TURING 图灵程序设计丛书

统计思维

程序员数学之概率统计（第2版）

Think Stats

Second Edition

[美] Allen B. Downey 著
金迎 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北　京

图书在版编目（CIP）数据

统计思维 : 程序员数学之概率统计 : 第2版 / (美)唐尼 (Downey,A.B.) 著 ; 金迎译. -- 2版. -- 北京 : 人民邮电出版社, 2015.9
（图灵程序设计丛书）
ISBN 978-7-115-40108-3

Ⅰ. ①统… Ⅱ. ①唐… ②金… Ⅲ. ①概率统计 Ⅳ. ①O211

中国版本图书馆CIP数据核字(2015)第176735号

内 容 提 要

这是一本以全新视角讲解概率统计的入门书。抛开经典的数学分析，Downey手把手教你用编程理解统计学。具体说来，本书通过一个案例研究，介绍探索性数据分析的全过程：从收集数据、生成统计信息，到发现模式、验证假设。同时研究分布、概率规则、可视化和其他多种工具及概念。此外，第2版新增了回归、时间序列分析、生存分析和分析方法等章节。

本书既适合作为教材，又适合作为程序员学习概率统计的参考书，也适合作为非程序员了解概率统计与编程的工具书。

◆ 著 [美] Allen B. Downey
译 金 迎
责任编辑 岳新欣
执行编辑 张 庆
责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京 印刷

◆ 开本：800×1000 1/16
印张：12.75
字数：302千字 2015年9月第2版
印数：9001－13 000册 2015年9月北京第1次印刷
著作权合同登记号 图字：01-2015-2497号

定价：49.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

版权声明

O’Reilly Media, Inc.介绍

O’Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O’Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O’Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O’Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O’Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O’Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项 O’Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O’Reilly Radar 博客有口皆碑。”

——*Wired*

“O’Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O’Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O’Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

目录

前言……xi

第 1 章　探索性数据分析……1

1.1　统计学方法……2
1.2　全国家庭增长调查……2
1.3　数据导入……3
1.4　DataFrame……4
1.5　变量……6
1.6　数据变换……6
1.7　数据验证……8
1.8　解释数据……9
1.9　练习……10
1.10　术语……11

第 2 章　分布……13

2.1　表示直方图……14
2.2　绘制直方图……14
2.3　全国家庭增长调查中的变量……15
2.4　离群值……18
2.5　第一胎……18
2.6　分布概述……20
2.7　方差……21
2.8　效应量……21
2.9　报告结果……22
2.10　练习……23
2.11　术语……23

第 3 章　概率质量函数……25
3.1　概率质量函数……25
3.2　绘制 PMF……26
3.3　绘制 PMF 的其他方法……28
3.4　课堂规模悖论……29
3.5　使用 DataFrame 进行索引……31
3.6　练习……33
3.7　术语……34
第 4 章　累积分布函数……35
4.1　PMF 的局限……35
4.2　百分位数……36
4.3　CDF……37
4.4　表示 CDF……38
4.5　比较 CDF……39
4.6　基于百分位数的统计量……40
4.7　随机数……41
4.8　比较百分位秩……42
4.9　练习……43
4.10　术语……44
第 5 章　分布建模……45
5.1　指数分布……45
5.2　正态分布……48
5.3　正态概率图……49
5.4　对数正态分布……51
5.5　Pareto 分布……53
5.6　随机数生成……56
5.7　为什么使用模型……56
5.8　练习……57
5.9　术语……59
第 6 章　概率密度函数……61
6.1　PDF……61
6.2　核密度估计……63
6.3　分布框架……65
6.4　Hist 实现……65
6.5　Pmf 实现……66
6.6　Cdf 实现……67
6.7　矩……68

6.8 偏度……69
6.9 练习……72
6.10 术语……73

第 7 章 变量之间的关系……75

7.1 散点图……75
7.2 描述关系特征……78
7.3 相关性……79
7.4 协方差……80
7.5 Pearson 相关性……81
7.6 非线性关系……82
7.7 Spearman 秩相关……82
7.8 相关性和因果关系……83
7.9 练习……84
7.10 术语……85

第 8 章 估计……87

8.1 估计游戏……87
8.2 猜测方差……89
8.3 抽样分布……90
8.4 抽样偏倚……93
8.5 指数分布……93
8.6 练习……95
8.7 术语……95

第 9 章 假设检验……97

9.1 经典假设检验……97
9.2 假设检验……98
9.3 检验均值差……100
9.4 其他检验统计量……101
9.5 检验相关性……102
9.6 检验比例……103
9.7 卡方检验……104
9.8 再谈第一胎……105
9.9 误差……106
9.10 功效……107
9.11 复现……108
9.12 练习……109
9.13 术语……109

第 10 章　线性最小二乘法……111
10.1　最小二乘法拟合……111
10.2　实现……112
10.3　残差……113
10.4　估计……114
10.5　拟合优度……116
10.6　检验线性模型……118
10.7　加权重抽样……119
10.8　练习……121
10.9　术语……121
第 11 章　回归……123
11.1　StatsModels……124
11.2　多重回归……125
11.3　非线性关系……127
11.4　数据挖掘……128
11.5　预测……129
11.6　Logistic 回归……131
11.7　估计参数……132
11.8　实现……133
11.9　准确度……134
11.10　练习……135
11.11　术语……136
第 12 章　时间序列分析……139
12.1　导入和清洗数据……139
12.2　绘制图形……141
12.3　线性回归……143
12.4　移动平均值……144
12.5　缺失值……146
12.6　序列相关……148
12.7　自相关……149
12.8　预测……150
12.9　参考书目……154
12.10　练习……154
12.11　术语……155
第 13 章　生存分析……157
13.1　生存曲线……157
13.2　危险函数……159

13.3 估计生存曲线 160
13.4 Kaplan-Meier 估计 161
13.5 婚姻曲线 162
13.6 估计生存函数 163
13.7 置信区间 164
13.8 群组效应 166
13.9 外推 168
13.10 预期剩余生存期 169
13.11 练习 171
13.12 术语 172

第 14 章 分析方法 173

14.1 正态分布 173
14.2 抽样分布 174
14.3 表示正态分布 175
14.4 中心极限定理 176
14.5 检验 CLT 177
14.6 应用 CLT 180
14.7 相关检验 181
14.8 卡方检验 183
14.9 讨论 184
14.10 练习 184

作者介绍 186

封面介绍 186

前言

本书介绍探索性数据分析的实用工具，书中章节按照我自己处理数据集时遵循的步骤进行组织。

- 导入和清洗：无论数据格式如何，我们通常都需要花费一些时间和精力进行数据的读取、清洗和变换，并进行检查，以确保在此过程中信息完好无损。
- 单变量探索：通常情况下，我会首先逐个检查变量，弄清变量的意义，分析变量值的分布，选择合适的汇总统计量。
- 成对探索：为了发现变量之间的关系，我会分析表格和散点图，计算相关性并进行线性拟合。
- 多变量分析：如果变量之间存在明显关系，我就要使用多元回归以增加控制变量，从而研究更复杂的关联关系。
- 估计和假设检验：在汇报统计结果时，有 3 个重要问题需要回答。效应规模如何？再次运行同一测量时，预期的变化性有多大？这个明显的效应是否可能是偶然产生的？
- 可视化：在数据探索中，可视化是寻找可能关系和效应的一个重要工具。如果一个明显的效应是统计显著的，那么可视化可以帮助我们有效地展示结果。

本书采用的是计算方法。相比数学方法，计算方法具有如下优点。

- 大多数概念用 Python 代码进行展示，而非数学符号。总体而言，Python 代码的可读性更好，而且这些代码是可执行的，读者可以下载、运行并进行修改。
- 每一章都附有练习，可以帮助读者扩展并巩固知识。编写程序时，你把自己对知识的理解表达为代码；调试代码时，这些理解也可以得到修正。
- 一些练习使用了实验检验统计行为。例如，你可以通过生成随机样本并计算它们的总和来探索中心极限定理（Central Limit Theorem，CLT）。练习得到的可视化结果展示了 CLT 的工作原理及适用条件。

- 一些概念很难从数学角度进行理解，却很容易通过模拟掌握。例如，通过运行随机模拟对 *p* 值进行近似，可以增强我们对 *p* 值含义的理解。
- 由于本书使用通用编程语言（Python），因此读者几乎可以从任何数据源导入数据，而不必受限于使用特定统计工具进行了清洗和格式化的数据集。

本书使用基于项目的方法。在我的课堂上，学生需要完成一个为期一个学期的项目。在项目中，学生要提出一个统计问题，寻找可以解决这个问题的数据集，并将学到的各种技术应用于这个数据集。

为了展示我采用的统计分析方法，本书将介绍一个贯穿各章的案例。这个案例使用的数据来自以下两方面资源。

- 全国家庭增长调查（National Survey of Family Growth，NSFG），这一调查由美国疾病控制和预防中心（Center for Disease Control and Prevention，CDC）开展，以收集“与家庭生活、婚姻状况、妊娠情况、生育情况、避孕情况，以及两性健康相关的信息”。参见 http://cdc.gov/nchs/nsfg.htm。
- 行为危险因素监测系统（Behavioral Risk Factor Surveillance System，BRFSS），由国家慢性病预防和健康促进中心（National Center for Chronic Disease Prevention and Health Promotion）主持，以“跟踪美国的健康状况及风险行为”。参见 http://cdc.gov/BRFSS/。

其他示例使用的数据来自美国国税局（IRS）、美国人口普查（U.S. Census）及波士顿马拉松赛（Boston Marathon）。

《统计思维》的第 2 版包含了第 1 版的各章，但对其中很多内容进行了大幅修改，并新增了关于回归、时间序列分析、生存分析和分析方法的章节。本书第 1 版没有使用 pandas、SciPy 和 StatsModels，所以这些内容也都是新增的。

写作思路

人们在编写新教材时，通常会参考已有教材。这样做的结果就是，大部分图书都采用相似的结构顺序叙述相似的内容。

我没有这样做。实际上，在撰写本书时出于以下几点考虑，我几乎没有使用任何纸质资料。

- 我的目的是探索新方法，因此不想过多介绍已有方法。
- 既然本书的授权是免费的，那么我希望书中所有的内容都不受版权限制。
- 我的很多读者都无法到提供纸质资料的图书馆去，因此我尽量引用互联网上的免费资源。

- 一些传统媒体的支持者认为，只使用电子资料是偷懒且不可靠的做法。关于偷懒，他们可能说对了，但是我不认为电子资料不可靠，因此希望对自己的理论进行验证。

我使用最多的资源是维基百科（Wikipedia）。总的来说，我在维基百科上读到的统计资料都很不错（但是我在后续也做了一些小的改动）。本书多处引用了维基百科页面，希望你能通过提供的链接阅读这些资料。很多时候，维基百科页面是本书内容的补充。除去我认为必要的修改，书中使用的术语和符号与维基百科基本一致。另外两个我觉得有用的资源是 Wolfram Mathworld 和 Reddit 统计论坛（http://www.reddit.com/r/statistics）。

本书代码

本书使用的代码和数据都可从 GitHub（htts://github.com/AllenDoweny/ThinkStats2）下载。Git 是一个版本管理系统，可以对项目文件进行跟踪。受 Git 管理的文件集称为*代码库*（repository）。GitHub 是一项托管服务，可以存储 Git 代码库，并提供一个便于使用的 Web 接口。

我的 GitHub 主页提供以下几种使用代码的方法。

- 你可以点击 Fork 按钮，在 GitHub 上创建该代码库的副本。如果你还没有 GitHub 账号，就需要创建一个。创建副本之后，你就在 GitHub 上拥有了自己的代码库，可以跟踪学习本书时编写的代码。之后你可以复制这个代码库，即将文件复制到自己的计算机上。
- 或者，你也可以复制我的代码库。这一操作不需要 GitHub 账号，但是你对代码所做的修改无法写回 GitHub。
- 如果你完全不想使用 Git，那么可以点击 GitHub 页面右下角的按钮，下载文件的 Zip 包。

本书所有代码都无需翻译即可在 Python 2 和 Python 3 中直接运行。

编写本书代码时，我使用的是 Continuum Analytics 的 Anaconda，这是一个免费的 Python 版本，其中带有运行本书代码所需的所有软件包（还有很多其他包）。Anaconda 很容易安装。默认情况下，Anaconda 进行用户级而非系统级安装，因此不需要管理员权限。Anaconda 同时支持 Python 2 和 Python 3，你可以从 Continuum（http://continuum.io/downloads）进行下载。

如果你不想使用 Anaconda，那么需要安装以下软件包。

- pandas，进行数据的表示和分析。下载地址为：http://pandas.pydata.org/。
- NumPy，支持基本的数字运算。下载地址为：http://www.numpy.org/。

- SciPy，进行科学计算，包括统计运算。下载地址为：http://www.scipy.org/。
- StatsModels，进行回归分析和其他统计分析。下载地址为：http://statsmodels.sourceforge.net/。
- matplotlib，支持可视化。下载地址为：http://matplotlib.org/。

虽然这些都是常用软件包，但并不是所有的 Python 安装都包含这些包，而且在有些环境下很难进行安装。如果你无法安装这些包，我强烈建议你使用 Anaconda，或者包含这些包的其他 Python 版本。

当你复制完代码库或者将 Zip 包解压后，会得到一个名为 ThinkStats2/code 的文件夹，其中有一个 nsfg.py 文件。运行 nsfg.py 会读取一个数据文件，运行一些测试，并输出一条消息，如“All tests passed”。如果你得到的是 import error，可能是因为缺少某些必要的软件包。

本书的大部分练习都使用 Python 脚本，但也有一些使用 IPython 记事本。如果你之前没有用过 IPython 记事本，可以访问文档 http://ipython.org/ipython-doc/stable/notebook/notebook.html 得到帮助。

本书读者应该熟悉 Python 的核心功能，包括面向对象的特征，但无需具备 pandas、NumPy 和 SciPy 知识。如果你已经熟知这些模块，可以跳过一些相关小节。

本书读者应该了解基本的数学知识，例如对数和求和。本书中有几处会涉及微积分概念，但你无需进行微积分运算。

如果你从未学习过统计学，本书会是一本很好的入门教材。如果你学习过传统的统计学课程，那么我希望本书能够修正你过去接受的一些错误观点。

—

Allen B. Downey 是一位计算机科学教授，执教于美国马萨诸塞州尼德姆的富兰克林欧林工程学院。

致谢

如果你有任何建议或者更正，请发送电子邮件至 downey@allendowney.com。如果我采纳了你的意见并对书中内容进行了修改，会将你的名字会加入致谢列表（除非你拒绝这样做）。

请在邮件中给出存在错误的句子，或句子的一部分，以便我进行搜索。当然，提供页码和章节号也可以，但还是以句子内容为佳。谢谢！

- Lisa Downey 和 June Downey 阅读了本书初稿，作出了很多更正，也提出了很多建议。
- Steven Zhang 发现了数处错误。

- Andy Pethan 和 Molly Farison 帮助调试了一些解决方案，Molly 还发现了数处拼写错误。
- Andrew Heine 发现了 error 函数中的一个错误。
- Dr. Nikolas Akerblom 知道一只始祖马有多大。
- Alex Morrow 对一个代码示例进行了说明。
- Jonathan Street 在关键时刻发现了一个错误。
- Gábor Lipták 发现了本书的一处拼写错误及接力赛问题的解决方案。
- 非常感谢 Kevin Smith 和 Tim Arnold 设计了 plasTeX，让我可以将本书转为 DocBook。
- George Caplan 的建议使本书结构更加清晰。
- Julian Ceipek 发现了一处错误和很多拼写错误。
- Stijn Debrouwere、Leo Marihart III、Jonathan Hammler 和 Kent Johnson 更正了本书第 1 印次中的错误。
- Dan Kearney 发现了一处拼写错误。
- Jeff Pickhardt 发现了一个损坏的链接和一处拼写错误。
- Jörg Beyer 发现了书中的拼写错误，并对书中代码的帮助文档进行了许多修正。
- Tommie Gannert 发送了一个补丁文件，其中包括许多更正。
- Alexander Gryzlov 对一个练习中的说明提出了建议。
- Martin Veillette 报告了一个 Pearson 相关性公式中的一处错误。
- Christoph Lendenmannt 提交了数个勘误。
- Haitao Ma 发现了一处拼写错误，并告诉了我。
- Michael Kearney 提出了很多极佳的建议。
- Alex Birch 提出了一些很有益的建议。
- Lindsey Vanderlyn、Griffin Tschurwald 和 Ben Small 阅读了本书的早期版本，并发现了很多错误。
- John Roth、Carol Willing 和 Carol Novitsky 进行了技术审阅，发现了很多错误，并提出了许多有益的建议。
- Rohit Deshpande 发现了一处排版错误。
- David Palmer 提出了很多建议，作出了不少更正。
- Erik Kulyk 发现了很多拼写错误。

Safari® Books Online

Safari Books Online 是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的首选资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。

成为 Safari Books Online 的会员，你即可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBMRedbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，请访问我们的网站（http://www.safaribooksonline.com）。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://bit.ly/think_stats_2e

对于本书的评论和技术性问题，请发送电子邮件到：bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

http://www.oreilly.com

我们在 Facebook 的地址如下：http://facebook.com/oreilly。

请关注我们的 Twitter 动态：http://twitter.com/oreillymedia。

我们的 YouTube 视频地址如下：http://www.youtube.com/oreillymedia。

第1章

探索性数据分析

如果能将数据与实际方法相结合，就可以在存在不确定性时解答问题并指导决策，这就是本书的主题。

举个例子。我的妻子在怀第一胎时，我听到了一个问题：第一胎是不是经常晚于预产期出生？下面所给出的案例研究就是由这个问题引出的。

如果用谷歌搜索这个问题，会看到大量的讨论。有人认为第一胎的生产日期确实经常晚于预产期，有人认为这是无稽之谈，还有人认为恰恰相反，第一胎常常会早产。

在很多此类讨论中，人们会提供数据来支持自己的观点。我发现很多论据是下面这样的。

> “我有两个朋友最近都刚生了第一个孩子，她们都是超过预产期差不多两周才出现临产征兆或进行催产的。”
> “我的第一个孩子是过了预产期两周才出生的，我觉得第二个孩子可能会早产两周！”
> “我认为这种说法不对，因为我姐姐是头生子，而且是早产儿。我还有好些表兄妹也是这样。”

这些说法都是基于未公开的数据，通常来自个人经验，因此称为轶事证据（anecdotal evidence）。在闲聊时讲讲轶事当然无可厚非，所以我并不是要批评以上那几个人。

但是，我们可能需要更具说服力的证据以及更可靠的回答。如果按照这个标准进行衡量，轶事证据通常都靠不住，原因有如下几点。

- 观测值数量较小
  如果第一胎的孕期的确偏长，这个时间差与正常的偏差相比可能很小。在这种情况下，我们可能需要比对大量的孕期数据，才能确定这种时间差确实存在。

- 选择数据时存在偏倚
  人们之所以参与这个问题的讨论，有可能是因为自己的第一个孩子出生较晚。这样的话，这个选择数据的过程就会对结果产生影响。

- 确认数据时存在偏倚
  赞同这种说法的人也许更可能提供例子进行佐证。持怀疑态度的人则更可能引用反例。

- 不精确
  轶事通常都是个人经验，经常会记错、误传或者误解等。

那我们该如何更好地回答这个问题呢？

1.1 统计学方法

为了解决轶事证据的局限性，我们将使用以下统计学工具。

- 数据收集
  我们将使用大型的全国性调查数据，这个调查专门设计用于对美国人口进行有效的统计推断。

- 描述性统计
  得出统计量，对数据进行简要的汇总，并评估可视化数据的不同方法。

- 探索性数据分析
  寻找各种模式、差异，以及其他能够解决我们感兴趣的问题的特征，同时还将检查数据的不一致性，发现局限性。

- 估计
  使用样本数据来估计一般总体的统计特征。

- 假设检验
  如果看到明显的效应，例如两个群组之间存在差异，将衡量该效应是否是偶然产生的。

谨慎执行上面的步骤，并避免各种错误，我们就可以获得合理性和准确性更高的结论。

1.2 全国家庭增长调查

从 1973 年起，美国疾病控制和预防中心（CDC）就开始进行全国家庭增长调查（NSFG，

http://cdc.ogv/nchs/nsfg.htm），以收集“与家庭生活、婚姻状况、妊娠情况、生育情况、避孕情况，以及两性健康相关的信息。此项调查的结果用于……进行健康服务和健康教育项目的规划，以及对家庭、生育及健康情况进行统计研究”。

我们将使用这项调查收集到的数据研究第一胎是否出生较晚，并解答一些其他问题。为了有效地使用这些数据，我们必须理解这项研究是如何设计的。

全国家庭增长调查是一项*横截面*（cross-sectional）研究，也就是说该研究捕获的是一个群组在某一时刻的快照。在横截面研究之外，最常见的是*纵向*（longitudinal）研究，指在一个时间段内重复观察一个群组。

全国家庭增长调查进行过 7 次，每一次都称为一个*周期*（cycle）。我们将使用第 6 次的数据，其时间段为 2002 年 1 月至 2003 年 3 月。

这项调查的目的是对一个*总体*（population）得出结论。全国家庭增长调查的目标总体是居住在美国、年龄在 15~44 岁的人。理想情况下，调查要收集这个总体中每个成员的数据，但这是不可能实现的。实际上，我们收集了这个总体的一个子集的数据，这个子集称为*样本*（sample）。参与调查的人称为*调查参与者*（respondent）。

通常来说，横截面研究应该是*有代表性*（representative）的，也就是说目标总体中每个成员参与调查的机会均等。这种理想条件在实践中很难实现，但是进行调查的人员会竭尽所能满足这个条件。

全国家庭增长调查不具有代表性，而是特意进行*过度抽样*（oversample）。这项研究的设计者招募了拉美裔美国人、非洲裔美国人和青少年 3 个群组的参与者，每个群组的招募比例都超过其在美国人口中所占的比例，以确保各群组的参与者数量足够多，从而进行有效的统计推断。

当然，过度抽样也有缺点，那就是不容易从调查的统计数据中得出关于总体的结论。我们稍后会对此进行讨论。

在使用这种调查数据时，我们必须熟悉*代码本*（codebook），这一点非常重要。代码本记录了一项研究的设计、使用的调查问题，以及调查中响应变量的编码。你可以从美国疾病控制和预防中心的网站（http://www.cdc.gov/nchs/nsfg/nsfg_cycle6.htm）下载全国家庭增长调查数据的代码本和使用手册。

1.3 数据导入

本书所用的代码和数据都可以通过 GitHub（https://github.com/AllenDowney/ThinkStats2）获取。前言中介绍了如何下载和使用这些代码。

下载代码后，你会得到一个名为 ThinkStats2/code 的文件夹，其中包含一个名为 nsfg.py

的文件。运行 nsfg.py 会读取数据文件，执行测试，然后打印出一条消息，例如“All test passed”。

让我们看看这个文件所执行的工作。第 6 次全国家庭增长调查的妊娠数据保存在名为 2002FemPreg.dat.gz 的文件中，这是一个纯文本（ASCII 码）形式的 gzip 压缩文件，有固定宽度的列。这个文件中的每一行都是一个记录（record），包含一次妊娠的数据。

2002FemPreg.dct 是一个 Stata 字典文件，记录了数据文件的格式。Stata 是一个统计软件。Stata“字典”是由变量名、变量类型及标识变量位置的索引值组成的列表。

下面几行摘自 2002FemPreg.dct：

```
infile dictionary {
  _column(1)  str12  caseid    %12s  "RESPONDENT ID NUMBER"
  _column(13) byte   pregordr   %2f  "PREGNANCY ORDER (NUMBER)"
}
```

这个字典描述了两个变量：caseid 是一个长度为 12 的字符串，代表调查参与者的 ID；pregorder 是一个单字节整数，说明这条记录描述的是这位调查参与者的第几次妊娠。

下载的代码包含一个 thinkstats2.py 文件，这是一个 Python 模块，包含了本书中用到的很多类和函数，其中有读取 Stats 字典和全国家庭增长调查数据文件的函数。这两个函数在 nsfg.py 中的用法如下：

```
def ReadFemPreg(dct_file='2002FemPreg.dct',
                dat_file='2002FemPreg.dat.gz'):
    dct = thinkstats2.ReadStataDct(dct_file)
    df = dct.ReadFixedWidth(dat_file, compression='gzip')
    CleanFemPreg(df)
    return df
```

ReadStataDct 的参数是字典文件名，返回值 dct 是一个 FixedWidthVariables 对象，其中包含从字典文件中得到的信息。dct 对象提供 ReadFixdWidth 方法进行数据文件的读取。

1.4 DataFrame

ReadFixedWidth 方法返回一个 DataFrame 对象。DataFrame 是 pandas 提供的基础数据结构。pandas 是一个 Python 数据和统计包，它的使用会贯穿本书。在 DataFrame 中，每个记录为一行（在我们的例子中就是每个妊娠数据为一行），每个变量为一列。

除了数据，DataFrame 还包含变量名和变量类型信息，并提供访问和修改数据的方法。

如果打印 df 对象，你会看到其中行列的部分数据和 DataFrame 的大小：13 593 行 / 记录，244 列 / 变量。

```
>>> import nsfg
>>> df = nsfg.ReadFemPreg()
>>> df
...
[13593 rows x 244 columns]
```

df 的 columns 属性将列名返回为一列 Unicode 字符串。

```
>>> df.columns
Index([u'caseid', u'pregordr', u'howpreg_n', u'howpreg_p', ... ])
```

df.columns 的结果是一个 Index 对象，Index 也是一个 pandas 数据结构。我们稍后会详细介绍 Index，现在可以暂时将其视为一个列表。

```
>>> df.columns[1]
'pregordr'
```

要访问 DataFrame 中的一列，你可以将列名作为键值。

```
>>> pregordr = df['pregordr']
>>> type(pregordr)
<class 'pandas.core.series.Series'>
```

其结果是一个 Series 对象，这又是一个 pandas 数据结构。Series 与 Python 列表类似，还能提供一些附加功能。打印一个 Series 对象会得到索引和对应的数值。

```
>>> pregordr
0     1
1     2
2     1
3     2
...
13590    3
13591    4
13592    5
Name: pregordr, Length: 13593, dtype: int64
```

这个示例中的索引是从 0 到 13 592 的整数，但通常索引可以使用任何可排序的数据类型。这个示例中的元素也是整数，但元素可以是任何类型的。

示例中的最后一行列出了变量名、Series 长度和数据类型。int64 是 NumPy 提供的类型之一。如果在 32 位机器上运行这个示例，得到的数据类型可能是 int32。

你可以使用整数的 index 和 slice 值访问 Series 中的元素。

```
>>> pregordr[0]
1
>>> pregordr[2:5]
2    1
3    2
4    3
Name: pregordr, dtype: int64
```

index 操作符的结果是 int64，slice 的结果还是一个 Series。

你也可以使用点标记法来访问 DataFrame 中的列。

```
>>> pregordr = df.pregordr
```

只有当列名为合法的 Python 标识符时（即以字母开头，不包含空格等），才能使用这种写法。

1.5 变量

我们已经使用了全国家庭增长调查数据集中的两个变量——caseid 和 pregordr，还看到数据集中共有 244 个变量。本书的探索性分析用到如下变量。

- caseid：调查参与者的整数 ID。
- prglength：妊娠周数，是一个整数。
- outcome：怀孕结果的整数代码。1 代表成功生产。
- pregordr：妊娠的顺序号。例如，一位调查参与者的第一次妊娠为 1，第二次为 2，以此类推。
- birthord：成功生产的顺序号，一位调查参与者的第一个孩子代码为 1，以此类推。对没有成功生产的其他妊娠结果，此字段为空。
- birthwgt_lb 和 birthwgt_oz：新生儿体重的磅部分数值和盎司部分数值。
- agepreg：妊娠结束时母亲的年龄。
- finalwgt：调查参与者的统计权重。这是一个浮点数，表示这位调查参与者在全美人口中代表的人数。

如果你仔细阅读了代码本，就会发现这些变量中很多都是*重编码*（recode），也就是说这些不是调查收集的*原始数据*（raw data），而是使用原始数据计算得到的。

例如，如果成功生产，prglngth 的值就与原始变量 wksgest（妊娠周数）相等；否则，prglngth 的值估算为 mosgest * 4.33（妊娠月数乘以一个月的平均周数）。

重编码通常都基于一定的逻辑，这种逻辑用于检查数据的一致性和准确性。一般情况下，如果数据中存在重编码，我们就直接使用，除非有特殊的原因需要自己处理原始数据。

1.6 数据变换

导入调查数据时，经常需要检查数据中是否存在错误，处理特殊值，将数据转换为不同的格式并进行计算。这些操作都称为*数据清洗*（data cleaning）。

nsfg.py 包含一个 CleanFemPreg 函数，用于清洗计划使用的变量。

```
def CleanFemPreg(df):
```

```
    df.agepreg /= 100.0

    na_vals = [97, 98, 99]
    df.birthwgt_lb.replace(na_vals, np.nan, inplace=True)
    df.birthwgt_oz.replace(na_vals, np.nan, inplace=True)

    df['totalwgt_lb'] = df.birthwgt_lb + df.birthwgt_oz / 16.0
```

agepreg 包含母亲在妊娠结束时的年龄。在数据文件中，agepreg 是以百分之一年为单位的整数值。因此 CleanFemPreg 的第一行将每个 agepreg 除以 100，从而获得以年为单位的浮点数值。

birthwgt_lb 和 birthwgt_oz 包含成功生产时的新生儿体重，分别是磅和盎司的部分。这两个变量还使用几个特殊的代码。

```
97 NOT ASCERTAINED
98 REFUSED
99 DON'T KNOW
```

用数字编码特殊值是一种*危险*的做法，因为如果没有进行正确的处理，这些数字可能产生虚假结果，例如，99 磅重的新生儿。replace 方法可以将这些值替换为 np.nan，这是一个特殊的浮点数值，表示“不是数字”。replace 方法使用 inplace 标识，说明直接修改现有的 Series 对象，而不是创建新对象。

IEEE 浮点数表示法标准中规定，在任何算术运算中，如果有参数为 nan，结果都返回 nan。

```
>>> import numpy as np
>>> np.nan / 100.0
nan
```

因此使用 nan 进行计算会得到正确的结果，而且大部分的 pandas 函数都能恰当地处理 nan。但我们经常需要处理数据缺失的问题。

CleanFemPreg 函数的最后一行生成一个新列 totalwgt_lb，将磅和盎司值结合在一起，得到一个以磅为单位的值。

需要注意的是，向 DataFrame 添加新列时，必须使用如下字典语法：

```
# 正确
df['totalwgt_lb'] = df.birthwgt_lb + df.birthwgt_oz / 16.0
```

而不是使用点标记：

```
# 错误!
df.totalwgt_lb = df.birthwgt_lb + df.birthwgt_oz / 16.0
```

使用点标记的写法会给 DataFrame 对象添加一个新属性，而不是创建一个新列。

1.7 数据验证

当数据从一个软件环境导出，再导入另一个环境时，可能会产生错误。如果不熟悉新数据集，可能会对数据进行不正确的解释，或者引入其他的误解。如果能抽出一些时间进行数据验证，就可以节省后续可能花费的时间，避免可能出现的错误。

验证数据的一种方法是计算基本的统计量，并与已发布的结果进行比较。例如，全国家庭增长调查的代码本为每个变量提供了概要表。outcome 变量对每个妊娠结果进行了编码，其概要表如下：

```
value label       Total
1 LIVE BIRTH          9148
2 INDUCED ABORTION    1862
3 STILLBIRTH           120
4 MISCARRIAGE         1921
5 ECTOPIC PREGNANCY    190
6 CURRENT PREGNANCY    352
```

Series 类提供了一个 value_counts 方法，可用于计算每个值出现的次数。如果得到 DataFrame 中的 outcome Series，我们可以使用 value_counts 方法，将结果与已发布的数据进行比较。

```
>>> df.outcome.value_counts().sort_index()
1    9148
2    1862
3     120
4    1921
5     190
6     352
```

value_counts 返回的结果是一个 Series 对象。sort_index 方法将 Series 对象按索引排序，使结果按序显示。

我们将得到的结果与官方发布的表格进行对比，outcome 变量的值似乎没有问题。类似地，已发布的关于 birthwgt_lb 的概要表如下：

```
value label                Total
. INAPPLICABLE           4449
0-5 UNDER 6 POUNDS         1125
6 6 POUNDS               2223
7 7 POUNDS               3049
8 8 POUNDS               1889
9-95 9 POUNDS OR MORE        799
```

birthwgt_lb 的 value_counts 结果如下：

```
>>> df.birthwgt_lb.value_counts(sort=False)
0        8
```

```
1       40
2       53
3       98
4      229
5      697
6     2223
7     3049
8     1889
9      623
10     132
11      26
12      10
13       3
14       3
15       1
51       1
```

数值 6、7、8 的出现次数是正确的。如果计算出 0~5 和 9~95 的次数，结果也是正确的。但是，如果再看仔细些，你会发现有一个数值肯定是错的——一个 51 磅的新生儿！

为了处理这个错误，可以在 `CleanFemPreg` 中加入一行代码。

```
df.birthwgt_lb[df.birthwgt_lb > 20] = np.nan
```

这行代码将非法值替换为 `np.nan`。方括号中的表达式产生一个 `bool` 类型的 Series 对象，值为 True 表示满足该条件。当一个布尔 Series 用作索引时，它只选择满足该条件的元素。

1.8 解释数据

要想有效使用数据，就必须同时在两个层面上思考问题：统计学层面和上下文层面。

例如，让我们看一看几位调查参与者的 `outcome` 序列。由于数据文件的组织方式，我们必须进行一些处理才能得到每位调查参与者的妊娠数据。以下函数实现了我们需要的处理：

```
def MakePregMap(df):
    d = defaultdict(list)
    for index, caseid in df.caseid.iteritems():
        d[caseid].append(index)
    return d
```

`df` 是包含妊娠数据的 DataFrame 对象。`iteritems` 方法遍历所有妊娠记录的索引（行号）和 `caseid`。

`d` 是将每个 `caseID` 映射到一列索引的字典。如果你不熟悉 `defaultdict`，可以到 Python 的 `collections` 模块中查看其定义。使用 `d`，我们可以查找一位调查参与者，获得其妊娠数据的索引。

下面的示例就查找了一位调查参与者，并打印出其妊娠结果列表：

```
>>> caseid = 10229
>>> indices = preg_map[caseid]
>>> df.outcome[indices].values
[4 4 4 4 4 4 1]
```

`indices` 是调查参与者 `10229` 的妊娠记录索引列表。

以这个列表为索引可以访问 `df.outcome` 中指定的行，获得一个 Series。上面的示例没有打印整个 Series 对象，而是选择输出 `values` 属性，这个属性是一个 NumPy 数组。

输出结果中的代码 `1` 表示成功分娩。代码 `4` 表示流产，即自发终止的妊娠，终止原因通常未知。

从统计学上看，这位调查参与者并无异常。流产并不少见，其他一些调查参与者的流产次数相同或者更多。

但是考虑到上下文，这个数据说明一位妇女怀孕 6 次，每次都以流产告终。她第 7 次也是最近一次怀孕成功产下了孩子。如果我们抱着同情心看待这些数据，就很容易被数据背后的故事感动。

全国家庭增长调查数据集中的每一条记录都代表一位参与者，这些参与者诚实地回答了很多非常私密而且难以回答的问题。我们可以使用这些数据解答与家庭生活、生育和健康相关的统计学问题。同时，我们有义务思及这些数据所代表的参与者，对他们心存敬意和感谢。

1.9 练习

- 练习 1.1

你下载的代码中应该有一个名为 chap01ex.ipynb 的文件，这是一个 IPython 记事本。你可以用如下命令从命令行启动 IPython 记事本：

```
$ ipython notebook &
```

如果系统安装了 IPython，会启动一个在后台运行的服务器，并打开一个浏览器查看记事本。如果你不熟悉 IPython，我建议你从 IPython 网站（http://ipython.org/ipython-doc/stable/notebook/notebook.html）开始学习。

你可以添加一个命令行选项，使图片在“行内”（即在记事本中）显示，而非弹出窗口：

```
$ ipython notebook --pylab=inline &
```

打开 chap01ex.ipynb。记事本中一些单元已经填好了代码，可以直接执行。其他单元列出了你应该尝试的练习。

本练习的参考答案在 chap01soln.ipynb 中。

- 练习 1.2

创建一个名为 chp01ex.py 的文件，编写代码，读取参与者文件 2001FemResp.dat.gz。你可以复制 nsfg.py 文件并对其进行修改。

变量 `pregnum` 是一个重编码，用于说明每位调查参与者有过多少次妊娠经历。打印这个变量中不同值的出现次数，将结果与全国家庭增长调查代码本中发布的结果进行比较。

你也可以将每位调查参与者的 `pregnum` 值与妊娠文件中的记录数进行比较，对调查参与者文件和妊娠文件进行交叉验证。

你可以使用 `nsfg.MakePregMap` 生成一个字典，将每个 `caseid` 映射到妊娠 DataFrame 的索引列表。

本练习的参考答案在 chp01soln.py 中。

- 练习 1.3

学习统计学的最好方法是使用一个你感兴趣的项目。你想研究“第一胎是否都会晚出生”这样的问题吗？

请思考一些你个人感兴趣的问题，可以是传统观点、争议话题或影响政局的问题，看是否可以构想出一个能以统计调查进行验证的问题。

寻找能帮助你回答这个问题的数据。公共研究的数据经常可以免费获取，因此政府网站是很好的数据来源，如 http://www.data.gov/ 和 http://www.science.gov/。如果想获得英国的数据，可以访问 http://data.gov.uk/。

我个人最喜爱的两个数据集是 General Social Survey（http://www3.norc.org/gss+website/）和 European Social Survey（http://www.europeansocialsurvey.org）。

如果有人看似已经解答了你的问题，那么仔细检查该回答是否合理。数据和分析中可能存在的缺陷都会使结论不可靠。如果发现别人的解答存在问题，你可以对同样的数据进行不同的分析，或者寻找更好的数据来源。

如果有一篇论文解答了你的问题，那么你应该能够获得论文使用的原始数据。很多论文作者会把数据放在网上供大家使用，但如果涉及敏感信息，你可能需要向作者写信索要，提供你计划如何使用这些数据的信息，或者同意某些使用条款。坚持就是胜利！

1.10 术语

- *轶事证据*（anecdotal evidence）

  随意收集，而非通过精心设计的研究获得的证据，通常是个人证据。

- 总体（population）
  在研究中，我们感兴趣的群组。“总体”经常指一组人，但这个词也可以用于其他对象。

- 横截面研究（cross-sectional study）
  收集一个总体在某个特定时间点的数据的研究。

- 周期（cycle）
  在重复进行的横截面研究中，每次研究称为一个周期。

- 纵向研究（longtitudinal study）
  在一段时间内跟踪一个总体的研究，从同一个群体重复收集数据。

- 记录（record）
  在数据集中，关于单个人或其他对象的信息集合。

- 调查参与者（respondent）
  参与调查的人。

- 样本（sample）
  总体中用于数据收集的一个子集。

- 有代表性（representative）
  如果总体中的每个成员被选入样本的机会都均等，那么这个样本就是有代表性的。

- 过度抽样（oversampling）
  一种通过增加一个子总体的样本数来避免因样本规模过小产生错误的技术。

- 原始数据（raw data）
  没有经过或只经过少许检查、计算或解释，直接收集和记录的值。

- 重编码（recode）
  通过计算和应用于原始数据的其他逻辑生成的值。

- 数据清洗（data cleaning）
  数据处理过程，包括数据验证、错误检查，以及数据类型和表示的转换等。