньшей мере, если вы хотите иметь оптимальное решение). К счастью, были созданы алгоритмы, которые будут справляться с этим за вас. В R можно применить функцию `block()` из пакета `blockTools`:

```
## R
stratified_data <- block(prepped_data, id.vars = c("ID"), n.tr = 3,
                         algorithm = "naiveGreedy", distance = "euclidean")
```

Параметры этой функции таковы:

`id.vars`
: Это переменная(ые), используемая(ые) для выявления индивидуумов в данных.

`n.tr`
: Это число экспериментальных групп, включая контрольную группу.

`algorithm`
: Указывает имя используемого алгоритма. Как и следовало ожидать, «optimal» будет продуцировать наилучшее паросочетание в целом, но этот алгоритм быстро может стать невыполнимым при больших объемах данных и лимитированной вычислительной мощности; «naiveGreedy» – наименее вычислительно требовательный и хорош для начала. «optGreedy», как правило, является неплохим компромиссом, когда вы готовы выполнить окончательное размещение.

`distance`
: Указывает на то, как рассчитать расстояния между отдельными индивидуумами. «euclidean» – это функция расстояния из средней школы, и она подходит для данных, которые мы подготовили.

Функция `block()` возвращает стратифицированное размещение в громоздком формате, поэтому я создал удобную обертку, которая преобразовывает ее результат в формат, который можно использовать. Без церемоний обратитесь к ее исходному коду в репо на GitHub[1]: ниже я просто покажу вам результат ее работы:

```
## R
> Nexp <- 4998 # Ограничиваем наши данные кратностью 3
> stratified_data <- block_wrapper_fun(prepped_data, Nexp)
Warning message:
attributes are not identical across measure variables;
they will be dropped
> head(stratified_data,3)
    ID  sq_ft avg_review  BPday tier.3 tier.2 tier.1  group
1  224 0.6932     0.8167 0.4964      1      0      0 treat1
2 3627 0.4143     0.9290 0.6084      1      0      0 treat1
3 4190 0.6686     0.5976 0.2820      1      0      0 treat1
```

Обратите внимание, что 5000 на 3 не делится, поэтому нам нужно случайным образом отбросить две строки, т. к. самое меньшее число, делящееся на 3, равно 4998. Сравнение двух типов случайных размещений показало бы, что экспериментальные группы, полученные с помощью стратифицированной рандомизации, гораздо более похожи, чем группы, полученные с помощью стандартной рандомизации.

Помимо помощи в сокращении шума в наших экспериментах, стратификация также полезна, если вы собираетесь проводить анализ подгрупп или

[1] См. https://oreil.ly/BehavioralDataAnalysisCh9.

модерации (которую мы обсудим в книге позже). Как говорится, «разбивайте на блоки [с помощью стратификации] все, что можно, и рандомизируйте все, что не можно» (Гербер и Грин, 2012).

Что делать пользователю Python?

Если вы являетесь пользователем Python, то в нем, к сожалению, эквивалента функции `block()` не существует[1]. Я написал функцию `stratified_assgnt_fun()`, которая выполняет эту работу для простых случаев; вы можете найти ее в репозитории книги на GitHub[2]. В качестве аргументов она принимает кадр данных с информацией о базе испытуемых, числе необходимых для эксперимента испытуемых и числе требуемых экспериментальных групп (включая контрольную, т. е. 2 для стандартного A/B-теста):

```
## Python
# Отбор из случайного месячного периода
per = random.sample(range(35), 1)[0] + 1
sample_df = hist_data_df.loc[hist_data_df.period == per].sample(5000)
stratified_data_df = stratified_assgnt_fun(sample_df, K=3)
```

Эта функция работает на моем ноутбуке около 30 секунд для 5000 испытуемых. Я также создал простую функцию для сравнения результатов стратифицированного размещения с чисто случайным размещением, `assgnt_comparison_fun()`, которая в качестве аргументов принимает кадр данных, который был сгенерирован предыдущей функцией, и имя числовой переменной, которую мы хотим использовать для сравнения, и возвращает стандартное отклонение (s.d.):

```
## Python
>> assgnt_comparison_fun(sample_df, 'sq_ft')
s.d. между групп для sq_ft равно 0.1 для стратифицированного размещения
s.d. между групп для sq_ft равно 4.0 для случайного размещения

assgnt_comparison_fun(sample_df, 'BPday')
s.d. между групп для BPday равно 0.0 для стратифицированного размещения
s.d. между групп для BPday равно 0.3 для случайного размещения
```

Хорошо видно, что даже очень простая функция, собранная наобум за несколько часов, может сделать ваши экспериментальные группы гораздо более похожими, чем чисто случайное размещение.

Стратифицированная рандомизация является эффективным и устойчивым подходом к экспериментальному размещению. Его устойчивость во многом обусловлена его прозрачностью: впоследствии вы всегда сможете проверить качество сбалансированности ваших экспериментальных групп с точки зрения средних значений числовых переменных и пропорций категорий категориальных переменных.

---

[1] Технически существуют функции для стратифицированного отбора, такие как `sklearn.utils.resample()`, но они не позволяют паросочетать данные на основе расстояния, как мы делаем здесь.

[2] См. https://oreil.ly/BehavioralDataAnalysis.

В дополнение к этому, поскольку каждый индивидуум в паре имеет одинаковую вероятность оказаться в любой экспериментальной группе, даже слабо или неправильно определенная функция расстояния не ухудшит ваше положение по сравнению с чисто случайным размещением. Главный риск исходит из включения слишком большого числа нерелевантных переменных, которые затем заглушают релевантные переменные. Это, однако, легко исправить, включая только те переменные, которые являются частью вашей причинно-следственной диаграммы, или главные демографические переменные. Не перегружайте другими переменными просто потому, что вы можете.

Категориальные переменные с большим числом категорий также иногда могут создавать шум в вашей стратификации из-за их грубости. Беря в качестве примера занятость, исследователь данных отличается от статистика, но интуитивно эта разница меньше, чем их общая разница, скажем, с пожарным. Очень гранулярные переменные игнорируют такие нюансы, и их лучше заменять более широкими категориями.

Из опасений, что эти предостережения вас обескуражат: стратификация является эффективной и устойчивой; не бойтесь ее использовать. Даже та стратификация, которая будет основываться только на нескольких ключевых демографических переменных, приведет к значительным улучшениям и должна быть вашим подходом, принятым на вооружение по умолчанию.

Теперь, когда мы определились с тем, как будем выполнять случайное размещение, давайте проведем анализ мощности, чтобы определить размер выборки.

Анализ мощности с помощью бутстраповских симуляций

После согласования с деловыми партнерами мы определяем, что хотим иметь 90%-ную мощность для увеличения чистой прибыли от брони в день на 2 доллара, поскольку это минимальный наблюдаемый эффект, в котором они были бы заинтересованы. Это транслируется в увеличение «сырой» прибыли от брони в день на 12 долларов для вмешательства в форме бесплатной уборки (процедура 1) и на 2 доллара для вмешательства в форме минимальной продолжительности (процедура 2). Это никоим образом не повлияет на наш анализ, так как мы можем просто изменить переменную результата, вычтя стоимость в размере 10 долларов из прибыли/день для объектов недвижимости в группе бесплатной уборки, но нам нужно будет иметь это в виду и не забывать это делать. Для простоты в наших аналитических расчетах мощности я буду обсуждать только вмешательство в форме минимальной продолжительности.

Методы симуляции по-настоящему блистают в ситуациях, подобных этой, где зачастую отсутствуют специальные формулы для расчета мощности или размеров выборки, либо имеющиеся формулы становятся ужасно сложными. Альтернативой было бы использовать стандартные формулы, которые игнорируют специфику ситуации (здесь стратификацию наших эксперименталь-

ных данных), и скрестить пальцы, чтобы все пошло хорошо (закон Мерфи: скорее всего, не пойдет).

Наш процесс будет таким же, как в главе 8.

1. Сначала мы определим метрическую функцию и функцию принятия решения.
2. Затем мы создадим функцию, которая симулирует один эксперимент для заданного размера выборки и размера эффекта.
3. Наконец, мы создадим функцию, которая симулирует большое число экспериментов, и подсчитаем, сколько из них приведут к истинно положительному результату, т. е. наша функция принятия решения точно улавливает эффект; процент истинно положительных результатов является нашей мощностью для этого размера выборки.

Однократная симуляция

Наша метрическая функция для экспериментальной процедуры минимальной продолжительности выглядит следующим образом:

```
## R
treat2_metric_fun <- function(dat){
    lin_model <- lm(BPday~sq_ft+tier+avg_review+group, data = dat)
   summ <- summary(lin_model)
    coeff <- summ$coefficients['grouptreat2', 'Estimate']
    return(coeff)}

## Python
def treat2_metric_fun(dat_df):
    model = ols("BPday~sq_ft+tier+avg_review+group", data=dat_df)
    res = model.fit(disp=0)
    coeff = res.params['group[T.treat2]']
    return coeff
```

Метрическая функция для экспериментальной процедуры 1 будет определена аналогичным образом.

Мы будем использовать функции `boot_CI_fun()` и `decision_fun()` из главы 8 повторно. Другими словами, наше правило принятия решения будет заключаться в имплементировании экспериментальной процедуры, если ее 90%-ный интервал уверенности строго выше нуля. Я повторяю их исходный код ниже, просто для справки:

```
## R
> boot_CI_fun <- function(dat, metric_fun, B = 100, conf.level = 0.9){
    # Задаем число бутстраповских выборок
    boot_metric_fun <- function(dat, J){
        boot_dat <- dat[J,]
        return(metric_fun(boot_dat))}
    boot.out <- boot(data=dat, statistic=boot_metric_fun, R=B)
    confint <- boot.ci(boot.out, conf = conf.level, type = c('perc'))
    CI <- confint$percent[c(4,5)]
    return(CI)}
```

```
> decision_fun <- function(dat, metric_fun){
    boot_CI <- boot_CI_fun(dat, metric_fun)
    decision <- ifelse(boot_CI[1]>0,1,0)
    return(decision)}

## Python
def boot_CI_fun(dat_df, metric_fun, B = 100, conf_level = 0.9):
    # Задаем размер выборки
    N = len(dat_df)
    coeffs = []
    for i in range(B):
        sim_data_df = dat_df.sample(n=N, replace = True)
        coeff = metric_fun(sim_data_df)
        coeffs.append(coeff)

    coeffs.sort()
    start_idx = round(B * (1 - conf_level) / 2)
    end_idx = - round(B * (1 - conf_level) / 2)
    confint = [coeffs[start_idx], coeffs[end_idx]]
    return(confint)

def decision_fun(dat_df, metric_fun, B = 100, conf_level = 0.9):
    boot_CI = boot_CI_fun(dat_df, metric_fun, B = B, conf_level = conf_level)
    decision = 1 if boot_CI[0] > 0 else 0
    return decision
```

Затем мы можем написать функцию выполнения одной симуляции со встроенной логикой, которую мы видели до этого:

```
## R
single_sim_fun <- function(dat, Nexp, eff_size){

    # Отфильтровать данные до случайного месяца ❶
    per <- sample(1:35, size=1)
    dat <- dat %>%
        filter(period == per)

    # Подготовить стратифицированное размещение ❷
    # для случайной выборки желаемого размера
    stratified_assgnt <- dat %>%
        slice_sample(n=Nexp) %>%
        # Стратифицированное размещение
        block_wrapper_fun() %>%
        # извлечь ИД и размещение в группе
        select(ID, group)

    sim_data <- dat %>%
        # Применить размещение к полным данным ❸
        inner_join(stratified_assgnt) %>%
        # Добавить размер целевого эффекта
        mutate(BPday = ifelse(group == 'treat2', BPday + eff_size, BPday))

    # Вычислить решение (мы хотим, чтобы оно равнялось 1) ❹
    decision <- decision_fun(sim_data, treat2_metric_fun)
    return(decision)}
```

```
## Python
def single_sim_fun(dat_df, metric_fun, Nexp, eff_size, B = 100,
                   conf_level = 0.9):

    # Отфильтровать данные до случайного месяца ❶
    per = random.sample(range(35), 1)[0] + 1
    dat_df = dat_df.loc[dat_df.period == per]
    dat_df = dat_df.sample(n=Nexp)

    # Подготовить стратифицированное размещение ❷
    # для случайной выборки желаемого размера
    assgnt = strat_assgnt_fun(dat_df, Nexp = Nexp)
    sim_data_df = dat_df.merge(assgnt, on='ID', how='inner')

    # Добавить размер целевого эффекта ❸
    sim_data_df.BPday = np.where(sim_data_df.group == 'treat2',
                                 sim_data_df.BPday + eff_size, sim_data_df.BPday)

    # Вычислить решение (мы хотим, чтобы оно равнялось 1) ❹
    decision = decision_fun(sim_data_df, metric_fun, B = B,
                            conf_level = conf_level)
    return decision
```

❶ **Выбрать месяц в случайном порядке, чтобы имитировать способ, которым мы будем выполнять наш фактический эксперимент (в нашем анализе мощности мы не хотим использовать данные с разницей в 10 лет).**

❷ **Сгенерировать стратифицированное случайное размещение для выборки желаемого размера.**

❸ **Применить размещение к данным и применить размер целевого эффекта к процедурной группе 2.**

❹ **Применить функцию принятия решения и вернуть ее результат.**

Симуляции в масштабе

С этого места мы можем применить ту же совокупную функцию симуляций мощности, что и в главе 8 (повторяется ниже для справки):

```
## R
power_sim_fun <- function(dat, Nexp, eff_size, Nsim){
    power_list <- vector(mode = "list", length = Nsim)
    for(i in 1:Nsim){
        power_list[[i]] <- single_sim_fun(dat, Nexp, eff_size)}
    power <- mean(unlist(power_list))
    return(power)}

## Python
def power_sim_fun(dat_df, metric_fun, Nexp, eff_size, Nsim, B = 100,
                  conf_level = 0.9):
    power_lst = []
    for i in range(Nsim):
        power_lst.append(single_sim_fun(dat_df, metric_fun = metric_fun,
                                        Nexp = Nexp, eff_size = eff_size,
                                        B = B, conf_level = conf_level))
    power = np.mean(power_lst)
    return(power)
```

Наш максимальный размер выборки составит 5000, потому что это суммарное число собственников объектов недвижимости, которые есть в AirCnC. Это число поддается управлению для симуляционных целей, поэтому давайте сначала проведем 100 симуляций с таким размером выборки. Нет смысла продвигаться дальше, если окажется, что для эксперимента нам понадобится использовать всю нашу популяцию. Мы находим мощность, равную 1, что утешает: требуемый размер выборки меньше, чем наша суммарная популяция. С этого места мы пробуем разные размеры выборки, поступательно увеличивая число симуляций, по мере того как приближаемся к размеру выборки с мощностью, равной 0.90 (рис. 9.2).

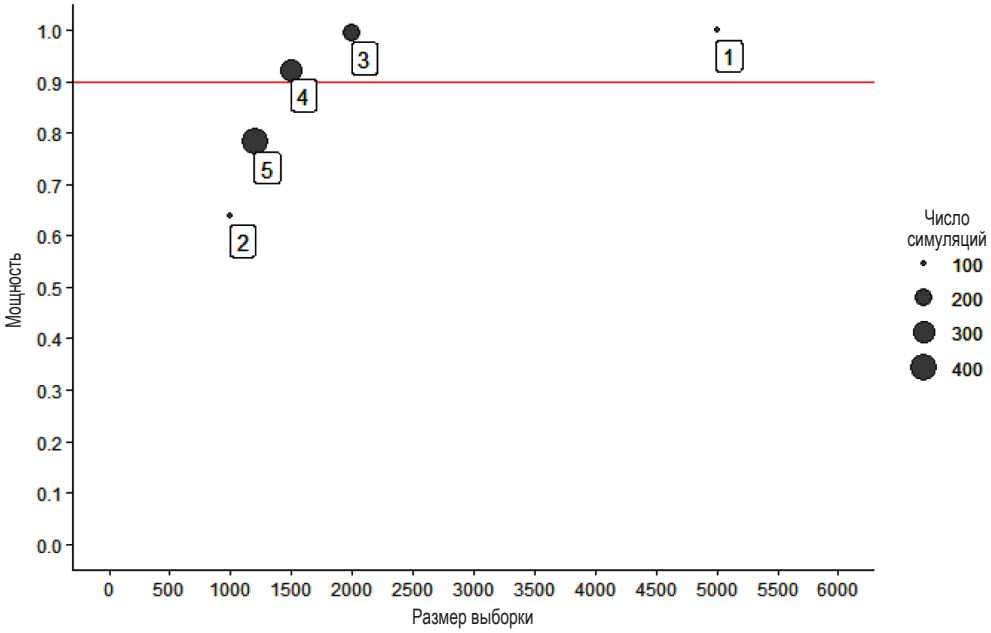

Рис. 9.2 ❖ Итеративные симуляции мощности с увеличивающимся числом симуляций, при этом метки обозначают величину прогонов

Похоже, что подойдет выборка в размере 1500. Как и в главе 8, давайте теперь определим кривую мощности для различных размеров эффекта при таком размере выборки (рис. 9.3).

По рис. 9.3 хорошо видно, что наша кривая мощности очень круто падает, переходя от размера эффекта в 2 доллара к 1 доллару, и наша мощность почти равна нулю для размеров эффекта меньше 1; то есть если мы допустим, что экспериментальная процедура увеличивает прибыль от бронирования в день на 1 доллар, то мы, скорее всего, получим интервал уверенности, который включает ноль, а затем сделаем вывод, что эффекта нет. В левом конце кривой наша симулированная значимость равна нулю вместо ожидаемых 5 %. Давайте обсудим вопросы, что является тому причиной и стоит ли нам об этом беспокоиться.

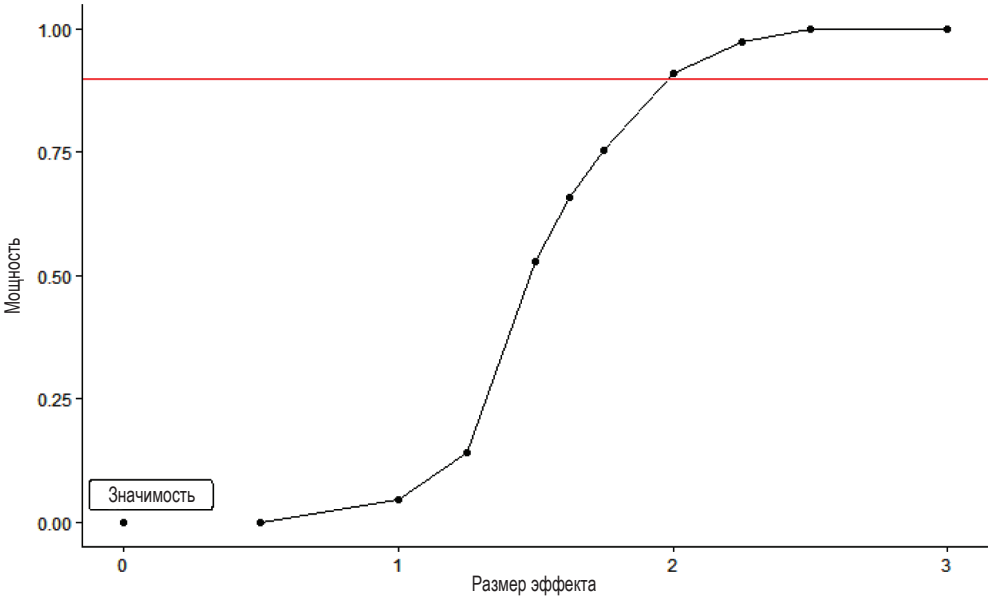

Рис. 9.3 ❖ Мощность в обнаружении различных размеров эффекта и значимости (размер выборки = 1500)

Понимание компромисса между мощностью и значимостью

Как я упоминал в предыдущей главе, если наши данные «хорошо себя ведут» (т. е. нормально распределены, размещены между экспериментальными группами чисто случайным образом и т. д.) и нет истинного эффекта, то мы ожидаем, что 90%-ный интервал уверенности будет включать ноль в 90 % случаев, будет строго выше него в 5 % случаев и будет строго ниже него в 5 % случаев. Здесь, из-за стратифицированной рандомизации, наша частота ложноположительных результатов оказывается ниже 5 %: я не наблюдал вообще ни одного в 500 симуляциях. В свою очередь, уменьшая шум в наших данных, стратифицированная рандомизация также снижает риск ложноположительных результатов.

Замечательно, но в данном случае это, возможно, будет больно уж хорошо выглядеть, потому что это также снижает кривую мощности для малых положительных эффектов вплоть до 1, как мы видим на рис. 9.3. Давайте сформулируем это по-другому: если бы у нас был 5%-ный шанс допустить, что эффект есть, когда его нет, то у нас был бы по меньшей мере 5%-ный шанс допустить, что эффект есть, когда он мал. В этом смысле значимость дает нам некоторую «бесплатную» мощность для размеров низких эффектов.

Давайте сравним предыдущую кривую мощности с кривыми мощности для более низких уровней уверенности. По определению, это даст нам более узкие интервалы уверенности, а значит, более высокую значимость и более высокую мощность, в особенности для размеров малых эффектов (рис. 9.4).

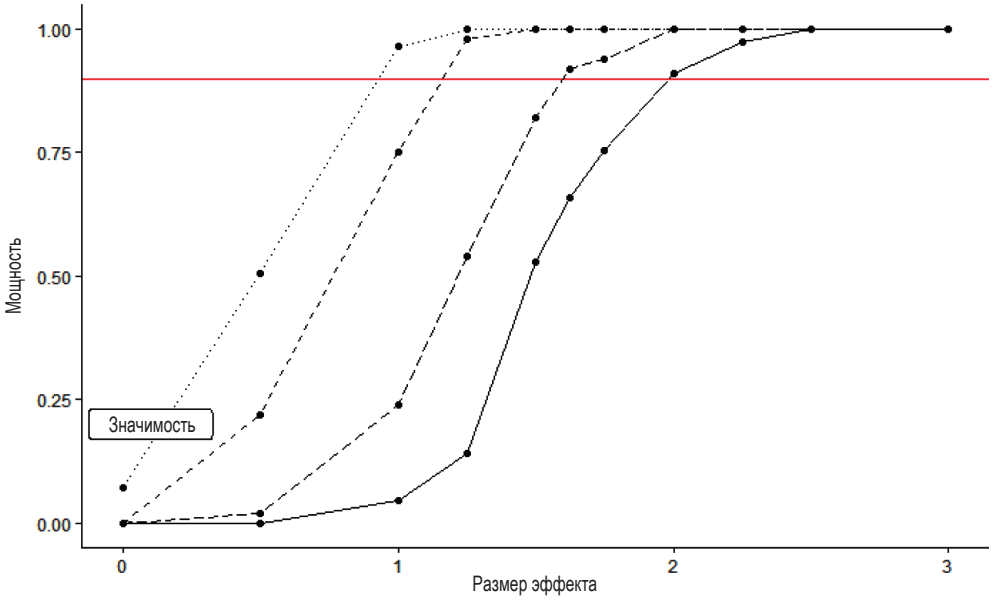

Рис. 9.4 ❖ Сравнение кривых мощности для уровней уверенности 0.90 (сплошная линия), 0.80 (длинная пунктирная линия), 0.60 (пунктирная линия) и 0.40 (пунктирная линия)

Хорошо видно, что при 40%-ном интервале уверенности мы получаем лишь малое увеличение значимости, но значительное увеличение мощности, причем для обнаружения эффекта размером 0.5, мощность приближенно достигает 50 %.

Означает ли это, что мы должны использовать 40%-ный интервал уверенности вместо 90%-го? Все зависит от ситуации. Давайте вернемся к деловой задаче. Наши деловые партнеры запросили мощность 90 % для размера эффекта, равного 2, потому что они не заинтересованы в том, чтобы с головой окунаться в суету по имплементированию любой из экспериментальных процедур, если выгоды ниже этого. Отсюда, с точки зрения бизнеса, улавливание истинного размера эффекта, равного 0.5, с интервалом уверенности, который не включает в себя ноль, или с интервалом уверенности, который включает в себя ноль, по сути, является одним и тем же. В любом случае, никакой экспериментальной процедуры имплементировано не будет. Следовательно, кривая мощности для 90%-го интервала уверенности отражает наши деловые цели лучше.

С другой стороны, при проведении эксперимента, подобного «кнопке выполнения в 1 клик» из главы 8, затраты и риски имплементации лимитированы. Порог мощности в 90 % – это всего лишь базовый уровень, и практически любой строго положительный эффект даст добро на имплементацию. В такой ситуации увеличение мощности для размеров малых эффектов, вероятно, заслуживает небольшого увеличения значимости.

В более широком смысле, когда ваши данные или ваш экспериментальный дизайн расходятся со стандартным каркасом, анализ мощности перестает

быть простой подстановкой обычных чисел в формулу и требует понимания того, что происходит, и вынесения суждений о правильных решениях. К счастью, кривые мощности предлагают отличный инструмент для визуализации возможных результатов эксперимента в условиях разных сценариев и разных правил принятия решения.

Анализ и интерпретация экспериментальных результатов

Проведя эксперимент, мы сможем приступить к анализу его результатов. Наша целевая метрика, средняя прибыль от брони в день, является непрерывной и недвоичной; следовательно, двумя надлежащими методами являются Т-тест средних и линейная регрессия. Если вы хотите узнать о Т-тесте больше, то я отошлю вас к Герберу и Грину (2012), а вот о линейной регрессии я расскажу далее.

Прежде чем приступить к количественному анализу, впомните, что мы не могли принуждать собственников иметь двухсуточную минимальную продолжительность или соглашаться на сокращение продолжительности временного окна для уборки в обмен на бесплатные услуги по уборке. Мы могли лишь предлагать им возможность согласиться на предложение, которым некоторые воспользовались, а другие нет. С технической точки зрения, этот подход называется побудительным дизайном, потому что мы побуждаем испытуемых принимать наше предложение.

Примеры побудительного дизайна очень распространены, но они вводят несколько дополнительных аспектов, потому что теперь у нас в процедурной группе есть две категории людей: те, кто сделал свой выбор в пользу предложения, и те, кто этого не сделал. Для практических целей это означает, что у нас есть два разных вопроса, на которые мы можем попытаться ответить:

- что произойдет, если мы предложим всей популяции наших собственников возможность согласиться на экспериментальную процедуру?
- что произойдет, если мы принудительно возложим экспериментальную процедуру на всю популяцию наших собственников, не предоставляя им возможности сделать свой выбор против?

Ответ на первый вопрос называется оценкой намерения относительно экспериментальной процедуры (intention-to-treat, аббр. ITT), потому что мы желаем, чтобы люди подвергались воздействию экспериментальной процедуры, но не делаем это в императивном порядке. Второй вопрос сложнее, и мы не можем ответить на него полностью, основываясь только на побудительном дизайне (или, по меньшей мере, он требует дополнительных допущений), но мы можем получить более близкую аппроксимацию, чем оценка ITT, с помощью оценки причинно-следственного эффекта среднего по соблюдающим испытуемым (complier average causal effect, аббр. CACE).

Давайте рассчитаем обе эти оценки по очереди.

Оценка намерения относительно экспериментальной процедуры для стимулирования вмешательства

Давайте сначала рассчитаем оценку ITT, что будет очень просто: это всего лишь коэффициент эффекта экспериментального размещения, как мы рассчитали в предыдущей главе. Должны ли мы учитывать тот факт, что большинство собственников в процедурных группах не сделали своего выбора в пользу предложения? Нет. Тот факт, что коэффициент ITT разбавляется людьми, которые сделали свой выбор против предложения, является функциональной особенностью, а не дефектом: такое же разбавление произошло бы в большем масштабе.

Давайте вычтем 10 долларов для собственников в группе уборки, которые сделали свой выбор в пользу предложения, чтобы учесть дополнительные расходы, а затем выполним линейную регрессию. Мы могли бы применить наши метрические функции отдельно, но я предпочитаю выполнять всю регрессию сразу, чтобы иметь возможность видеть другие коэффициенты:

```
## Python (результат не показан)
exp_data_reg_df = exp_data_df.copy()
exp_data_reg_df.BPday = np.where((exp_data_reg_df.compliant == 1) & \
                                 (exp_data_reg_df.group == 'treat2'),
                                 exp_data_reg_df.BPday -10,
                                 exp_data_reg_df.BPday)
print(ols("BPday~sq_ft+tier+avg_review+group",
          data=exp_data_reg_df).fit(disp=0).summary())

## R
> exp_data_reg <- exp_data %>%
    mutate(BPday = BPday - ifelse(group=="treat2" & compliant, 10,0))
> lin_model <- lm(BPday~sq_ft+tier+avg_review+group, data = exp_data_reg)
> summary(lin_model)

...
Coefficients:
             Estimate Std. Error t value        Pr(>|t|)
(Intercept) 19.232831   3.573522   5.382 0.0000000854103 ***
sq_ft        0.006846   0.003726   1.838          0.0663 .
tier2        1.059599   0.840598   1.261          0.2077
tier1        5.170473   1.036066   4.990 0.0000006728868 ***
avg_review   1.692557   0.253566   6.675 0.0000000000347 ***
grouptreat1  0.966938   0.888683   1.088          0.2767
grouptreat2 -0.172594   0.888391  -0.194          0.8460
...
```

Мы выполняем регрессию интересующей нас переменной, прибыли от брони в день (BPday), на квадратном метраже недвижимости, городскому ярусу, среднему значению отзывов клиентов и экспериментальным группам. Коэффициент для `grouptreat1` относится к экспериментальной процедуре,

связанной с минимальной продолжительностью, в то время как `grouptreat2` относится к экспериментальной процедуре бесплатной уборки.

Первая экспериментальная процедура увеличивает прибыль от брони в день в среднем примерно на 0.97 доллара, но *p*-значение является умеренно высоким, примерно 0.27. Оно говорит о том, что коэффициент может отличаться от нуля совсем незначительно, и действительно, соответствующий бутстраповский 90%-ный интервал уверенности составляет приближенно [0.002; 2.66].

Если бы вы проводили Т-тест, сравнивая первую процедурную группу с контрольной группой, то вы бы обнаружили, что абсолютное значение тестовой статистики составляет 0.96, что близко к коэффициенту регрессии, которую мы только что выполнили. Схожим образом сырая разница в средних значениях прибыли от брони в день между контрольной группой и первой процедурной группой составляет приближенно 0.85. Должно ли это нас удивлять? Нет. Благодаря стратификации наши экспериментальные группы очень хорошо сбалансированы, и поэтому другие независимые переменные оказывают одинаковый средний эффект по группам. Из этого следует, что даже не объясняющие ковариаты метрики являются несмещенными (однако их *p*-значения отклонятся, поскольку они не учитывают стратификацию).

Вторая экспериментальная процедура снижает прибыль от брони в день примерно на 0.17 доллара, после вычета затрат, не очень выгодное предложение. Соответствующий интервал уверенности равен [–2.23; 1.61].

Помните, что наши деловые партнеры заинтересованы в имплементировании вмешательства только в том случае, если оно приносит дополнительные 2 доллара в сутки сверх затрат. На первый взгляд это исключило бы имплементирование вмешательства в форме минимальной продолжительности – не только потому, что статистическая значимость является пограничной, но в первую очередь из-за отсутствия экономической значимости. Даже если бы нижняя граница интервала уверенности была прямо выше нуля, это все равно не изменило бы решения наших деловых партнеров.

Если бы на этом история закончилась, то наши деловые партнеры не стали бы имплементировать ни одно из побудительных вмешательств. Однако, возможно, стоит подумать о том, что произойдет, если мы принудительно возложим экспериментальную процедуру в форме минимальной продолжительности по всем направлениям, не предлагая людям возможности делать свой выбор против.

Оценка причинно-следственного эффекта среднего по соблюдающим требования испытуемым в целях обязательного вмешательства

При наличии у нас побудительного дизайна можем ли мы оценить эффект от обязательной экспериментальной процедуры? Заманчивым, но неправильным ответом на этот вопрос было сравнение значения дело-

вой метрики для категории сделавших свой выбор в пользу предложения (т. н. «подвергающихся экспериментальной процедуре»), с одной стороны, со значением для категории сделавших свой выбор против предложения и контрольной группы, с другой стороны, собрав последних двух вместе как «не подвергающихся экспериментальной процедуре». Можно было бы допустить, что это сравнение отразит ожидаемый результат принудительного возложения экспериментальной процедуры по всем направлениям, т. е. навязывания двухсуточного минимума горячим рынкам или одностороннего сокращения времени уборки, предлагая бесплатную уборку, независимо от предпочтений собственников. Это не так, потому что выбор в пользу экспериментальной процедуры не является рандомизированным и, скорее всего, будет спутан. Внутри процедурных групп вполне вероятно, что собственники, которые делают выбор в пользу предложения, в некоторых отношениях отличаются от собственников, которые этого не делают, например у них есть финансовые потребности или другие характеристики, которые побуждают их уделять больше внимания и усилий своему объекту недвижимости (рис. 9.5).

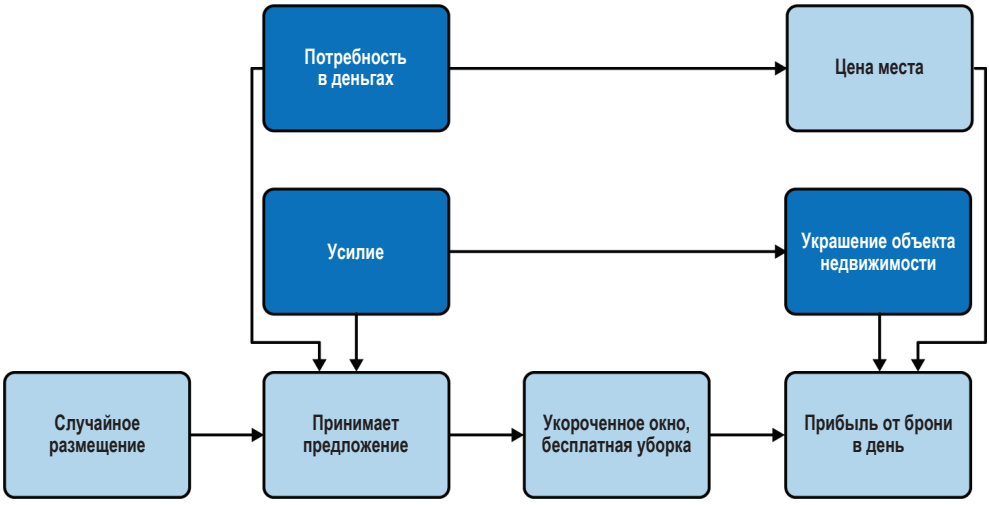

Рис. 9.5 ❖ Экспериментальное размещение рандомизировано, но принятие экспериментальной процедуры бесплатной уборки таким не является

Если эта причинно-следственная диаграмма верна, то принятие предложения о бесплатной уборке коррелирует с поведениями, которые увеличивают прибыль от брони в день, и увеличивает наш коэффициент. Другими словами, мы ошибочно приписали бы предложению некоторый эффект от таких поведений, если бы сравнивали людей, которые делают выбор в пользу предложения, с людьми, которые этого не делают. Случайное размещение обеспечивает, чтобы сравнения между экспериментальными группами были несмещенными, но это ничего не гарантирует для подгрупп дальше по цепочке.

В случае A/B-тестов электронной почты этот предел на эффект рандомизации означает, что все частотные показатели (например, частота открытия, кликабельность и т. д.) должны в качестве числителя иметь число людей в экспериментальной группе, а не число людей с предыдущего этапа. Если 50 % людей открывают вашу электронную почту и 50 % тех, кто открывает ссылку в почтовом сообщении, то коэффициент кликабельности должен быть выражен как 25 %, а не 50 %.

В побудительном дизайне мы желаем, чтобы люди в процедурной группе делали свой выбор в пользу предложения и подвергались процедуре, но мы также желаем, чтобы люди в контрольной группе не подвергались процедуре. Однако в определенных ситуациях мы не можем помешать им получать доступ к экспериментальной процедуре. В нашем настоящем примере экспериментальная процедура бесплатной уборки имеет функциональные особенности, которые полностью находятся под нашим контролем: собственники объектов недвижимости могут пользоваться профессиональными услугами по уборке, но они должны за это платить, а окно между бронированиями зашито в программно-информационном обеспечении. Следовательно, никто за пределами этой процедурной группы не сможет получать доступ конкретно к той экспериментальной процедуре. Однако с процедурой в форме двухсуточного минимума все выглядит менее ясным: собственники объектов недвижимости, не входящие в эту процедурную группу, вполне возможно, неофициально навяжут двухсуточный минимум, отказав в запрашиваемых односуточных бронированиях (рис. 9.6).

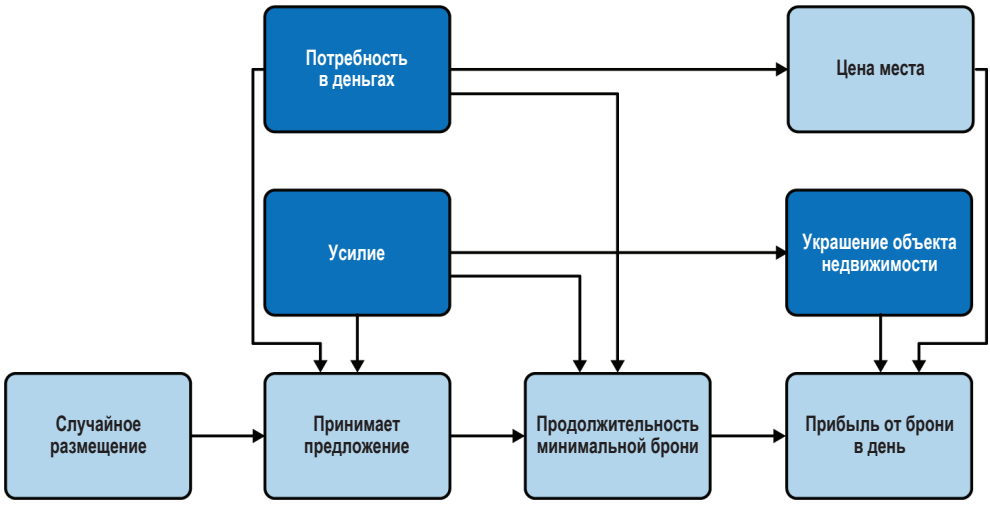

Рис. 9.6 ❖ Экспериментальное размещение рандомизировано, но принятие экспериментальной процедуры минимального бронирования таким не является, и это может происходить за пределами процедурной группы

При экспериментальной процедуре минимального бронирования мы можем наблюдать четыре возможных случая для собственников:

A) находиться в контрольной группе и не иметь двухсуточный минимум;

B) находиться в контрольной группе и тем не менее иметь двухсуточный минимум;
C) находиться в процедурной группе и иметь двухсуточный минимум;
D) находиться в процедурной группе и тем не менее не иметь двухсуточный минимум.

Эта категоризация еще не отвечает на наш вопрос, но она дает нам несколько важных строительных блоков, позволяющих различать эффект экспериментальной процедуры как таковой и эффект ненаблюдаемых факторов, способствующих установлению двухсуточного минимума. Давайте на секунду вообразим, что мы могли бы эти факторы наблюдать и ранжировать всех собственников в убывающем порядке этих факторов. Из-за рандомизации мы можем допустить, что контрольная и процедурная группы разумно идентичны с точки зрения распределения ненаблюдаемых факторов (рис. 9.7).

Все собственники в контрольной группе, значение которых для ненаблюдаемых факторов достаточно велико, будут имплементировать двухсуточный минимум (группа B), а все остальные собственники в контрольной группе не будут (группа A). Для процедурной группы мы можем разумно допустить, что собственники с высоким значением коэффициента по-прежнему имплементируют двухсуточный минимум и что наше побуждающее вмешательство просто снижает порог, привлекая некоторых собственников, которые в противном случае не стали бы его имплементировать (вместе они образуют группу C). Наконец, собственники, у которых слишком низкий уровень, по-прежнему его не имплементируют, несмотря на все наши усилия (группа D).

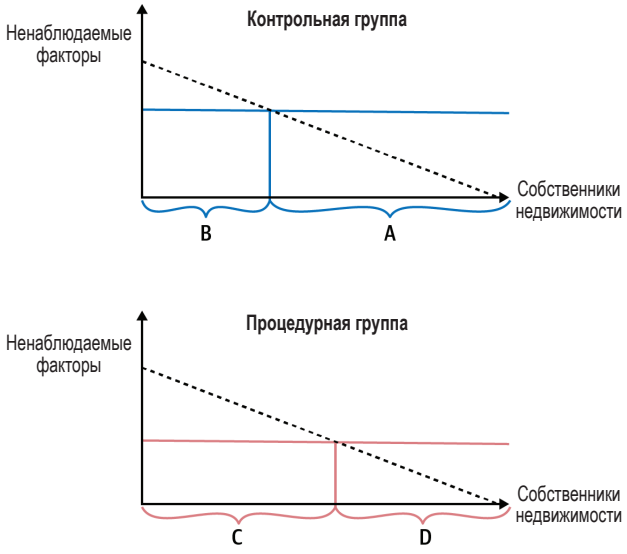

Рис. 9.7 ❖ Распределение ненаблюдаемых факторов и наблюдаемых поведений в двух группах

На языке эконометрии испытуемые, которых всегда будут подвергать экспериментальной процедуре независимо от их экспериментального размещения (группа B в контрольной группе и соответствующая часть группы C в процедурной группе), называются «всегда принимающими» (always-taker). Испытуемые, которых никогда не будут подвергать экспериментальной процедуре, независимо от их экспериментального размещения (группа D в процедурной группе и соответствующая часть группы A в контрольной группе), предсказуемо называются «никогда не принимающими» (never-taker). Испытуемые, которых подвергают экспериментальной процедуре тогда и только тогда, когда они входят в процедурную группу (наложение между группой A и группой C), называются «соблюдающими требования» (complier).

Теоретически у вас может быть четвертая категория, а именно испытуемые, которых подвергают экспериментальной процедуре тогда и только тогда, когда они входят в контрольную группу. Они называются «непокорными» (defier), потому что всегда делают прямо противоположное тому, что мы от них хотим. Техническим термином для такого поведения в психологии является реактивное сопротивление (reactance). Хотя это может происходить в реальной жизни (подростки, кхе-кхе), это редко вызывает беспокойство в деловых условиях, если только вы не пытаетесь заставить людей делать то, что они не хотят делать, и я вам в этом не помощник.

По определению, в нашем эксперименте мы не можем наблюдать ненаблюдаемые факторы, а это означает, что мы можем с уверенностью выявить только две группы: всегда принимающих испытуемых, отнесенных в контрольную группу (B), и никогда не принимающих испытуемых, отнесенных в процедурную группу (D). Мы не можем знать, являются ли собственники в процедурной группе, имплементирующие двухсуточный минимум (C), всегда принимающими испытуемыми или испытуемыми, соблюдающими требования, и мы не можем знать, являются ли собственники в контрольной группе, не имплементирующими его (A), испытуемыми, соблюдающими требования, или никогда не принимающими испытуемыми. Однако, и в этом весь фокус, мы можем вычленить всегда принимающих и никогда не принимающих испытуемых в разных экспериментальных группах, чтобы измерить эффект экспериментальной процедуры на соблюдающих требования испытуемых, который называется причинно-следственным эффектом средним по всем соблюдающим испытуемым (CACE). Формула для CACE очень проста[1]:

$$CACE = \frac{ITT}{P(\textit{подвергнут процедуре}|TG) - P(\textit{подвергнут процедуре}|CG)}.$$

Другими словами, чтобы определить эффект от экспериментальной процедуры на соблюдающих требования испытуемых, нам просто нужно взвесить нашу упомянутую выше оценку ITT на меру несоблюдаемости в нашем экс-

[1] Если вам интересно знать, откуда взялась эта формула, то ее математическое выведение доступно в репозитории книги на GitHub.

перименте: если у нас есть полная соблюдаемость в обеих группах, а значит, никто не получает доступ к экспериментальной процедуре в контрольной группе ($P(подвергнут\ процедуре|CG) = 0$), и все подвергаются процедуре в процедурной группе ($P(подвергнут\ процедуре|TG) = 1$), то это упрощает оценку ITT. Очень часто с помощью вариантов побудительного дизайна мы можем мешать людям из контрольной группы получать доступ к экспериментальной процедуре, но только малая доля людей из процедурной группы действительно подвергается процедуре. В этом случае CACE кратен ITT: если только 10 % людей в процедурной группе подвергаются процедуре, то наш эффект очень ослаблен, и наш CACE равен 10-кратному ITT.

Метрика CACE очень полезна в двух отношениях: во-первых, она дает нам оценку эффекта от имплементирования экспериментальной процедуры по всем направлениям без возможности отказа. Во-вторых, рассмотрение взаимосвязи между ITT и CACE позволяет нам различать две возможные ситуации:

- оценка ITT низкая, но вероятность $P(подвергнут\ процедуре|TG) - P(подвергнут\ процедуре|CG)$ высокая, а значит, вмешательство оказывает низкое влияние на испытуемых, соблюдающих требования, но имеет высокий уровень соблюдаемости;
- и наоборот, оценка ITT высокая, но вероятность $P(подвергнут\ процедуре|TG) - P(подвергнут\ процедуре|CG)$ низкая, а значит, вмешательство оказывает большое влияние на испытуемых, соблюдающих требования, но имеет низкий уровень соблюдаемости.

В первом случае мы сосредоточили бы наши усилия на повышении эффективности вмешательства, тогда как во втором случае мы сосредоточились бы на увеличении коэффициента охвата, возможно, сделав вмешательство обязательным. Эти идеи также помогут нам разведать альтернативные варианты дизайна: возможно, 8 часов – это слишком мало, но 12 часов было бы приемлемо? Может быть, нам не нужно предлагать собственникам бесплатную уборку, достаточно просто предложить авторитетного поставщика услуг?

В нашем настоящем эксперименте уровни принятия (uptake rates) в процедурных группах довольно низкие, в среднем около 20 %:

```
## R (результат не показан)
> exp_data_reg %>%
    group_by(group) %>%
    summarise(compliance_rate = mean(compliant))

## Python
exp_data_reg_df.groupby('group').agg(compliance_rate = ('compliant', 'mean'))
Out[15]:
        compliance_rate
group
ctrl              1.000
treat1            0.238
treat2            0.166
```

Это означает, что наша оценка CACE эффекта экспериментальной процедуры в форме минимальной продолжительности будет значительно выше, чем оценка ITT:

$$CACE_1 = ITT_1 / СтепеньСоблюдаемости_1 = 0.97/0.24 \approx 4.06.$$

Теперь это гораздо более интересное значение. Низкий уровень принятия и высокий уровень CACE свидетельствуют о том, что наше вмешательство является фундаментально обоснованным и действительно приносит пользу при его имплементации. Мы можем либо попытаться увеличить уровень принятия, изменив дизайн, либо сделать вмешательство обязательным.

CACE имеет очень опрятную, но узкую интерпретацию: поскольку мы (неявно) сравниваем одних и тех же людей, т. е. тех, кто соблюдает требования, в контрольной и процедурной группах, наша оценка эффекта экспериментальной процедуры не смещена. Мы не без умысла улавливаем влияние других факторов. Однако мы измеряем его только для этой узкой части нашей популяции, поэтому обобщаемость не является данностью. Соблюдающие требования испытуемые могли бы иметь характеристики, которые взаимодействуют с нашей экспериментальной процедурой. То есть у них могут быть черты, которые влияют не на то (или не только на то), примут они экспериментальную процедуру или нет, но и на то, насколько экспериментальная процедура на них влияет. Вот где нам нужно перейти от причинно-следственной связи к поведенческой: является ли наша экспериментальная процедура приливом, который поднимает все лодки, или же имеют значение люди, которые в этом участвуют? Например, в следующей главе мы рассмотрим пример пути разговора в кол-центрах. И в этом случае соблюдение требований не просто означает применение экспериментальной процедуры, но и приложение усилий, чтобы делать это убедительно, а не просто проходить по списку телодвижений.

Здесь наше вмешательство имплементируется через веб-сайт AirCnC. Двухсуточный минимум – он и в Африке двухсуточный минимум, у кого бы вы ни снимали. Это означает, что мы можем быть уверены в том, что наша экспериментальная процедура будет имплементирована в соответствии с планом, если она будет разворачиваться по всем направлениям, и мы можем дать нашим деловым партнерам для этого зеленый свет.

Выводы

В предыдущей главе нам приходилось рандомизировать наше экспериментальное размещение «на лету», когда клиенты подсоединялись к веб-сайту. В этой главе мы смогли выполнить случайное размещение сразу и, следовательно, стратифицировать (иначе говоря, разбивать на блоки) нашу выборку, создавая пары похожих испытуемых, один из которых относился к контрольной группе, а другой – к процедурной группе. Хотя это добавило дополнительный уровень сложности, это также значительно повысило эффективность (в статистических терминах – мощность) нашего эксперимента. Как

только вы познакомитесь со стратификацией, вы по достоинству оцените способность извлекать глубинную информацию даже из малых выборок.

Мы также ввели второе осложнение: наше экспериментальное вмешательство было побудительной экспериментальной процедурой. Мы предлагали собственникам возможности, но мы не могли заставлять их принимать, и принятие не было случайным. В подобных ситуациях мы можем легко измерять эффект побудительного вмешательства как таковой, но измерять эффект принятия предложения (или выбора в пользу экспериментальной процедуры) становится заковыристей. К счастью, в лице CACE у нас есть несмещенная оценка этого эффекта для соблюдающих требования испытуемых в нашей экспериментальной популяции. Когда мы можем допустить отсутствие взаимодействия между личностными характеристиками и экспериментальной процедурой, CACE способен обобщаться на всю нашу экспериментальную популяцию. Даже если мы не можем обобщить эту оценку так далеко, она дает менее смещенную оценку, чем простое сравнение контрольной и процедурной групп (т. е. оценку намерения относительно экспериментальной процедуры).

Наконец, у нас было несколько экспериментальных процедур. Это ничего принципиально не изменило, но и добавило немного сложности. Я бы рекомендовал начать свое экспериментальное путешествие только с одной экспериментальной процедуры, но я убежден, что в долгосрочной перспективе вы по достоинству оцените организационные «постоянные затраты» на проведение эксперимента: получение одобрения от всех интересантов (деловых партнеров, юридического отдела и т. д.) и введение в эксплуатацию технологии и конвейера данных занимают едва ли больше времени с двумя экспериментальными процедурами, чем с одной. Таким образом, проведение экспериментов сразу с несколькими экспериментальными процедурами является ключевым шагом в увеличении числа таких процедур, тестируемых в течение года.

Глава 10

Кластерная рандомизация и иерархическое моделирование

Наш последний эксперимент, хотя и концептуально простой, проиллюстрирует несколько логистических и статистических трудностей экспериментов в бизнесе. AirCnC имеет 10 кол-центров, занимающихся обслуживанием клиентских телефонных звонков по всей стране, где представители решают любые проблемы, которые могут возникать в ходе бронирования (например, оплата не прошла, объект недвижимости не похож на фотографии и т. д.). Прочитав статью в Harvard Business Review (HBR)[1] об обслуживании клиентов, вице-президент по обслуживанию клиентов решил внести изменения в свод стандартных операционных процедур: вместо того чтобы постоянно извиняться, когда что-то пошло не так, представители кол-центра должны извиняться в начале взаимодействия, затем переходить в «режим решения проблем», а потом заканчивать предложением клиенту нескольких вариантов.

Этот эксперимент проявляет целый ряд трудностей: из-за логистических ограничений мы сможем рандомизировать экспериментальную процедуру только на уровне кол-центров, а не представителей, и у нас возникнут трудности с обеспечением и измерением соблюдения требований. Это, конечно, не означает, что мы не можем или не должны проводить эксперимент!

Что касается рандомизационного ограничения, то мы увидим, что оно делает стандартный линейно-регрессионный алгоритм неуместным и что вместо него мы должны использовать иерархическое линейное моделирование (hierarchical linear modeling, аббр. HLM).

---

[1] «Извинений недостаточно» (*Sorry' Is Not Enough*), Harvard Business Review, январь-февраль 2018 года.

Как и прежде, наш подход будет таков:

- планирование эксперимента;
- определение случайного размещения и размера/мощности выборки;
- анализ эксперимента.

Планирование эксперимента

В этом разделе я кратко изложу нашу теорию изменения, чтобы предоставить вам некоторый необходимый контекст и поведенческую основу.

1. Сначала деловая цель и целевая метрика.
2. Далее определение нашего вмешательства.
3. Наконец, поведенческая логика, которая их соединяет.

Деловая цель и целевая метрика

Основываясь на статье в журнале HBR, наш критерий успеха, или целевая метрика, выглядит прямолинейно: удовлетворенность клиентов (CSAT), измеренная с помощью одновопросного опроса, проводимого по электронной почте после телефонного звонка. Однако через минуту мы увидим, что тут есть сложности, поэтому нам нужно будет вернуться к этой теме после обсуждения предмета тестирования.

Определение вмешательства

Тестируемая нами экспериментальная процедура будет заключаться в проверке того, что представители обучены новому своду стандартных операционных процедур и проинструктированы о его применении.

Первая трудность заключается в имплементировании экспериментальной процедуры. Из прошлого опыта мы знаем, что просить представителей применять разные стандартные операционные процедуры к разным клиентам очень сложно: если просить представителей произвольно переключать процессы между между телефонными звонками, то они будут увеличивать свою когнитивную нагрузку и риск несоблюдения требований. Поэтому нам придется обучить нескольких представителей и проинструктировать их использовать новый свод стандартных операционных процедур для всех их звонков, оставив при этом других представителей со старым сводом.

Даже при такой коррекции соблюдение требований остается под угрозой: представители в процедурной группе могут имплементировать новый свод бессистемно или даже вообще его не имплементировать, в то время как представители в контрольной группе тоже могут применять старый свод бессистемно. Очевидно, что это загрязнило бы наш анализ и сделало бы экспериментальную процедуру внешне менее отличимой от контрольной группы, чем она есть на самом деле. Один из путей смягчения этой проб-

лемы состоит в том, чтобы сначала наблюдать за текущей соблюдаемостью имеющегося свода, принимая звонки, затем провести пилотное исследование, в ходе которого мы отбираем нескольких представителей, обучаем их и прослеживаем за соблюдением ими нового свода. Постфактумный опрос представителей в пилотном исследовании поможет выявить недоразумения и препятствия на пути к соблюдаемости требований. К сожалению, как правило, невозможно добиться 100%-ной соблюдаемости в эксперименте, в котором люди доставляют или выбирают экспериментальную процедуру. Лучшее, что можно сделать, – это попытаться измерить соблюдаемость и принять ее во внимание при выведении заключений.

Наконец, существует риск «утечки» между нашей контрольной и нашей процедурной группами. Представители – это люди, а представители в данном кол-центре взаимодействуют и переговариваются. Учитывая, что представители стимулируются за свои звонки показателем среднемесячной удовлетворенности клиентов (CSAT), если представители в процедурной группе начали видеть значительно более высокие результаты, то существует риск того, что представители в контрольной группе того же кол-центра начнут изменять свою стандартную процедуру. Если несколько людей из контрольной группы применят экспериментальную процедуру, то это загрязнит сравнения двух групп и сделает разницу меньше, чем она есть на самом деле. Поэтому мы будем применять экспериментальную процедуру на уровне кол-центра: все представители в данном кол-центре будут либо в процедурной группе, либо в контрольной группе.

Применение экспериментальной процедуры на уровне кол-центра, а не на уровне телефонного звонка влияет на наш критерий успеха. Если нашей единицей рандомизации является кол-центр, то должны ли мы измерять удовлетворенность клиентов на уровне кол-центра? Это казалось бы логичным, но означало бы, что мы не можем использовать какую-либо информацию об отдельных представителях или отдельных звонках. С другой стороны, измерение средней удовлетворенности клиентов на уровне представителя или даже на уровне звонка позволило бы нам использовать больше информации, но это проблематично по двум причинам:

- во-первых, если бы мы проигнорировали тот факт, что рандомизация не проводилась на уровне звонков, и использовали стандартный анализ мощности, то наши результаты были бы систематически смещенными, поскольку рандомизация неизбежно коррелирует с переменной кол-центра; добавление большего числа звонков в нашу выборку не изменило бы тот факт, что у нас всего 10 кол-центров и, следовательно, только 10 единиц рандомизации;
- во-вторых, при анализе наших данных мы столкнулись бы с неприятностями из-за вложенной природы данных: если допустить, что каждый представитель принадлежит одному и только одному кол-центру, то будет наблюдаться мультиколлинеарность между нашей переменной кол-центра и нашей переменной представителя (например, мы можем добавить 1 в коэффициент для первого кол-центра и вычесть 1 из коэффициентов для всех представителей в этом кол-центре без

изменения результатов регрессии; поэтому коэффициенты регрессии, по существу, не решены).

К счастью, есть простое решение этой проблемы: мы будем использовать иерархическую модель, которая распознает вложенную структуру данных и обрабатывает ее надлежащим образом, позволяя нам использовать объясняющие переменные вплоть до уровня телефонного звонка[1]. Для наших целей мы не будем вдаваться в статистические подробности, а только посмотрим, как выполнять соответствующий исходный код и интерпретировать результаты. Иерархическая модель – это общий каркас, который может применяться к линейной и логистической регрессии, поэтому мы будем попрежнему находиться на известной территории.

Поведенческая логика

Наконец, логика успеха этого эксперимента проста: новый свод стандартных операционных процедур позволит клиентам чувствовать себя лучше во время взаимодействия, что приведет к более высокому уровню удовлетворенности клиентов (рис. 10.1).

Рис. 10.1 ❖ Причинно-следственная логика для нашего эксперимента

Данные и пакеты

Папка этой главы в репозитории на GitHub содержит два CSV-файла с переменными, перечисленными в табл. 10.1. Флажок (✓) обозначает переменные, присутствующие в этом файле, тогда как крестик (×) обозначает переменные, которых нет.

Обратите внимание, что два упомянутых набора данных также содержат двоичную переменную *6МесячныйРасход* (*M6Spend*), сумму, израсходованную на последующие бронирования в течение шести месяцев после данного бронирования. Эта переменная будет использоваться только в главе 11.

В этой главе мы будем использовать следующие ниже пакеты в дополнение к обычным:

```
## R
library(blockTools)  # Для функции block()
library(caret)       # Для функции кодирования с одним активным состоянием dummyVars()
library(scales)      # Для функции rescale()
```

[1] Если вы хотите узнать об этом типе моделей больше, то работа Гельмана и Хилла (2006) является классическим справочным материалом по данной теме.

```
library(lme4)        # Для иерархического моделирования
library(lmerTest)    # Для дополнительной диагностики для иерархического моделирования
library(nbpMatching) # Для использования алгоритма 'optimal' в стратифицированной
                     # рандомизации
library(binaryLogic) # For function as.binary()

## Python
# Для перешкалирования числовых переменных
from sklearn.preprocessing import MinMaxScaler
# Для преобразования категориальных переменных в кодировку с одним активным состоянием
from sklearn.preprocessing import OneHotEncoder"
```

Таблица 10.1. Переменные в наших данных

	Описание переменной	chap10-historical_data.csv	chap10-experimental_data.csv
Center_ID (ИД центра)	Категориальная переменная для 10 кол-центров	✓	✓
Rep_ID (ИД представителя)	Категориальная переменная для 193 представителей кол-центров	✓	✓
Age (Возраст)	Возраст звонящего клиента, 20–60	✓	✓
Reason (Основание)	Основание для звонка, «оплата»/«объект недвижимости» (payment/property)	✓	✓
Call_CSAT (Удовлетворенность клиента звонком)	Удовлетворенность клиента звонком, 0–10	✓	✓
Group (Группа)	Экспериментальное размещение, контрольная/процедурная группы (ctrl/treat)	×	✓

Введение в иерархическое моделирование

Иерархические модели можно использовать тогда, когда в ваших данных есть категориальные переменные:

- транзакции клиентов в нескольких магазинах;
- аренда объектов недвижимости в нескольких штатах
- и т. д.

В некоторых ситуациях требуются иерархические модели, и это связано с тем, что невозможно использовать традиционные категориальные переменные. Главная из них – это если у вас есть категориальная переменная, которая зависит от другой категориальной переменной (например, Вегетарианский = {«да», «нет»} и Вкус = {«ветчина», «индейка», «тофу», «сыр»}), иначе именуемые «вложенными» категориальными переменными. Тогда проблемы мультиколлинеарности делают иерархические модели самыми подходящими из возможных. Вот, кстати, почему они называются «иерархическими» моделями, хотя их можно применять и к невложенным категориям.

Помимо этого, иерархические модели также предлагают более устойчивую альтернативу, если у вас есть категориальная переменная с большим числом категорий, таких как ИД представителя кол-центра в нашем примере, и в особенности если в некоторых категориях очень мало строк в ваших данных. Если не вдаваться в подробности, то указанная устойчивость обусловливается тем, как коэффициенты в иерархических моделях встраивают некоторую информацию из других строк, что приближает их к совокупному среднему значению. Давайте вообразим, что в наших данных был представитель кол-центра, ответивший только на один телефонный звонок, с исключительно плохой удовлетворенностью клиента, что явно является выбросом. Имея только один звонок у этого представителя, мы не знаем, что конкретно является выбросом: представитель или же звонок. Категориальная переменная назначила бы 100 % «выброса» представителю, тогда как иерархическая модель разделила бы его между представителем и звонком, т. е. мы ожидали бы, что представитель будет иметь показатель удовлетворенности клиентов ниже среднего по сравнению с другими звонками, но не такой экстремальный, как наблюдавшийся звонок.

Наконец, в ситуациях, когда могут применяться и категориальные переменные, и иерархические модели (что в основном относится к любой ситуации, когда у вас есть категориальная переменная с несколькими невложенными категориями!), в интерпретации существует несколько нюансов, которые могут побудить вас предпочесть ту или иную. Концептуально категориальная переменная представляет собой разделение ваших данных на группы с внутренними различиями между ними, которые мы хотим понять, в то время как иерархическая модель трактует группы как случайные извлечения из потенциально бесконечного распределения групп. AirCnC имеет 30 кол-центров, но вместо этого могло быть 10 или 50, и нас не интересуют различия между кол-центром № 3 и кол-центром № 28. С другой стороны, мы хотели бы знать, имеют ли звонки по вопросам оплаты более высокий или более низкий средний показатель удовлетворенности клиентов, чем звонки по вопросам объектов недвижимости, и мы не были бы удовлетворены, просто зная, что стандартное отклонение между группами составляет 0.3. Но опять же, это нюансы интерпретации, так что не слишком об этом заморачивайтесь.

Исходный код на R

Давайте проведем обзор синтаксиса иерархического моделирования в простом контексте, обратившись к детерминантам удовлетворенности клиентов звонком в кол-центр в наших исторических данных, оставив пока в стороне переменную *Rep_ID*. Исходный код на R выглядит следующим образом:

```
## R
> hlm_mod <- lmer(data=hist_data, call_CSAT ~ reason + age + (1|center_ID))
> summary(hlm_mod)
Linear mixed model fit by REML. t-tests use Satterthwaite's method
   ['lmerModLmerTest']
```

```
Formula: call_CSAT ~ reason + age + (1 | center_ID)
   Data: hist_data

REML criterion at convergence: 2052855

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.3238 -0.6627 -0.0272 0.6351 4.3114

Random effects:
   Groups Name Variance Std.Dev.
   center_ID (Intercept) 1.406 1.186
   Residual 1.122 1.059
Number of obs: 695205, groups: center_ID, 10

Fixed effects:
                   Estimate    Std. Error         df      t value   Pr(>|t|)
(Intercept)       3.8990856     0.3749857      9.0938797   10.40 0.00000238 ***
reasonproperty    0.1994487     0.0026669 695193.0006122   74.79    < 2e-16 ***
age               0.0200043     0.0001132 695193.0008798  176.75    < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) rsnprp
reasnprprty  0.000
age         -0.011 -0.236
```

Синтаксис функции `lmer()` аналогичен синтаксису традиционной функции `lm()`, за одним исключением: нам нужно ввести переменную кластеризации, здесь `center_ID`, между двумя круглыми скобками и предваряемую символами `1|`. Это позволяет коэффициенту пересечения нашей регрессии варьироваться от одного кол-центра к другому. Следовательно, у нас есть один коэффициент для каждого кол-центра; вы можете смотреть на эти коэффициенты как на аналогичные коэффициенты, которые мы получили бы в стандартной линейной регрессии с фиктивной переменной для каждого кол-центра[1].

Раздел случайных эффектов (Random effects) результатов относится к переменной(ым) кластеризации. Коэффициенты для каждого ИД кол-центра в сводных результатах не показываются (к ним можно получить доступ с помощью команды `coef(hlm_mod)`). Вместо этого мы получаем меры вариабельности наших данных внутри кол-центров и между кол-центрами в виде дисперсии и стандартного отклонения. Здесь стандартное отклонение наших данных между кол-центрами составляет 1.185; другими словами, если бы мы рассчитали среднее значение удовлетворенности клиентов для каждого кол-центра, а затем рассчитали стандартное отклонение средних, то мы бы получили то же самое значение, что можно проверить самостоятельно:

---

[1] Если вы действительно хотите знать, то эти коэффициенты рассчитываются как средневзвешенное значение средней удовлетворенности клиентов (CSAT) в колцентре и средней удовлетворенности клиентов по всем нашим данным.

```
## R
> hist_data %>%
    group_by(center_ID)%>%
    summarize(call_CSAT = mean(call_CSAT)) %>%
    summarize(sd = sd(call_CSAT))
`summarise()` ungrouping output (override with `.groups` argument)
# Тиббл: 1 x 1
     sd
  <dbl>
1  1.18
```

Стандартное отклонение остатков, здесь 1.059, показывает, сколько вариабельности осталось в наших данных после учета эффекта кол-центров. Сравнивая два стандартных отклонения, мы видим, что эффекты кол-центров демонстрируют более половины вариабельности наших данных.

Раздел фиксированных эффектов (Fixed effects) результатов должен выглядеть знакомо: в нем указаны коэффициенты для переменных уровня звонка. Здесь мы видим, что у клиентов, обращающихся по вопросам объектов недвижимости, удовлетворенность клиентов в среднем на 0.199 выше, чем у клиентов, обращающихся по вопросам оплаты, и что каждый дополнительный год возраста наших клиентов добавляет в среднем 0.020 в удовлетворенность клиентов звонком.

Затем давайте включим в состав переменную `rep_ID` в качестве переменной кластеризации, вложенной в переменную `center_ID`:

```
## R
> hlm_mod2 <- lmer(data=hist_data,
                   call_CSAT ~ reason + age + (1|center_ID/rep_ID),
                   control = lmerControl(optimizer ="Nelder_Mead"))
> summary(hlm_mod2)
Linear mixed model fit by REML. t-tests use Satterthwaite's method
   ['lmerModLmerTest']
Formula: call_CSAT ~ reason + age + (1 | center_ID/rep_ID)
   Data: hist_data
Control: lmerControl(optimizer = "Nelder_Mead")

REML criterion at convergence: 1320850

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.0373  -0.6712 -0.0003 0.6708 4.6878

Random effects:
  Groups Name Variance Std.Dev.
  rep_ID:center_ID (Intercept) 0.7696 0.8772
  center_ID (Intercept) 1.3582 1.1654
  Residual 0.3904 0.6249
Number of obs: 695205, groups: rep_ID:center_ID, 193; center_ID, 10

Fixed effects:
                 Estimate   Std. Error            df   t value  Pr(>|t|)
(Intercept)    3.90099487   0.37397956      8.73974599   10.43 0.00000316 ***
```

```
reasonproperty  0.19952547   0.00157368 695010.05594912  126.79    < 2e-16 ***
age             0.01992162   0.00006678 695010.05053170  298.30    < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) rsnprp
reasnprprty  0.000
age         -0.007 -0.236
```

Хорошо видно, что это делается путем добавления `rep_ID` в качестве переменной кластеризации после `center_ID`, разделяя их символом /. Также обратите внимание на то, что я получал предупреждение о том, что модель не сошлась, поэтому изменил алгоритм-оптимизатор на `"Nelder_Mead"`[1]. Коэффициенты для фиксированных эффектов немного отличаются, но не настолько сильно.

Исходный код на Python

Хотя он и более лаконичен, исходный код на языке Python работает аналогично. Главное различие заключается в том, что группы выражаются с помощью `groups = hist_data_df["center_ID"]`:

```
## Python
mixed = smf.mixedlm("call_CSAT ~ reason + age", data = hist_data_df,
                    groups = hist_data_df["center_ID"])
print(mixed.fit().summary())
```

```
            Mixed Linear Model Regression Results
============================================================
Model:              MixedLM Dependent Variable: call_CSAT
No. Observations:   695205  Method:             REML
No. Groups:         10      Scale:              1.1217
Min. group size:    54203   Log-Likelihood:     -1026427.7247
Max. group size:    79250   Converged:          Yes
Mean group size:    69520.5
------------------------------------------------------------
                   Coef. Std.Err.    z    P>|z| [0.025 0.975]
------------------------------------------------------------
Intercept          3.899    0.335  11.641 0.000  3.243  4.556
reason[T.property] 0.199    0.003  74.786 0.000  0.194  0.205
age                0.020    0.000 176.747 0.000  0.020  0.020
Group Var          1.122    0.407
============================================================
```

[1] Как всегда при численном моделировании, ваш пробег может отличаться. Спасибо Джессике Якубовски (Jessica Jakubowski) за предложенную альтернативную спецификацию: `lmerControl(optimizer ="bobyqa", optCtrl=list(maxfun=2e5))`.

Коэффициенты для фиксированных эффектов (т. е. пересечения, основания для звонка и возраста) идентичны исходному коду на R. Коэффициент дисперсии случайного эффекта выражен внизу фиксированных эффектов. При 1.122 он слегка отличается от значения на R из-за различий в алгоритмах, но это не повлияет на коэффициенты, которые нас интересуют.

Использование вложенных переменных кластеризации также имеет другой синтаксис на Python. Нам нужно выражать вложенную переменную более низкого уровня в отдельной формуле («формуле компонентов дисперсии», от англ. словосочетания variance components formula, которое я сократил как vcf):

```
## Python
vcf = {"rep_ID": "0+C(rep_ID)"}
mixed2 = smf.mixedlm("call_CSAT ~ reason + age",
                     data = hist_data_df,
                     groups = hist_data_df["center_ID"],
                     vc_formula=vcf)
print(mixed2.fit().summary())

            Mixed Linear Model Regression Results
============================================================
Model:             MixedLM Dependent Variable: call_CSAT
No. Observations:  695205  Method:             REML
No. Groups:        10      Scale:              0.3904
Min. group size:   54203   Log-Likelihood:     -660498.6462
Max. group size:   79250   Converged:          Yes
Mean group size:   69520.5
------------------------------------------------------------
                   Coef. Std.Err.    z    P>|z| [0.025 0.975]
------------------------------------------------------------
Intercept          3.874    0.099  38.992 0.000  3.679  4.069
reason[T.property] 0.200    0.002 126.789 0.000  0.196  0.203
age                0.020    0.000 298.301 0.000  0.020  0.020
rep_ID Var         1.904    0.303
============================================================
```

Синтаксис формулы компонентов дисперсии немного эзотеричен, но интуиция проста. Сама формула представляет собой словарь, в котором каждая вложенная переменная служит в качестве ключа. Прикрепленное к каждому ключу значение обозначает, что конкретно мы хотим, чтобы эта переменная имела: случайный коэффициент пересечения либо случайный коэффициент наклона (случайный здесь означает «варьирующийся по категориям»). Случайное пересечение является эквивалентом иерархической модели категориальной переменной и выражается как `"0+C(var)"`, где `var` – это имя вложенной переменной, т. е. то же, что и ключ. Случайные наклоны выходят за рамки этой книги, но, например, если вы хотите, чтобы связь между возрастом и удовлетворенностью звонком имела разный наклон для каждого представителя, то формулой компонентов дисперсии будет `vcf = {"rep_ID": "0+C(rep_ID)", "age":"0+age"}`, без `C()` во втором случае.

Определение случайного размещения и размера/мощности выборки

Теперь, когда мы спланировали качественные аспекты нашего эксперимента, нам нужно определить случайное размещение, которое мы будем использовать, а также наши размер и мощность выборки. В двух предыдущих экспериментах (глава 8 и глава 9) у нас был некий размер целевого эффекта и статистическая мощность, и мы выбирали размер выборки соответствующим образом. Здесь мы добавим морщинку, допустив, что наши деловые партнеры готовы проводить эксперимент только в течение месяца[1], а минимальный обнаруживаемый эффект, который они заинтересованы улавливать, составляет 0.6 (т. е. они хотят убедиться, что у вас есть достаточно мощности для улавливания эффекта такого размера, но они готовы рискнуть в том, что размер эффекта будет ниже).

При таких ограничениях возникает вопрос: сколько мощности у нас есть, чтобы уловить разницу этой величины с этой выборкой? Другими словами, если допустить, что разница действительно равна 0.6, то какова вероятность того, что наше правило принятия решения придет к выводу, что экспериментальная процедура действительно лучше, чем контрольная?

Как упоминалось ранее, для анализа наших данных мы будем использовать иерархическую регрессию, и это немного усложнит наш анализ мощности, но давайте сначала проведем краткий обзор процесса случайного размещения.

Случайное размещение

Даже если мы заранее не знаем, какие клиенты будут звонить, для случайного размещения это не имеет значения, потому что мы будем делать это на уровне кол-центра. Следовательно, мы можем сделать это заранее, назначив контрольную и процедурную группы сразу. В подобного рода кластеризованных экспериментах стратификация особенно полезна, потому что у нас столь мало фактических единиц для рандомизации. Здесь мы рандомизируем на уровне кол-центров, поэтому мы хотели бы провести стратификацию на основе характеристик центров, таких как число представителей и средние значения метрик звонков. Соответствующий исходный код представляет собой прямолинейную версию исходного кода главы 9, разбавленную функцией подготовки данных и оберткой для функции разделения на блоки (пример 10.1).

Пример 10.1 ❖ Стратифицированное случайное размещение кол-центров

```
## R

# Функция для подготовки данных
strat_prep_fun <- function(dat){
```

[1] Разве не дрянь дело для вашего экспериментального дизайна? Полностью. Разве это нереально? К сожалению, абсолютно нет. Как мы говорили, когда я был консультантом, клиент всегда остается клиентом.

```
    # Извлекаем переменные уровня свойств
    dat <- dat %>%
        group_by(center_ID) %>% ❶
        summarise(nreps = n_distinct(rep_ID),
                  avg_call_CSAT = mean(call_CSAT),
                  avg_age = mean(age),
                  pct_reason_pmt = sum(reason == 'payment')/n()) %>%
        ungroup()

    # Изолируем разные компоненты наших данных
    center_ID <- dat$center_ID # Center identifier
    dat <- dat %>% select(-center_ID)
    num_vars <- dat %>%
        # Отбираем числовые переменные
        select_if(function(x) is.numeric(x)|is.integer(x))

    # Нормализуем числовые переменные ❷
    num_vars_out <- num_vars %>%
        mutate_all(rescale)

    # Кладем переменные снова вместе
    dat_out <- cbind(center_ID, num_vars_out) %>%
        mutate(center_ID = as.character(center_ID)) %>%
        mutate_if(is.numeric, function(x) round(x, 4)) # Округляем для читаемости
    return(dat_out)}

block_wrapper_fun <- function(dat){

    prepped_data <- strat_prep_fun(dat)

    # Получаем стратифицированное размещение
    assgt <- prepped_data %>% ❸
        block(id.vars = c("center_ID"), n.tr = 2,
              algorithm = "optimal", distance = "euclidean") %>%
        assignment()
    assgt <- assgt$assg$`1`
    assgt <- assgt %>%
        select(-'Distance')

    assgt <- as.matrix(assgt) %>% apply(2, function(x) as.integer(x))
    return(assgt)} ❹
```

❶ **Мы группируем по center_ID и резюмируем нашу переменную кластеризации: мы берем число представителей в разбивке по центру, вычисляем среднюю удовлетворенность звонком и возраст клиентов и определяем процент звонков, основанием для которых является оплата, 'payment'.**

❷ **Мы выполняем перешкалирование всех переменных кластеризации в интервал от 0 до 1.**

❸ **Мы применяем функцию block() из blockTools, используя алгоритм 'optimal' из пакета nbpMatching (при таком малом числе кол-центров мы можем позволить себе лишние вычисления).**

❹ **Мы извлекаем таблицу паросочетаний из результата на выходе из функции block().**

Результирующая таблица паросочетаний такова:

```
## R
     Treatment 1 Treatment 2
[1,]           2           3
```

```
[2,]          8          9
[3,]          7          6
[4,]          1          5
[5,]         10          4
```

Как упоминалось в предыдущей главе, в Python нет эквивалента пакету `block`, поэтому для данной цели мы будем использовать две функции, которые я описал в предыдущей главе, с небольшими корректировками (например, у нас нет категориальных переменных на уровне центра, поэтому нам не нужно их кодировать с одним активным состоянием):

```
## Python
def strat_prep_fun(dat_df):
    ...

def stratified_assgnt_fun(dat_df, K = 2):
    ...

stratified_assgnt_df = stratified_assgnt_fun(hist_data_df, K=2)
```

Анализ мощности

Использовать стандартную статистическую формулу для анализа мощности (в данном случае это была бы формула для Т-теста) было бы крайне ошибочным, поскольку она не учитывала бы существующую в данных корреляцию. Гельман и Хилл (2006) приводят несколько конкретных статистических формул для иерархических моделей, но я не хочу опускаться в кроличью нору накопления возрастающе сложных и узких формул. Как обычно, в качестве нашего защищенного от неосторожного обращения подхода к анализу мощности мы будем выполнять симуляции.

Давайте сначала определим нашу метрическую функцию:

```
## R
hlm_metric_fun <- function(dat){
    # Оцениваем процедурный коэффициент с помощью иерархической регрессии
    hlm_mod <- lmer(data=dat,
                    call_CSAT ~ reason + age + group + (1|center_ID/rep_ID)
                    ,control = lmerControl(optimizer ="Nelder_Mead")
                   )
    metric <- fixef(hlm_mod)["grouptreat"]
    return(metric)}

## Python
def hlm_metric_fun(dat_df):
    vcf = {"rep_ID": "0+C(rep_ID)"}
    h_mod = smf.mixedlm("call_CSAT ~ reason + age + group",
                        data = dat_df,
                        groups = dat_df["center_ID"],
                        re_formula='1',
                        vc_formula=vcf)
    coeff = h_mod.fit().fe_params.values[2]
    return coeff
```

Эта функция возвращает из нашей иерархической модели коэффициент для процедурной группы. Как мы делали в предыдущих главах, давайте теперь выполним симуляции относительно нашего анализа мощности, с которым вы, будем надеяться, уже знакомы. Единственное, здесь нам нужно принять во внимание одну дополнительную вещь – это то, что наши данные стратифицированы, иначе говоря, кластеризованы. Это имеет два последствия.

Первое состоит в том, что мы не можем просто случайно извлекать звонки из исторических данных. В нашем эксперименте мы ожидаем, что у каждого представителя будет почти одинаковое число звонков; с другой стороны, по-настоящему случайное извлечение будет генерировать некоторую значительную вариацию в числе звонков в расчете на одного представителя. Мы ожидаем, что представители будут обрабатывать около 1200 звонков в месяц; ситуация, когда один представитель обрабатывает 1000 звонков, а другой – 1400, гораздо более вероятнее при действительно случайном извлечении, чем в реальности. К счастью, с точки зрения программирования, это можно легко урегулировать, сгруппировав наши исторические данные на уровне кол-центра и представителя, перед тем как делать случайное извлечение:

```
## R
sample_data  %< %- filter(dat, month==m)  %> %dplyr::group_by(rep_ID) %>%
    slice_sample(n = Nexp) %>% dplyr::ungroup()

## Python
sample_data_df = sample_data_df.groupby('rep_ID').sample(n=Ncalls_rep)\
                               .reset_index(drop = True)
```

Использование перестановок, когда случайность «лимитирована»

Второе последствие относится к статистическому уровню и является более глубинным. Мы используем стратификацию для соединения похожих кол-центров в пары и назначаем по одному из каждой пары в контрольную группу, а другой – в процедурную группу. Это хорошо тем, что мы снижаем риск спутывания нашего анализа некоторыми характеристиками кол-центра. Но в то же время это приводит к фиксированному эффекту в наших симуляциях: предположим, что кол-центры 1 и 5 соединены в пару, потому что они очень похожи. Тогда, сколько бы симуляций мы ни проводили, одна из них будет в контрольной группе, а другая – в процедурной группе; мы сократили суммарное число возможных комбинаций. При полностью свободной рандомизации имеется $10!/(5! \cdot 5!) \approx 252$ разных размещения 10 кол-центров в равноразмерных экспериментальных группах, что уже не так много[1]. Со стратификацией существует всего 2^5 ≈ 32 разных размещения, потому что для каждой из пяти пар возможны два размещения: (контрольная группа, процедурная группа) и (процедурная группа, контрольная группа). Это озна-

[1] Восклицательный знак обозначает математический оператор факториала. Обратитесь к указанной в скобках странице Википедии (https://oreil.ly/I5PTW), если хотите понять лучше лежащую в его основе математику.

чает, что даже если бы вы выполняли 32 000 симуляций, вы бы увидели только 32 разных случайных размещения на уровне кол-центров. Более того, имея исторические данные всего за три месяца, мы можем генерировать только три совершенно разные (т. е. взаимоисключающие) выборки в расчете на представителя, в общей сложности 32 · 3 = 96 разных симуляций.

Это не означает, что мы не должны использовать стратификацию; напротив, стратификация тем более важна, чем меньше становится наша экспериментальная популяция! Однако из этого не следует, что в значительной степени бессмысленно и потенциально дезориентирующе выполнять много больше симуляций, чем у вас действительно разных размещений.

В целях понимания причины давайте воспользуемся метафорой: вообразите студента, который решает увеличить свой словарный запас перед тестом (например, тестом LSAT[1], сдаваемым перед поступлением в юридическое учебное заведение). Он покупает словарь для учащихся и планирует читать в нем определение случайного слова десять раз в день, пока не сделает это тысячу раз, чтобы выучить тысячу слов. Но вот в чем загвоздка: в его словаре всего 96 слов! Это означает, что сколько бы раз студент ни искал нужное слово, его словарный запас не сможет увеличиться более чем на 96 слов. Безусловно, в чтении определения слова более одного раза, чтобы понять и запомнить его лучше, есть своя ценность, но это отличается от чтения определений большего числа слов. Это также означает, что случайный просмотр определений является очень неэффективным образом действий. Гораздо лучше просто пройтись по 96 словам по порядку.

Указанная логика точно так же применима и к симуляциям: мы обычно черпаем из наших исторических данных наугад, чтобы построить набор симулированных экспериментальных данных, и мы (правильно) трактуем вероятность того, что несколько симуляций будут идентичными, как незначительную. В данном случае если бы у нас была сотня кол-центров, каждый из которых имел тысячу представителей и данные за десять лет, то мы могли бы уверенно просимулировать сотни или даже тысячи экспериментов, совершенно не беспокоясь. С нашим лимитированным числом кол-центров и представителей нам будет лучше, если мы будем систематически использовать лимитированное число возможностей.

Давайте посмотрим, как это сделать в исходном коде. У нас есть соединенные в пары кол-центры (см. рис. 10.2 в предыдущем подразделе), и нам нужно просмотреть 32 возможные перестановки этих пар. Первая пара состоит из кол-центров № 7 и № 2, поэтому в половине симуляций № 7 будет в контрольной группе и № 2 – в процедурной группе, тогда как в другой половине № 2 будет в контрольной группе и № 7 в процедурной группе и т. д. Таким образом, первая симуляция может иметь в качестве контрольной группы кол-центры (7, 9, 3, 10, 4), тогда как вторая симуляция имеет в качестве контрольной группы (2, 9, 3, 10, 4).

Мы используем трюк, который помогает нам легко проходить перестановки. На самом деле это не сложно, но он опирается на свойства двоичных чисел, которые не воспринимаются интуитивно, так что приготовьтесь и потерпите. Любое целое число может быть выражено по двоичному основанию

[1] LSAT от англ. Law School Admission Test. – *Прим. перев.*

в виде последовательности нулей и единиц. 0 равно 0, 1 равно 1, 2 равно 10, 3 равно 11 и т. д. Они могут дополняться нулями, чтобы иметь постоянное число цифр. Мы хотим, чтобы число цифр было равно числу пар, здесь 5. Это означает, что 0 равно 00000, 1 равно 00001, 2 равно 00010, а 3 равно 00011. Самое большое целое число, которое мы можем выразить с помощью 5 цифр, равно 31. Обратите внимание, что, и это не совпадение, включая 0 как 00000, мы можем выразить 32 разных целых числа с помощью 5 двоичных цифр и что 32 – это число перестановок, которые мы хотим имплементировать. Следовательно, мы можем решить, что первая симуляция, которую мы назовем «симуляция 00000», имеет в качестве контрольной группы (7, 9, 3, 10, 4) из рис. 10.2. После этого мы будем менять местами контрольную и процедурную группы в паре всякий раз, когда цифра, соответствующая паре в двоичной форме симуляционного числа, равна 1. Поэтому, например, для симуляции 10 000 мы поменяли бы местами кол-центры № 7 и № 2, получив контрольную группу (2, 9, 3, 10, 4). И вот где происходит волшебство: перейдя от 00000 к 11111, мы увидим все возможные перестановки 5 пар!

Исходный код для перестановок

Из-за различий в индексации между Python и R (в первом она начинается с 0, а во втором – с 1) исходный код на Python немного проще, поэтому давайте начнем с соответствующего фрагмента кода:

```
## Python
for perm in range(Nperm):
    bin_str = f'{perm:0{Npairs}b}' ❶
    idx = np.array([[i for i in range(Npairs)], ❷
                    [int(d) for d in bin_str]]).T
    treat = [stratified_pairs[tuple(idx[i])] for i in range(Npairs)] ❸

    sim_data_df = sample_data_df.copy()
    sim_data_df['group'] = 'ctrl'  ❹
    sim_data_df.loc[(sim_data_df.center_ID.isin(treat)),'group']\
        = 'treat
```

❶ **Мы конвертируем счетчик перестановок perm в двоичный строковый литерал. В Python это можно сделать несколькими способами. Я сделал это здесь с помощью F-строки, которая синтаксически оформляется через f'{exp}', где выражение exp вычисляется перед форматированием в виде строкового значения. В выражении переменная Npairs тоже находится между фигурными скобками, поэтому перед передачей в выражение она сначала оценивается; после этой первой оценки exp равно perm:05b. Первый член слева от двоеточия – это число для форматирования; буква после двоеточия указывает используемый формат, здесь b означает двоичный; число непосредственно слева от буквы указывает общее число используемых цифр (здесь 5); и, наконец, любой символ слева от этого числа должен использоваться для заполнения (здесь 0).**

❷ **Мы соотносим цифры двоичного строкового литерала со счетчиком пар внутри матрицы idx. Таким образом, после транспонирования на Python «00000» становится**

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 3 & 0 \\ 4 & 0 \end{pmatrix}.$$

❸ **Мы передаем строки матрицы idx в качестве индексов, чтобы указать, какой элемент каждой пары входит в процедурную группу. То есть, чтобы указать, что первый элемент первой пары должен попасть в процедурную группу, мы передаем [0, 0]. В перестановке 00000 мы всегда помещаем первый элемент каждой пары в процедурную группу. В последней перестановке, 11111, мы помещаем второй элемент каждой пары в процедурную группу, зеркально отражая размещение для 00000. Если взять усложненный пример, то для перестановочного числа 7, двоичный формат которого равен 00111, мы поместили бы в контрольную группу первый элемент для первых двух пар и второй элемент для последних трех пар.**

❹ **Наконец, мы обновляем наш набор симулированных экспериментальных данных, относя каждую строку матрицы либо в контрольную группу, либо в процедурную группу на основе ее ИД центра.**

Указанный процесс идентичен на языке R с некоторыми различиями в синтаксисе:

```
## R
permutation_gen_fun <- function(i, stratified_pairs){
    Npairs <- nrow(stratified_pairs)
    bin_str <- as.binary(i, n=Npairs) ❶
    idx <- matrix(c(1:Npairs, bin_str), nrow = Npairs)
    idx[,2] <- idx[,2] + 1 ❷
    treat <- stratified_pairs[idx] ❸
    return(treat)}
```

❶ **Конвертирование perm в двоичный формат на R выполняется с помощью функции as.binary(), которая в качестве первого аргумента принимает конвертируемое число, а в качестве второго – суммарное число цифр, которые мы хотим (т. е. число пар, здесь 5).**

❷ **Поскольку индексация начинается с 1, а не с 0, на R нам нужно прибавить 1 ко всем элементам второго столбца в матрице idx. Таким образом, для первой перестановки 00000, где первый элемент каждой пары входит в контрольную группу, матрица idx такова:**

$$\begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{pmatrix}.$$

Для перестановки 11111 второй столбец будет состоять из двоек, а для 00111 он будет 11222.

❸ **Мы передаем строки матрицы idx в качестве индексов, чтобы указать, какой элемент каждой пары входит в процедурную группу.**

Функция `permutation_gen_fun()` возвращает список ИД центров для процедурной группы, которые затем могут использоваться в функции случайного размещения.

Кривая мощности

Теперь, когда у нас есть решение проблемы лимитированных возможных выборок, мы можем вернуться к нашему анализу мощности. Вспомните, что деловые партнеры хотят проводить эксперимент не более месяца, а значит, объем выборки будет около 230 000 звонков. Вместо того чтобы вычислять необходимый размер выборки для порогового значения, равного 0.6 пункта удовлетворенности клиентов и желаемой мощности, нам нужно взять заданный размер выборки и рассчитать мощность, которую мы можем иметь для этого порогового значения.

Давайте сначала посмотрим на статистическую значимость. Вспомните, что в предыдущей главе наш оценщик был «недостаточно уверенным»: 90%-ный интервал уверенности содержал ноль более чем в 90 % случаев. Даже использование 40%-го интервала уверенности привело лишь к небольшому числу ложноположительных результатов. Здесь мы сталкиваемся с противоположной проблемой: наш оценщик «чрезмерно уверен», поскольку 90%-ный интервал уверенности содержит ноль гораздо реже, чем в 90 % случаев, и на самом деле он его не содержит совсем: наш охват равен нулю. На рис. 10.2 показаны 96 интервалов уверенности, ранжированных от самого низкого до самого высокого.

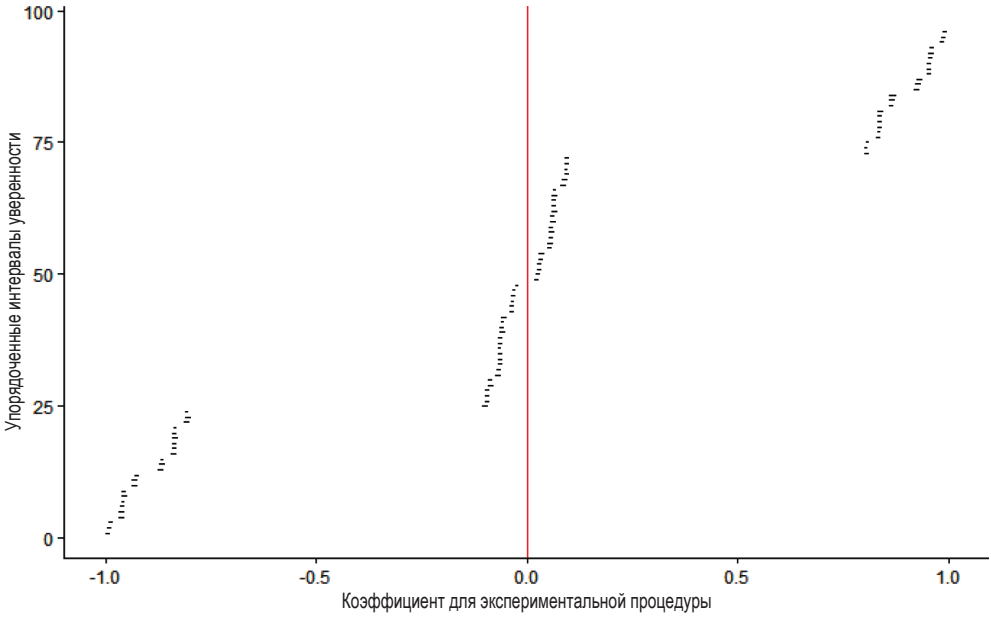

Рис. 10.2 ❖ 90%-ные интервалы уверенности без эффекта

Ситуация, которую мы видим на рис. 10.3, аналогична той, что мы видели в главе 7, где наличие очень лимитированных данных привело к разрывам в наших графиках. Здесь случайные ошибки ни разу не выстраиваются таким образом, чтобы привести к интервалу уверенности, содержащему ноль. Вместо этого у нас есть четыре плотных кластера интервалов уверенности, хотя распределение интервалов уверенности симметрично вокруг нуля (т. е. наша оценка не смещена), и половина из них очень близка к нему. С практической точки зрения это означает, что если мы проведем наш эксперимент, то мы не должны ожидать, что истинное значение будет включено в наш интервал уверенности.

Из этого не следует, что наш эксперимент обречен, но он означает, что мы не должны доверять границам интервалов уверенности и вместо этого должны опираться на наше правило принятия решения. С используемым по умолчанию таким правилом о принятии любого интервала уверенности,

который является строго положительным, наша значимость составляет 50 %: поскольку половина наших интервалов уверенности ниже нуля, а половина выше, в половине случаев мы бы наблюдали отрицательный коэффициент и справедливо пришли к выводу, что процедурная группа не лучше контрольной группы. На рис. 10.3 показана кривая мощности с этим правилом принятия решения для разных размеров эффекта.

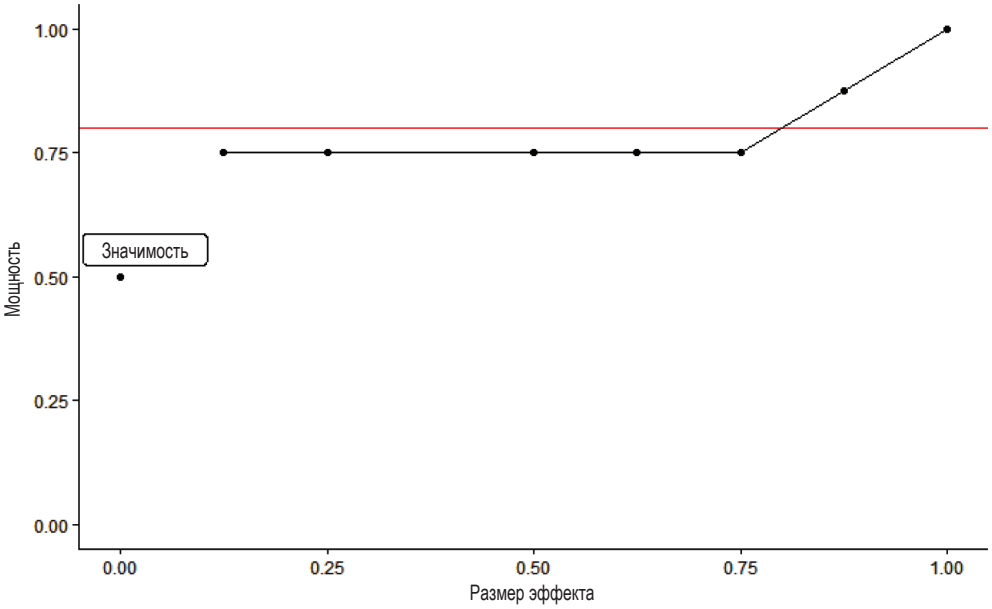

Рис. 10.3 ❖ Кривая мощности с порогом принятия решения, равным 0, для разных размеров эффекта

Как вы видите, наша мощность достигает 75 % очень быстро, в основном как только кластер интервалов уверенности, который был чуть ниже нуля, смещается чуть выше него. После этого наша мощность остается постоянной в интервале значений, включая наш пороговый размер эффекта 0.6, до тех пор, пока кластер сильно отрицательных интервалов уверенности, в свою очередь, не будет сдвинут выше нуля. Тогда наша мощность становится близкой к 100 % для размеров эффекта, равных 1 или выше. То есть если истинный эффект равен 1 или выше, то мы крайне маловероятно увидим отрицательный интервал уверенности.

Мы могли бы вернуться к нашим деловым партнерам и сказать им, что наши интервалы уверенности ненадежны и, следовательно, наш риск ложноположительных результатов велик, но наш риск ложноотрицательных результатов очень низок. В данном случае мы можем добиться большего успеха, установив более строгое правило принятия решения и применяя вмешательство только в том случае, если мы наблюдаем размер эффекта, равный 0.25 или выше.

На рис. 10.4 показана кривая мощности для этого правила принятия решения.

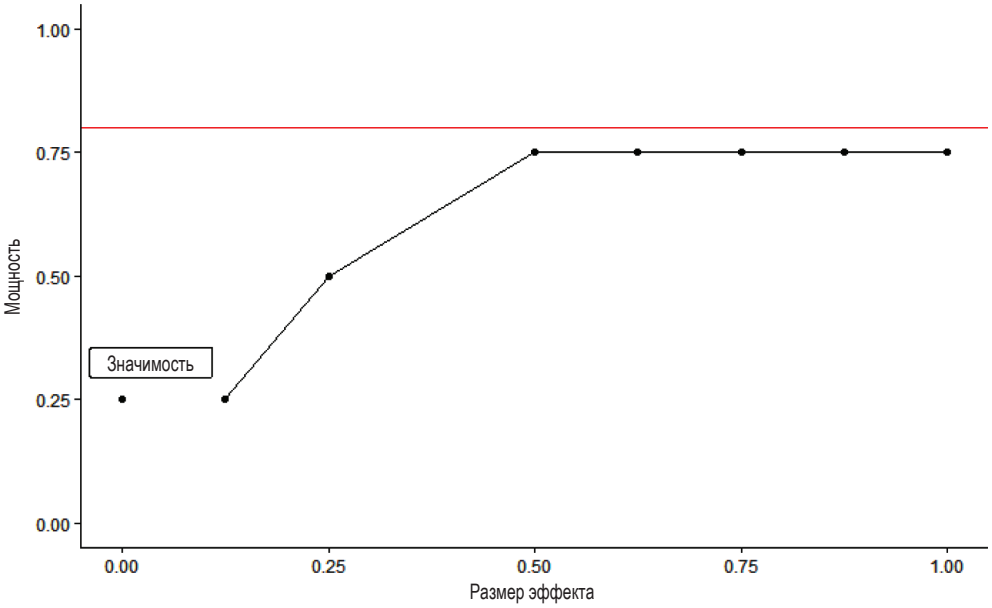

Рис. 10.4 ❖ Кривая мощности для разных размеров эффекта с порогом принятия решения, равным 0.25

Как мы видим на рис. 10.4, увеличив порог принятия решения, мы снизили левую часть кривой мощности. Из этого вытекает меньшая значимость (т. е. меньший риск ложноположительных результатов) за счет меньшей мощности (т. е. более высокий риск ложноотрицательных результатов) для небольших размеров эффекта. Однако правая часть нашей кривой мощности остается в основном неизменной, а значит, наша способность обнаружить эффект, равный 0.6, остается на уровне 75 %.

Давайте подытожим то, что сказал нам наш анализ мощности. Поскольку мы планируем использовать стратифицированное рандомизированное размещение с лимитированным числом эффективных экспериментальных единиц (т. е. кол-центров), наш эксперимент имеет жесткую структуру, которая ограничивает возможные исходы. Это делает наши интервалы уверенности ненадежными сами по себе. Однако мы можем скорректировать наше правило принятия решения на более высокий порог (т. е. мы будем имплементировать наше вмешательство только в том случае, если наблюдаем эффект, равный 0.25 или выше). Поступая таким образом, мы можем снизить риск ложноположительных результатов для нулевого размера эффекта, поддерживая при этом нашу мощность для целевого размера эффекта достаточно высокой. Этот эксперимент остается недостаточно эффективным, но это лучшее, что мы можем предложить как экспериментаторы, и нашим деловым партнерам придется решить, как относиться к таким шансам.

Обратите внимание на разницу между нашим порогом принятия решения 0.25 и нашим целевым эффектом 0.6. По определению, мощность на пороге принятия решения всегда равна 0.5, и мы поставили перед собой цель получить как можно больше мощности для размера эффекта 0.6.

Анализ эксперимента

Закончив проведение нашего эксперимента, мы сможем собрать и проанализировать данные. Определив ранее метрическую функцию, анализ теперь просто сводится к ее применению к экспериментальным данным, а затем получению бутстраповского 90%-го интервала уверенности ее значения для наших экспериментальных данных:

```
## R (результат не показан)
> coeff <- hlm_metric_fun(exp_data)
> print(coeff)
> hlm_CI <- boot_CI_fun(exp_data, hlm_metric_fun)
> print(hlm_CI)

## Python
coeff = hlm_metric_fun(exp_data_df)
print(coeff)
hlm_CI = boot_CI_fun(exp_data_df, hlm_metric_fun)
print(hlm_CI)
0.477903237163797
[0.47434045128179986, 0.4815858577196438]
```

Наш интервал уверенности очень узок и находится прямо выше 0.25. Основываясь на нашем анализе мощности, истинный размер эффекта вряд ли будет на самом деле в пределах этого интервала уверенности, но он, скорее всего, будет как ниже, так и выше, поэтому наш ожидаемый размер эффекта равен 0.48. Поскольку это превышает наш порог принятия решения, мы бы имплементировали вмешательство, даже если ожидаемый размер эффекта был меньше, чем наша цель. Интересно, что этот интервал уверенности намного меньше, чем тот, который мы получили бы на основе нормального приближения (т. е. коэффициента $\pm 1.96 \cdot$ коэффициент стандартной ошибки), отчасти из-за стратифицированной рандомизации.

Выводы

На этом мы завершаем наш тур по экспериментальному дизайну. В последней части книги мы увидим продвинутые инструменты, которые позволят нам разбираться в анализе экспериментальных данных глубже, но эксперимент с кол-центрами, который мы только что видели, является примерно таким же сложным, как эксперименты в реальной жизни. Невозможность рандомизации на самом низком уровне и наличие предопределенного количества времени для проведения эксперимента являются неприятными, но нередкими случаями. Рандомизация на уровне офиса или магазина вместо клиентов или сотрудников является обычным явлением, чтобы избежать логистических осложнений и «утечки» между экспериментальными группами. Задействование симуляций для анализа мощности и стратификации для случайного размещения становится практически неизбежным, если вы хотите

получить полезные результаты от своего эксперимента; будем надеяться, теперь вы должны быть подготовлены для этого полностью.

Разработка и проведение экспериментов, на мой взгляд, являются одной из самых увлекательных частей бихевиористики. Когда все идет хорошо, вы можете четко измерить влияние деловой инициативы или бихевиористического вмешательства. Но добиться того, чтобы все шло хорошо, – само по себе немалый подвиг. Популярные СМИ и коммерческие поставщики часто создают впечатление, что эксперименты бывают такими же простыми, как «включи и играй, проверь значимость на отметке 5 %, и все готово!». Но это заявление вводит в заблуждение, и я попытался устранить несколько неправильных представлений, которые возникают в результате этого.

Во-первых, статистическая значимость и мощность нередко понимаются неправильно, что может приводить к напрасным экспериментам и субоптимальным решениям. Я считаю, что отказ от *p*-значений в пользу бутстраповских интервалов уверенности приводит к результатам и интерпретациям, которые являются как более правильными, так и более релевантными применяемым условиям.

Во-вторых, рассматривать эксперименты как чистую технологическую задачу и задачу анализа данных проще, но менее плодотворно, чем применять причинно-поведенческий подход. Использование причинно-следственных диаграмм позволяет вам четче формулировать, что будет успехом и что заставляет вас верить, что ваша экспериментальная процедура будет успешной.

Проведение эксперимента в полевых условиях сопряжено с трудностями (дополнительные ресурсы см. в библиографии), и, к сожалению, каждый эксперимент отличается, поэтому я могу дать вам только несколько общих советов:

- проведение полевых экспериментов – это искусство и наука, и ничто не может заменить опыт в конкретном контексте. Сначала начните с более мелких и простых экспериментов;
- начните с имплементирования экспериментальной процедуры в малой экспериментальной группе, за которой вы затем некоторое время понаблюдаете и которую подробно опросите. Это позволит вам максимально убедиться в том, что люди эту процедуру понимают и применяют ее в какой-то мере правильно и последовательно;
- попытайтесь вообразить все возможные варианты того, как все может пойти не так, и их предотвратить;
- осознайте, что тем не менее все пойдет не так, и привнесите в свой эксперимент гибкость (например, запланируйте «буферы» времени, потому что все займет больше времени, чем вы думаете, – людям может потребоваться неделя, чтобы правильно имплементировать экспериментальную процедуру, данные могут прийти с опозданием и т. д.).

Часть V

ПРОДВИНУТЫЕ ИНСТРУМЕНТЫ АНАЛИЗА ПОВЕДЕНЧЕСКИХ ДАННЫХ

Это последняя часть книги, где все сходится воедино. Мы увидим три мощных инструмента анализа поведенческих данных: сначала модерация в главе 11, а затем опосредование и его производное – инструментальные переменные – в главе 12.

Модерация представляет собой универсальный математический инструмент, который позволяет нам понимать эффекты взаимодействия, а также эффективно и прозрачно сегментировать нашу клиентскую аудиторию. Опосредование позволяет нам заглядывать в черный ящик причинно-следственных связей и понимать принцип, по которому одна переменная влияет на другую. Наконец, я буду использовать инструментальные переменные, чтобы выполнить свое обещание измерить влияние удовлетворенности клиентов на последующие поведения клиентов.

Модерация, опосредование и инструментальные переменные – это мясистые темы, которые являются предметом оживленных методологических дебатов и целых книг. Но инструменты, которые мы представили в книге ранее, значительно облегчат это путешествие. Во-первых, использование причинно-следственных диаграмм даст интуитивные интерпретации для всех трех из них. Во-вторых, применение бутстрапа позволит нам строить интервалы уверенности напрямую и полностью обходить методологические осложнения, связанные с *p*-значениями. Как следствие мы сможем получать глубокие и действенные представления о поведении с помощью однострочного исходного кода.

Глава 11

Введение в модерацию

Один из наиболее приятных аспектов сочетания причинно-следственной и поведенческой точек зрения состоит в том, что вещи, которые, возможно, выглядят совершенно не связанными в рамках одной из них, могут оказаться совершенно одинаковыми в рамках другой. Проще говоря, когда у вас есть правильный молоток, многие вещи действительно являются гвоздями.

До сих пор мы использовали причинно-следственные диаграммы, чтобы понимать, что именно движет поведениями в среднем: если температура повышается на один градус, сохраняя все другие участвующие переменные постоянными, то на сколько увеличиваются продажи мороженого в киосках C-Mart? Но очень часто нас интересует не просто это общее среднее значение, а мы хотели бы разложить его дальше:

- распространяется ли это число в равной степени на киоски в Техасе и Висконсине? Если нет, то это означает, что наши данные показывают возможность для сегментации;
- относится ли это число в равной степени к шоколадному и ванильному мороженому? Если нет, то это означает, что существует взаимодействие между температурой и вкусом мороженого;
- применяется ли это число одинаково при низких и высоких температурах? Если нет, то это означает, что существует нелинейность во влиянии температур на продажи.

Молоток, который мы увидим в этой главе, социологами называется модерационным анализом, и он позволит нам ответить на эти три типа вопросов в точно таком же ключе.

В первом разделе после обзора данных и пакетов этой главы мы проведем экскурс по модерации и посмотрим, как она может применяться к различным поведенческим ситуациям. Поскольку математика остается во всех случаях неизменной, в последнем разделе я собрал все практические и технические соображения для проведения обзора.

Данные и пакеты

Папка этой главы в репозитрии на GitHub[1] содержит CSV-файл *chap11-historical_data.csv* с переменными, перечисленными в табл. 11.1.

Таблица 11.1. Переменные в наших данных

Имя переменной	Описание переменной
Day (День)	Индекс дня, 1–20
Store (Магазин)	Индекс магазина, 1–50
Children (Дети)	Двоичная 0/1, имеет ли покупатель с собой маленького ребенка
Age (Возраст)	Возраст покупателя, 20–80
VisitDuration (Продолжительность посещения)	Продолжительность посещения магазина в минутах, 3–103
PlayArea (Игровая площадка)	Двоичная 0/1, уровень магазина, имеет ли магазин игровую площадку
GroceriesPurchases (Покупки продовольственных товаров)	Сумма в долларах, израсходованная на покупки продовольственных товаров во время посещения, 0–324

В этой главе мы будем использовать только общие пакеты, поэтому специфичные для главы пакеты отсутствуют.

Поведенческие разновидности модерации

Формальное определение модерации является чрезвычайно простым: это включение в регрессию умножения между двумя предсказателями. Например, ранее я предположил, что продажи мороженого могут увеличиваться более или менее в зависимости от температуры в Техасе и Висконсине; это было бы выражено математически следующим образом:

$$\textit{ПродажиМороженого} = \beta_t \cdot \textit{Температура} + \beta_s \cdot \textit{Штат} + \beta_{ts} \cdot (\textit{Температура} \times \textit{Штат}).$$

Модерация может использоваться для понимания всех следующих ниже поведенческих явлений, которые мы рассмотрим по очереди:

- сегментация;
- взаимодействия;
- нелинейности (т. н. самомодерация).

Сегментация

Строительство релевантных клиентских сегментов является ключевой задачей маркетинговой аналитики и, в более широком смысле, деловой ана-

[1] См. https://oreil.ly/BehavioralDataAnalysisCh11.

литики. Мы проведем обзор выполнения этой задачи с наблюдательными данными, а затем с экспериментальными данными.

Сегментирование наблюдательных данных

Нашей отправной точкой будет пример C-Mart: эта компания недавно внедрила игровые площадки в некоторых своих магазинах и заинтересована в понимании того, как это повлияло на продолжительность посещения покупателями магазинов. Регрессионный анализ, подкрепленный причинно-следственными диаграммами, дает нам средний причинно-следственный эффект: каково влияние наличия игровой площадки в магазине на продолжительность его посещения по всем посетителям в наших данных? Однако средние значения могут дезориентировать и скрывать большие различия между сегментами нашей популяции. Например, имеет смысл допустить, что наличие игровой площадки в большей степени влияет на продолжительность посещения покупателями с детьми. Как это следует учитывать в нашей регрессии? Можно подумать, что простое включение *Детей* в качестве еще одного предсказателя *ПродолжительностиПосещения* достигло бы цели, как показано на рис. 11.1.

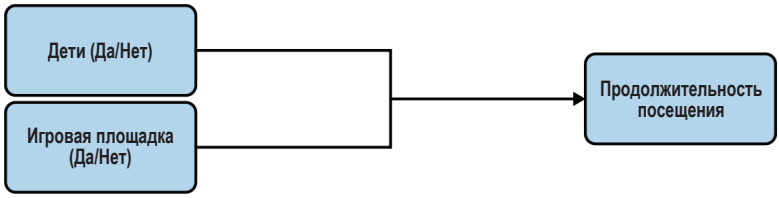

Рис. 11.1 ❖ Включение детей в качестве предсказателя *ПродолжительностиПосещения*

Проблема с таким подходом заключается в том, что он объясняет влияние *Детей* на *ПродолжительностьПосещения* вне зависимости от наличия или отсутствия игровой площадки: при выполнении регрессии каждая переменная инспектируется независимо. Каждый коэффициент задается таким, что совокупные расстояния остатков минимизируются, но коэффициенты должны быть одинаковыми для каждой переменной независимо от значений других переменных. Из этого вытекает, что коэффициент и, следовательно, измеряемый эффект *ИгровойПлощадки* на *ПродолжительностьПосещения* математически вынуждены быть средневзвешенным значением его эффекта для покупателей с детьми и без детей. Схожим образом эффект от покупателей, которые имеют с собой детей, измеряется как средневзвешенное значение эффектов, когда игровая площадка присутствует и когда ее нет. Это можно подтвердить, посмотрев на соответствующие уравнения. Уравнение для причинно-следственной диаграммы, которую мы только что начертили, показано в уравнении (11.1) (включая постоянный коэффициент β_0, который мы обычно для простоты опускаем).

$$\textit{ПродолжительностьПосещения} = \beta_0 + \beta_p \cdot \textit{ИгроваяПлощадка} + \beta_c \cdot \textit{Дети}. \quad (11.1)$$

Поскольку *ИгроваяПлощадка* и *Дети* являются двоичными переменными, у нас есть четыре возможных случая, в зависимости от того, равен ли каждый из них 0 или 1:

1) β_0 – это средняя продолжительность посещения магазина без игровой площадки покупателями без детей (для краткости мы будем сокращать как $C = 0, P = 0$);
2) $\beta_0 + \beta_c$ – это средняя продолжительность посещения магазина без игровой площадки покупателями с детьми ($C = 1, P = 0$);
3) $\beta_0 + \beta_p$ – это средняя продолжительность посещения магазина с игровой площадкой покупателями без детей ($C = 0, P = 1$);
4) $\beta_0 + \beta_p + \beta_c$ – это средняя продолжительность посещения магазина с игровой площадкой покупателями с детьми ($C = 1, P = 1$).

Из этого следует, что наличие или отсутствие у покупателя с собой детей не воздействует на влияние добавления игровой площадки, как можно легко проверить:

- если *Дети* = 0, то влияние добавления игровой площадки равно:

  ПродолжительностьПосещения($C = 0, P = 1$)
  – *ПродолжительностьПосещения*($C = 0, P = 0$) = $(\beta_0 + \beta_p) - (\beta_0) = \beta_p$;

- если *Дети* = 1, то влияние добавления игровой площадки равно:

  ПродолжительностьПосещения($C = 1, P = 1$)
  – *ПродолжительностьПосещения*($C = 1, P = 0$)
  = $(\beta_0 + \beta_c + \beta_p) - (\beta_0 + \beta_c) = \beta_p$.

Разница между этими двумя уравнениями (т. е. насколько больше добавление игровой площадки увеличивает продолжительность посещения магазина покупателями с детьми по сравнению с покупателями без детей) по определению равна:

$$[(\beta_0 + \beta_c + \beta_p) - (\beta_0)] - [(\beta_0 + \beta_p) - (\beta_0)] = \beta_p - \beta_p = 0.$$

Эквивалентный способ взглянуть на проблему состоит в том, что у нас есть четыре уравнения с четырьмя соответствующими средними значениями, но только с тремя коэффициентами. Если, задав β_0, β_c и β_p на основе первых трех уравнений, окажется, что средняя продолжительность посещения ($C = 1$, $P = 1$) не равна $\beta_0 + \beta_p + \beta_c$, то мы застрянем. Наш алгоритм сделает все возможное, чтобы отыскать значения, которые минимизируют в нашей регрессии ошибку, но наши оценки будут систематически смещенными. К сожалению, это именно тот случай, который мы пытаемся объяснить! Другими словами, простое добавление *Детей* в качестве переменной в нашу регрессию не объясняет взаимодействие между *Детьми* и *ИгровойПлощадкой*. С помощью этих уравнений мы не способны подтвердить, что наличие игровой площадки влияет на продолжительность посещения больше для покупателей с детьми.

Вот тут-то и выходит на первый план модерация. Мы можем решить нашу проблему, добавив четвертый коэффициент, как взаимодействие *ИгровойПлощадки* и *Детей* (показано в уравнении (11.2)).

$$\textit{ПродолжительностьПосещения} = \beta_0 + \beta_p \cdot \textit{ИгроваяПлощадка} + \beta_c \cdot \textit{Дети} + \beta_i \cdot (\textit{ИгроваяПлощадка} \times \textit{Дети}). \quad (11.2)$$

Уравнение для (C = 1, P = 1) становится $\textit{ПродолжительностьПосещения} = \beta_0 + \beta_p + \beta_c + \beta_i$, и мы можем скорректировать коэффициент β_i для объяснения эффекта взаимодействия. Теперь мы имеем:

- если *Дети* = 0, то влияние добавления игровой площадки равно $(\beta_0 + \beta_p) - (\beta_0) = \beta_p$;
- если *Дети* = 1, то влияние добавления игровой площадки равно $(\beta_0 + \beta_c + \beta_p + \beta_i) - (\beta_0 + \beta_c) = \beta_p + \beta_i$.

Разница между этими двумя уравнениями равна:

$$[(\beta_0 + \beta_c + \beta_p + \beta_i) - (\beta_0)] - [(\beta_0 + \beta_p + \beta_i) - (\beta_0)] = \beta_p + \beta_i - \beta_p = \beta_i.$$

Добавление игровой площадки увеличивает продолжительность посещения на β_i минут больше для покупателей с детьми, чем для покупателей без детей.

Мультипликативный член в уравнении (11.2) традиционно представляется на причинно-следственной диаграмме стрелкой, заканчивающейся посередине изначальной стрелки (рис. 11.2). В этом случае переменная *Дети* называется модератором, и взаимосвязь между *ИгровойПлощадкой* и *ПродолжительностьюПосещения* модерирована.

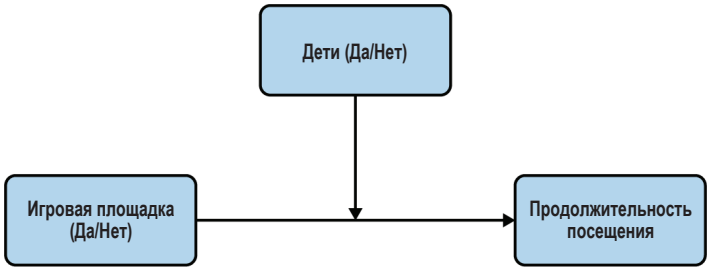

Рис. 11.2 ❖ Эффект *ИгровойПлощадки* на *ПродолжительностьПосещения* модерируется покупателями с детьми или без детей

Регрессионное программно-информационное обеспечение способно распознавать модерацию и обычно содержит укороченный путь: если вы включаете только произведение двух переменных, то программа распознает, что вы также хотите рассчитать коэффициенты для отдельных переменных:

```
## Python (результат не показан)
ols("duration~play_area * children", data=hist_data_df).fit().summary()

## R
> summary(lm(duration~play_area * children, data=hist_data))
...
Coefficients:
```

```
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         19.98760    0.01245  1605.5   <2e-16 ***
play_area1           3.95907    0.02097   188.8   <2e-16 ***
children1           10.01527    0.02017   496.6   <2e-16 ***
play_area1:children1 20.98663    0.03343   627.8   <2e-16 ***
...
```

При использовании описанной в предыдущем разделе укороченной формы выбранное вами программное обеспечение добавит *ИгровуюПлощадку* и *Детей* в качестве отдельных предсказателей *ПродолжительностиПосещения* и в результате вычислит уравнение (11.2). Не лучше ли вместо этого вычислять уравнение без детей как отдельный член? То есть:

$$\textit{ПродолжительностьПосещения} = \beta_0 + \beta_p \cdot \textit{ИгроваяПлощадка} + \beta_i \cdot (\textit{ИгроваяПлощадка} \times \textit{Дети}).$$

Короткий ответ: нет. Более подробную информацию по этому вопросу можно получить, обратившись к справочным материалам, таким как работа Хайеса (2017), но в сущности для того, чтобы коэффициенты означали то, что вы хотите, чтобы они означали, включайте в свою регрессию модератор и модерированные переменные в качестве отдельных переменных, даже если их коэффициенты не являются экономически или статистически значимыми. Не переопределяйте свое программное обеспечение путем их удаления; это не дефект, а функциональная особенность.

Коэффициенты в этой регрессии соответствуют средним значениям для четырех перечисленных нами случаев:

- средняя продолжительность посещения покупателем без детей магазина без игровой площадки составляет β_0, коэффициент пересечения, т. е. 20 минут;
- средняя продолжительность посещения покупателем без детей магазина с игровой площадкой составляет $\beta_0 + \beta_p$, сумма коэффициентов пересечения и *ИгровойПлощадки*, т. е. примерно 20 + 4 = 24 минуты;
- средняя продолжительность посещения покупателем с детьми магазина без игровой площадки составляет $\beta_0 + \beta_c$, сумма коэффициентов пересечения и *Детей*, т. е. примерно 20 + 10 = 30 минут;
- средняя продолжительность посещения покупателем с детьми магазина с игровой площадкой составляет $\beta_0 + \beta_c + \beta_p + \beta_i$, сумма коэффициентов пересечения, *ИгровойПлощадки*, *Детей* и члена взаимодействия между *ИгровойПлощадкой* и *Детьми*, примерно 20 + 4 + 10 + 21 = 55 минут.

Другими словами, наличие игровой площадки оказывает большое влияние на среднюю продолжительность посещения магазина покупателями с детьми и гораздо меньшее, но не пренебрежимо малое, влияние на среднюю продолжительность посещения магазина покупателями без детей (возможно, передышка от приступов детской истерики при покупке).

Это можно показать наглядно, как на рис. 11.3. Значения для *Детей* варьируются по оси *x*, а *ПродолжительностьПосещения* – по оси *y*, и у нас есть две линии, по одной для каждого возможного значения *ИгровойПлощадки*. Другими словами, четыре точки на концах двух линий демонстрируют че-

тыре случая, которые мы только что видели, и их значения y соответствуют коэффициентам.

Если бы не было эффекта взаимодействия между нашими двумя переменными, то две линии были бы параллельными, так как наличие игровой площадки сдвинуло бы среднюю продолжительность посещения вверх на тот же шаг приращения. Тот факт, что они не параллельны, демонстрирует, что *ИгроваяПлощадка* оказывает большее влияние на покупателей с детьми.

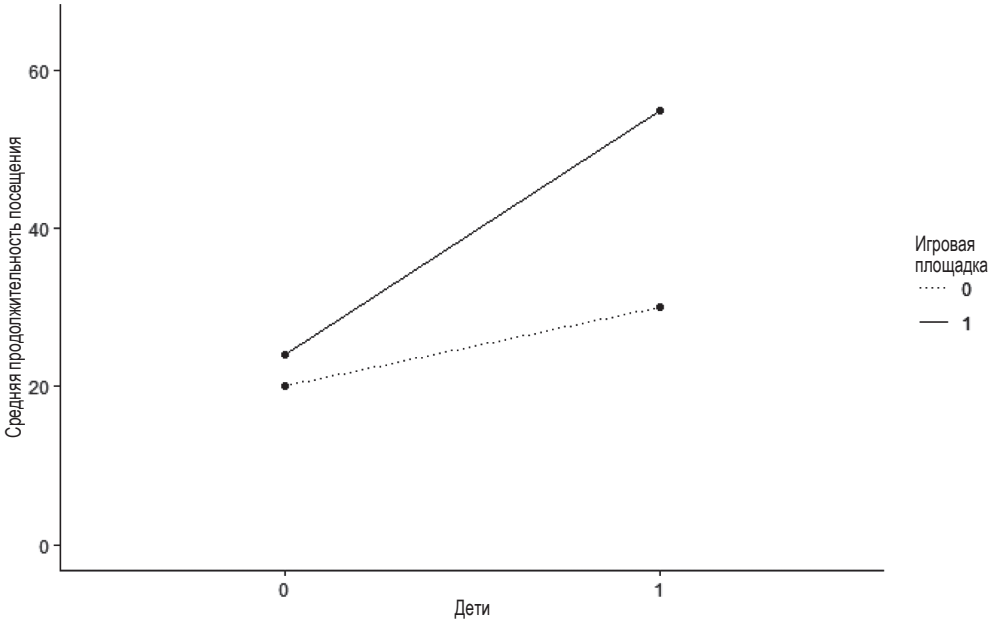

Рис. 11.3 ❖ Наглядная демонстрация модерации

Если вы хотите соединить это изображение с нашей математикой, то расстояние между двумя точками слева (для $C = 0$) равно β_p, тогда как расстояние между двумя точками справа ($C = 1$) равно $\beta_p + \beta_i$.

Выводы из модерационного анализа позволяют нам лучше нацеливать наши действия, например в случае если C-Mart хочет определить, в каких магазинах в следующий раз строить игровые площадки. Хотя отдельный коэффициент β_p релевантен для получения подтверждения, что игровая зона окупится или не окупится в целом, он совершенно не релевантен для приоритизации магазина. Выбор между магазинами требует определения характеристик магазина и покупателей, которые наиболее сильно взаимодействуют с переменной *ИгроваяПлощадка*, а затем определения магазина, который даст наибольший «подъем» от добавления игровой площадки.

В качестве альтернативы в магазинах, в которых уже есть игровая площадка, модерационный анализ позволил бы C-Mart определить, на каких потенциальных покупателей нацеливаться с использованием почтовой рассылки, расхваливающей наличие игровой площадки.

Сегментирование экспериментальных данных

Процесс сегментирования экспериментальных данных в значительной степени идентичен процессу, который мы только что использовали с наблюдательными данными. Поэтому я не буду повторять квантитативный анализ и просто укажу на несколько нюансов и тонкостей, которые следует иметь в виду.

Проводя эксперимент, мы часто заботимся не только об измерении среднего эффекта экспериментальной процедуры в нашей выборке, но и об определении групп, для которых эффект особенно силен или слаб. Отправка письма или обучение сотрудника может обходиться дорого; даже если финансовые издержки незначительны, как в случае с электронной почтой, могут возникать нематериальные издержки, такие как вызов раздражения у покупателя. Как следствие мы обычно хотим применять экспериментальную процедуру только к тем людям, для которых она работает.

Помимо стоимостных соображений, персонализация является ключевым основанием для нацеливания конкретных сообщений или экспериментальных процедур на конкретные сегменты нашей клиентской базы. Сама идея персонализированной передачи сообщений заключается в том, что люди за пределами целевого сегмента не будут реагировать или даже, возможно, будут реагировать на сообщение негативно. Если у вас есть сообщение, которое обращено ко всем, то это хорошо для вас, но это не персонализация. Например, купон на профессиональные коньки для катания на льду может раздражать подавляющее большинство людей, которым они не нужны. Реклама, нацеленная на тех, кто предпочитает активное времяпрепровождение на открытом воздухе, возможно, будет порождать антагонизм у тихих книжных червей, и наоборот. Персонализация означает, что вы обмениваете повышение эффективности в одной подгруппе на снижение эффективности в другой.

В маркетинговой аналитике этот подход часто называется анализом, или моделированием, «подъемной силы» (uplift), потому что мы пытаемся определить группы покупателей, для которых данная услуга или экспериментальная процедура повышает их предрасположенность к действию в наибольшей степени (например, покупке или голосованию), независимо от их первоначальной предрасположенности. Этот последний момент часто является источником путаницы, поэтому его стоит уточнить: выявление клиентов с высокой предрасположенностью может иметь свои преимущества, но само по себе оно не должно использоваться в качестве основы для нацеливания (т. н. таргетирования).

Предположим, вы сравниваете молодых клиентов (моложе 30 лет) и пожилых клиентов (старше 60 лет):

- у первой группы вероятность того, что она предпримет какие-либо действия, составляет 20 %, если им не будет направлено сообщение по электронной почте, и 40 %, если им будет направлено сообщение по электронной почте;
- у второй группы вероятность того, что она предпримет какие-либо действия, составляет 80 %, если им не будет направлено сообщение по

электронной почте, и 90 %, если им будет направлено сообщение по электронной почте.

Отправка сообщения по электронной почте первой группе, молодым клиентам, в среднем будет намного эффективнее, чем отправка сообщения по электронной почте второй группе, пожилым клиентам: она увеличит суммарное число клиентов, предпринимающих действия, на более крупное число.

Однако в реальной жизни этот факт нередко бывает затушеван отсутствием надлежащей контрольной группы. Если бы вы отправили электронное письмо только второй группе и сравнили их поведение с остальной частью популяции, то это значительно повысило бы очевидную эффективность рекламной кампании по электронной почте.

Математически выявление группы с высокой эффективностью экспериментальной процедуры выливается в отыскание демографических переменных, которые являются модераторами эффекта переменной экспериментальной процедуры на интересующем эффекте. В этом смысле модерационный анализ предлагает нам прочный и унифицированный концептуальный каркас для анализа подъемной силы, персонализации и в более общем случае марктетингового нацеливания.

Взаимодействия

Причинно-следственная диаграмма, которую мы начертили для изображения сегментации, была асимметричной: переменная *ИгроваяПлощадка* имеет стрелку, ведущую непосредственно к *ПродолжительностиПосещения*, тогда как переменная *Дети* имеет стрелку, ведущую к первой стрелке. Однако наше регрессионное уравнение является идеально симметричным; ничто не указывает на то, какая из двух переменных является модератором, а какая – модерируется. Технически говоря, мы могли бы интерпретировать уравнение (11.2) как означающее, что *Дети* являются причиной, а *ИгроваяПлощадка* – модератором.

Является ли одно из этих представлений более «верным», чем другое? Когда у нас в модерации участвует отдельная характеристика и деловая либо поведенческая переменная, как в данном случае, мы обычно представляем отдельную характеристику (например, наличие детей) в качестве модератора и ссылаемся на нее как на сегментацию. То есть эффект игровой площадки отличается между сегментом покупателей с детьми и сегментом покупателей без детей.

С другой стороны, если у нас есть модерация между двумя переменными одного и того же типа, такими как две отдельные характеристики или два деловых вмешательства, то нет смысла вводить между ними асимметрию. Например, можно вообразить, что наличие игровой площадки и наличие зоны отдыха увеличивают продолжительность посещения независимо (игровая площадка для покупателей с детьми, а зона отдыха для покупателей без детей), но наличие обеих вместе имеет больший эффект, чем сумма их отдельных эффектов, потому что покупатели с детьми теперь могут пользоваться зоной отдыха.

В таких обстоятельствах мы можем изобразить модерацию, соединив стрелки вместе, как на рис. 11.4, и ссылаться на нее как на взаимодействие.

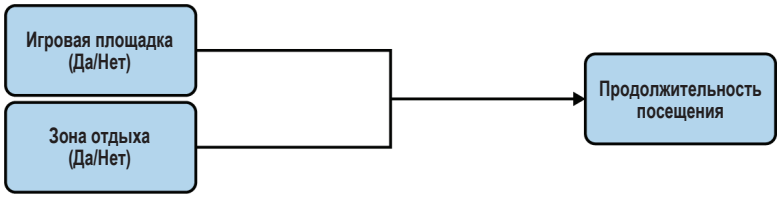

Рис. 11.4 ❖ Демонстрация симметричных взаимодействий

Уравнение для этой причинно-следственной диаграммы будет таким:

$$\begin{aligned} \textit{ПродолжительностьПосещения} = \beta_0 &+ \beta_p \cdot \textit{ИгроваяПлощадка} \\ &+ \beta_l \cdot \textit{ЗонаОтдыха} \\ &+ \beta_{pl} \cdot (\textit{ИгроваяПлощадка} \times \textit{ЗонаОтдыха}). \end{aligned}$$

Хорошо видно, что уравнение является в точности таким же, как и для сегментации, и если бы мы хотели, то могли бы назвать *ЗонуОтдыха* модератором эффекта *ИгровойПлощадки* на *ПродолжительностьПосещения* (или наоборот). Концептуально взаимодействия позволяют нам думать о ситуациях, в которых «целое – это больше, чем сумма частей», таких как комплементарные поведения.

В качестве замечания на полях: я считаю, что часть мощи новых и более сложных методов машинного обучения, таких как случайные леса, XGBoost или нейронные сети, заключается в их способности улавливать такие взаимодействия. Включая взаимодействия в регрессию, мы можем сокращать разрыв в результативности, сохраняя при этом интерпретируемость причинно-следственных связей, которые мы имеем с регрессией.

Нелинейности

Во многих обстоятельствах взаимосвязь между причиной и следствием не является линейной. Она может иметь то, что экономисты называют «уменьшающимися финансовыми возвратами»: вы получаете все меньше и меньше «ништяков за свою копеечку». Например, удовлетворенный покупатель может купить больше, чем неудовлетворенный, но восторженный покупатель может купить не намного больше, чем счастливый. Отправка покупателю одного маркетингового электронного письма в месяц может увеличить число покупок, но отправка 11 электронных писем вместо 10, вероятно, не сильно поможет, как показано на левой панели рис. 11.5.

И наоборот, причинно-следственная связь может иметь «возрастающие финансовые возвраты», такие как сетевые эффекты, расхваливаемые стартапами: чем больше объектов недвижимости AirCnC размещает на своем вебсайте, тем больше клиентов она привлечет; но тогда чем шире ее клиентская

база, тем выше стимул для собственников предлагать свою недвижимость на ее веб-сайте и т. д., как показано на правой панели на рис. 11.5.

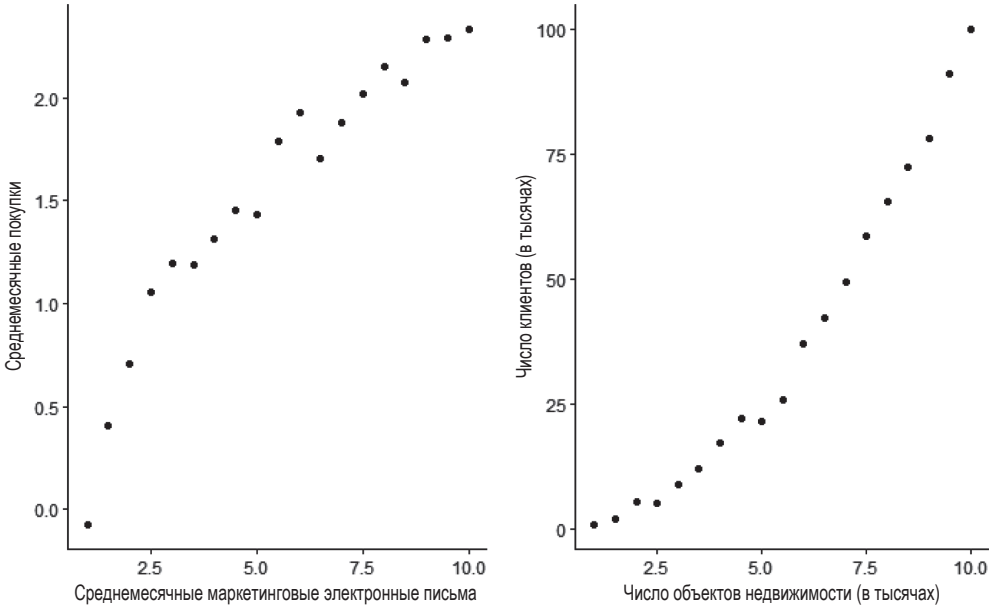

Рис. 11.5 ❖ Нелинейные взаимосвязи между переменными: слева уменьшающиеся возвраты, а справа увеличивающиеся возвраты

Математически кривая слева выпуклая, а кривая справа вогнутая. Распространенной ошибкой является мнение, что такие связи невозможно представить с помощью линейной регрессии. На самом деле для того, чтобы регрессия была «линейной», важно, чтобы предсказываемая переменная имела линейную связь не с переменными, а с коэффициентами. $Y = \beta_1 \cdot e^{X_1} + \beta_2 \cdot e^{X_2}$ является правильной линейной регрессией, потому что умножение каждого коэффициента на 2 умножит Y на 2. И наоборот, $Y = e^{\beta_1 \cdot X_1} + \beta_2 X_2$ не является правильной линейной регрессией, потому что Y не имеет линейной связи с коэффициентами.

Мы можем улаживать нелинейные взаимосвязи между переменными, добавляя объясняющую переменную, возведенную в квадрат (т. е. квадратичный член). Например, взаимосвязь между маркетинговыми электронными письмами и покупками, которые мы только что обсуждали, можно смоделировать как

$$Покупки = \beta_0 + \beta_1 \cdot ЭлектронныеПисьма + \beta_2 \cdot ЭлектронныеПисьма^2.$$

Добавление квадратичного члена может значительно повышать точность регрессии. На рис. 11.6 хорошо видно, что сплошная кривая, представляющая линию наилучшей подгонки для линейной регрессии с квадратичным членом, намного ближе к точкам данных, чем пунктирная прямая, представляющая стандартную регрессию без квадратичного члена. Здесь дополнительные ежемесячные электронные письма оказывают уменьшающееся

влияние, транслируясь в то, что квадратичный член имеет отрицательный коэффициент в регрессии.

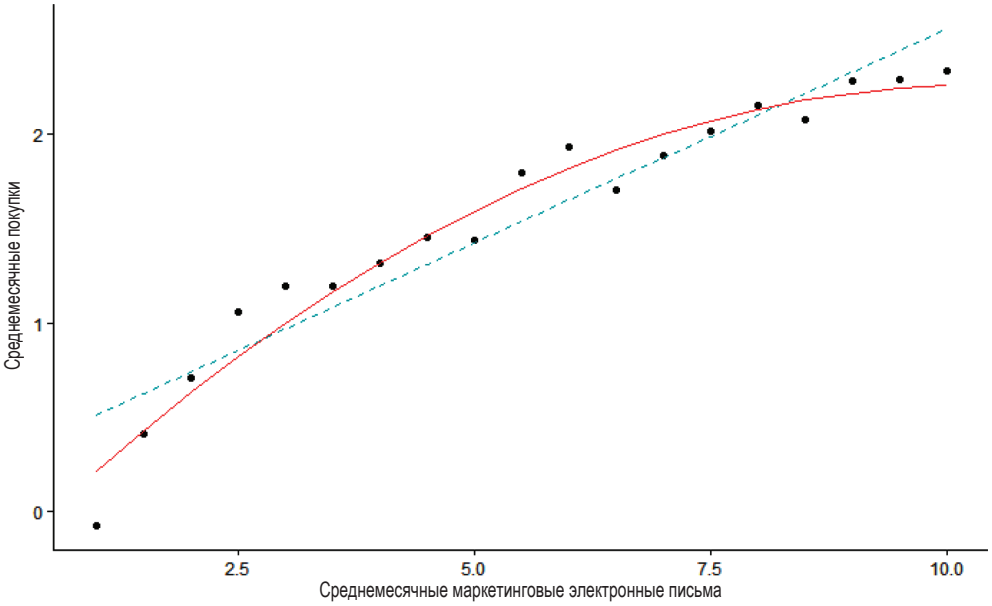

Рис. 11.6 ❖ Линейная (пунктирная) прямая и квадратичная (сплошная) кривая наилучшей подгонки

Однако квадратичный член – это не что иное, как взаимодействие переменной с самой собой. Другими словами, нелинейная причинно-следственная связь между двумя переменными может быть переформулирована как самомодерация (рис. 11.7).

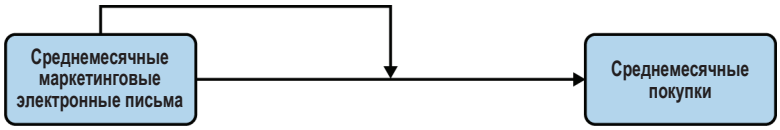

Рис. 11.7 ❖ Представление самомодерации

Концептуально это означает, что добавление лишнего электронного письма в месяц оказывает разное влияние в зависимости от текущего традиционного числа отправляемых электронных писем.

Синтаксис в исходном коде, содержащем самомодерацию, таков:

```
## R
summary(lm(Purchases ~ Emails + I(Emails^2), data=dat))

## Python
model = ols("Purchases ~ Emails + I(Emails**2)", data=dat_df)
print(model.fit().summary())
```

И на R, и на Python это делается с помощью функции тождественного отображения `I()`, которая предотвращает попытку линейно-регрессионного алгоритма интерпретировать квадратичный член и просто передает его общему решателю. На языке R квадратичный член выражается кареткой[1], тогда как на Python он выражается двумя символами умножения. Из этого также вытекает, что самомодерация подтверждается точно так же, как и традиционная модерация: мы строим бутстраповский интервал уверенности и определяем, содержит ли он ноль и является ли он экономически значимым.

Подводя итог нашему разведывательному анализу модерации, мы увидели три ее разновидности, которые математически идентичны, но имеют разные интерпретации, основанные на типах задействованных переменных:

сегментация.
: В рамках сегментации личностные характеристики (например, демографические переменные) модерируют эффект делового поведения, такого как экспериментальное вмешательство (и тогда это называется анализом подъемной силы);

взаимодействие.
: В рамках взаимодействия мы наблюдаем модерацию между переменными одной и той же природы, такими как две демографические или две поведенческие переменные;

нелинейности.
: В рамках нелинейностей переменная самомодерирует свое причинно-следственное влияние на другую переменную.

Теперь, когда у нас есть четкое понимание поведенческой интерпретации модерации, давайте обратимся к деталям ее применения.

Как применять модерацию

Как обсуждалось в предыдущем разделе, модерация может использоваться для улавливания различных поведенческих эффектов, просто добавляя произведение двух переменных в регрессию. В этом разделе мы обратимся к техническим соображениям:

- когда следует отыскивать модерацию;
- как ее подтверждать;
- модерированная модерация;
- как интерпретировать коэффициенты для отдельных переменных в модерируемой регрессии.

---

[1] Каретка – это символ на клавише с цифрой 6 на вашей клавиатуре.

Когда следует искать модерацию?

При таком большом числе возможных применений модерации может возникнуть соблазн отыскивать ее повсюду, но как второпорядковый эффект (т. е. эффект на эффект) модерация обычно порождает малые коэффициенты, и риск ложноположительных результатов высок. Это особенно верно в отношении экспериментальных данных, где сильна мотивация отыскивать содержательный эффект: «Понятное дело, средний эффект от рекламной кампании по электронной почте почти равен нулю, но посмотрите на частоту ответов тридцатилетних мужчин в Канзасе!»

Допустим, вы начинаете анализировать наблюдательные данные или работаете над дизайном эксперимента. В какой момент вам следует задуматься о модерации, и как вы должны интегрировать ее в свой анализ? Сначала я расскажу о стадии экспериментального дизайна. Процесс на стадии анализа данных одинаков независимо от того, носят ли данные наблюдательный или экспериментальный характер, поэтому после этого я рассмотрю два случая вместе. Наконец, я расскажу о нелинейностях, что будет легко: поскольку речь идет только об одной переменной, вам в принципе не нужно беспокоиться о потенциальных рисках, и вы можете свободно включать нелинейность в свой анализ.

Включение модерации на стадии экспериментального дизайна

Я буду различать две ситуации, которые потребуют разных подходов:

- первичным объектом вашего анализа является главный эффект, независимо от модерации, а модерация является вторичным объектом вашего анализа;
- модерация является первичным объектом вашего анализа.

Если вы работаете над дизайном эксперимента, в котором первичной целью не является измерение модерации, то моя рекомендация будет простой: используйте возможность модерации для уточнения своей теории изменения, но не пытайтесь корректировать размер выборки.

В части IV мы увидели, что перед проведением эксперимента вы должны четко формулировать свою теорию изменения с помощью причинно-следственных диаграмм. В данном случае все внимание сосредоточивается на средних причинно-следственных эффектах, то есть на среднем эффекте экспериментальной процедуры по всем направлениям, но во многих случаях мы можем эту логику усовершенствовать с помощью модерации.

В главе 8 мы разработали эксперимент, в ходе которого стремились повысить частоту бронирования в AirCnC, предложив «бронирование в 1 клик». Вполне возможно, что этот эффект мог бы быть промодерирован (рис. 11.8).

Поведенческая логика нашей теории изменения заключалась в том, что кнопка выполнения в 1 клик сократит продолжительность процесса бронирования, что само по себе влияет на вероятность завершения бронирования. Это означает, что с точки зрения модерации есть две возможности: возраст

модерирует эффект кнопки выполнения в 1 клик на продолжительность бронирования, либо он модерирует эффект продолжительности бронирования на вероятность завершения бронирования, либо и то, и другое (рис. 11.9).

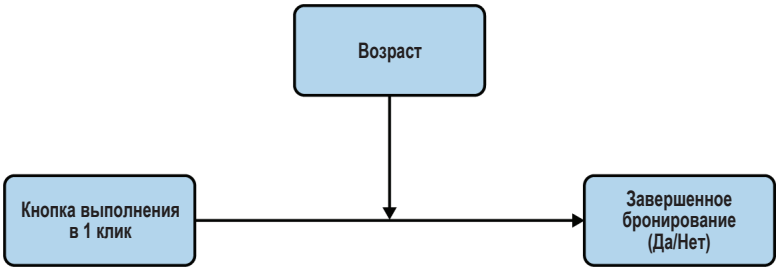

Рис. 11.8 ❖ Экспериментальная процедура, модерированная по возрасту

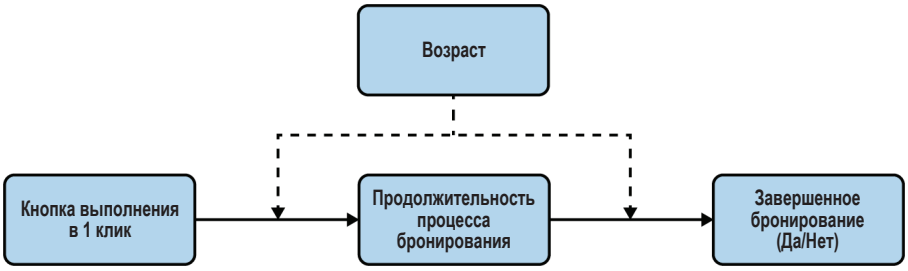

Рис. 11.9 ❖ Возраст может модерировать две причинно-следственные связи

Сведение воедино нашей поведенческой логики и потенциального модератора несет очень мощный потенциал, потому что оно позволяет нам продумывать поведенческие последствия, а в некоторых случаях выполнять аналитические расчеты до того, как состоится наш эксперимент.

Давайте начнем с первой связи слева от причинно-следственной диаграммы, между кнопкой выполнения в 1 клик и продолжительностью бронирования. У нас в наших исторических данных нет никаких данных о кнопке выполнения в 1 клик, поэтому мы не сможем измерить этот модерационный эффект напрямую. Однако если в наших исторических данных мы видим (неспутанную) корреляцию между возрастом и продолжительностью бронирования, то это придает некоторую достоверность идее, поскольку она намекает на то, что на участвующие в игре когнитивные процессы влияет возраст. С другой стороны, если возраст не коррелирован с продолжительностью бронирования, то модерация будет опираться на то, что молодые и пожилые клиенты реагируют на саму кнопку по-разному. Хотя и не невозможный, этот путь, будучи «более узким» поведенческим путем, в какой-то мере менее вероятен в соответствии с бритвой Оккама – самое простое объяснение, как правило, является правильным. Что еще важнее, так это тестируемый поведенческий путь. Например, если вы принесете выборку бэбибумеров в лабораторию исследования обыта пользователей (UX) и обнаружите, что

они не доверяют процессу выполнения в 1 клик и хотят пройти все его шаги, то модерация является весьма вероятной, и вы можете нацелить свой эксперимент на более молодых клиентов или, по меньшей мере, провести их избыточный отбор.

Что касается второй связи, то между продолжительностью бронирования и завершением бронирования у нас в наших исторических данных есть все необходимые переменные, а значит, мы можем подтвердить или отклонить модерацию еще до выполнения нашего эксперимента с гораздо большей точностью, чем при лимитированном размере выборки одного эксперимента. Опять же, если мы подтвердим наличие модерации, то мы можем скорректировать наш экспериментальный дизайн соответствующим образом (например, нацеливаясь только на более молодых или пожилых клиентов, в зависимости от направления модерации).

Здесь есть более широкий вывод: формулируя поведенческую логику вмешательства, мы нередко можем расширять цепочку между нашим вмешательством и интересующим нас эффектом на причинно-следственной диаграмме, выявляя между ними одного или нескольких посредников, по которым у нас уже есть данные. Затем мы можем провести разведку на предмет модерирования взаимосвязи между посредником(ами) и окончательным эффектом. В дополнение к этому мы можем разведать эту взаимосвязь на предмет самомодерирования. Здесь в нашем примере, возможно, сумма брони до определенного момента остается без воздействия на нее со стороны продолжительности бронирования, но затем она резко падает, например клиентам на самом деле все равно, если бронирование занимает 30 или 45 секунд, но они массово выходят из процесса, если он занимает более 2 минут (рис. 11.10).

Рис. 11.10 ❖ Взаимосвязь между посредником и окончательным эффектом самомодерируется

Во время указанного процесса выявления модераторов в ваших исторических данных вам придется иметь в виду, что тут применимо правило «корреляция не есть каузация», и рассматриваемые вами связи могут быть спутанными, как показано на рис. 11.11.

В такой ситуации спутывающий эффект дохода будет смещать взаимосвязь между *ПродолжительностьюБронирования* и *ЗавершеннымБронированием* по сравнению с реальным причинно-следственным эффектом (в соответствии с которым вы будете действовать посредством эксперимента с кнопкой выполнения в 1 клик). Следовательно, во время оценивания модерации вам нужно будет контролировать спутывающий эффект.

В случае переменной наподобие *Возраста*, которая наблюдалась до нашего экспериментального размещения, мы можем использовать ее непо-

средственно для самого размещения. Например, вместо того чтобы брать случайную выборку из всей нашей клиентской базы, мы можем ограничить наш эксперимент клиентами младше определенного возраста, где мы ожидаем наибольшего эффекта.

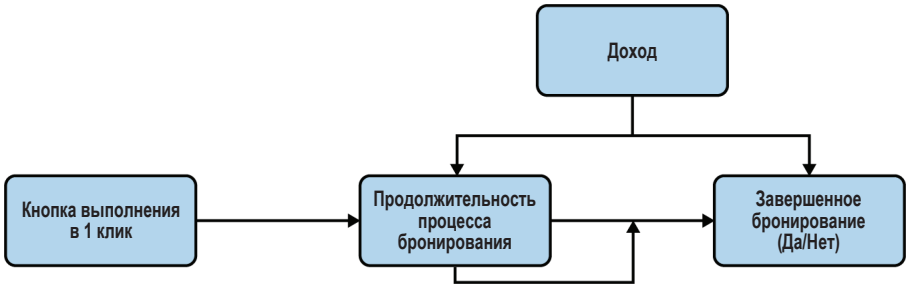

Рис. 11.11 ❖ Доход является спутывающим фактором взаимосвязи между продолжительностью бронирования и суммой брони

Очевидно, что мы не можем наблюдать *ПродолжительностьБронирования* до того, как клиент начнет процесс бронирования, а значит, мы не сможем использовать ее в самом экспериментальном размещении. Тем не менее мы можем попытаться выявить косвенные индикаторы для целевой переменной: например, может быть, что в выходные дни бронирования тяготеют к тому, чтобы быть продолжительнее, чем в будние дни. Затем мы можем использовать эти косвенные индикаторы для определения нашей экспериментальной популяции и, скажем, проводить наш эксперимент только с клиентами выходного дня. Это будет хорошо работать, если косвенный индикатор и целевая переменная тесно связаны, например в выходные дни также есть несколько более коротких бронирований, но 90 % более продолжительных бронирований приходятся на выходные. И наоборот, если эти две переменные связаны лишь слабо, например 60 % более продолжительных бронирований приходятся на выходные, то наблюдаемые в вашем эксперименте результаты, возможно, будут не очень хорошо обобщаться на более продолжительные бронирования в будние дни.

Это все, что я рекомендую вам сделать, если модерация не является первичным акцентом вашего эксперимента, т. е. вы еще не измерили немодерируемый эффект. Это связано с тем, что способность обнаруживать модерацию и взаимодействия значительно ниже, чем способность обнаруживать главные эффекты, а это означает, что размер выборки, необходимый для измерения модерации, будет значительно больше (думаю, в 10–20 раз больше или больше того). Это уйма времени, израсходованного на возможность модерации для среднего эффекта, который вы еще даже не подтвердили.

Поэтому я рекомендую вам провести свой первый эксперимент с акцентом на точное измерение среднего немодерируемого эффекта, а затем измерить модерационные эффекты в результатах. Если вы обнаружите какой-либо многообещающий модерационный эффект, то он, скорее всего, будет иметь большой интервал уверенности, который пересекает ноль, вследствие

того, что эксперимент не был организован для его улавливания. Если этот модерационный эффект выглядит достаточно многообещающим с экономической точки зрения, то проведите второй эксперимент, чтобы надлежаще его измерить.

Узнав размер немодерируемого эффекта, вы можете принять решение о проведении эксперимента, посвященного измерению или подтверждению модерации. Процесс определения надлежащего размера выборки остается прежним: вы устанавливаете целевую мощность и размеры гипотетического/целевого эффекта, а затем определяете частоту ложноотрицательных результатов в многократных симуляциях при разных размерах выборки, как мы делали в предыдущих аналитических расчетах мощности. Единственное отличие заключается в том, что вы будете использовать свои предварительные знания для определения размера главного эффекта и устанавливать целевой размер эффекта только для модерации.

Включение модерации на стадии анализа данных

У вас на руках есть немного данных, наблюдательных по своей природе либо полученных в результате проведенного вами эксперимента. Как выявить надлежащие модерационные эффекты, не сталкиваясь с ложноположительными результатами? Вам придется пойти заняться зондированием почвы, то есть попробовать включить модерационный член с одной переменной, затем с другой и т. д. Ниже я приведу несколько руководящих рекомендаций по минимизированию риска ложноположительных результатов.

Сначала я предоставлю вам грубую, но мощную проверку исправности, когда вы надеетесь модерировать влияние категориальной переменной на числовую переменную. Необходимое условие числового эффекта просто означает, что вы обращаетесь к линейной регрессии, а не к логистической. Необходимое условие категориальной причины, возможно, покажется очень ограничительным, но оно применимо ко всем экспериментальным размещениям (т. е. экспериментальное размещение всегда является двоичной либо категориальной переменной). В дополнение к этому если ваша причина является числовой, то вы можете ее дискретизировать для этой цели, взяв ее квартили:

```
## R
hist_data <- hist_data %>% mutate(age_quart = ntile(age, 4))

## Python
hist_data_df['age_quart'] = pd.cut(hist_data_df['age'], 4,
                                   labels=['q4', 'q3', 'q2', 'q1'],
                                   include_lowest=True)
```

Проверка исправности заключается в сравнении стандартного отклонения интересующего эффекта по группам, определяемым интересующей вас причиной. Если стандартное отклонение содержательно выше в процедурной группе (для экспериментальных данных) или отличается по всем группам (для данных результатов наблюдений), то это говорит о возможном наличии модерации, и вы можете продолжить свое зондирование почвы с разумной

уверенностью. Если стандартные отклонения по всем группам похожи, то это говорит о том, что модерации нет; вы все еще можете попробовать несколько потенциальных модераторов, если хотите, но у вас должно быть сильное теоретическое обоснование для них.

Что в этом контексте означает «содержательно выше» или «похожий»? Если вы хотите иметь строгое обоснование, то вы можете выполнить статистические тесты, которые помогут вам определить наличие или отсутствие статистической необычности наблюдаемой разницы, например используя тест Брауна–Форсайта[1]. Лично я бы рекомендовал просто пристально приглядеться к наличию или отсутствию экономически содержательной разницы по отношению к разнице в средних по всем группам.

Возвращаясь к примеру игровых площадок в магазинах C-Mart, исходный код будет выглядеть следующим образом:

```
## R (результат не показан)
> hist_data %>% group_by(play_area) %>% summarize(mean = mean(duration),
                                                  sd = sd(duration))

## Python
hist_data_df.groupby('play_area').agg(M = ('duration', lambda x: x.mean()),
                                      SD = ('duration', lambda x: x.std()))
Out[22]:
                   M         SD
play_area
0          23.803928   6.970786
1          36.360939  17.111469
```

В приведенном выше примере 10-минутная разница в стандартных отклонениях – это определенно то, что C-Mart было бы интересно разведать, учитывая, что разница в средних значениях по всем группам составляет около 13 минут.

Очевидно, что риск ложноположительных результатов возрастает с увеличением числа категорий. Если вашей первичной причиной является штат или профессия, то вы, скорее всего, увидите некоторые вариации в стандартных отклонениях по всем направлениям, просто из-за случайности и особых случаев. Поэтому я бы рассматривал модерацию для такой переменной только в том случае, если бы у меня в первую очередь было довольно сильное основание.

Смягчить риск ложноположительных результатов и сделать любую модерацию, которую вы можете отыскать, более содержательной можно путем замены вашей категориальной переменной квартилями релевантной числовой переменной. Например, политические взгляды или средний доход по штатам, доля женщин или средний уровень профессионального образования. Высказывание о том, что 25 % штатов с самым низким средним доходом имеют более низкое стандартное отклонение покупок, чем 25 % штатов с самым высоким средним доходом, является гораздо более устойчивым и действенным пониманием, чем высказывание о том, что в Калифорнии стандартное отклонение выше, чем в Миссисипи.

---

[1] Доступен в R в качестве функции `bf.test()` из пакета `onewaytests` (https://oreil.ly/iq7KN) и в Python в качестве функции `stats.level()` из пакета `scipy` с параметром `center='median'`.

Если допустить, что ваши данные проходят эту первую проверку исправности, то вторым шагом будет установление верхних границ на модерируемых эффектах. Ключевая интуиция здесь заключается в том, что модерация может только «перераспределить» средний эффект; она его не увеличивает. Наглядная иллюстрация сделает это яснее, взяв пример *Детей* в качестве потенциального модератора влияния *ИгровойПлощадки* на *ПродолжительностьПосещения*. На рис. 11.12 показан средний эффект игровой площадки по всем нашим данным (11.92 минуты), при этом ширина прямоугольников отражает долю покупателей без детей и с детьми в нашей популяции.

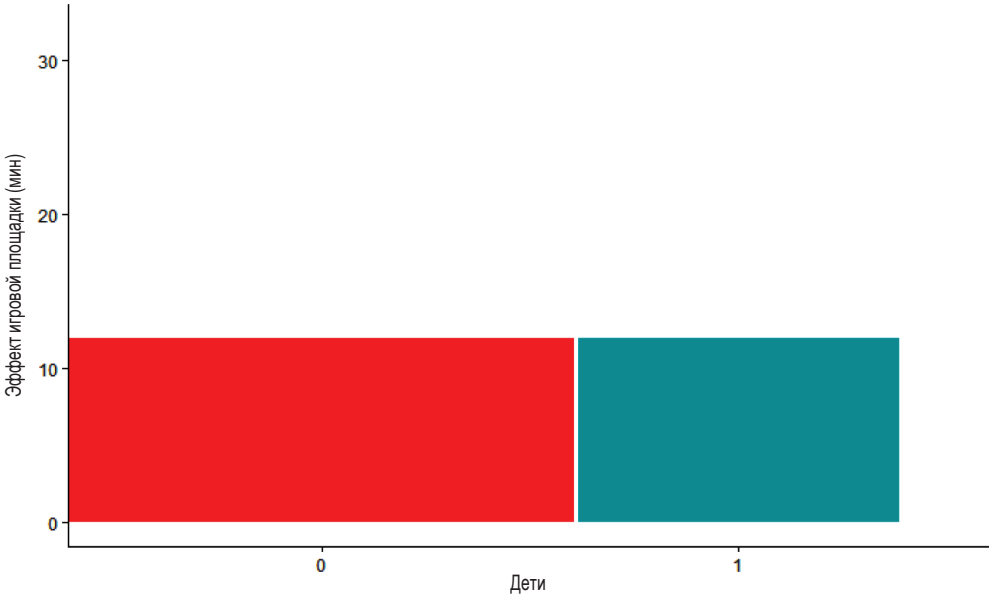

Рис. 11.12 ❖ Средний эффект игровой площадки по всем покупателям без детей и с детьми

Давайте на секунду допустим, что наличие игровой площадки в худшем случае имеет нулевой эффект на покупателей без детей (т. е. оно не может иметь отрицательного эффекта), это правдоподобное допущение с поведенческой точки зрения, если разумность наличия игровой площадки надлежаще подтверждена. Из этого следует, что, самое большее, весь средний эффект исходит от покупателей с детьми, и модерация перераспределяет всю область левого прямоугольника на правый прямоугольник, как показано на рис. 11.13.

В условиях этого «самого экстремального сценария» давайте рассчитаем размер эффекта в группе покупателей с детьми. По определению среднего значения мы имеем:

Средний размер эффекта = (*Сумма отдельных размеров эффекта*) / (*Суммарное число покупателей*).

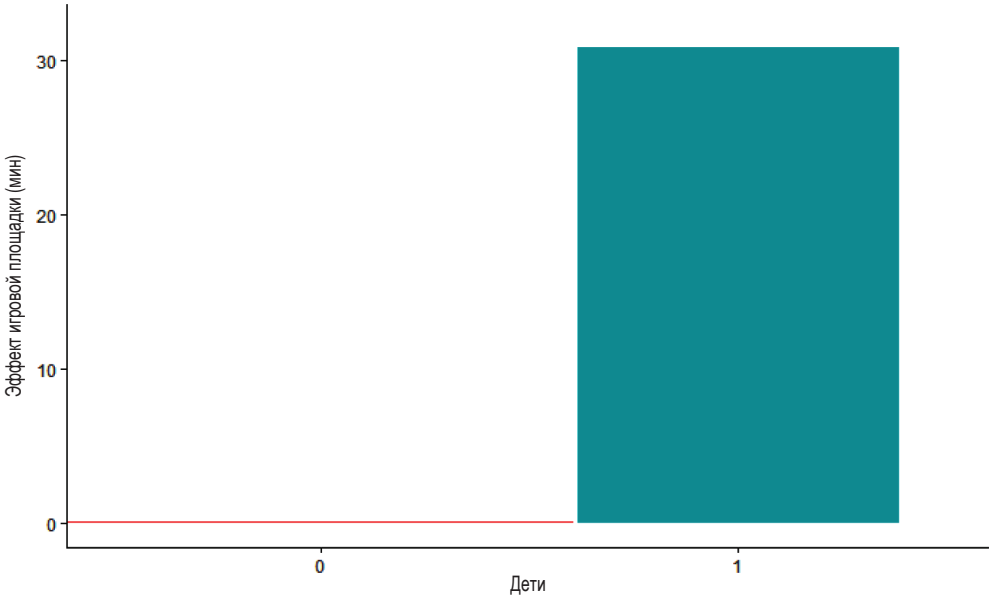

Рис. 11.13 ❖ Средний эффект, полностью исходящий от покупателей с детьми

Давайте выделим покупателей с детьми и без детей в числителе:

Средний размер эффекта = (*Сумма отдельных размеров эффекта для покупателей без детей* + *Сумма отдельных размеров эффекта для покупателей с детьми*) / (*Суммарное число покупателей*).

Если мы допустим отсутствие эффекта на покупателей без детей, то первый член в числителе равен нулю:

Средний размер эффекта = (*Сумма отдельных размеров эффекта для покупателей с детьми*) / (*Суммарное число покупателей*).

Давайте умножим обе стороны уравнения на суммарное число покупателей:

Средний размер эффекта × *Суммарное число покупателей* = *Сумма отдельных размеров эффекта для покупателей с детьми*.

А затем разделим обе стороны на число покупателей с детьми:

Средний размер эффекта × *Суммарное число покупателей* / *Число покупателей с детьми* = *Средний размер эффекта для покупателей с детьми*;

Средний размер эффекта × 1 / *Доля покупателей с детьми* = *Средний размер эффекта для покупателей с детьми*.

Поменяв местами левую и правую стороны для удобства чтения и вставив числа из предыдущего примера, мы получим:

Средний размер эффекта для покупателей с детьми = 11.92 × 1 / 0.387 = 30.8.

Другими словами, средний размер эффекта для покупателей с детьми не может превышать 30.8 дополнительной минуты для *ПродолжительностиПосещения*, учитывая совокупный средний размер эффекта. Если это число слишком мало, чтобы представлять экономический интерес, то нет смысла измерять модерацию между *ИгровойПлощадкой* и *Детьми*. Помимо этого конкретного примера, формула *Средний размер эффекта/Доля покупателей* в сегменте может применяться к любому потенциальному модератору: если у вас в вашей клиентской базе есть равная разбивка на мужчин и женщин, то это означает, что, самое большее, один пол имеет двойной средний эффект, а другой – никакого эффекта. Надеяться на то, что каким-то образом один пол будет иметь втрое больший средний эффект или что модерация увеличит эффект для обоих полов, – это принимать желаемое за действительное, и это математически невозможно. Другими словами, мы по понятным причинам склонны фокусироваться на группах с размерами эффекта выше среднего, но противоположностью должны быть группы с размерами эффекта ниже среднего.

Это означает, что ваш поиск потенциальных модераторов должен начинаться с переменных, которые имеют сильное поведенческое обоснование или были признаны эффективными в ходе прошлых аналитических расчетов и которые создают подгруппы, достаточно большие, чтобы что-то значить. Помимо определенной степени неравенства в размерах групп, поиск становится упражнением в тщетности: если у вас есть переменная, которая разбивает вашу клиентскую базу в пропорции 90/10 %, то даже если группа, представляющая 10 % вашей клиентской базы, не покажет никакого эффекта вообще, это увеличит эффект в группе 90 % максимум до 1/0.9 = 111 % от среднего эффекта, давая увеличение на 11 %.

В более широком смысле модерация не может спасти посредственный средний эффект. Следует стремиться лишь к тому, чтобы добавлять в вашу экспериментальную процедуру чуть-чуть больше крутизны; например, если вы находитесь на уровне 90 % от точки равновесия или целевого значения для него или если вы находитесь в высокообъемной среде и пытаетесь извлечь любой прирост эффективности, который вы можете извлечь.

Для каждой переменной, которую вы хотите протестировать, процесс будет одним и тем же. Сначала выполнить регрессию со взаимодействием между переменной экспериментальной процедуры и потенциальным модератором. Затем, если обнаружится, что эффект взаимодействия достаточно велик, подтвердить его, построив бутстраповский интервал уверенности вокруг оцененного эффекта.

Существует несколько правил, таких как поправка Бонферрони, для сокращения риска ложноположительных результатов при проверке большого числа гипотез в таком ключе. Однако я бы не рекомендовал их по двум причинам:

1) они обычно опираются, явно или неявно, на каркас статистического тестирования нулевой гипотезы с допущениями о нормальности;
2) они бывают чрезмерно консервативными и могут неприемлемо увеличивать риск ложноотрицательных результатов.

Вместо этого я бы рекомендовал подтверждать перспективные подгруппы с помощью последующих экспериментов, когда это возможно. Если модерационный эффект достаточно велик, чтобы представлять ценность для бизнеса, то его следует считать достаточно большим, чтобы давать право на дальнейшее подтверждение. Если вы работаете с экспериментальными данными, повторять эксперимент концептуально несложно. Если вы работаете с данными результатов наблюдений, то бывает не сразу понятно, какой эксперимент вам следует проводить. Однако для того, чтобы ваш модерационный эффект имел какую-либо экономическую ценность, он должен означать, что вы планируете делать что-то по-другому. В противном случае это просто интересный лакомый кусочек для коктейльных вечеринок. Что бы вы ни делали по-другому, вы, вероятно, сможете это каким-то образом рандомизировать, и вот у вас уже есть эксперимент.

Ключевым фактором успеха в этом процессе является надлежащее доведение до деловых партнеров информации о том, что результаты такого многократного тестирования следует рассматривать как ориентировочные гипотезы, а не как доказанный факт. Они не должны чувствовать, что вы не выполнили поставленную задачу, когда эксперимент дает нулевой результат, потому что в любом случае это был выстрел на дальнюю дистанцию. Также помните, что ввиду природы процесса ваш окончательный размер эффекта, скорее всего, будет меньше, чем тот, который вы обнаружили при первом проходе.

Нелинейности

Нелинейности, т. н. самомодерация, являются особым случаем по сравнению с другими формами модерации, поскольку, по определению, рассматривается только один модератор, что ограничивает риск ложноположительных результатов. В дополнение к этому последствия ложноположительного результата, как правило, лимитированы, если вы не делаете выводов за пределами диапазона доступных данных. Например, если бы вы измерили эффект дохода на покупки, основываясь главным образом на покупателях с годовым доходом от 25 000 до 75 000 долларов, а затем сделали выводы для покупателей с годовым доходом в 250 000 долларов, как риски, так и последствия ложноположительного результата были бы высоки (даже без учета модерации, экстраполирование столь далеко за пределы в любом случае было бы ужасной идеей).

Поэтому в основном нормально, когда интересующая вас причина рутинно тестируется на самомодерацию, если у вас в ваших данных есть по меньшей мере несколько сотен строк. Переменная также должна быть числовой, потому что самомодерирующаяся категориальная переменная была бы бессмысленной. Затем вы должны подтвердить эффект самомодерации,

построив бутстраповский интервал уверенности, как мы увидим чуть позже в этом разделе.

Помимо улучшения подгонки вашей регрессии и объяснения интуитивных поведенческих эффектов, таких как снижающиеся возвраты, включение самомодерации в вашу регрессию может предупреждать вас о присутствии скрытого модератора.

Давайте обратимся к примеру C-Mart: взаимосвязи между продолжительностью посещения и покупками продовольственных товаров. Вполне возможно, что очень короткие посещения представляют собой целенаправленные походы по магазинам для покупки конкретного товара, тогда как более продолжительные набеги, скорее всего, будут походами за продовольственными товарами. Это сделало бы основание для посещения спутывающим фактором взаимосвязи между *ПродолжительностьюПосещения* и *ПокупкамиПродовольственныхТоваров* (рис. 11.14).

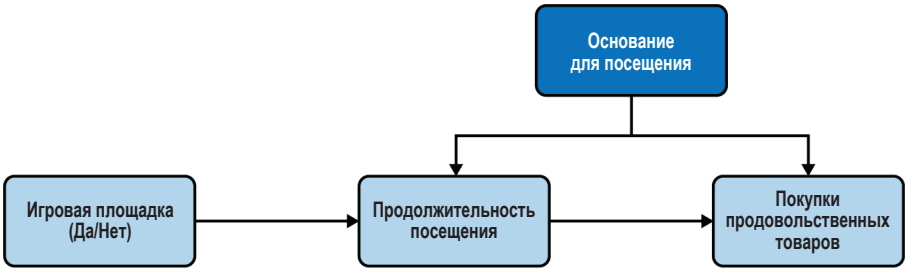

Рис. 11.14 ❖ Основание для посещения является спутывающим фактором взаимосвязи между *ПродолжительностьюПосещения* и *ПокупкамиПродовольственныхТоваров*

Но в то же время *ПродолжительностьПосещения* может, предположительно, модерировать эффект *ПродолжительностиПосещения* на *ПокупкиПродовольственныхТоваров*. Если вы идете в магазин, чтобы захватить подарок на день рождения или молоток, то удержание вас в магазине дольше (например, добавлением игровой площадки) вряд ли побудит вас купить вяленые томаты, как это могло бы быть, если бы вы ходили в поход за продовольственными товарами (рис. 11.15).

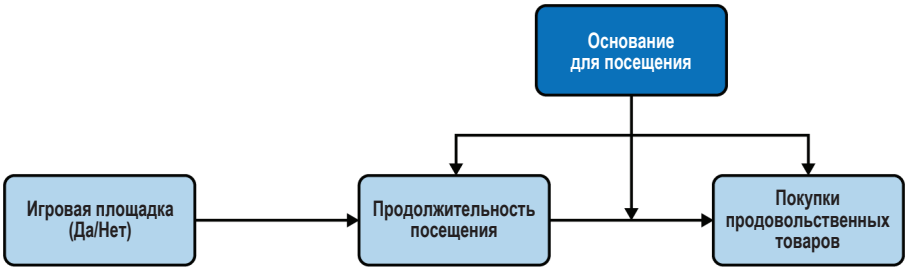

Рис. 11.15 ❖ Основание для визита модерирует эффект *ПродолжительностиПосещения* на *ПокупкиПродовольственныхТоваров*

Если бы мы могли наблюдать за причиной посещения, мы смогли бы добавить ее на нашу причинно-следственную диаграмму, и это стало бы простым случаем спутывания + модерация. Если допустить, что мы не можем наблюдать причину посещения, то мы остаемся со следующими ниже наблюдаемыми фактами:

- *ПокупкиПродовольственныхТоваров* положительно коррелируют с *ПродолжительностьюПосещения* (как из-за истинного эффекта, так и из-за спутывающего эффекта причины посещения);
- увеличение *ПродолжительностиПосещения* оказывает большее влияние на *ПокупкиПродовольственныхТоваров* для более продолжительных посещений, чем для более коротких посещений.

Другими словами, взаимосвязь между *ПродолжительностьюПосещения* и *ПокупкамиПродовольственныхТоваров* демонстрирует свою нелинейность. Самомодерационный член будет повышать точность нашей регрессии, поэтому нам следует его включить.

В более общем случае всякий раз, когда у вас есть переменная, которая, по-видимому, самомодерируется без четкого поведенческого основания, вам следует разведать возможность скрытой переменной, которая является как причиной, так и модератором этой переменной.

Напомним, что отыскание релевантных модераторов является важной частью поведенческого анализа, но, как и большинство наших инструментов, это часто больше искусство, чем наука. На стадии экспериментального дизайна мы можем отточить нашу поведенческую логику путем отыскания модерируемых посредников в наших исторических данных либо путем тестирования нашей экспериментальной процедуры с помощью UX. На стадии анализа данных зондирование почвы может обнаруживать многообещающие потенциальные модераторы; в то время как бутстраповские интервалы уверенности помогают сокращать риск ложноположительных результатов, последующие эксперименты в конечном счете являются вашей лучшей гарантией успеха. И наоборот, самомодерация безопасна для проведения разведки и включения в рутинном порядке.

Несколько модераторов

Для простоты до сих пор я показывал только по одному модератору за раз, но эффект может иметь несколько модераторов. Переход от одного модератора к нескольким прост; тут главная тонкость, которую следует учитывать, заключается в наличии или отсутствии взаимодействия модераторов друг с другом.

Параллельные модераторы

Возвращаясь к нашему примеру с C-Mart, мы могли бы вообразить, что еще одна демографическая переменная, *Возраст*, тоже отдельно модерирует эффект *ИгровойПлощадки* на *ПродолжительностьПосещения*, как показано на рис. 11.16.

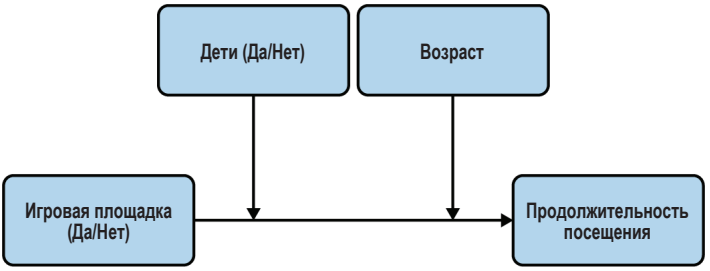

Рис. 11.16 ❖ Влияние *ИгровойПлощадки* на *ПродолжительностьПосещения* имеет два модератора

Соответствующее регрессионное уравнение показано в уравнении (11.3).

$$\begin{aligned} \textit{ПродолжительностьПосещения} = {} & \beta_0 + \beta_p \cdot \textit{ИгроваяПлощадка} + \beta_c \cdot \textit{Дети} \\ & + \beta_a \cdot \textit{Возраст} + \beta_{pc} \cdot (\textit{ИгроваяПлощадка} \\ & \times \textit{Дети}) + \beta_{pa} \cdot (\textit{ИгроваяПлощадка} \\ & \times \textit{Возраст}). \end{aligned} \tag{11.3}$$

Обратите внимание, что указанное уравнение включает отдельный член для *Возраста*, даже если стрелка из *Возраста* в *ПродолжительностьПосещения* отсутствует. Приведенное ранее предупреждение по-прежнему здесь применимо, и вам необходимо включить в свою регрессию все отдельные члены.

Интерпретация уравнения (11.3) будет заключаться в том, что:

- влияние *ИгровойПлощадки* на *ПродолжительностьПосещения* отличается для покупателей с детьми и без детей;
- влияние *ИгровойПлощадки* на *ПродолжительностьПосещения* отличается для молодых и пожилых покупателей. Эти два модерирующих эффекта не зависят друг от друга.

Возраст является числовой переменной, поэтому его интерпретация должна быть соответствующим образом скорректирована: коэффициент для *ИгровойПлощадки* × *Возраст* представляет собой разницу в эффекте *ИгровойПлощадки* на *ПродолжительностьПосещения* между покупателями, у которых разница в возрасте составляет один год. В зависимости от текущей деловой задачи мы можем:

- либо сохранять ее в числовом формате, если мы хотим получать прецизионную, причинно-следственно обоснованную оценку *ПродолжительностиПосещения*, например для определения увеличения продаж, которого нам следует ожидать от добавления *ИгровойПлощадки* в определенный магазин;
- либо сделать ее категориальной, надлежащим образом сгруппировав ее в «корзины». Например, мы могли бы конвертировать *Возраст* в такие корзины, как «менее 20», «от 20 до 40», «40+», либо любую другую разбивку, делая интерпретацию соответствующих коэффициентов проще для целей сегментации.

Взятые вместе, *Дети* и *Возраст* создают двухмерную демографическую сегментацию, которая позволяет нам сравнивать среднюю продолжительность посещения, скажем 28-летнего с детьми и 45-летнего без детей.

Поскольку два модерационных эффекта независимы друг от друга, мы могли бы подтвердить каждый из них независимо путем бутстрапирования регрессии для уравнения (11.3) и глядя на интервал уверенности для каждого модератора (см. следующий ниже подраздел).

Из этого также вытекает, что между двумя модераторами нет порядка, и их можно чертить взаимозаменяемо: на рис. 11.16 на первом месте стоят *Дети*, но там может быть и *Возраст*. Это не имеет значения.

Логика нескольких модераторов применима аналогично для взаимодействий между переменными одной и той же природы. Например, мы могли бы иметь ситуацию, когда *Дети* взаимодействуют как с *Возрастом*, так и с *Полом* (рис. 11.17).

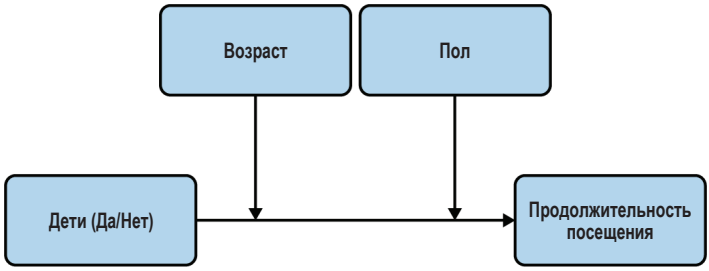

Рис. 11.17 ❖ И *Возраст*, и *Пол* взаимодействуют с *Детьми*

Соответствующее регрессионное уравнение таково:

$$ПродолжительностьПосещения = \beta_0 + \beta_c \cdot Дети + \beta_a \cdot Возраст + \beta_g \cdot Пол + \beta_{ca} \cdot (Дети \times Возраст) + \beta_{cg} \cdot (Дети \times Пол).$$

Наконец, регрессия может включать несколько переменных, каждая из которых самомодерируется. Здесь опять же анализ будет проводиться независимо для каждой из них.

В целом добавлять несколько независимых модераторов очень просто, но иногда также имеет смысл допускать, что модераторы взаимодействуют друг с другом, к чему мы сейчас и обратимся.

Взаимодействующие модераторы

Рассматривая эффект *ИгровойПлощадки* на *ПродолжительностьПосещения*, вполне разумно допустить, что он будет модерироваться как *Детьми*, так и *Возрастом*. Но мы также можем допустить, что среди покупателей с детьми увеличение продолжительности посещения зависит от возраста покупателя, например бабушки и дедушки с меньшей вероятностью оставляют

своих внучат на игровой площадке, чем родители оставляют своих детей. В таком случае модерирующий эффект *Детей* на эффект *ИгровойПлощадки* на *ПродолжительностьПосещения* сам по себе будет модерироваться *Возрастом*, что в социальных науках называется «модерируемой модерацией» (рис. 11.18).

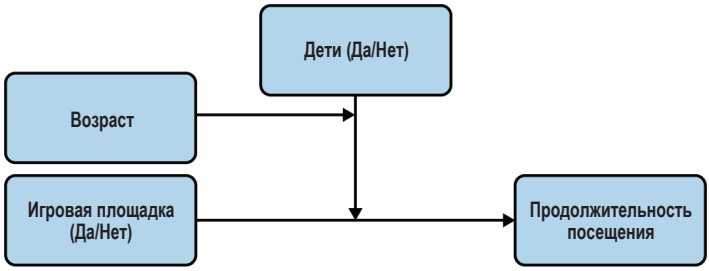

Рис. 11.18 ❖ Модерируемая модерация

На рис. 11.19 показано участие в игре модерируемой модерации, причем каждый подграфик демонстрирует возрастную группу среди наших покупателей. Мы видим, что при переходе от более молодых покупателей к более пожилым влияние *Детей* на влияние *ИгровойПлощадки* на *ПродолжительностьПосещения* уменьшается (довольно изрядно!), поскольку расстояние между двумя точками для $C = 1$ уменьшается по всем подграфикам.

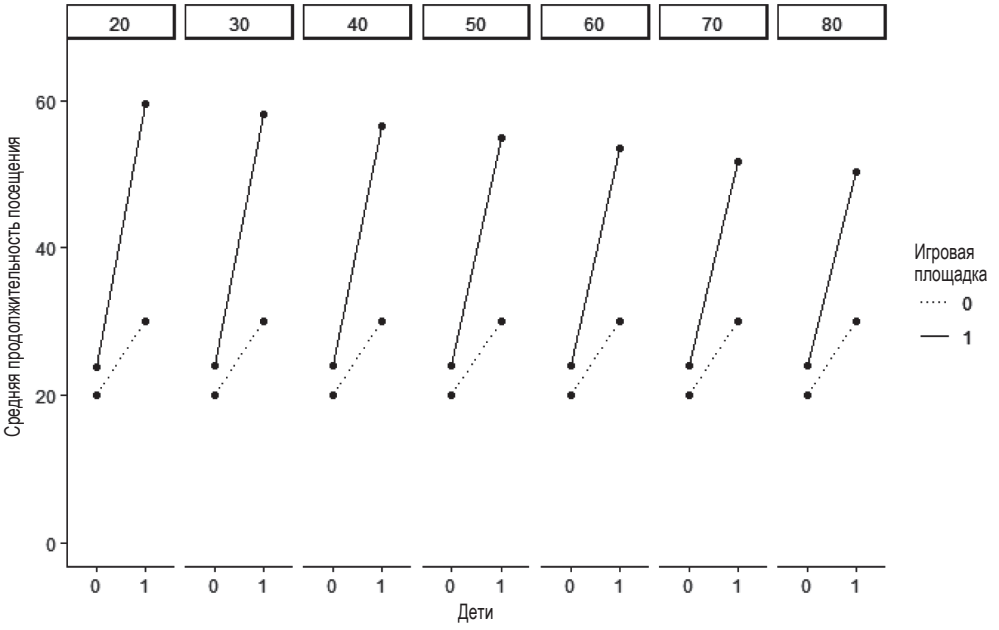

Рис. 11.19 ❖ Модерируемая модерация по разным возрастным группам

Соответствующее уравнение будет выглядеть так, как показано ниже:

$$\textit{ПродолжительностьПосещения} = \beta_0 + \beta_p \cdot \textit{ИгроваяПлощадка} + \beta_c \cdot \textit{Дети} + \beta_a \cdot \textit{Возраст} + \beta_{pc} \cdot (\textit{ИгроваяПлощадка} \times \textit{Дети}) + \beta_{pa} \cdot (\textit{ИгроваяПлощадка} \times \textit{Возраст}) + \beta_{ca} \cdot (\textit{Дети} \times \textit{Возраст}) + \beta_{pca} \cdot (\textit{ИгроваяПлощадка} \times \textit{Дети} \times \textit{Возраст}).$$

Это уравнение идентично уравнению (11.3) с добавлением трехпутного члена в конце. Давайте выполним регрессию (обратите внимание, как я только ввел в регрессию трехпутный член взаимодействия – и программно-информационное обеспечение добавило все отдельные переменные и двухпутные члены взаимодействия автоматически):

```
## R (результат не показан)
> summary(lm(duration~play_area * children * age, data=hist_data))

## Python
ols("duration~play_area * children * age", data=hist_data_df).fit().summary()

...
                         coef     std err    t        P>|t|    [0.025   0.975]
------------------------------------------------------------------------------
Intercept              20.0166    0.037  534.906    0.000    19.943    20.090
play_area               3.9110    0.063   62.014    0.000     3.787     4.035
children                9.9983    0.061  165.012    0.000     9.880    10.117
play_area:children     29.1638    0.101  290.105    0.000    28.967    29.361
age                    -0.0006    0.001   -0.820    0.412    -0.002     0.001
play_area:age           0.0010    0.001    0.806    0.420    -0.001     0.003
children:age            0.0003    0.001    0.297    0.767    -0.002     0.003
play_area:children:age -0.1637    0.002  -86.139    0.000    -0.167    -0.160
...
```

Коэффициент для нашего трехпутного члена, последнего в распечатке, является отрицательным, а также экономически содержательным, и 90%-ный интервал уверенности составляет приблизительно [–0.1671; –0.1590], что достаточно узко, чтобы дать нам уверенность в том, что он не будет близок к нулю.

Модерируемая модерация подчиняется той же логике и правилам, что и простая модерация. Поэтому она симметрична, а значит, мы можем интерпретировать *Возраст* как то, что он модерирует модерирующий эффект *Детей*, либо мы можем смотреть на него как на то, что *Дети* модерируют модерирующий эффект *Возраста*, который расположен последним в распечатке.

С поведенческой точки зрения интерпретация модерируемой модерации зависит от того, какая логика лежит в ее основе: логика сегментации либо логика взаимодействия:

- мы бы интерпретировали ее как сегментацию, если бы у нас были две переменные с личностными характеристиками, определяющие деловую характеристику или деловое поведение. Так было, например,

в случае с игровой площадкой, где *Дети* и *Возраст* модерируют эффект переменной *ИгроваяПлощадка*. Интуитивно это означает, что у нас есть двухмерная сегментация, где модерирующий эффект размерности увеличивается или уменьшается в другой размерности (например, модерирующий эффект наличия детей уменьшается с увеличением *Возраста*);

- с тремя переменными одинаковой природы мы бы интерпретировали ее как трехпутное взаимодействие между тремя переменными, как показано на рис. 11.20.

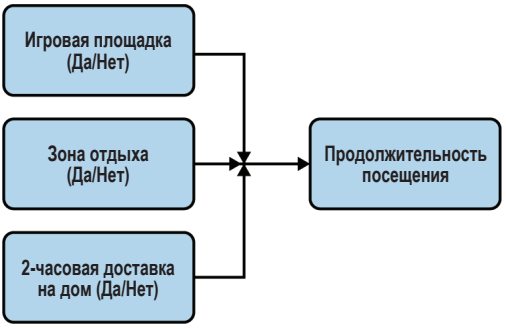

Рис. 11.20 ❖ Трехпутное взаимодействие между тремя переменными деловых характеристик

Смысл этого различия не в том, чтобы быть педантичным, а в том, чтобы напомнить вам о необходимости отслеживать цели вашего анализа. Отыскание и подтверждение модерации между предсказателями в регрессии – все это хорошо и ладно, но что вы собираетесь делать с этой информацией? Вы можете изменить поведения бизнеса, а иногда и деловые характеристики, но вы можете нацеливаться только на личностные характеристики покупателей. Схожим образом, если вы обнаруживаете, что определенное поведение покупателей модерирует эффект делового поведения (например, рекламная кампания по электронной почте с перекрестными продажами лучше всего работает с покупателями, которые были в магазине недавно), эта информация может использоваться либо для нацеливания на покупателей, которые демонстрировали такое поведение, либо для побуждения покупателей к такому поведению в первую очередь.

Технически говоря, модерируемая модерация может также применяться к нелинейностям, имея кубический член (например, *Покупки* = $\beta_0 + \beta_1 \cdot$ *ЭлектронныеПисьма* + $\beta_2 \cdot$ *ЭлектронныеПисьма*^{2} + $\beta_3 \cdot$ *ЭлектронныеПисьма*^{3}), но это бывает крайне редко и скорее является проявлением любопытства, лишенного практических применений.

В этом месте у вас, возможно, появятся признаки зарождающейся головной боли или даже полноценная головная боль. Модерация может быстро становиться нарочито сложной. Я показал вам трехпутные взаимодействия, чтобы вы знали, что это возможно, но я бы не пошел по этому пути, если бы

у вас не было для этого достаточно веских ранее существовавших поведенческих оснований. Помимо этого, теоретически мы могли бы иметь 5- или даже 12-путные взаимодействия («модерация модерации…»), но простое вбрасывание как можно большего числа членов взаимодействия является рецептом ложноположительных результатов и в конечном счете ослепительной, но бессмысленной модели. Прямолинейная и простая модерация – это нередко все, что вам нужно, и она может добавлять в ваш анализ значительную крутизну при разумной стоимости с точки зрения добавленной сложности.

Подтверждение модерации с помощью бутстрапа

До сих пор мы рассматривали коэффициенты регрессии только для модерационных членов, а не для *p*-значений. Но, как и любой другой коэффициент регрессии, наши коэффициенты для модерации подвержены неопределенности и изменчивости при отборе. Учитывать неопределенность тем более важно именно с модерацией, потому что она является «второпорядковым» эффектом (т. е. эффектом на эффект, а не прямым эффектом на переменную); эти эффекты обычно намного меньше, чем «первопорядковые» эффекты.

Давайте вернемся к нашему примеру с игровыми площадками в C-Mart. Наш оценочный коэффициент для модерации между *ИгровойПлощадкой* и *Детьми* составил $\beta_{pc} = 21$. Как обычно, мы будем использовать бутстраповские симуляции, чтобы определить степень нашей уверенности в значениях, которые мы наблюдали. Давайте начнем с того, что нарисуем 1000 случайных выборок по 10 000 строк каждая из располагаемых нами исторических данных и каждый раз будем выполнять ту же регрессию, что и в предыдущем разделе. Распределение значений коэффициента взаимодействия показано на рис. 11.21.

В главе 7 мы видели, что увеличение числа выборок повышает гладкость гистограммы и точность оценок интервалов уверенности, а увеличение размера выборок уменьшает дисперсию значений. Нужно ли нам делать что-либо из приведенного ниже, зависит от рассматриваемого делового вопроса:

- если вопрос звучит так: «Существует ли вообще какая-либо модерация?» (т. е. отличается ли коэффициент β_{pc} от нуля?), то рис. 11.21 уже позволяет нам однозначно ответить «да», и нет необходимости копать глубже;
- если вопрос является более неопределенным, например «Насколько мы уверены, что коэффициент для модерации превышает 20.5?», то нам придется увеличить размер выборки. Прямо сейчас гистограмма выходит за пределы 20.5 влево (и весьма). Каждое значение ниже 20.5 представляет собой бутстраповскую симуляцию, в которой коэффициент для модерации был ниже этого порогового значения. Затем давайте увеличим размер бутстраповских выборок до 200 000 строк (рис. 11.22).

Хорошо видно, что значения теперь гораздо уже сосредоточены вокруг 21 и находятся на безопасном расстоянии от 20.5. Этой второй симуляции было достаточно, чтобы ответить на наш деловой вопрос.

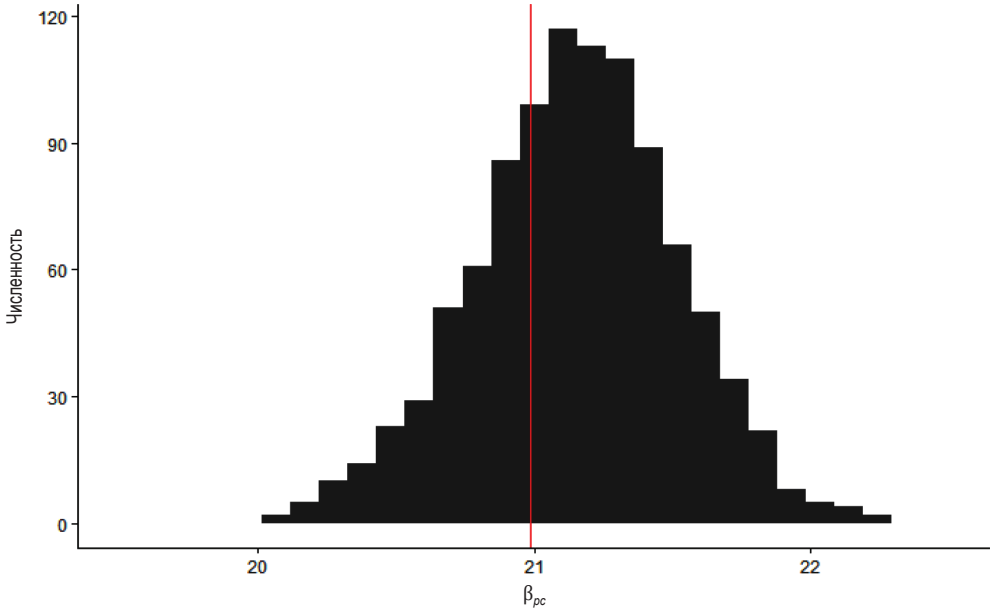

Рис. 11.21 ❖ Распределение бутстрапированных значений для коэффициента взаимодействия (1k выборок, состоящих из 10k строк каждая)

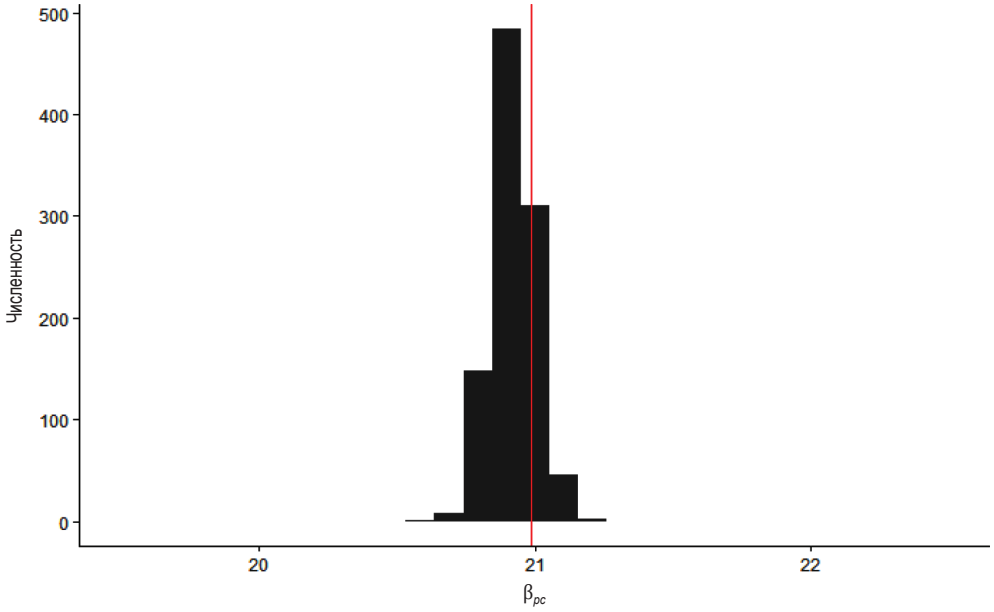

Рис. 11.22 ❖ Распределение бутстрапированных значений для коэффициента взаимодействия (1k выборок, состоящих из 200k строк каждая)

В указанном случае у нас в наших исторических данных всего около 600 000 строк, поэтому после подтверждения исходного кода с помощью

пробного прогона мы могли бы без особых неудобств перейти непосредственно к этому размеру выборки (помните, что вы никогда не должны использовать размер бутстраповской выборки выше размера данных, из которых вы черпаете). Но что делать, если бы ваши исторические данные содержали 10 или 100 миллионов строк? Вместо того чтобы ответить на свой деловой вопрос за считанные минуты, вы потратили бы часы или дни, ожидая результатов симулирования тысячи выборок такого размера; определенно, ваш окончательный интервал уверенности был бы чрезвычайно узким, например [20.9999; 21.0001], но это было бы совершенно пустой тратой времени, если бы вопрос заключался только в том, «превышает ли коэффициент 20.5 или нет?». Вот почему я хотел показать вам процесс поступательного увеличения размера бутстраповских выборок, вместо того чтобы сразу переходить к размеру ваших исторических данных.

Подводя итог: после выполнения небольшого бутстраповского симулирования, чтобы убедиться, что ваш исходный код работает правильно, вам следует увеличивать число выборок либо размер выборок согласно потребности, чтобы ответить на ваш деловой вопрос. Этот итеративный процесс имеет добавочную выгоду в том, что он задействует ваше критическое чутье и не позволяет вам выполнять анализ на автопилоте. Это относится ко всем формам модерации: сегментации (в том числе экспериментальных данных), взаимодействию и самомодерации.

Интерпретирование отдельных коэффициентов

В этой главе я несколько раз упоминал о том, что участвующие в модерации переменные также должны включаться в регрессию в качестве отдельных переменных, даже если вы не планируете их использовать либо они не выглядят значимыми. Однако если вы хотите их использовать, здесь есть несколько тонкостей, обзор которых мы сейчас проведем.

Давайте начнем со сравнения следующих ниже двух регрессий (для простоты только включая *Возраст* сверх *ИгровойПлощадки*):

$$\mathit{ПродолжительностьПосещения} = \beta_0 + \beta_{p0} \cdot \mathit{ИгроваяПлощадка} + \beta_{a0} \cdot \mathit{Возраст}$$

и

$$\mathit{ПродолжительностьПосещения} = \beta_1 + \beta_{p1} \cdot \mathit{ИгроваяПлощадка} + \beta_{a1} \cdot \mathit{Возраст} + \beta_{pa1} \cdot (\mathit{ИгроваяПлощадка} \times \mathit{Возраст}).$$

На первый взгляд может показаться, что интерпретация второго уравнения идентична первому, с единственным добавлением модерационного члена. Это не так. β_{p0} и β_{p1} не равны и не имеют одинакового смысла, и аналогичным образом для β_{a0} и β_{a1}.

Давайте визуализируем разницу, для удобства чтения начертив выборку, состоящую из 1000 точек данных, и представив их на плоскости *Возраст* × *Посещаемость* (рис. 11.23 и 11.24). На обоих рисунках показаны две линии регрессии: одна для покупателей с игровой площадкой и одна для покупателей без игровой площадки.

На рис. 11.23 представлено первое уравнение без модерации. Две линии имеют одинаковый наклон, равный $\beta_{a0} = -0.024$, а (постоянное) расстояние между этими двумя линиями равно $\beta_{p0} = 12.56$.

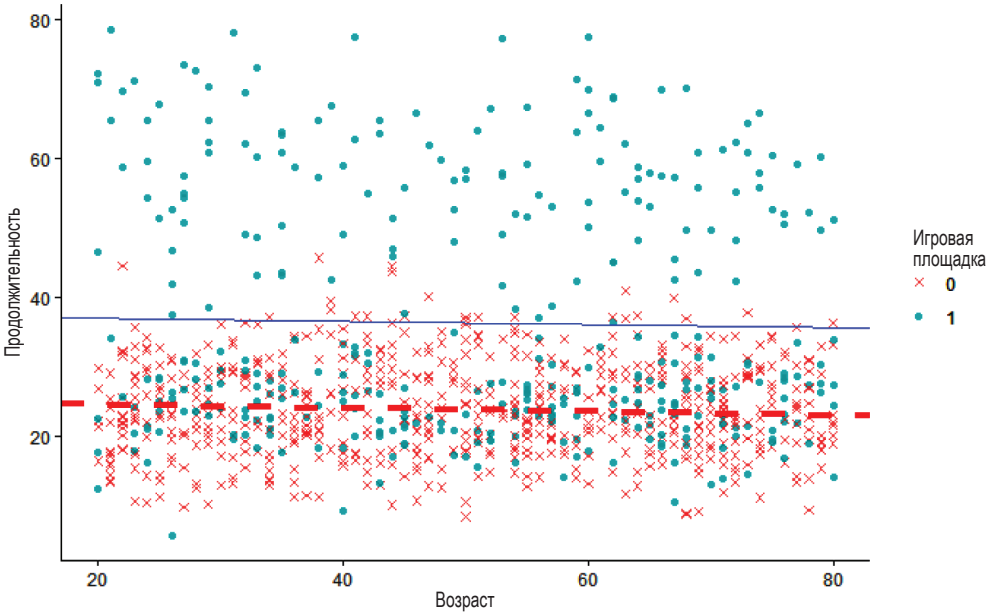

Рис. 11.23 ❖ Выборка размером 1000 точек данных и линии регрессии с игровой площадкой (полные точки, сплошная линия) и без нее (крестики, пунктирная линия), без модерационного члена

На рис. 11.24 показано соответствующее изображение для второго уравнения. Линии не параллельны. Из этого вытекает, что мы не можем просто сказать: «β_{p1} показывает вертикальное расстояние между двумя линиями регрессии», мы должны указать, при каком возрасте мы измеряем это вертикальное расстояние. Схожим образом мы не можем просто сказать, что «β_{a1} показывает наклон этих линий регрессии», мы должны выбрать, на какую из этих линий мы ссылаемся (т. е. мы должны указать, для какого значения *ИгровойПлощадки* мы измеряем этот наклон).

Это имеет важные последствия с точки зрения бизнеса. Если деловой партнер спросит «Каково влияние игровой площадки на продолжительность посещения?» и вы опираетесь на первое уравнение без модерационного члена, то вы можете ответить, что «это влияние равно β_{p0}». Однако если вы определили, что между этими двумя переменными существует значительный модерационный эффект, и вы хотите опираться на второе уравнение (как и следует делать), то вашим ответом на самом деле должно быть «Тут все зависит от ситуации», которого деловые партнеры боятся.

К счастью, эту проблему можно легко урегулировать с небольшой осторожностью, двумя возможными путями:

- установкой содержательных опорных точек;
- расчетом эффектов на уровне деловых решений.

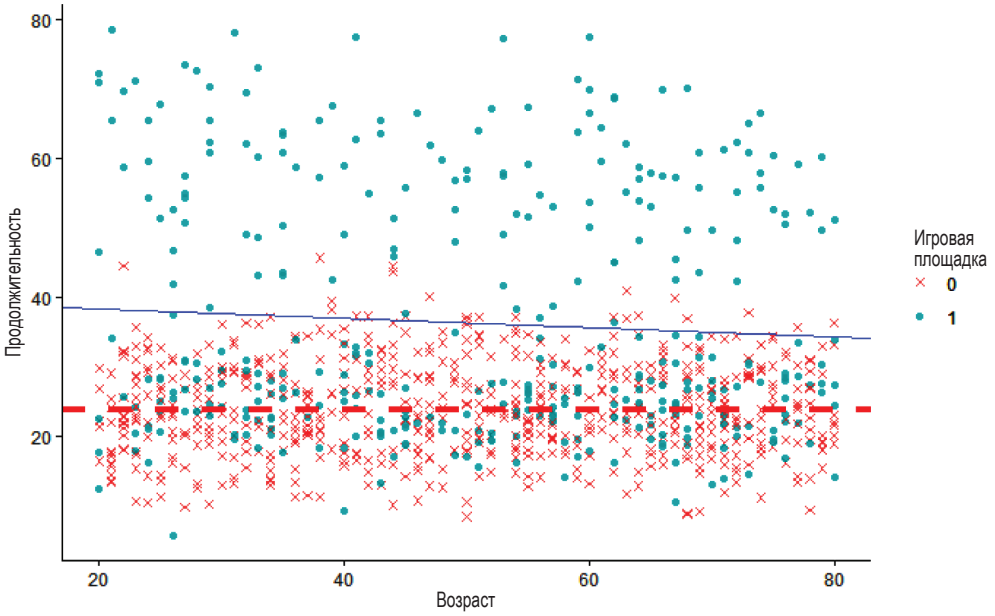

Рис. 11.24 ❖ Выборка в размере 1000 точек данных и линии регрессии с игровой площадкой (полные точки, сплошная линия) и без нее (крестики, пунктирная линия), с модерационным членом

Установка содержательных опорных точек

Первое решение состоит в установлении содержательных опорных точек для наших переменных. Сначала давайте вернемся к нашему уравнению с модерацией:

$$\textit{ПродолжительностьПосещения} = \beta_1 + \beta_{p1} \cdot \textit{ИгроваяПлощадка} + \beta_{a1} \cdot \textit{Возраст} + \beta_{pa1} \cdot (\textit{ИгроваяПлощадка} \times \textit{Возраст}).$$

Если мы установим возраст покупателя равным нулю, то ожидаемая продолжительность посещения у нашего новорожденного покупателя составит β_1 без игровой площадки и $\beta_1 + \beta_{p1}$ с игровой площадкой. Следовательно, β_{p1} достоверно является влиянием игровой площадки на продолжительность посещения для нулевого возраста. Однако новорожденные покупатели не являются нашей первостепенной целью, и мы можем добиться большего успеха, нормализовав переменную *Возраста*. Типичный подход состоит в установлении его равным среднему возрасту в нашем наборе данных:

```
## R
> centered_data <- hist_data %>% mutate(age = age - mean(age))
```

```
## Python
centered_data_df = hist_data_df.copy()
centered_data_df['age'] = centered_data_df['age'] \
    .subtract(centered_data_df['age'].mean())
```

Поступая так, мы сокращаем коэффициент для *ИгровойПлощадки* с 15.85 до 12.56, и мы можем сказать, что «влияние игровой площадки на продолжительность посещения составляет 12.56 дополнительной минуты для среднего возраста покупателей в наших данных». Схожим образом, поскольку игровая площадка является двоичной переменной, β_{a1} по умолчанию является наклоном линии для покупателей без игровой площадки. Это можно изменить, инвертировав уровни *ИгровойПлощадки* (т. е. установив по умолчанию 1 или «*Y*» и в качестве изменения 0 или «*N*», в зависимости от того, как сформулирована ваша двоичная переменная):

```
## R
> centered_data <- hist_data %>%
    mutate(play_area = factor(play_area, levels=c('1','0')))

## Python
centered_data_df['play_area'] = centered_data_df['play_area']
```

Изменение используемого по умолчанию уровня для *ИгровойПлощадки* меняет коэффициент для β_{a1} с 0 на –0.07 (наклон верхней сплошной линии на рис. 11.24).

Каким образом следует устанавливать значения по умолчанию для своих переменных? Это зависит от текущей деловой задачи, а также от природы переменной:

- в некоторых ситуациях для числовой переменной по умолчанию используется не среднее значение, а минимальное, максимальное или среднее значение в релевантной подгруппе (например, средний возраст покупателей с детьми или средний возраст покупателей в магазинах с игровой площадкой, в отличие от глобального среднего). В частности, когда вы имеете дело со счетными переменными, такими как число детей или число телефонных звонков, то ноль бывает лучшей опорной точкой, чем среднее значение;
- для двоичных переменных соответствующим принятым по умолчанию значением обычно является статус-кво. Здесь, например, для *ИгровойПлощадки* вы бы установили значение по умолчанию, равное 0, если вы рассматриваете возможность добавления новых игровых площадок, и значение 1, если вы рассматриваете несколько существующих;
- для категориальных переменных, таких как пол или штат, если нет содержательной опорной точки, то вы можете по умолчанию назначать наиболее распространенную категорию.

Установление содержательных опорных точек для всех участвующих в модерации переменных имеет преимущество вычислительной простоты и нередко является прямолинейным. Однако бывает, что оно быстро становится громоздким по мере увеличения числа участвующих переменных. Давайте

вообразим, например, что мы добавляем в нашу регрессию взаимодействие между штатом и полом:

$$\begin{aligned} \textit{ПродолжительностьПосещения} = {} & \beta_1 + \beta_{p1} \cdot \textit{ИгроваяПлощадка} + \beta_{a1} \cdot \textit{Возраст} \\ & + \beta_{pa1} \cdot (\textit{ИгроваяПлощадка} \times \textit{Возраст}) \\ & + \beta_{g1} \cdot \textit{Пол} + \beta_{s1} \cdot \textit{Штат} + \beta_{gs1} \cdot (\textit{Пол} \times \textit{Штат}). \end{aligned}$$

Теперь β_{p1} – это влияние игровой площадки для покупателей опорного возраста и пола в опорном штате. Высказывание «Эффект игровой площадки равен 10 для калифорнийских покупателей женского пола в возрасте 43 лет» начинает звучать труднопроизносимо и может стать довольно бессмысленным, если переменные не являются независимыми: тот факт, что у нас больше покупателей женского пола, чем мужского, тот факт, что у нас больше покупателей в Калифорнии, чем в любом другом штате, и тот факт, что совокупный средний возраст составляет 43 года, не означают, что у нас много 43-летних калифорнийских покупателей женского пола или кого-либо еще в этом отношении. Это подводит нас к другому решению – вычислению среднего эффекта в нашей выборке.

Расчет эффектов на уровне деловых решений

С научной точки зрения, одним из главных преимуществ предыдущего подхода является то, что он обеспечивает одно-единственное число для коэффициента, которое предположительно может применяться к совершенно другим обстоятельствам или, по меньшей мере, по сравнению с числами, полученными в других обстоятельствах. Однако цель прикладной аналитики состоит не в том, чтобы измерять вещи ради измерения, а в том, чтобы направлять деловые решения (если вы проводите анализ данных и не знаете, какие возможные решения могут из него выйти, то вам нужно поговорить со своим менеджером, потому что по меньшей мере один из вас неправильно выполняет свою работу). Поэтому альтернативный подход состоит в том, чтобы рассчитать значение интересующей вас переменной эффекта с этим решением и без него.

Давайте вообразим, например, что деловое решение для C-Mart состоит в выборе следующего магазина, в котором будет установлена игровая площадка. В целях ответа на этот вопрос нам не нужно определять один-единственный «средний» эффект игровой площадки, и попытка сделать это на самом деле была бы контрпродуктивной. Вместо этого для каждого магазина, в котором сегодня нет игровой площадки, мы можем напрямую определить, какова была бы средняя продолжительность дополнительного посещения, если бы мы добавили игровую площадку. Процесс заключается в следующем (номера выносок относятся как к R, так и к Python):

```
## Python (результат не показан)
def business_metric_fun(dat_df):
    model = ols("duration~play_area * (children + age)", data=dat_df) ❶
    res = model.fit(disp=0)
    action_dat_df = dat_df[dat_df.play_area == 0].copy() ❷
```

```
    action_dat_df['pred_dur0'] = res.predict(action_dat_df) ❸
    action_dat_df.play_area = 1 ❹
    action_dat_df['pred_dur1'] = res.predict(action_dat_df) ❺
    action_dat_df['pred_dur_diff'] = \ ❻
        action_dat_df.pred_dur1 - action_dat_df.pred_dur0
    action_res_df = action_dat_df.groupby(['store_id']) \ ❼
        .agg(mean_dur_diff=('pred_dur_diff', 'mean'),
            tot_dur_diff=('pred_dur_diff', 'sum'))
    return action_res_df

action_res_df = business_metric_fun(hist_data_df)
action_res_df.describe()
```

```
## R
business_metric_fun <- function(dat){
    mod_model <- lm(duration~play_area * (children + age), data=dat) ❶
    action_dat <- dat %>%
        filter(play_area == 0) ❷
    action_dat <- action_dat %>%
        mutate(pred_dur0 = predict(mod_model, action_dat)) %>% ❸
        mutate(play_area = factor('1', levels=c('0', '1'))) ❹
    action_dat <- action_dat %>%
        mutate(pred_dur1 = predict(mod_model, action_dat)) %>% ❺
        mutate(pred_dur_diff = pred_dur1 - pred_dur0) %>% ❻
        dplyr::group_by(store_id) %>% ❼
        summarise(mean_d = mean(pred_dur_diff), sum_d = sum(pred_dur_diff))
    return(action_dat)}

action_summ_dat <- business_metric_fun(hist_data)
summary(action_summ_dat)
```

```
    store_id      mean_d          sum_d
3       : 1   Min.   :10.41   Min.   :109941
4       : 1   1st Qu.:11.26   1st Qu.:129817
5       : 1   Median :11.80   Median  :143079
7       : 1   Mean   :11.95   Mean    :144616
8       : 1   3rd Qu.:12.25   3rd Qu. :155481
9       : 1   Max.   :14.43   Max.    :207647
(Other):27
```

❶ **Выполнить и сохранить модель, чтобы использовать ее для предсказаний.**
❷ **Выбрать магазины, которые сегодня не имеют игровой площадки.**
❸ **Добавить предсказанную продолжительность посещения в текущих обстоятельствах, pred_dur0.**
❹ **Поменять значение двоичной переменной *ИгровойПлощадки* с 0 на 1.**
❺ **Определить предсказанную продолжительность посещения с помощью добавленной игровой площадки, pred_dur1.**
❻ **Вычислить разницу между обоими.**
❼ **Агрегировать на уровне магазина, либо обратившись к среднему значению, либо к суммарной дополнительной продолжительности (среднее значение воспринимается интуитивнее, но общая сумма прямее связана с результатами бизнеса путем предпочтения более крупных магазинов).**

Затем мы можем выбрать магазин(ы) с наибольшими преимуществами игровой площадки. Важно отметить, что вы можете сами убедиться в том, что

центрирование числовых переменных в начале процесса приводит к точно таким же окончательным выводам. Математически это происходит потому, что мы вычитаем одну и ту же величину из двух членов в разности:

$$
\begin{aligned}
&(\textit{ПродолжительностьПосещения}_{i1} \\
&\quad - \text{среднее}(\textit{ПродолжительностьПосещения})) \\
&\quad - (\textit{ПродолжительностьПосещения}_{i0} \\
&\quad - \text{среднее}(\textit{ПродолжительностьПосещения})) \\
&\quad = \textit{ПродолжительностьПосещения}_{i1} \\
&\quad - \textit{ПродолжительностьПосещения}_{i0},
\end{aligned}
$$

где $\textit{ПродолжительностьПосещения}_{i1}$ и $\textit{ПродолжительностьПосещения}_{i0}$ – это предсказанная продолжительность посещения соответственно с игровой площадкой и без нее (независимо от текущих условий). Следовательно, с ориентированной на принятие решения точки зрения, опорные точки и центрирование либо их отсутствие не являются релевантыми, и вам больше не нужно об этом беспокоиться.

Резюмируя изложенное: добавление модерационного члена изменяет значение и интерпретацию отдельных коэффициентов для участвующих переменных. Это происходит потому, что по самому определению модерации указанные коэффициенты не одинаковы «везде», и модерация изменяет базовоуровневые значения, в которых они измеряются. Таким образом, отдельные коэффициенты должны интерпретироваться либо по отношению к соответствующим опорным точкам участвующих переменных (которые вы можете корректировать посредством центрирования), либо по всему нашему набору данных, но в зависимости от рассматриваемого решения.

Выводы

Одним из ключевых принципов бихевиористики является то, что «поведение есть функция человека и окружающей среды». Это высказывание обычно понимается как означающее, что мы можем влиять на поведения, изменяя окружающую среду. Это, безусловно, верно, но мне также нравится видеть в этом напоминание о том, что средние значения – это просто средние значения, и прилежному поведенческому аналитику надлежит копать глубже, орудуя модерационным анализом. В частности, я думаю, что некоторые из недавних безуспешных попыток реплицирования классических психологических экспериментов лучше всего интерпретировать не как означающие, что «эффекта нет», а что «эффект сильно модерируется характеристиками популяции (т. е. человеком) и экспериментальными условиями (т. е. окружающей средой)».

Может показаться, что этот вопрос относится лишь к академической сфере, но это не так. В деловых условиях многие жаркие дискуссии, в которых обе стороны использовали разрозненные подтверждения из жизни, могут быть с пользой переформулированы с точки зрения модерации. То, что верно на Восточном побережье, может оказаться не так на Западном побережье. Про-

грамма обучения бывает эффективной с неопытными сотрудниками, но не с опытными сотрудниками, или наоборот. По определению, регрессия без модерации не смогла бы пролить свет ни на один из этих двух случаев.

Это делает модерационный анализ очень ценным, но нередко игнорируемым дополнением к поясу инструментов поведенческой аналитики. В этой главе мы увидели, что он может применяться как к наблюдательным, так и к экспериментальным данным в очень простом ключе: надо лишь добавить в свою регрессию мультипликативный член между двумя переменными. В следующей (и последней) главе мы обратимся к еще одному ключевому инструменту анализа поведенческих данных – опосредованию.

Глава 12

Опосредование и инструментальные переменные

В предыдущей главе мы увидели, что модерация позволяет нам открывать черный ящик причинно-следственной связи, выявляя группы, для которых эта связь сильнее или слабее. Опосредование относится к наличию промежуточной переменной между двумя переменными в цепочке; оно предлагает другой способ прощупывать этот черный ящик путем понимания участвующего в игре причинно-следственного механизма – отвечая на вопрос «как» в отношении причинно-следственного эффекта.

Это имеет ряд выгод как с причинно-следственной, так и с поведенческой сторон нашего каркаса. С причинно-следственной точки зрения опосредование снижает риск ложноположительных результатов, а отсутствие адекватного их объяснения может систематически смещать наши аналитические расчеты. С поведенческой точки зрения опосредование помогает нам улучшать экспериментальный дизайн и глубже понимать эксперименты. В некотором смысле опосредование не является чем-то новым, и большинство аргументов в этой главе можно было бы резюмировать вот так: «расширяй цепочки на своих причинно-следственных диаграммах настолько, насколько сможешь, по меньшей мере в начале». Но я считаю, что такое упрощение оказало бы вам медвежью услугу, потому что отыскание посредников лежит в основе многих научных открытий. Вопрос «Но почему?» является одним из лучших последующих вопросов после подтверждения причинно-следственной взаимосвязи между двумя переменными. Удовлетворенность клиентов повышает удержание, но почему? Происходит ли это потому, что она снижает вероятность поиска альтернатив, или потому, что она повышает мнение клиента о компании?

Опосредование также предлагает хорошую ступеньку для последнего инструмента, который мы увидим в этой книге, инструментальных переменных. Инструментальные переменные, которые подобны посредничеству на стероидах, позволяют нам отвечать на вопросы, которые в противном случае

были бы непрослеживаемыми. Как и было обещано в начале книги, мы получим несмещенную оценку влияния удовлетворенности клиентов на последующее поведение клиентов, и инструментальные переменные делают это возможным.

В следующем далее разделе я познакомлю вас с опосредованием в контексте примера игровых площадок сети магазинов C-Mart из главы 11 и покажу, как опосредование может повышать эффективность анализа причинно-следственных связей. Затем во втором разделе мы перейдем к торжественному финалу: инструментальным переменным.

Опосредование

Давайте продолжим пример с C-Mart из предыдущей главы и допустим, что C-Mart теперь заинтересована в измерении эффекта игровых площадок на покупки продовольственных товаров (рис. 12.1).

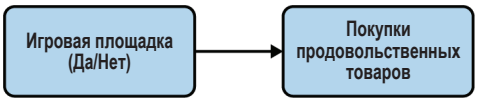

Рис. 12.1 ❖ Интересующая нас взаимосвязь

Основываясь на предыдущих аналитических расчетах, руководство C-Mart считает, что продолжительность посещения является ключевым причинно-следственным механизмом в этой взаимосвязи. То есть они считают, что *ИгроваяПлощадка* является причиной *ПродолжительностиПосещения*, которая сама по себе является причиной *ПокупокПродовольственныхТоваров*. Очевидно, что мы могли бы пренебречь этой гипотезой и проанализировать взаимосвязь, показанную на рис. 12.1, непосредственно, построив соответствующую регрессию, интервалы уверенности и т. д. Тем не менее мы собираемся подтвердить и измерить этот причинно-следственный механизм, потому что его объяснение имеет ряд выгод:

- опосредование позволяет нам понимать действующие механизмы и генерировать практические идеи;
- при определенных обстоятельствах необъяснение опосредования могло бы систематически смещать причинно-следственную оценку.

Давайте проведем обзор этих выгод подробнее в следующих далее двух подразделах, а затем я перейду к техническим соображениям измерения опосредования.

Понимание причинно-следственных механизмов

Первая выгода выявления и измерения опосредования заключается в том, что она дает объяснение имеющегося причинно-следственного механизма. Кор-

реляция все-таки не (всегда) есть каузация, но понимание того, что на самом деле происходит с поведенческой точки зрения, является прочной защитой от мнимых корреляций. Если у вас есть две коррелированные переменные, но вы не уверены в том, что эта корреляция является причинно-следственной, то отыскание и подтверждение взаимосвязи между ними дает очень веские доказательства того, что связь является причинно-следственной. В том месте наиболее вероятным источником ошибки будет обратная причинно-следственная связь – она течет в противоположном направлении. (Альтернативой является то, что каждая из трех переменных просто имеет мнимую корреляцию с двумя другими; это целые три мнимые корреляции из чистой случайности вместо одной, что является крайне маловероятным событием.)

Опосредование также является очень эффективным дополнением к модерации как в фазе обнаружения, так и в фазе дизайна. В фазе обнаружения (то, что в предыдущей главе я назвал «зондированием почвы») выявление вероятных посредников помогает процессу мозгового штурма. Даже если посредник не поддается наблюдению (например, мнение или эмоция), простое его рассмотрение может приводить вас к измеримому модератору и давать ему обоснование. В главе 11 мы видели, что взаимосвязи между *ИгровойПлощадкой* и *ПокупкамиПродовольственныхТоваров* модерируются *Детьми*, что имеет непосредственный и интуитивный смысл. Однако распознание того, что эта взаимосвязь опосредуется *ПродолжительностьюПосещения*, может дать нам ключ к другим возможным модераторам. Например, если опосредование является полным, то совокупный эффект, скорее всего, будет слабее для посещений ближе ко времени закрытия магазина, что не было бы сразу очевидно, если бы мы игнорировали роль *ПродолжительностиПосещения* (рис. 12.2).

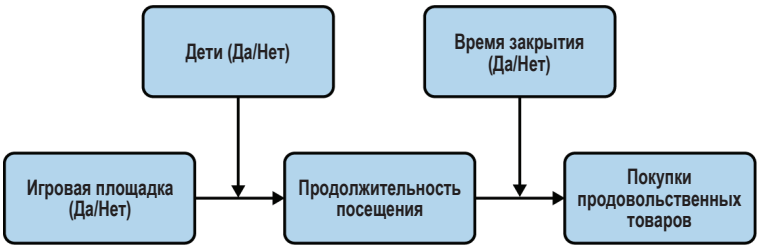

Рис. 12.2 ❖ Взаимосвязь между *ИгровойПлощадкой* и *ПокупкамиПродовольственныхТоваров*

Мы могли бы прощупать эту гипотезу, проверив, что эффект *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров* является слабее для посещений, которые начинаются ближе к закрытию, что дало бы дополнительную информацию о природе эффекта.

Опосредование также бывает полезно для конструирования или совершенствования деловых процессов и сообщений. Слишком часто упражнения по сегментации заканчиваются словами «сегмент А демонстрирует более сильный эффект, чем сегмент В». Это полезно только для более точного нацеливания на уже существующие экспериментальные процедуры, такие как

маркетинговые кампании. Понимание механизма, посредством которого модерация действует, дает нам полезную информацию для порождения идей. Подтвердив, что модерирующий эффект *Детей* на *ИгровуюПлощадку* опосредуется *ПродолжительностьюПосещения*, мы могли бы на это опереться, предложив льготу, например дополнительную проверку наличия парковочного места или закусочной рядом с игровой площадкой. Мы также могли бы заменить игровые площадки (потенциально более дешевыми) мини-кинотеатрами, в которых показывают мультфильмы.

Причинно-следственные систематические смещения

Опосредование – это не просто инструмент, который «неплохо иметь». В некоторых обстоятельствах если его не учитывать, то в наши причинно-следственные оценки будет внесено систематическое смещение.

Простейший случай, когда это происходит, – это ситуация, когда мы пытаемся измерить (суммарный) эффект одной переменной на еще одну, но мы невольно включаем в нашу регрессию посредника в качестве контрольной переменной. Давайте допустим, что мы измерили эффект наличия игровой площадки на покупки продовольственных товаров на отдельном уровне в соответствии с рекомендациями из главы 2. Вполне возможно, что игровая площадка не только влияет на продолжительность посещения магазина покупателями, которые пришли бы независимо ни от чего, но и привлекает новых покупателей. В целях объяснения этого причинно-следственного пути нам нужно перечертить нашу причинно-следственную диаграмму на уровне магазина (рис. 12.3).

Рис. 12.3 ❖ Переформулирование причинно-следственной диаграммы на уровне магазина

Обратите внимание, что хотя средние покупки продовольственных товаров на уровне покупателей полностью опосредуются средней продолжительностью посещения, это не относится к суммарным продажам продовольственных товаров на уровне магазина. Уровень, на котором вы измеряете поведения, имеет значение!

На рис. 12.3 видно, что *ЧислоПокупателей* является посредником эффекта *ИгровойПлощадки* на *ПродажиПродовольственныхТоваров*. Тем не менее мы легко можем вообразить, что кто-то, расследующий этот эффект, решил бы включить *ЧислоПокупателей* в качестве контрольной переменной

в эту регрессию: *ПродажиПродовольственныхТоваров* = $\beta_p \cdot$ *ИгроваяПлощадка* + $\beta_c \cdot$ *ЧислоПокупателей*. В конце концов, размер клиентской базы магазина определенно влияет на суммарные продажи продовольственных товаров, и это, возможно, повлияло на выбор магазинов, в которых будут построены игровые площадки (рис. 12.4).

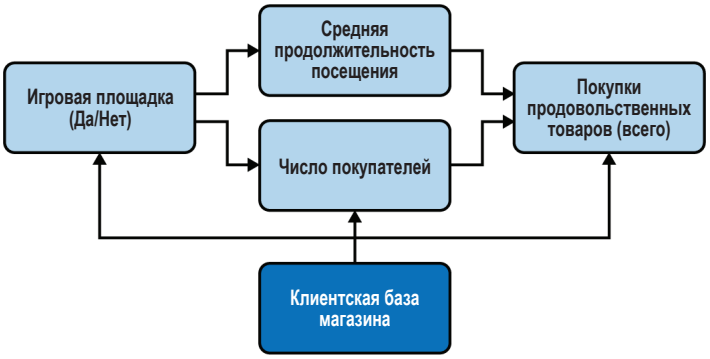

Рис. 12.4 ❖ Клиентская база магазина является спутывающим фактором

На рис. 12.4 представлена причинно-следственная головоломка: клиентская база магазина, измеряемая числом потенциальных покупателей, находящихся в пределах досягаемости магазина, может быть спутывающим фактором взаимосвязи между *ИгровойПлощадкой* и *ПродажамиПродовольственныхТоваров*. Но в то же время *ЧислоПокупателей* является посредником указанной взаимосвязи. Эта ситуация требует продуманного контроля за соответствующими регрессиями, то есть всеми регрессиями, в которых зависимой переменной является *ЧислоПокупателей* или *ПродажиПродовольственныхТоваров*.

Этого можно достичь, добавив в эти регрессии продуманные контрольные переменные. Например, число покупателей за год до установления игровой площадки является неплохим косвенным индикатором для клиентской базы и по определению не улавливает никаких эффектов установления игровой площадки. В качестве альтернативы мы могли бы выбрать еще один косвенный индикатор, такой как категориальная переменная для *СельскогоРайона*/*ГородскогоРайона*.

Это еще раз иллюстрирует опасности подхода «все, что есть, и кухонная раковина в придачу» к включению переменных: только потому, что текущее число покупателей имеется в распоряжении и соответствует нашей ситуации, не означает, что оно должно автоматически включаться в качестве элемента контрольной переменной.

Выявление опосредования

Знакомя со строительными блоками причинно-следственных диаграмм в главе 3, я упомянул, что посредник (медиатор) – это переменная между двумя другими переменными в цепочке, как показано на рис. 12.5.

Рис. 12.5 ❖ Эффект наличия игровой площадки на покупки продовольственных товаров опосредуется продолжительностью посещения

ПродолжительностьПосещения – это следствие *ИгровойПлощадки* и причина *ПокупокПродовольственныхТоваров*. В силу этого на данной причинно-следственной диаграмме она является посредником эффекта *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров*.

Если допустить, что причинно-следственная диаграмма на рис. 12.5 рассказывает нам всю историю, то *ИгроваяПлощадка* не имеет эффекта на *ПокупкиПродовольственныхТоваров*, кроме как через путь, проходящий по *ПродолжительностиПосещения*; изменение *ИгровойПлощадки* при удержании *ПродолжительностиПосещения* постоянной не изменит *ПокупкиПродовольственныхТоваров*. Принято говорить, что эффект *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров* «идеально» или «полностью» опосредуется. В качестве аьтернативы мы могли бы вообразить ситуацию, когда *ИгроваяПлощадка* также имеет прямой эффект на *ПокупкиПродовольственныхТоваров*, помимо пути из *ПродолжительностиПосещения* (рис. 12.6).

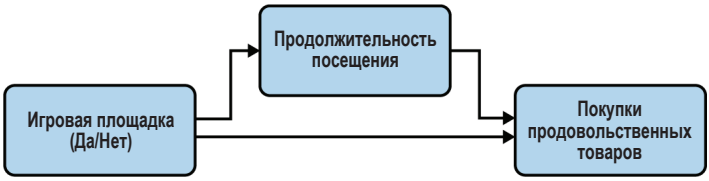

Рис. 12.6 ❖ Частичное опосредование

Это называется «частичным» опосредованием. *ПродолжительностьПосещения* по-прежнему является посредником, даже если она не объясняет суммарный эффект *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров*. Прямой эффект от *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров* может быть по-настоящему прямым – в нем не участвует промежуточная переменная – либо он может представлять одного или нескольких других посредников, о которых мы не знаем или в анализе которых не заинтересованы (и в таком случае мы свернули соответствующую цепочку или цепочки).

Как и поиск потенциальных модераторов, поиск посредников является поиском методом проб и ошибок, но область поиска (и, следовательно, риск ложноположительных результатов) гораздо более лимитирована. Поскольку потенциальный посредник должен расследоваться только в том случае, если для этого есть веское поведенческое основание, число кандидатов значительно ограничено. В дополнение к этому подтверждение посредника предусматривает несколько регрессий, что также снижает риск ложноположительных результатов.

Учитывая выгоды от объяснения посредников и риски, связанные с их отсутствием, вы всегда должны включать в свой анализ релевантных посредников, по меньшей мере в качестве первого шага. Определив цепочки, которые вы собираетесь анализировать, и те, которые вы собираетесь игнорировать, вы можете безопасно свернуть последние (например, посредников между переменными, которые играют в вашем анализе лишь второстепенную роль).

Как и в случае с модераторами, процесс поиска включает в себя поиск потенциального кандидата, а затем его подтверждение посредством бутстраповского интервала уверенности. В нашем примере *ПродолжительностьПосещения* является очевидным кандидатом, поэтому давайте посмотрим, как мы сможем подтвердить и измерить его посредническую роль.

Измерение опосредования

Проводить измерение опосредования довольно просто, но немного громоздко. Указанное измерение сводится к выполнению нескольких регрессий, чтобы оценить следующее:

- суммарный эффект *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров*;
- эффект *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров*, который опосредуется *ПродолжительностьюПосещения*, т. н. косвенный эффект;
- эффект игровой площадки на *ПокупкиПродовольственныхТоваров*, который не опосредуется *ПродолжительностьюПосещения*, т. н. прямой эффект.

Если мы не отыщем подтверждающих данных для косвенного, опосредованного пути, то мы должны отклонить нашего предварительного посредника. И наоборот, если мы не отыщем подтверждающих данных прямого пути, то эффект полностью опосредован. Процент суммарного опосредованного эффекта является распространенным и полезным способом резюмирования этих подтверждающих данных. Мы завершим данный раздел рассмотрением особого случая, в котором посредником является двоичная переменная.

Суммарный эффект

Сначала мы определяем суммарный эффект, выполнив регрессию *ПокупокПродовольственныхТоваров* на *ИгровойПлощадке* (без объяснения *ПродолжительностиПосещения*):

```
## R (результат не показан)
summary(lm(grocery_purchases~play_area, data=hist_data))

## Python
ols("grocery_purchases~play_area", data=hist_data_df).fit().summary()
...
              coef       std err    t           P>|t|   [0.025 0.975]
Intercept 49.1421    0.047      1036.494    0.000   49.049 49.235
play_area 27.6200    0.079       349.485    0.000   27.465 27.775
...
```

Суммарный эффект составляет примерно 27.6, а значит, добавление игровой площадки увеличивает сумму, расходуемую на продовольственные товары, в среднем на 27.6 доллара, не держа *ПродолжительностьПосещения* постоянной.

Опосредованный эффект

Эффект *ИгровойПлощадки* на *ПокупкиПродовольственныхТоваров*, опосредованный *ПродолжительностьюПосещения*, можно получить путем умножения эффекта *ИгровойПлощадки* на *ПродолжительностьПосещения* и эффекта *ПродолжительностиПосещения* на *ПокупкиПродовольственныхТоваров*. Это имеет интуитивный смысл: если игровая площадка увеличивает среднюю продолжительность посещения на X минут, а каждая дополнительная минута продолжительности посещения увеличивает сумму, расходуемую на продовольственные товары, на Y долларов, тогда добавление игровой площадки увеличивает сумму, расходуемую на продовольственные товары, на $X \times Y$ долларов.

Первая регрессия относится к стрелке между *ИгровойПлощадкой* и *ПродолжительностьюПосещения*. Это дает коэффициент, равный примерно 12.6 (наличие игровой площадки добавляет примерно 12.6 минуты к средней продолжительности посещения):

```
## R (результат не показан)
summary(lm(duration~play_area, data=hist_data))

## Python
ols("duration~play_area", data=hist_data_df).fit().summary()
...
              coef       std err    t           P>|t|   [0.025 0.975]
Intercept 23.8039    0.018     1287.327     0.000   23.768 23.840
play_area 12.5570    0.031      407.397     0.000   12.497 12.617
...
```

Вторая регрессия относится к стрелке между *ПродолжительностьюПосещения* и *ПокупкамиПродовольственныхТоваров*. Тем не менее в эту регрессию я также включаю *ИгровуюПлощадку*. Глядя снова на рис. 12.6 и помня определение термина «спутывающий фактор», мы видим, что если опосредование является только частичным (т. е. имеется прямая стрелка из *ИгровойПлощадки* в *ПокупкиПродовольственныхТоваров*), то *ИгроваяПлощадка* является спутывающим фактором взаимосвязи между *ПродолжительностьюПосещения* и *ПокупкамиПродовольственныхТоваров*. Поэтому он должен быть включен в регрессию по умолчанию. Выполнение регрессии с нашей первичной причиной и нашим посредником в качестве объясняющих переменных дает коэффициенты соответственно 0.16 (добавление игровой площадки добавляет примерно 0.16 доллара к средним покупкам продовольственных товаров за посещение, держа продолжительность посещения постоянной) и 2.2 (добавление одной минуты к продолжительности посещения добавляет примерно 2.20 доллара к средним покупкам продовольственных товаров за посещение):

```
## Python (результат не показан)
ols("grocery_purchases~duration+play_area", data=hist_data_df).fit().summary()

## R
summary(lm(grocery_purchases~duration+play_area, data=hist_data))
...
Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) -2.917728   0.047329  -61.647  < 2e-16 ***
duration     2.187025   0.001695 1290.410  < 2e-16 ***
play_area1   0.157477   0.046419    3.393 0.000693 ***
...
```

В приведенном выше очень простом примере эти три переменные являются единственными участвующими переменными, но в реальной жизни вам пришлось бы также включать в каждую свою регрессию любую другую переменную со стрелкой в сторону зависимой переменной.

Выше я уже упоминал, что первичная причина должна включаться в регрессию «по умолчанию». Однако иногда первичная причина и посредник могут быть настолько тесно коррелированы, что включение их обоих в регрессию создает мультиколлинеарность. Как правило, это обусловлено полной опосредованностью и подтверждается подозрительно крупными коэффициентами в противоположных направлениях (то есть первичная причина и посредник в основном компенсируют друг друга) с большими *p*-значениями. В худших случаях ваше аналитическое программно-информационное обеспечение, возможно, даже откажет и выдаст сообщение об ошибке, не завершив регрессию. Всякий раз, когда включение первичной причины в регрессию приводит к тому, что коэффициент для посредника идет наперекосяк, включать первичную причину не следует.

Наконец, вы также, возможно, столкнетесь с более сложными ситуациями, такими как два посредника с лишней стрелкой между ними, то есть один из посредников также является причиной другого (рис. 12.7). Это не просто теоретическая возможность; это действительно время от времени происходит, в особенности с поведенческими данными. Подобного рода ситуации бросают вызов укороченным путям, и вам нужно будет помнить то, что мы узнали в части II: критерий боковой двери для распутывания и другие правила причинно-следственных диаграмм по-прежнему применимы и позволят вам узнавать то, какие переменные вам следует и не следует включать в свою регрессию.

Опосредованный эффект равен произведению двух коэффициентов по цепочке распространения (т. е. коэффициенту регрессии *ПродолжительностиПосещения* на *ИгровойПлощадке* и коэффициенту регрессии *ПокупокПродовольственныхТоваров* на *ПродолжительностиПосещения*):

$$ОпосредованныйЭффект \approx 12.6 * 2.2 \approx 27.5.$$

С этого места мы можем рассчитать процент суммарного опосредованного эффекта:

ПроцентОпосредованнности = *ОпосредованныйЭффект* / *СуммарныйЭффект* ≈ 27.5/27.6 ≈ 99.5 %.

Бутстраповский 90%-ный интервал уверенности для этого процента приближенно составляет [0.9933; 0.9975].

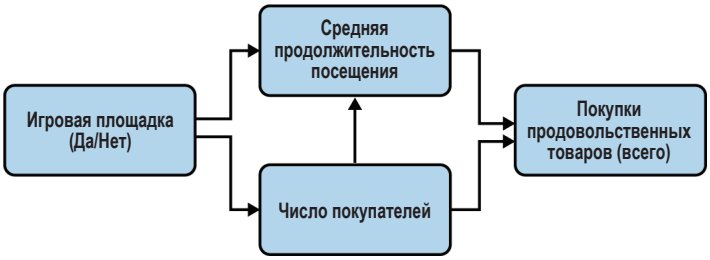

Рис. 12.7 ❖ Один из посредников влияет на другого посредника

Прямой эффект

Прямой, неопосредованный эффект равен коэффициенту для *ИгровойПлощадки* в регрессии *ПокупокПродовольственныхТоваров* на *ИгровойПлощадке* и *ПродолжительностиПосещения*: *НеопосредованныйЭффект* ≈ 0.16. С этого места мы можем рассчитать процент суммарного неопосредованного эффекта:

ПроцентНеопосредованности = *НеопосредованныйЭффект* / *СуммарныйЭффект* ≈ 0.16/27.6 ≈ 0.5 %.

Другими словами, суммарный эффект технически опосредован не полностью. Однако для практических целей мы можем пренебречь неопосредованным эффектом. Обратите внимание, что «опосредованный эффект» всегда выражается по отношению к конкретному посреднику. Если бы у вас было два посредника, которые вместе полностью опосредовали бы суммарный эффект, то опосредованный эффект первого был бы равен опосредованному эффекту второго, и наоборот.

В ситуациях, когда вы сталкиваетесь с мультиколлинеарностью и не можете включить первичную причину в регрессию, как описано ранее, эффект, скорее всего, является полностью опосредованным. Вы можете и должны убедиться, что это так, рассчитав процент суммарного эффекта, который не является опосредованным, как остаток опосредованного эффекта:

ПроцентНеопосредованности = (*СуммарныйЭффект* – *ОпосредованныйЭффект*) / *СуммарныйЭффект*.

Если вы получаете неопосредованный эффект, который благодаря такому подходу является экономически значимым, то это означает, что у вас, скорее всего, более сложная причинно-следственная структура, чем вы думали.

Тогда самое время пересмотреть свою причинно-следственную диаграмму критическим взглядом. Может быть, ваши первичная причина и посредник делят между собой дальнейшую общую причину? Или, возможно, есть несколько посредников со взаимосвязями между ними.

Посредник в этом примере был положительно коррелирован как с его причиной, так и с его следствием. Посредники также могут иметь отрицательный эффект, если ровно один из двух коэффициентов вдоль цепочки отрицателен. Если это происходит и между интересующими нас причиной и следствием есть еще один посредник либо прямой эффект, то посредник будет сокращать суммарный эффект. В этом случае ваш первый посредник мог представлять, скажем, –25 % от суммарного эффекта, тогда как прямой эффект (или еще один посредник) представлял бы 125 % от суммарного эффекта, но сумма пропорций эффектов все равно будет составлять 100 %. Указанная ситуация является совершенно нормальной и ожидаемой, так что не позволяйте ей сбивать вас с толку.

Если она вас смущает, то, значит, вы в хорошей компании. Социологи десятилетиями спорили о том, должно ли наличие значительного суммарного эффекта быть предпосылкой для анализа опосредованности. Но существует масса совершенно законных регулирующих или саморегулирующихся явлений, в которых посредник компенсирует прямой эффект. Например, C-Mart, возможно, обнаружит, что изменения цен не влияют на ее объем продаж, как ожидалось, поскольку они являются причиной соответствующих изменений цен у ее конкурентов.

Когда посредник является двоичной переменной

Посредник в этом примере был числовой переменной, поэтому мы смогли легко получить опосредованный эффект путем умножения коэффициентов для двух участвующих стрелок. Когда посредник является двоичной переменной, можно по-прежнему квантифицировать опосредованность аналитически, т. е. с помощью уравнений, но формулы становятся запутаннее.

Если мы обозначим интересующую причину через X, посредника – через M и интересующий эффект – через Y, то регрессионные уравнения для посредника и окончательного эффекта примут следующий ниже вид:

$$P(M = 1) = \text{logistic}(\alpha_0 + a_X \cdot X);$$

$$Y = \beta_0 + \beta_X \cdot X + \beta_M \cdot M.$$

Обратите внимание, что первое уравнение теперь представляет собой логистическую регрессию, соответствующую двоичной переменной. Вместо того чтобы предсказывать значение M, как при линейной регрессии, мы теперь предсказываем вероятность того, что оно примет значение 1, $P(M = 1)$. Мы можем подставить эту вероятность во второе уравнение:

$$Y = \beta_0 + \beta_X \cdot X + \beta_M \cdot P(M = 1).$$

Прямой эффект по-прежнему легко вычислить: если X увеличивается на 1, то прямой эффект увеличивает Y на β_X. Но косвенный эффект теперь представляет дополнительную трудность, поскольку влияние X на M не является линейным. Следовательно, влияние X на M должно быть выявлено для опре-

деленного значения *X*. Этот вопрос аналогичен тому, с которым мы столкнулись, работая с модерацией в главе 11, и возможные решения те же:

- определить глобальную опорную точку, такую как среднее значение *X* в наших данных,
- либо рассчитать опосредованный эффект и процент опосредованности для каждой строки в наших данных, а затем рассчитать их соответствующие средние значения.

Как и в случае с главой 11, я рекомендую второй подход, модифицируемый согласно потребности, для того чтобы вписываться в рассматриваемое деловое решение.

Инструментальные переменные

Опосредование само по себе представляет собой отличное дополнение к инструментарию анализа поведенческих данных, но оно также является ступенькой к еще одному мощному инструменту, именуемому инструментальными переменными (instrumental variable, аббр. IV). В двух словах: инструментальные переменные задействуют известные посреднические связи, чтобы сокращать спутывающие систематические смещения в наших коэффициентах.

Одним из наиболее эффективных вариантов их использования является применение эксперимента для ответа на более широкий и нередко более сложный вопрос. Я проиллюстрирую это применение примером, связанным с удовлетворенностью клиентов, одной из наиболее наблюдаемых деловых метрик, но, по общему признанию, одной из самых сложных для измерения.

Как впервые упоминалось в главе 2, руководство AirCnC хочет знать влияние удовлетворенности клиентов (CSAT) на один из их ключевых индикаторов результативности – сумму, расходуемую в течение шести месяцев после данного бронирования, *6МесячныйРасход* (*M6Spend*). Мы повторно используем данные эксперимента главы 10, в которой мы занимались разведкой влияния изменений в процедурах кол-центра на удовлетворенность клиентов: а именно: «Вместо того чтобы постоянно извиняться, когда что-то пошло не так, представители кол-центра должны извиняться в начале взаимодействия, затем переходить в “режим решения проблем”, а потом заканчивать предложением клиенту нескольких вариантов».

Данные

Папка этой главы в репозитории на GitHub[1] содержит копию экспериментальных данных из главы 10. На этот раз мы включим в наш анализ переменную *6МесячныйРасход* (*M6Spend*). В табл. 12.1 приведены переменные в наших данных для этой главы.

[1] См. https://oreil.ly/BehavioralDataAnalysisCh12.

Таблица 12.1. Переменные в наших данных

	Описание переменной	chap10-experimental_data.csv
Center_ID (ИД центра)	Категориальная переменная для 10 кол-центров	✓
Rep_ID (ИД представителя)	Категориальная переменная для 193 представителей кол-центров	✓
Age (Возраст)	Возраст звонящего клиента, 20–60	✓
Reason (Основание)	Основание для звонка, оплата/объект недвижимости (payment/property)	✓
Call_CSAT (Удовлетворенность клиента звонком)	Удовлетворенность клиента звонком, 0–10	✓
Group (Группа)	Экспериментальное размещение, контрольная/процедурная группы (ctrl/treat)	✓
M6Spend (6МесячныйРасход)	Сумма, израсходованная на брони в пределах 6 месяцев с момента данного бронирования	✓

Пакеты

Далее мы будем использовать следующие ниже специальные пакеты для инструментальных переменных:

```
## Python
from linearmodels.iv import IV2SLS

## R
library(ivreg)
```

Понимание и применение инструментальных переменных

После того как вы познакомитесь с причинно-следственными диаграммами и опосредованием, выражение идеи, лежащей в основе инструментальных переменных, станет относительно простым:

> Давайте допустим, что у вас есть полностью опосредованная взаимосвязь между двумя переменными, а взаимосвязь между посредником и окончательной переменной запутанна. Тогда вы можете получить объективную оценку этой взаимосвязи, разделив коэффициент для суммарного эффекта на коэффициент для первого звена опосредования (т. е. взаимосвязь между первой переменной и посредником).

В целях ознакомления с тем, как это выглядит в нашем примере, давайте начнем с рисования причинно-следственной диаграммы, интересующих нас переменных. Мы хотим измерить причинно-следственную взаимосвязь между *УдовлетворенностьюКлиентов* и *6МесячнымРасходом*. Вполне вероятно, что высокая *УдовлетворенностьКлиентов* увеличивает сумму,

расходуемую на брони в последующие месяцы, но связь также подвержена воздействию со стороны неизмеримых спутывающих факторов, в том числе таких личностных черт, как открытость. Наконец, у нас есть данные о нашей экспериментальной процедуре, которая, как мы знаем, влияет на *УдовлетворенностьКлиентов* благодаря нашему эксперименту из главы 10 (рис. 12.8).

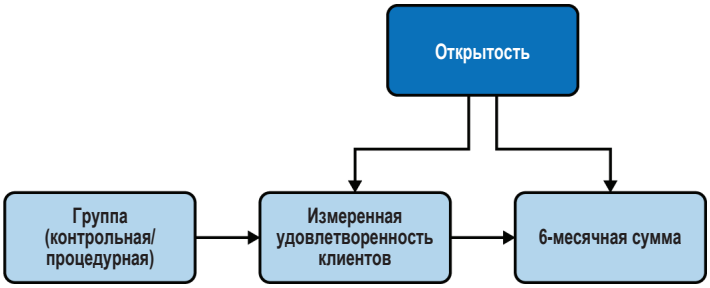

Рис. 12.8 ❖ Причинно-следственная диаграмма для интересующих нас переменных

История инструментальных переменных, т. н. корни причинно-следственных диаграмм

Инструментальные переменные были изобретены Филиппом и Сьюоллом Райтами (Philip Wright, Sewall Wright), отцом и сыном, которые также были среди изобретателей путевого анализа, предшественника причинно-следственных диаграмм. К сожалению, эта история происхождения потерялась на десятилетия, и экономисты использовали инструментальные переменные в качестве самостоятельного инструмента. Например, авторы ведущих вводных книг по эконометрии, Ангрист и Пишке (2009, 2014), вводят инструментальные переменные, ни разу не вводя причинно-следственные диаграммы (хотя они упоминают Райтов).

По моему опыту, использование инструментальных переменных без причинно-следственных диаграмм является упражнением по чесанию затылка и разглядыванию стен. Вам нужно постулировать потенциальный инструмент, который, казалось бы, возникает из воздуха, а затем обосновывать необходимые допущения в ходе длительных дискуссий. Это определенно можно делать, но для этого требуется неявное знание деловой или экономической ситуации, которое приходит только с опытом и которое трудно передавать. И наоборот, как мы увидим ниже, если эти знания выражены в форме тщательно проработанной и повторяющейся причинно-следственной диаграммы, то применение инструментальных переменных становится само собой разумеющимся делом.

Хорошо видно, что на этой причинно-следственной диаграмме *УдовлетворенностьКлиентов* является посредником между *Группой* и *6МесячнымРасходом*, но взаимосвязь между *УдовлетворенностьюКлиентов* и *6МесячнымРасходом* систематически смещена вверх из-за смешивающего эффекта *Открытости*.

В идеальном мире у нас были бы данные о переменной *Открытость*, и мы могли бы выполнить две истинные регрессии (уравнения (12.1) и (12.2)).

$$\textit{УдовлетворенностьКлиентов} = \beta_{g1} \cdot \textit{Группа} + \beta_{o1} \cdot \textit{Открытость}. \quad (12.1)$$

$$\textit{6МесячныйРасход} = \beta_{c2} \cdot \textit{УдовлетворенностьКлиентов} + \beta_{o2} \cdot \textit{Открытость}. \quad (12.2)$$

Но даже если бы мы и получили конкретные данные об *Открытости* (например, в ходе опроса), то каким образом можно было бы быть уверенным в том, что за углом не скрывается еще один спутывающий фактор?

Отступив на секунду от математики, этот вопрос подводит нас к самому сердцу поведенческой аналитики: удовлетворенность клиентов является важным критерием успеха в бизнесе, но поскольку на него столь сильно влияет мириада ненаблюдаемых индивидуальных характеристик, влияющих на поведения индивидуумов, мы не можем надеяться выявить и контролировать все эти характеристики. Указанная проблема не может быть удовлетворительно решена вне причинно-поведенческого каркаса, но ее легко решить внутри него.

Вернемся к математике – давайте пройдемся по изложенной ранее интуиции.

1. Рассчитать коэффициент для крайней левой связи между *Группой* и *УдовлетворенностьюКлиентов*.
2. Рассчитать коэффициент для суммарного эффекта *Группы* на *6МесячныйРасход*.
3. Рассчитать коэффициент влияния *УдовлетворенностиКлиентов* на *6МесячныйРасход*, разделив суммарный эффект из шага 2 на коэффициент для самой левой связи из шага 1.

Шаг 1: крайняя левая связь

Поскольку *Группа* является случайным размещением, не связанным с *Открытостью*, то мы можем выполнить следующую ниже регрессию вместо уравнения (12.1):

$$\textit{УдовлетворенностьКлиентов} = \beta_{g1} \cdot \textit{Группа}.$$

Наша оценка для β_{g1} является несмещенной и может быть вставлена в уравнение (12.1) согласно потребности, поскольку это истинный причинно-следственный коэффициент.

Шаг 2: суммарный эффект

Уравнение для суммарного эффекта (12.3) называется редуцированной регрессией (и мы будем индексировать ее буквой «*r*»), потому что она сворачивает цепочку между переменными.

Уравнение R

$$\textit{6МесячныйРасход} = \beta_{gr} \cdot \textit{Группа} \text{ (Уравн. } R\text{)}. \quad (12.3)$$

По тем же причинам, что и для первого звена опосредования, наша оценка β_{gr} является несмещенной.

Шаг 3: интересующая связь

Вот где происходит волшебство: как обсуждалось в предыдущем разделе об опосредовании, $\beta_{gr} = \beta_{c2} \times \beta_{g1}$. Мы можем переписать это уравнение как $\beta_{c2} = \beta_{gr} / \beta_{g1}$. Поскольку все переменные справа от знака равенства являются несмещенными, то и та, что слева от него, также является несмещенной. Эта оценка для β_{c2} является несмещенной.

Другими словами, мы можем распутать связь между двумя переменными, если сможем найти переменную (именуемую инструментальной), которая является причиной интересующей нас причины, но в остальном не связана как со спутывающим фактором, так и с интересующим нас эффектом:

- первое условие (именуемое допущением о независимости) необходимо для того, чтобы редуцированная регрессия была несмещенной, но, к счастью, при случайном распределении это всегда верно;
- второе условие (именуемое ограничением исключения) можно перефразировать следующим образом: связь между инструментом и интересующим следствием должна быть полностью опосредована интересующей причиной. Необходимо, чтобы уравнение $\beta_{gr} = \beta_{c2} \times \beta_{g1}$ было истинным. К сожалению, это невозможно доказать математически (для этого потребуется знать коэффициент для второго звена посредничества, который мы как раз и ищем!) и должно быть принято, основываясь на качественных причинно-следственных соображениях – в данном случае, например, размещение в экспериментальную группу вряд ли повлияет на *6МесячныйРасход* за пределами цепочки *УдовлетворенностиКлиентов*.

Измерение

Это была интуиция. Мы определенно могли бы рассчитать все соответствующие регрессии вручную, но, как мы уже забегали вперед, испортив всю игру, у аналитиков данных XXI века для этого есть пакет.

Прежде всего мы проведем две проверки исправности, выполнив линейные регрессии для первого звена опосредования и для суммарного эффекта (т. е. редуцированного уравнения). Если любая из них даст коэффициент, очень близкий к нулю (как определено бутстраповским интервалом уверенности), то это поставит под угрозу нашу регрессию инструментальных переменных. В большинстве случаев вы захотите включить какие-то другие ковариаты, которые причинно-следственно связаны с интересующими вас переменными. В нашем примере из главы 10 *Возраст* (возраст вызывающего абонента) и *Основание* (основание для звонка) были предсказательными для *УдовлетворенностиКлиентов* и включенными внутрь *Группы*. Мы должны включить их и сюда:

```
## Python (результат не показан)
ols("call_CSAT~group+age+reason", data=exp_data_df).fit(disp=0).summary()
ols("M6Spend~group+age+reason", data=exp_data_df).fit(disp=0).summary()

## R
summary(lm(call_CSAT~group+age+reason, data=exp_data))
summary(lm(M6Spend~group+age+reason, data=exp_data))
..
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.103826  0.011790   348.07   <2e-16 ***
Grouptreat     0.540633  0.006291    85.94   <2e-16 ***
age            0.020202  0.000280    72.14   <2e-16 ***
reasonproperty 0.200590  0.006600    30.39   <2e-16 ***
...
Coefficients:
               Estimate Std. Error  t value         Pr(>|t|)
(Intercept)    99.93195    0.43976  227.242          < 2e-16 ***
grouptreat      1.61687    0.23465    6.891 0.00000000000557 ***
age            -1.46785    0.01044 -140.536          < 2e-16 ***
reasonproperty  0.44458    0.24615    1.806           0.0709 .
...
```

К счастью, оба коэффициента безопасно отличаются от нуля, поэтому мы можем перейти к нашей регрессии инструментальных переменных.

Исходный код на Python

На языке Python мы будем использовать пакет `linearmodels`:

```
## Python
iv_mod = IV2SLS.from_formula('M6Spend ~ 1 + age + reason + [call_CSAT ~ group]',
                             exp_data_df).fit()
iv_mod.params
Out[8]:
Intercept             87.658610
reason[T.property]    -0.155326
age                   -1.528264
call_CSAT              2.990706
Name: parameter, dtype: float64
```

Синтаксис функции `IV2SLS.from_formula()` почти такой же, как и для `ols()`. Интересующая нас предсказываемая переменная находится слева от знака тильды («~»), а предсказатели – справа от нее, причем регрессия первого этапа записана в скобках. Тут следует отметить две вещи:

- вам необходимо включить константу («1») в явной форме в число предсказателей;
- другие ковариаты, связанные с интересующими вас переменными (здесь *Возраст* и *Основание*), также должны быть сюда включены, за скобками. Обратите внимание, что они автоматически будут включены

в регрессию первого этапа. Ваша формула для регрессии первого этапа должна включать только посредника, т. е. интересующую нас причину и инструмент.

Распечатка функции Python урезана, но это все, что нам нужно. Эффект *УдовлетворенностиКлиентов* от звонка (*call_CSAT*) на *6МесячныйРасход* составляет около 2.99 доллара за единицу, примерно на 1 доллар меньше, чем предполагала бы наивная, систематически смещенная регрессия *6МесячныхРасходов* на *УдовлетворенностиКлиентов* от звонка:

```
## Python
ols("M6Spend~call_CSAT+age+reason", data=exp_data_df).fit(disp=0).summary()
...
                     coef    std err      t      P>|t|    [0.025  0.975]
Intercept           83.2283   0.536    155.302   0.000   82.178  84.279
reason[T.property] -0.3582    0.245     -1.461   0.144   -0.839   0.122
call_CSAT           4.0019    0.076     52.767   0.000    3.853   4.151
age                -1.5488    0.010   -147.549   0.000   -1.569  -1.528
...
```

Бутстраповский 90%-ный интервал уверенности для несмещенного эффекта составляет приближенно [2.26; 3.89].

Исходный код на R

На языке R мы будем использовать пакет `ivreg`:

```
## R
> iv_mod <- ivreg::ivreg(M6Spend~call_CSAT + age + reason | group + age + reason,
    data=exp_data)
> summary(iv_mod)

Call:
ivreg::ivreg(formula = M6Spend ~ call_CSAT + age + reason | group +
    age + reason, data = exp_data)

Residuals:
   Min      1Q Median     3Q    Max
-86.82  -35.01 -17.94  19.92 706.58

Coefficients:
               Estimate Std. Error  t value         Pr(>|t|)
(Intercept)    87.65861    1.93745   45.244          < 2e-16 ***
call_CSAT       2.99071    0.43165    6.929 0.00000000000426 ***
age            -1.52826    0.01358 -112.540          < 2e-16 ***
reasonproperty -0.15533    0.25968   -0.598             0.55

Diagnostic tests:
                    df1    df2 statistic p-value
Weak instruments      1 231655  7384.847  <2e-16 ***
Wu-Hausman            1 231654     5.667  0.0173 *
Sargan 0 NA NA NA
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.12 on 231655 degrees of freedom
Multiple R-Squared: 0.0925,     Adjusted R-squared: 0.09249
Wald test:  7019 on 3 and 231655 DF,  p-value: < 2.2e-16
```

К счастью, авторы пакета `ivreg` предприняли сознательные усилия, чтобы сделать функцию `ivreg()` как можно более похожей на `lm()` как по синтаксису, так и по распечатке. Единственное отличие в формуле состоит в том, что теперь есть два списка регрессоров, которые разделяются вертикальной чертой «|». Вот куда нужно направить переменные:

- интересующая причина, здесь *УдовлетворенностьКлиентов* звонком, появляется только слева от вертикальной черты;
- инструмент(ы), здесь *Группа*, появляется только справа от вертикальной черты;
- другие объясняющие переменные, которые вносят вклад в интересующие нас причину или следствие, здесь *Возраст* и *Основание*, появляются по обе стороны от вертикальной черты.

Распечатка `ivreg()` также очень похожа на распечатку `lm()`. Интересующим нас значением является коэффициент для *УдовлетворенностиКлиентов* звонком. `ivreg()` также возвращает результаты нескольких диагностик для инструментальной переменной, которые тестируют прочность различных связей в нашей модели. Здесь опять-таки эффект *УдовлетворенностиКлиентов* звонком на *6МесячныйРасход* составляет около 2.99 доллара за единицу. Соответствующий бутстраповский 90%-ный интервал уверенности составляет приближенно [2.26; 3.89].

В нашем простом примере эта разница между двумя оценками обусловлена Открытостью. В реальной жизни маловероятно, что мы смогли бы уверенно выявить все участвующие в игре спутывающие факторы, которых мы могли бы вместо этого, если хотите, обозначить как «неизвестный психологический материал». Однако, даже не измеряя эти спутывающие факторы, мы можем измерить их влияние на *6МесячныйРасход* с помощью *УдовлетворенностиКлиентов*: изменение в этих спутывающих факторах, являющееся причиной роста *УдовлетворенностиКлиентов* на 1 пункт, также является причиной роста *6МесячногоРасхода* примерно на 1 доллар. Теперь мы можем провести опросы, чтобы измерить открытость или любой другой неизвестный психологический материал, зная, что, так сказать, спутывание стоимостью в 1 доллар на *6МесячномРасходе* в расчете на пункт *УдовлетворенностиКлиентов* требует объяснения. Мы знаем то, чего не знаем, а это, если вы меня спросите, довольно круто.

Применение инструментальных переменных: часто задаваемые вопросы

В предыдущем примере инструментальные переменные использовались для нижестоящего анализа экспериментальных данных: использование экспериментальных данных для распутывания причинно-следственных связей. Это одно из их самых простых и эффективных применений, но оно не единст-

венное. Познакомившись с инструментальными переменными, вы начнете задавать себе вопросы: «А что делать, если…?» Строить полные примеры для каждого потенциального варианта использования было бы излишним, но целесообразно назвать наиболее распространенные из них:

Можно ли использовать инструментальные переменные с чисто наблюдательными данными?

Да, и процесс точно такой же, как и с экспериментальными данными, но поскольку допущение о независимости в этом случае не является заданным, вам нужно будет обеспечить, чтобы оно соблюдалось. Я часто придумывал потенциальный инструмент только для того, чтобы немного позже осознать, что между инструментом и окончательным эффектом на заднем плане притаилась еще одна взаимосвязь.

Можно ли использовать инструментальные переменные с двоичным окончательным эффектом?

В таком случае взаимосвязи между коэффициентами значительно усложняются, чем при использовании линейных регрессий на протяжении всего процесса. Пакет языка R `ivprobit()` позволяет выполнять пробит-регрессию с инструментом, но, насколько я знаю, такого решения для логистической регрессии не существует.

Выводы

Вот мы и на месте. В начале книги я пообещал, что мы измерим причинно-следственное влияние удовлетворенности клиентов на деловую метрику, и мы как раз это и сделали: «Увеличение заявленной удовлетворенности клиентов на одну единицу увеличивает расходы в последующие шесть месяцев на 2.99 доллара». Никаких длинных предостережений и сносок, никаких размахиваний руками, дескать, «корреляция не есть каузация» – это самый четко очерченный результат, какой только может быть. И надеюсь, в вашем сознании открывается целый мир возможностей. Удовлетворенность клиентов, членство на основе программ лояльности, восприятие бренда: измерение деловых последствий для всех этих туманных и систематически смещенных концепций находится в пределах вашей досягаемости. И скорее всего, у вас уже есть необходимые данные. Проводил ли кто-нибудь из маркетологов эксперимент два года назад, предлагая скидку клиентам, если они подписались на вашу программу лояльности? Тогда это просто вопрос извлечения соответствующих данных и применения однострочной формулы для регрессии инструментальных переменных. Разумеется, эта простота не возникает из ниоткуда. Процесс становления занял немало времени, так как он требует:

- четко определенных и понятых переменных для удовлетворенности клиентов и расходования, как мы увидели в части I;
- правильной причинно-следственной диаграммы, как мы обнаружили в части II;

- инструментов, которые позволяют нам справляться с неопределенностью без необходимости запоминать кучу статистических тестов, как мы увидели в части III;
- хорошо продуманных и хорошо проанализированных экспериментов, как мы разведали в части IV;
- наконец, понимания модерации и опосредованности, как мы узнали в части V.

Заканчивая на несколько менее формальной ноте: часто отмечается, что дети проявляют бесконечное любопытство к окружающему их миру («Почему небо голубое?»). Конечно же, это любопытство на самом деле не бесконечно, и в какой-то момент большинство детей перестают задавать столь много вопросов. Я искренне надеюсь, что данная книга вновь разожжет в вас то детское любопытство настолько, что вы позволите себе снова быть заинтригованными миром (в особенности людьми) вокруг вас и будете задаваться вопросом: «Но почему?» Разумеется, прежде чем претендовать на какую-либо заслугу в этом, мне придется поразмыслить над вполне реальной возможностью того, что я понял причинно-следственную связь неправильно и что все это с самого начала отложилось у вас (рис. 12.9).

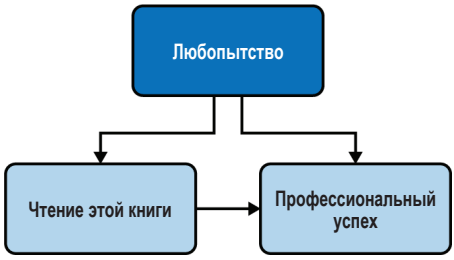

Рис. 12.9 ❖ Я разве не упоминал, что корреляция не есть каузация?

Библиография

Аберсон, Кристофер Л. Прикладной анализ мощности для бихевиористики (Aberson, Christopher L. *Applied Power Analysis for the Behavioral Sciences*. Abingdon, UK: Routledge, 2019).

Ангрист, Джошуа Д. и Йорн-Штеффен Пишке. В основном безобидная эконометрия: компаньон Эмпирика (Angrist, Joshua D., and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press, 2009). Настольный справочник по прикладной эконометрии с математикой и статистикой для выпускников.

Ангрист, Джошуа Д. и Йорн-Штеффен Пишке. Овладение «метриками»: Путь от причины к следствию (Angrist, Joshua D., and Jörn-Steffen Pischke. *Mastering "Metrics": The Path from Cause to Effect*. Princeton, NJ: Princeton University Press, 2014). Более доступная версия их классики 2009 года с добавлением классных справочных материалов!

Антонио, Нуно, Ана де Алмейда и Луис Нуньес. Бронирование отелей требует наборов данных (Antonio, Nuno, Ana de Almeida, and Luis Nunes. Hotel booking demand datasets. *Data in Brief 22* (Feb. 2019): 41–49, https://doi.org/10.1016/j.dib.2018.11.126).

Бертран, Марианна и Сендил Муллайнатан. Являются ли Эмили и Грег более привлекательными, чем Лакиша и Джамал? Полевой эксперимент по дискриминации на рынке труда (Bertrand, Marianne, and Sendhil Mullainathan. Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. *American Economic Review* 94, no. 4 (2004): 991–1013).

Коэн, Джейкоб. Статистический анализ мощности для бихевиористики, 2-е изд. (Cohen, Jacob. *Statistical Power Analysis for the Behavioral Sciences*, 2nd Edition. Abingdon, UK: Routledge, 2013). Классика по анализу мощности. Как и большинство классических книг, она несколько устарела в компьютерной имплементации, но все же является отличным ресурсом для более глубокого осмысления того, что такое статистическая мощность.

Каннингем, Скотт. Причинно-следственный вывод: микстейп (Cunningham, Scott. *Causal Inference: The Mixtape*. New Haven, CT: Yale University Press, 2021). Очень доступная вариация книг Ангриста/Пишке, все же предназначенная в первую очередь для академической сферы.

Дэвисон, А. С. и Д. В. Хинкли. Методы бутстрапирования и их применение (Davison, A. C., and D. V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge, UK: Cambridge University Press, 1997).

Эфрон, Брэдли и Р. Дж.Тибширани. Введение в бутстрап (Efron, Bradley, and R. J. Tibshirani. *An Introduction to the Bootstrap*. Abingdon, UK: Chapman and Hall/CRC, 1994). Эфрон является «изобретателем» бутстрапа, и эта книга представ-

ляет собой хорошее в него введение для людей, по меньшей мере имеющих представление о математике и статистике.

Эйял, Нир. Зацепленный: как создавать продукты, формирующие привычки (Eyal, Nir. *Hooked: How to Build Habit-Forming Products*. New York: Portfolio, 201).

Фандер, Дэвид К. Головоломка личности (David C. *The Personality Puzzle*. New York: W. W. Norton & Company, 2016). Отличное «продвинутое введение» в психологию личности, которое устраняет разрыв между популярной наукой и академическими исследованиями.

Гельман, Эндрю и Дженнифер Хилл. Анализ данных с использованием регрессии и многоуровневой/иерархической модели (Gelman, Andrew, and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Model*. Cambridge, UK: Cambridge University Press, 2006). Обеспечивает более глубокий анализ многоуровневых (т. е. кластеризованных) данных за счет более высокого порога математических и статистических знаний.

Гербер, Алан С. и Дональд П. Грин. Полевые эксперименты: дизайн, анализ и интерпретация (Gerber, Alan S., and Donald P. Green. *Field Experiments: Design, Analysis and Interpretation*. New York: W. W. Norton & Company, 2012). Моя настольная книга для более глубокого понимания экспериментального дизайна. Она содержит целый ряд статистических и математических формул, но они должны быть понятны любому, даже обладающему умеренным квантитативным образованием. Мой рекомендуемый ресурс, если вы хотите анализировать экспериментальные данные с помощью статистических тестов.

Гленнерстер, Рейчел и Кудзай Такавараша. Проведение рандомизированных оцениваний: практическое руководство (Glennerster, Rachel, and Kudzai Takavarasha. *Running Randomized Evaluations: A Practical Guide*. Princeton, NJ: Princeton University Press, 2013). Доступное введение в экспериментирование в контексте экономики развития и помощи развитию. Содержит обширные обсуждения эмпирических проблем, связанных с проведением экспериментов в реальном мире (т. е. не в онлайновом режиме).

Гордон, Бретт Р. и соавт. Сравнение подходов к измерению рекламы: подтверждающие данные из крупных полевых экспериментов в Facebook (Gordon, Brett R., et al. A Comparison of Approaches to Advertising Measurement: Evidence from Big Field Experiments at Facebook. *Marketing Science*, INFORMS, 38, no. 2 (2019): 193–225).

Хейс, Эндрю Ф. Введение в анализ опосредованности, модерации и условных процессов: подход на основе регрессии, 2-е изд. (Hayes, Andrew F. *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*, 2nd Edition. New York: Guilford Press, 2017).

Эрнан, Мигель. Причинно-следственные диаграммы: прими допущения, прежде чем делать выводы (Hernan, Miguel. *Causal Diagrams: Draw Your Assumptions Before Your Conclusions*). Онлайновый курс edX.org[1] (доступен с 6 мая 2021).

Хаббард, Дуглас У. Как измерять что-либо: определение ценности нематериальных активов в бизнесе (Hubbard, Douglas W. *How to Measure Any-*

[1] См. https://oreil.ly/dPtnt.

thing: Finding the Value of Intangibles in Business. Hoboken, NJ: Wiley, 2010). Очень проницательный и доступный взгляд на проведение измерений в бизнесе.

Хосе, Пол Э. Занимаясь статистическим опосредованием и модерацией (Jose, Paul E. *Doing Statistical Mediation & Moderation*. New York: Guilford Press, 2013). Полезное дополнение к Хейс (2017), хотя оно больше ориентировано на научные исследования и использует более мудреное программно-информационное обеспечение (то есть мудреное за пределами академических кругов и смежных областей!).

Жосс, Джули, Николас Тирни и Натали Виаланикс. Обзор задач CRAN: пропущенные данные. Всеобъемлющая сеть архивов R (Josse, Julie, Nicholas Tierney, and Nathalie Vialaneix. CRAN Task View: Missing Data. The Comprehensive R Archive Network, https://cran.r-project.org/web/views/MissingData.html).

Канеман, Дэниел. Думая, быстро и медленно (Kahneman, Daniel. *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux, 2013). Классик бихевиористики, автор одного из громких названий в этой области.

Литтл, Родерик Дж. А. и Дональд Б. Рубин. Статистический анализ с пропущенными данными. 3-е изд. (Little, Roderick J. A., and Donald B. Rubin. *Statistical Analysis with Missing Data*, 3rd Edition. Hoboken, NJ: Wiley, 2019). Третье издание классической книги по анализу пропущенных данных, написанной видными исследователями в этой области. Полезна для обзора статистической теории.

Медоуз, Донелла Х. Мышление системами: учебник (Meadows, Donella H. *Thinking in Systems: A Primer*. White River Junction, VT: Chelsea Green Publishing, 2008). Отличное общее введение в системное мышление.

Перл, Джуди. Причинно-следственная связь (Pearl, Judea. *Causality*. Cambridge, UK: Cambridge University Press, 2009). Более ранняя книга Перл о причинно-следственной связи с подробной математикой на уровне выпускника.

Перл, Джуди и Дана Маккензи. Книга вопросов «почему»: новая наука о причине и следствии (Pearl, Judea, and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. New York: Basic Books, 2018). Наиболее доступное введение в причинно-следственный анализ и причинно-следственные диаграммы, с которым я сталкивался до сих пор, написанное одними из передовых исследователей в этой области.

Сенге, Питер М. Пятая дисциплина: искусство и практика обучающейся организации (Senge, Peter M. *The Fifth Discipline: The Art & Practice of the Learning Organization*. New York: Currency, 2010).

Шипли, Билл. Причина и корреляция в биологии: руководство пользователя по анализу путей, структурным уравнениям и причинно-следственным выводам с использованием R. 2-е изд. (Shipley, Bill. *Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference with R*, 2nd Edition. Cambridge, UK: Cambridge University Press, 2016). Вы не биолог? И я тоже. Эта книга все же помогла мне углубить понимание причинно-следственных диаграмм, и при ограниченном числе книг по этой теме нищие не могут выбирать.

Тейлор, Аарон Б. и Дэвид П. Маккиннон. Четыре применения методов перестановки для тестирования модели с одним посредником (Taylor, Aaron B.,

and David P. MacKinnon. Four applications of permutation methods to testing a single-mediator model. *Behavioral Research Methods* 44, no. 3 (Sep. 2012): 806–44).

Талер, Ричард Х. и Касс Р. Санштейн. Толчок в верном направлении: совершенствование решений о здоровье, богатстве и счастье (Thaler, Richard H., and Cass R. Sunstein. *Nudge: Improving Decisions About Health, Wealth, and Happiness*. New York: Penguin, 2009).

ван Бюрен, Стеф. Гибкое вменение пропущенных данных. 2-е изд. (van Buuren, Stef. *Flexible Imputation of Missing Data*, 2nd Edition. Abingdon, UK: Chapman and Hall/CRC, 2018). Очень доступная презентация автора пакета mice для R с большим числом конкретных примеров. Автор разместил книгу в свободном доступе на своей личной странице, https://stefvanbuuren.name/fimd.

Вандервил, Тайлер. Объяснение в причинно-следственном выводе: методы опосредования и взаимодействия (VanderWeele, Tyler. *Explanation in Causal Inference: Methods for Mediation and Interaction*. Oxford, UK: Oxford University Press, 2015). Отличный мост между литературой по причинно-следственному анализу и литературой по социальным наукам о средствах массовой информации и опосредовании. Охватывает значительное число более сложных материалов, которых нет в других справочных материалах.

Вендел, Стивен. Дизайн для изменения поведения: применение психологии и экономики поведения. 2-е изд. (Wendel, Stephen. *Designing for Behavior Change: Applying Psychology and Behavioral Economics*, 2nd Edition. Sebastopol: O'Reilly, 2020).

Уилкокс, Рэнд Р. Основы современных статистических методов: существенное повышение мощности и точности. 2-е изд. (Wilcox, Rand R. *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy*, 2nd Edition. New York: Springer, 2010). Мы рутинно допускаем, что наши данные нормально распределены «в достаточной мере». Уилкокс показывает, что это допущение не обосновано и может серьезно смещать аналитические расчеты. Очень легко читаемая книга на эту продвинутую тему.

Предметный указатель

Q

R

Б

В

Г

Д

Ж

З

И

К

Л

М

Н

О

П

P

C

Т

У

Ф

Х

Ц

Ч

Ш

Э

Книги издательства «ДМК ПРЕСС»
можно купить оптом и в розницу
в книготорговой компании «Галактика»
(представляет интересы издательств
«ДМК ПРЕСС», «СОЛОН ПРЕСС», «КТК Галактика»).
Адрес: г. Москва, пр. Андропова, 38;
тел.: **(499) 782-38-89**, электронная почта: **books@alians-kniga.ru**.
При оформлении заказа следует указать адрес (полностью),
по которому должны быть высланы книги;
фамилию, имя и отчество получателя.
Желательно также указать свой телефон и электронный адрес.
Эти книги вы можете заказать и в интернет-магазине: **http://www.galaktika-dmk.com/**.

Флоран Бюиссон

Анализ поведенческих данных на R и Python

Главный редактор *Мовчан Д. А.*
dmkpress@gmail.com
Зам. главного редактора *Сенченкова Е. А.*
Перевод *Логунов А. В.*
Корректор *Синяева Г. И.*
Верстка *Чаннова А. А.*
Дизайн обложки *Мовчан А. Г.*

Гарнитура PT Serif. Печать цифровая.

Усл. печ. л. 29,9. Тираж 200 экз.

Веб-сайт издательства: **www.dmkpress.com**

O'REILLY

Задействуйте всю мощь поведенческих данных в своей компании, используя инструменты, специально разработанные для их анализа. Общепринятые алгоритмы науки о данных и инструменты предсказательной аналитики трактуют данные о поведении клиентов, такие как клики на веб-сайте или покупки в супермаркете, аналогично любым другим данным.
Однако в этой книге представлены мощные методы, специально приспособленные для анализа поведенческих данных.

«Эта книга уникальна тем, что она начинается с вопросов и задач и задействует технические приемы и языки программирования как настоящие инструменты. Читатели научатся решать невероятно важные и сложные задачи. Она стоит вашего времени и вложений».

Эрик Вебер,
руководитель
отдела экспериментов, Yelp

Усовершенствованный экспериментальный дизайн позволяет вам получать максимальную отдачу от ваших A/B-тестов, тогда как причинно-следственные диаграммы позволяют выявлять причины поведений, даже если вы не можете проводить эксперименты.
Книга написана в доступном стиле для исследователей данных, бизнес-аналитиков и бихевиористов. Приведены полные примеры и упражнения на языках R и Python, которые помогут вам получать более глубокую информацию о ваших данных — и не откладывая в долгий ящик.
Вы научитесь:

- понимать специфику поведенческих данных;
- разведывать различия между результатом измерения и результатом предсказания;
- очищать и подготавливать поведенческие данные;
- планировать и анализировать эксперименты для принятия оптимальных деловых решений;
- использовать поведенческие данные для понимания и измерения причинно-следственных связей;
- сегментировать клиентов тщательным образом.

Флоран Бюиссон – поведенческий экономист с 10-летним опытом работы в бизнесе, аналитике и бихевиористике. Еще недавно он основал и в течение четырех лет возглавлял научную группу по бихевиористике в страховой компании Allstate. Флоран публикует научные статьи в таких журналах, как рецензируемый журнал Journal of Real Estate Research, посвященный исследованиям в сфере недвижимости. Он имеет степень магистра эконометрии, а также степень доктора философии в области поведенческой экономики в Университете Сорбонны в Париже.

ISBN 978-5-97060-992-7

Интернет-магазин: www.dmkpress.com

Оптовая продажа: КТК «Галактика»
books@alians-kniga.ru

ДМК ПРЕСС издательство
www.дмк.рф