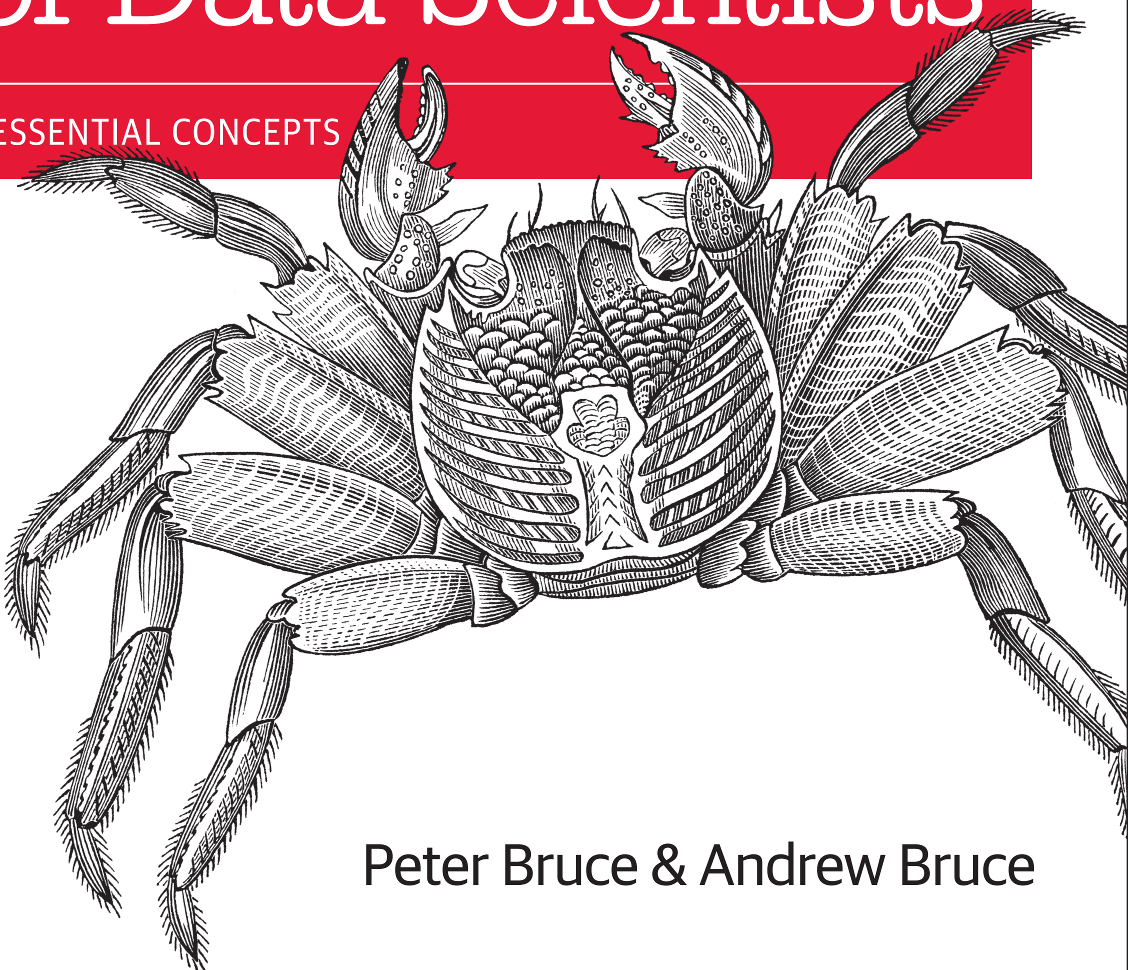
O'REILLY®

Practical Statistics for Data Scientists

50 ESSENTIAL CONCEPTS



Peter Bruce & Andrew Bruce

O'REILLY®

Practical Statistics for Data Scientists

Statistical methods are a key part of data science, yet very few data scientists have any formal statistics training. Courses and books on basic statistics rarely cover the topic from a data science perspective. This practical guide explains how to apply various statistical methods to data science, tells you how to avoid their misuse, and gives you advice on what's important and what's not.

Many data science resources incorporate statistical methods but lack a deeper statistical perspective. If you're familiar with the R programming language, and have some exposure to statistics, this quick reference bridges the gap in an accessible, readable format.

With this book, you'll learn:

- Why exploratory data analysis is a key preliminary step in data science
- How random sampling can reduce bias and yield a higher quality dataset, even with big data
- How the principles of experimental design yield definitive answers to questions
- How to use regression to estimate outcomes and detect anomalies
- Key classification techniques for predicting which categories a record belongs to
- Statistical machine learning methods that "learn" from data
- Unsupervised learning methods for extracting meaning from unlabeled data

"This book is not another statistics textbook nor a machine learning manual. It's much better: it makes the connection between useful statistical terms and principles and today's data mining lingo and practices, with clear explanations and plenty of examples. This is a terrific reference for data science beginners and old timers."

—Galit Shmueli
Lead author of the bestselling series *Data Mining for Business Analytics*, and Distinguished Professor, National Tsing Hua University, Taiwan

Peter Bruce founded and grew the Institute for Statistics Education at Statistics.com, which now offers about 90 courses in statistics, roughly half of which are aimed at data scientists.

Andrew Bruce has over 30 years of experience in statistics and data science in academia, government, and business. With a PhD in statistics from the University of Washington, he has published several papers in refereed journals.

US $39.99 **CAN $52.99**
ISBN: 978-1-491-95296-2

Twitter: @oreillymedia
facebook.com/oreilly

Practical Statistics for Data Scientists

50 Essential Concepts

Peter Bruce and Andrew Bruce

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Practical Statistics for Data Scientists

by Peter Bruce and Andrew Bruce

Printed in the United States of America.

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

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com/safari*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Shannon Cutt
Production Editor: Kristen Brown
Copyeditor: Rachel Monaghan
Proofreader: Eliahu Sussman
Indexer: Ellen Troutman-Zaig
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

May 2017: First Edition

Revision History for the First Edition

2017-05-09: First Release
2017-06-23: Second Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491952962* for release details.

978-1-491-95296-2

[LSI]

We would like to dedicate this book to the memories of our parents Victor G. Bruce and Nancy C. Bruce, who cultivated a passion for math and science; and to our early mentors John W. Tukey and Julian Simon, and our lifelong friend Geoff Watson, who helped inspire us to pursue a career in statistics.

Table of Contents

Preface

This book is aimed at the data scientist with some familiarity with the R programming language, and with some prior (perhaps spotty or ephemeral) exposure to statistics. Both of us came to the world of data science from the world of statistics, so we have some appreciation of the contribution that statistics can make to the art of data science. At the same time, we are well aware of the limitations of traditional statistics instruction: statistics as a discipline is a century and a half old, and most statistics textbooks and courses are laden with the momentum and inertia of an ocean liner.

Two goals underlie this book:

- To lay out, in digestible, navigable, and easily referenced form, key concepts from statistics that are relevant to data science.
- To explain which concepts are important and useful from a data science perspective, which are less so, and why.

What to Expect

Key Terms

Data science is a fusion of multiple disciplines, including statistics, computer science, information technology, and domain-specific fields. As a result, several different terms could be used to reference a given concept. Key terms and their synonyms will be highlighted throughout the book in a sidebar such as this.

Conventions Used in This Book

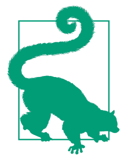
The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

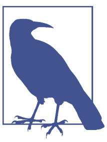
`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.



This element signifies a tip or suggestion.



This element signifies a general note.

This element indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://github.com/andrewgbruce/statistics-for-data-scientists*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Practical Statistics for Data Scientists* by Peter Bruce and Andrew Bruce (O'Reilly). Copyright 2017 Peter Bruce and Andrew Bruce, 978-1-491-95296-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government, education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like 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, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://bit.ly/practicalStats_for_DataScientists*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

The authors acknowledge the many people who helped make this book a reality.

Gerhard Pilcher, CEO of the data mining firm Elder Research, saw early drafts of the book and gave us detailed and helpful corrections and comments. Likewise, Anya McGuirk and Wei Xiao, statisticians at SAS, and Jay Hilfiger, fellow O'Reilly author, provided helpful feedback on initial drafts of the book.

At O'Reilly, Shannon Cutt has shepherded us through the publication process with good cheer and the right amount of prodding, while Kristen Brown smoothly took our book through the production phase. Rachel Monaghan and Eliahu Sussman corrected and improved our writing with care and patience, while Ellen Troutman-Zaig prepared the index. We also thank Marie Beaugureau, who initiated our project at O'Reilly, as well as Ben Bengfort, O'Reilly author and statistics.com instructor, who introduced us to O'Reilly.

We, and this book, have also benefited from the many conversations Peter has had over the years with Galit Shmueli, coauthor on other book projects.

Finally, we would like to especially thank Elizabeth Bruce and Deborah Donnell, whose patience and support made this endeavor possible.

CHAPTER 1
Exploratory Data Analysis

As a discipline, statistics has mostly developed in the past century. Probability theory—the mathematical foundation for statistics—was developed in the 17th to 19th centuries based on work by Thomas Bayes, Pierre-Simon Laplace, and Carl Gauss. In contrast to the purely theoretical nature of probability, statistics is an applied science concerned with analysis and modeling of data. Modern statistics as a rigorous scientific discipline traces its roots back to the late 1800s and Francis Galton and Karl Pearson. R. A. Fisher, in the early 20th century, was a leading pioneer of modern statistics, introducing key ideas of *experimental design* and *maximum likelihood estimation*. These and many other statistical concepts live largely in the recesses of data science. The main goal of this book is to help illuminate these concepts and clarify their importance—or lack thereof—in the context of data science and big data.

This chapter focuses on the first step in any data science project: exploring the data. *Exploratory data analysis*, or *EDA*, is a comparatively new area of statistics. Classical statistics focused almost exclusively on *inference*, a sometimes complex set of procedures for drawing conclusions about large populations based on small samples. In 1962, John W. Tukey (Figure 1-1) called for a reformation of statistics in his seminal paper "The Future of Data Analysis" [Tukey-1962]. He proposed a new scientific discipline called *data analysis* that included statistical inference as just one component. Tukey forged links to the engineering and computer science communities (he coined the terms *bit*, short for binary digit, and *software*), and his original tenets are suprisingly durable and form part of the foundation for data science. The field of exploratory data analysis was established with Tukey's 1977 now-classic book *Exploratory Data Analysis* [Tukey-1977].

Figure 1-1. John Tukey, the eminent statistician whose ideas developed over 50 years ago form the foundation of data science.

With the ready availablility of computing power and expressive data analysis software, exploratory data analysis has evolved well beyond its original scope. Key drivers of this discipline have been the rapid development of new technology, access to more and bigger data, and the greater use of quantitative analysis in a variety of disciplines. David Donoho, professor of statistics at Stanford University and former undergraduate student of Tukey's, authored an excellent article based on his presentation at the Tukey Centennial workshop in Princeton, New Jersey [Donoho-2015]. Donoho traces the genesis of data science back to Tukey's pioneering work in data analysis.

Elements of Structured Data

Data comes from many sources: sensor measurements, events, text, images, and videos. The *Internet of Things* (IoT) is spewing out streams of information. Much of this data is unstructured: images are a collection of pixels with each pixel containing RGB (red, green, blue) color information. Texts are sequences of words and nonword characters, often organized by sections, subsections, and so on. Clickstreams are sequences of actions by a user interacting with an app or web page. In fact, a major challenge of data science is to harness this torrent of raw data into actionable information. To apply the statistical concepts covered in this book, unstructured raw data must be processed and manipulated into a structured form—as it might emerge from a relational database—or be collected for a study.

Key Terms for Data Types

Continuous
: Data that can take on any value in an interval.

  Synonyms
  : interval, float, numeric

Discrete
: Data that can take on only integer values, such as counts.

  Synonyms
  : integer, count

Categorical
: Data that can take on only a specific set of values representing a set of possible categories.

  Synonyms
  : enums, enumerated, factors, nominal, polychotomous

Binary
: A special case of categorical data with just two categories of values (0/1, true/false).

  Synonyms
  : dichotomous, logical, indicator, boolean

Ordinal
: Categorical data that has an explicit ordering.

  Synonyms
  : ordered factor

There are two basic types of structured data: numeric and categorical. Numeric data comes in two forms: *continuous*, such as wind speed or time duration, and *discrete*, such as the count of the occurrence of an event. *Categorical* data takes only a fixed set of values, such as a type of TV screen (plasma, LCD, LED, etc.) or a state name (Alabama, Alaska, etc.). *Binary* data is an important special case of categorical data that takes on only one of two values, such as 0/1, yes/no, or true/false. Another useful type of categorical data is *ordinal* data in which the categories are ordered; an example of this is a numerical rating (1, 2, 3, 4, or 5).

Why do we bother with a taxonomy of data types? It turns out that for the purposes of data analysis and predictive modeling, the data type is important to help determine the type of visual display, data analysis, or statistical model. In fact, data science software, such as R and Python, uses these data types to improve computational performance. More important, the data type for a variable determines how software will handle computations for that variable.

Software engineers and database programmers may wonder why we even need the notion of *categorical* and *ordinal* data for analytics. After all, categories are merely a collection of text (or numeric) values, and the underlying database automatically handles the internal representation. However, explicit identification of data as categorical, as distinct from text, does offer some advantages:

- Knowing that data is categorical can act as a signal telling software how statistical procedures, such as producing a chart or fitting a model, should behave. In particular, ordinal data can be represented as an `ordered.factor` in R and Python, preserving a user-specified ordering in charts, tables, and models.
- Storage and indexing can be optimized (as in a relational database).
- The possible values a given categorical variable can take are enforced in the software (like an enum).

The third "benefit" can lead to unintended or unexpected behavior: the default behavior of data import functions in R (e.g., `read.csv`) is to automatically convert a text column into a `factor`. Subsequent operations on that column will assume that the only allowable values for that column are the ones originally imported, and assigning a new text value will introduce a warning and produce an `NA` (missing value).

Key Ideas

- Data is typically classified in software by type.
- Data types include continuous, discrete, categorical (which includes binary), and ordinal.
- Data typing in software acts as a signal to the software on how to process the data.

Further Reading

- Data types can be confusing, since types may overlap, and the taxonomy in one software may differ from that in another. The R-Tutorial website covers the taxonomy for R.

- Databases are more detailed in their classification of data types, incorporating considerations of precision levels, fixed- or variable-length fields, and more; see the W3Schools guide for SQL.

Rectangular Data

The typical frame of reference for an analysis in data science is a *rectangular data* object, like a spreadsheet or database table.

Key Terms for Rectangular Data

Data frame
: Rectangular data (like a spreadsheet) is the basic data structure for statistical and machine learning models.

Feature
: A column in the table is commonly referred to as a *feature*.

  Synonyms
  : attribute, input, predictor, variable

Outcome
: Many data science projects involve predicting an *outcome*—often a yes/no outcome (in Table 1-1, it is "auction was competitive or not"). The *features* are sometimes used to predict the *outcome* in an experiment or study.

  Synonyms
  : dependent variable, response, target, output

Records
: A row in the table is commonly referred to as a *record*.

  Synonyms
  : case, example, instance, observation, pattern, sample

Rectangular data is essentially a two-dimensional matrix with rows indicating records (cases) and columns indicating features (variables). The data doesn't always start in this form: unstructured data (e.g., text) must be processed and manipulated so that it can be represented as a set of features in the rectangular data (see "Elements of Structured Data" on page 2). Data in relational databases must be extracted and put into a single table for most data analysis and modeling tasks.

In Table 1-1, there is a mix of measured or counted data (e.g., duration and price), and categorical data (e.g., category and currency). As mentioned earlier, a special form of categorical variable is a binary (yes/no or 0/1) variable, seen in the rightmost column in Table 1-1—an indicator variable showing whether an auction was competitive or not.

Table 1-1. A typical data format

Category	currency	sellerRating	Duration	endDay	ClosePrice	OpenPrice	Competitive?
Music/Movie/Game	US	3249	5	Mon	0.01	0.01	0
Music/Movie/Game	US	3249	5	Mon	0.01	0.01	0
Automotive	US	3115	7	Tue	0.01	0.01	0
Automotive	US	3115	7	Tue	0.01	0.01	0
Automotive	US	3115	7	Tue	0.01	0.01	0
Automotive	US	3115	7	Tue	0.01	0.01	0
Automotive	US	3115	7	Tue	0.01	0.01	1
Automotive	US	3115	7	Tue	0.01	0.01	1

Data Frames and Indexes

Traditional database tables have one or more columns designated as an index. This can vastly improve the efficiency of certain SQL queries. In *Python*, with the `pandas` library, the basic rectangular data structure is a `DataFrame` object. By default, an automatic integer index is created for a `DataFrame` based on the order of the rows. In `pandas`, it is also possible to set multilevel/hierarchical indexes to improve the efficiency of certain operations.

In *R*, the basic rectangular data structure is a `data.frame` object. A `data.frame` also has an implicit integer index based on the row order. While a custom key can be created through the `row.names` attribute, the native R `data.frame` does not support user-specified or multilevel indexes. To overcome this deficiency, two new packages are gaining widespread use: `data.table` and `dplyr`. Both support multilevel indexes and offer significant speedups in working with a `data.frame`.

Terminology Differences

Terminology for rectangular data can be confusing. Statisticians and data scientists use different terms for the same thing. For a statistician, *predictor variables* are used in a model to predict a *response* or *dependent variable*. For a data scientist, *features* are used to predict a *target*. One synonym is particularly confusing: computer scientists will use the term *sample* for a single row; a *sample* to a statistician means a collection of rows.

Nonrectangular Data Structures

There are other data structures besides rectangular data.

Time series data records successive measurements of the same variable. It is the raw material for statistical forecasting methods, and it is also a key component of the data produced by devices—the Internet of Things.

Spatial data structures, which are used in mapping and location analytics, are more complex and varied than rectangular data structures. In the *object* representation, the focus of the data is an object (e.g., a house) and its spatial coordinates. The *field* view, by contrast, focuses on small units of space and the value of a relevant metric (pixel brightness, for example).

Graph (or network) data structures are used to represent physical, social, and abstract relationships. For example, a graph of a social network, such as Facebook or LinkedIn, may represent connections between people on the network. Distribution hubs connected by roads are an example of a physical network. Graph structures are useful for certain types of problems, such as network optimization and recommender systems.

Each of these data types has its specialized methodology in data science. The focus of this book is on rectangular data, the fundamental building block of predictive modeling.

Graphs in Statistics

In computer science and information technology, the term *graph* typically refers to a depiction of the connections among entities, and to the underlying data structure. In statistics, *graph* is used to refer to a variety of plots and *visualizations*, not just of connections among entities, and the term applies just to the visualization, not to the data structure.

Key Ideas

- The basic data structure in data science is a rectangular matrix in which rows are records and columns are variables (features).
- Terminology can be confusing; there are a variety of synonyms arising from the different disciplines that contribute to data science (statistics, computer science, and information technology).

Further Reading

- Documentation on data frames in R
- Documentation on data frames in Python

Estimates of Location

Variables with measured or count data might have thousands of distinct values. A basic step in exploring your data is getting a "typical value" for each feature (variable): an estimate of where most of the data is located (i.e., its central tendency).

Key Terms for Estimates of Location

Mean
: The sum of all values divided by the number of values.

  Synonyms
  : average

Weighted mean
: The sum of all values times a weight divided by the sum of the weights.

  Synonyms
  : weighted average

Median
: The value such that one-half of the data lies above and below.

  Synonyms
  : 50th percentile

Weighted median
: The value such that one-half of the sum of the weights lies above and below the sorted data.

Trimmed mean
: The average of all values after dropping a fixed number of extreme values.

  Synonyms
  : truncated mean

Robust
: Not sensitive to extreme values.

  Synonyms
  : resistant

Outlier
: A data value that is very different from most of the data.

  Synonyms
  : extreme value

At first glance, summarizing data might seem fairly trivial: just take the *mean* of the data (see "Mean" on page 9). In fact, while the mean is easy to compute and expedient to use, it may not always be the best measure for a central value. For this reason, statisticians have developed and promoted several alternative estimates to the mean.

Metrics and Estimates

Statisticians often use the term *estimates* for values calculated from the data at hand, to draw a distinction between what we see from the data, and the theoretical true or exact state of affairs. Data scientists and business analysts are more likely to refer to such values as a *metric*. The difference reflects the approach of statistics versus data science: accounting for uncertainty lies at the heart of the discipline of statistics, whereas concrete business or organizational objectives are the focus of data science. Hence, statisticians estimate, and data scientists measure.

Mean

The most basic estimate of location is the mean, or *average* value. The mean is the sum of all the values divided by the number of values. Consider the following set of numbers: {3 5 1 2}. The mean is (3 + 5 + 1 + 2) / 4 = 11 / 4 = 2.75. You will encounter the symbol $\bar{x}$ (pronounced "x-bar") to represent the mean of a sample from a population. The formula to compute the mean for a set of n values $x_1, x_2, ..., x_N$ is:

$$\text{Mean} = \bar{x} = \frac{\sum_i^n x_i}{n}$$

N (or n) refers to the total number of records or observations. In statistics it is capitalized if it is referring to a population, and lowercase if it refers to a sample from a population. In data science, that distinction is not vital so you may see it both ways.

A variation of the mean is a *trimmed mean*, which you calculate by dropping a fixed number of sorted values at each end and then taking an average of the remaining values. Representing the sorted values by $x_{(1)}, x_{(2)}, ..., x_{(n)}$ where $x_{(1)}$ is the smallest value and $x_{(n)}$ the largest, the formula to compute the trimmed mean with p smallest and largest values omitted is:

$$\text{Trimmed mean} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p}$$

A trimmed mean eliminates the influence of extreme values. For example, in international diving the top and bottom scores from five judges are dropped, and the final score is the average of the three remaining judges [Wikipedia-2016]. This makes it difficult for a single judge to manipulate the score, perhaps to favor his country's contestant. Trimmed means are widely used, and in many cases, are preferable to use instead of the ordinary mean: see "Median and Robust Estimates" on page 10 for further discussion.

Another type of mean is a *weighted mean*, which you calculate by multiplying each data value x_i by a weight w_i and dividing their sum by the sum of the weights. The formula for a weighted mean is:

$$\text{Weighted mean} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_i^n w_i}$$

There are two main motivations for using a weighted mean:

- Some values are intrinsically more variable than others, and highly variable observations are given a lower weight. For example, if we are taking the average from multiple sensors and one of the sensors is less accurate, then we might downweight the data from that sensor.
- The data collected does not equally represent the different groups that we are interested in measuring. For example, because of the way an online experiment was conducted, we may not have a set of data that accurately reflects all groups in the user base. To correct that, we can give a higher weight to the values from the groups that were underrepresented.

Median and Robust Estimates

The *median* is the middle number on a sorted list of the data. If there is an even number of data values, the middle value is one that is not actually in the data set, but rather the average of the two values that divide the sorted data into upper and lower halves. Compared to the mean, which uses all observations, the median depends only on the values in the center of the sorted data. While this might seem to be a disadvantage, since the mean is much more sensitive to the data, there are many instances in which the median is a better metric for location. Let's say we want to look at typical household incomes in neighborhoods around Lake Washington in Seattle. In comparing the Medina neighborhood to the Windermere neighborhood, using the mean would produce very different results because Bill Gates lives in Medina. If we use the median, it won't matter how rich Bill Gates is—the position of the middle observation will remain the same.

For the same reasons that one uses a weighted mean, it is also possible to compute a *weighted median*. As with the median, we first sort the data, although each data value has an associated weight. Instead of the middle number, the weighted median is a value such that the sum of the weights is equal for the lower and upper halves of the sorted list. Like the median, the weighted median is robust to outliers.

Outliers

The median is referred to as a *robust* estimate of location since it is not influenced by *outliers* (extreme cases) that could skew the results. An outlier is any value that is very distant from the other values in a data set. The exact definition of an outlier is somewhat subjective, although certain conventions are used in various data summaries and plots (see "Percentiles and Boxplots" on page 20). Being an outlier in itself does not make a data value invalid or erroneous (as in the previous example with Bill Gates). Still, outliers are often the result of data errors such as mixing data of different units (kilometers versus meters) or bad readings from a sensor. When outliers are the result of bad data, the mean will result in a poor estimate of location, while the median will be still be valid. In any case, outliers should be identified and are usually worthy of further investigation.

Anomaly Detection

In contrast to typical data analysis, where outliers are sometimes informative and sometimes a nuisance, in *anomaly detection* the points of interest are the outliers, and the greater mass of data serves primarily to define the "normal" against which anomalies are measured.

The median is not the only robust estimate of location. In fact, a trimmed mean is widely used to avoid the influence of outliers. For example, trimming the bottom and top 10% (a common choice) of the data will provide protection against outliers in all but the smallest data sets. The trimmed mean can be thought of as a compromise between the median and the mean: it is robust to extreme values in the data, but uses more data to calculate the estimate for location.

Other Robust Metrics for Location

Statisticians have developed a plethora of other estimators for location, primarily with the goal of developing an estimator more robust than the mean and also more *efficient* (i.e., better able to discern small location differences between data sets). While these methods are potentially useful for small data sets, they are not likely to provide added benefit for large or even moderately sized data sets.

Example: Location Estimates of Population and Murder Rates

Table 1-2 shows the first few rows in the data set containing population and murder rates (in units of murders per 100,000 people per year) for each state.

Table 1-2. A few rows of the data.frame state of population and murder rate by state

	State	Population	Murder rate
1	Alabama	4,779,736	5.7
2	Alaska	710,231	5.6
3	Arizona	6,392,017	4.7
4	Arkansas	2,915,918	5.6
5	California	37,253,956	4.4
6	Colorado	5,029,196	2.8
7	Connecticut	3,574,097	2.4
8	Delaware	897,934	5.8

Compute the mean, trimmed mean, and median for the population using R:

```
> state <- read.csv(file="/Users/andrewbruce1/book/state.csv")
> mean(state[["Population"]])
[1] 6162876
> mean(state[["Population"]], trim=0.1)
[1] 4783697
> median(state[["Population"]])
[1] 4436370
```

The mean is bigger than the trimmed mean, which is bigger than the median.

This is because the trimmed mean excludes the largest and smallest five states (`trim=0.1` drops 10% from each end). If we want to compute the average murder rate for the country, we need to use a weighted mean or median to account for different populations in the states. Since base R doesn't have a function for weighted median, we need to install a package such as `matrixStats`:

```
> weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.445834
```

```
> library("matrixStats")
> weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.4
```

In this case, the weighted mean and median are about the same.

Key Ideas

- The basic metric for location is the mean, but it can be sensitive to extreme values (outlier).
- Other metrics (median, trimmed mean) are more robust.

Further Reading

- Michael Levine (Purdue University) has posted some useful slides on basic calculations for measures of location.
- John Tukey's 1977 classic *Exploratory Data Analysis* (Pearson) is still widely read.

Estimates of Variability

Location is just one dimension in summarizing a feature. A second dimension, *variability*, also referred to as *dispersion*, measures whether the data values are tightly clustered or spread out. At the heart of statistics lies variability: measuring it, reducing it, distinguishing random from real variability, identifying the various sources of real variability, and making decisions in the presence of it.

Key Terms for Variability Metrics

Deviations
: The difference between the observed values and the estimate of location.

  Synonyms
  : errors, residuals

Variance
: The sum of squared deviations from the mean divided by $n - 1$ where n is the number of data values.

  Synonyms
  : mean-squared-error

Standard deviation
: The square root of the variance.

Synonyms
: l2-norm, Euclidean norm

Mean absolute deviation
: The mean of the absolute value of the deviations from the mean.

Synonyms
: l1-norm, Manhattan norm

Median absolute deviation from the median
: The median of the absolute value of the deviations from the median.

Range
: The difference between the largest and the smallest value in a data set.

Order statistics
: Metrics based on the data values sorted from smallest to biggest.

Synonyms
: ranks

Percentile
: The value such that *P* percent of the values take on this value or less and (100–P) percent take on this value or more.

Synonyms
: quantile

Interquartile range
: The difference between the 75th percentile and the 25th percentile.

Synonyms
: IQR

Just as there are different ways to measure location (mean, median, etc.) there are also different ways to measure variability.

Standard Deviation and Related Estimates

The most widely used estimates of variation are based on the differences, or *deviations*, between the estimate of location and the observed data. For a set of data {1, 4, 4}, the mean is 3 and the median is 4. The deviations from the mean are the differences: $1 - 3 = -2$, $4 - 3 = 1$, $4 - 3 = 1$. These deviations tell us how dispersed the data is around the central value.

One way to measure variability is to estimate a typical value for these deviations. Averaging the deviations themselves would not tell us much—the negative deviations offset the positive ones. In fact, the sum of the deviations from the mean is precisely zero. Instead, a simple approach is to take the average of the absolute values of the deviations from the mean. In the preceding example, the absolute value of the deviations is {2 1 1} and their average is (2 + 1 + 1) / 3 = 1.33. This is known as the *mean absolute deviation* and is computed with the formula:

$$\text{Mean absolution deviation} = \frac{\sum_{i=1}^{n} \left| x_i - \bar{x} \right|}{n}$$

where $\bar{x}$ is the sample mean.

The best-known estimates for variability are the *variance* and the *standard deviation*, which are based on squared deviations. The variance is an average of the squared deviations, and the standard deviation is the square root of the variance.

$$\text{Variance} = s^2 = \frac{\sum (x - \bar{x})^2}{n - 1}$$

$$\text{Standard deviation} = s = \sqrt{\text{Variance}}$$

The standard deviation is much easier to interpret than the variance since it is on the same scale as the original data. Still, with its more complicated and less intuitive formula, it might seem peculiar that the standard deviation is preferred in statistics over the mean absolute deviation. It owes its preeminence to statistical theory: mathematically, working with squared values is much more convenient than absolute values, especially for statistical models.

Degrees of Freedom, and *n* or *n* – 1?

In statistics books, there is always some discussion of why we have $n - 1$ in the denominator in the variance formula, instead of n, leading into the concept of *degrees of freedom*. This distinction is not important since n is generally large enough that it won't make much difference whether you divide by n or $n - 1$. But in case you are interested, here is the story. It is based on the premise that you want to make estimates about a population, based on a sample.

If you use the intuitive denominator of n in the variance formula, you will underestimate the true value of the variance and the standard deviation in the population. This is referred to as a *biased* estimate. However, if you divide by $n - 1$ instead of n, the standard deviation becomes an *unbiased* estimate.

To fully explain why using n leads to a biased estimate involves the notion of degrees of freedom, which takes into account the number of constraints in computing an estimate. In this case, there are $n - 1$ degrees of freedom since there is one constraint: the standard deviation depends on calculating the sample mean. For many problems, data scientists do not need to worry about degrees of freedom, but there are cases where the concept is important (see "Choosing K" on page 217).

Neither the variance, the standard deviation, nor the mean absolute deviation is robust to outliers and extreme values (see "Median and Robust Estimates" on page 10 for a discussion of robust estimates for location). The variance and standard deviation are especially sensitive to outliers since they are based on the squared deviations.

A robust estimate of variability is the *median absolute deviation from the median* or MAD:

$$\text{Median absolute deviation} = \text{Median}\left(\left|x_1 - m\right|, \left|x_2 - m\right|, \ldots, \left|x_N - m\right|\right)$$

where m is the median. Like the median, the MAD is not influenced by extreme values. It is also possible to compute a trimmed standard deviation analogous to the trimmed mean (see "Mean" on page 9).

The variance, the standard deviation, mean absolute deviation, and median absolute deviation from the median are not equivalent estimates, even in the case where the data comes from a normal distribution. In fact, the standard deviation is always greater than the mean absolute deviation, which itself is greater than the median absolute deviation. Sometimes, the median absolute deviation is multiplied by a constant scaling factor (it happens to work out to 1.4826) to put MAD on the same scale as the standard deviation in the case of a normal distribution.

Estimates Based on Percentiles

A different approach to estimating dispersion is based on looking at the spread of the sorted data. Statistics based on sorted (ranked) data are referred to as *order statistics*. The most basic measure is the *range*: the difference between the largest and smallest number. The minimum and maximum values themselves are useful to know, and helpful in identifying outliers, but the range is extremely sensitive to outliers and not very useful as a general measure of dispersion in the data.

To avoid the sensitivity to outliers, we can look at the range of the data after dropping values from each end. Formally, these types of estimates are based on differences between *percentiles*. In a data set, the *P*th percentile is a value such that at least *P* percent of the values take on this value or less and at least (100 – *P*) percent of the values take on this value or more. For example, to find the 80th percentile, sort the data. Then, starting with the smallest value, proceed 80 percent of the way to the largest value. Note that the median is the same thing as the 50th percentile. The percentile is essentially the same as a *quantile*, with quantiles indexed by fractions (so the .8 quantile is the same as the 80th percentile).

A common measurement of variability is the difference between the 25th percentile and the 75th percentile, called the *interquartile range* (or IQR). Here is a simple example: 3,1,5,3,6,7,2,9. We sort these to get 1,2,3,3,5,6,7,9. The 25th percentile is at 2.5, and the 75th percentile is at 6.5, so the interquartile range is 6.5 – 2.5 = 4. Software can have slightly differing approaches that yield different answers (see the following note); typically, these differences are smaller.

For very large data sets, calculating exact percentiles can be computationally very expensive since it requires sorting all the data values. Machine learning and statistical software use special algorithms, such as [Zhang-Wang-2007], to get an approximate percentile that can be calculated very quickly and is guaranteed to have a certain accuracy.

Percentile: Precise Definition

If we have an even number of data (n is even), then the percentile is ambiguous under the preceding definition. In fact, we could take on any value between the order statistics $x_{(j)}$ and $x_{(j+1)}$ where j satisfies:

$$100 * \frac{j}{n} \leq P < 100 * \frac{j+1}{n}$$

Formally, the percentile is the weighted average:

$$\text{Percentile}(P) = (1 - w)x_{(j)} + wx_{(j+1)}$$

for some weight w between 0 and 1. Statistical software has slightly differing approaches to choosing w. In fact, the R function `quantile` offers nine different alternatives to compute the quantile. Except for small data sets, you don't usually need to worry about the precise way a percentile is calculated.

Example: Variability Estimates of State Population

Table 1-3 (repeated from Table 1-2, earlier, for convenience) shows the first few rows in the data set containing population and murder rates for each state.

Table 1-3. A few rows of the data.frame state of population and murder rate by state

	State	Population	Murder rate
1	Alabama	4,779,736	5.7
2	Alaska	710,231	5.6
3	Arizona	6,392,017	4.7
4	Arkansas	2,915,918	5.6
5	California	37,253,956	4.4
6	Colorado	5,029,196	2.8
7	Connecticut	3,574,097	2.4
8	Delaware	897,934	5.8

Using R's built-in functions for the standard deviation, interquartile range (IQR), and the median absolution deviation from the median (MAD), we can compute estimates of variability for the state population data:

```
> sd(state[["Population"]])
[1] 6848235
> IQR(state[["Population"]])
```

```
[1] 4847308
> mad(state[["Population"]])
[1] 3849870
```

The standard deviation is almost twice as large as the MAD (in R, by default, the scale of the MAD is adjusted to be on the same scale as the mean). This is not surprising since the standard deviation is sensitive to outliers.

Key Ideas

- The variance and standard deviation are the most widespread and routinely reported statistics of variability.
- Both are sensitive to outliers.
- More robust metrics include mean and median absolute deviations from the mean and percentiles (quantiles).

Further Reading

1. David Lane's online statistics resource has a section on percentiles.
2. Kevin Davenport has a useful post on deviations from the median, and their robust properties in R-Bloggers.

Exploring the Data Distribution

Each of the estimates we've covered sums up the data in a single number to describe the location or variability of the data. It is also useful to explore how the data is distributed overall.

Key Terms for Exploring the Distribution

Boxplot
A plot introduced by Tukey as a quick way to visualize the distribution of data.

Synonyms
Box and whiskers plot

Frequency table
A tally of the count of numeric data values that fall into a set of intervals (bins).

Histogram
A plot of the frequency table with the bins on the x-axis and the count (or proportion) on the y-axis.

Density plot
A smoothed version of the histogram, often based on a *kernal density estimate*.

Percentiles and Boxplots

In "Estimates Based on Percentiles" on page 17, we explored how percentiles can be used to measure the spread of the data. Percentiles are also valuable to summarize the entire distribution. It is common to report the quartiles (25th, 50th, and 75th percentiles) and the deciles (the 10th, 20th, …, 90th percentiles). Percentiles are especially valuable to summarize the *tails* (the outer range) of the distribution. Popular culture has coined the term *one-percenters* to refer to the people in the top 99th percentile of wealth.

Table 1-4 displays some percentiles of the murder rate by state. In R, this would be produced by the `quantile` function:

```
quantile(state[["Murder.Rate"]], p=c(.05, .25, .5, .75, .95))
   5%   25%   50%   75%   95%
1.600 2.425 4.000 5.550 6.510
```

Table 1-4. Percentiles of murder rate by state

5%	25%	50%	75%	95%
1.60	2.42	4.00	5.55	6.51

The median is 4 murders per 100,000 people, although there is quite a bit of variability: the 5th percentile is only 1.6 and the 95th percentile is 6.51.

Boxplots, introduced by Tukey [Tukey-1977], are based on percentiles and give a quick way to visualize the distribution of data. Figure 1-2 shows a boxplot of the population by state produced by R:

```
boxplot(state[["Population"]]/1000000, ylab="Population (millions)")
```

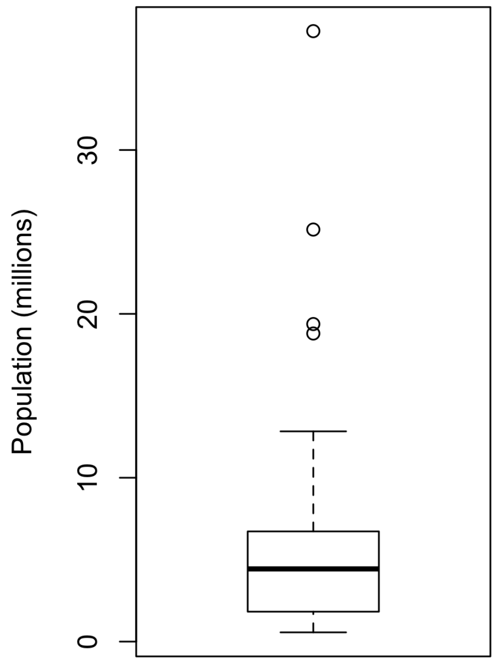


Figure 1-2. Boxplot of state populations

The top and bottom of the box are the 75th and 25th percentiles, respectively. The median is shown by the horizontal line in the box. The dashed lines, referred to as *whiskers*, extend from the top and bottom to indicate the range for the bulk of the data. There are many variations of a boxplot; see, for example, the documentation for the R function `boxplot` [R-base-2015]. By default, the R function extends the whiskers to the furthest point beyond the box, except that it will not go beyond 1.5 times the IQR (other software may use a different rule). Any data outside of the whiskers is plotted as single points.

Frequency Table and Histograms

A frequency table of a variable divides up the variable range into equally spaced segments, and tells us how many values fall in each segment. Table 1-5 shows a frequency table of the population by state computed in R:

```
breaks <- seq(from=min(state[["Population"]]),
                to=max(state[["Population"]]), length=11)
pop_freq <- cut(state[["Population"]], breaks=breaks,
                right=TRUE, include.lowest = TRUE)
table(pop_freq)
```

Table 1-5. A frequency table of population by state

BinNumber	BinRange	Count	States
1	563,626–4,232,658	24	WY,VT,ND,AK,SD,DE,MT,RI,NH,ME,HI,ID,NE,WV,NM,NV,UT,KS,AR,MS,IA,CT,OK,OR
2	4,232,659–7,901,691	14	KY,LA,SC,AL,CO,MN,WI,MD,MO,TN,AZ,IN,MA,WA
3	7,901,692–11,570,724	6	VA,NJ,NC,GA,MI,OH
4	11,570,725–15,239,757	2	PA,IL
5	15,239,758–18,908,790	1	FL
6	18,908,791–22,577,823	1	NY
7	22,577,824–26,246,856	1	TX
8	26,246,857–29,915,889	0	
9	29,915,890–33,584,922	0	
10	33,584,923–37,253,956	1	CA

The least populous state is Wyoming, with 563,626 people (2010 Census) and the most populous is California, with 37,253,956 people. This gives us a range of 37,253,956 – 563,626 = 36,690,330, which we must divide up into equal size bins—let's say 10 bins. With 10 equal size bins, each bin will have a width of 3,669,033, so the first bin will span from 563,626 to 4,232,658. By contrast, the top bin, 33,584,923 to 37,253,956, has only one state: California. The two bins immediately below California are empty, until we reach Texas. It is important to include the empty bins; the fact that there are no values in those bins is useful information. It can also be useful to experiment with different bin sizes. If they are too large, important features of the distribution can be obscured. It they are too small, the result is too granular and the ability to see bigger pictures is lost.

> Both frequency tables and percentiles summarize the data by creating bins. In general, quartiles and deciles will have the same count in each bin (equal-count bins), but the bin sizes will be different. The frequency table, by contrast, will have different counts in the bins (equal-size bins).

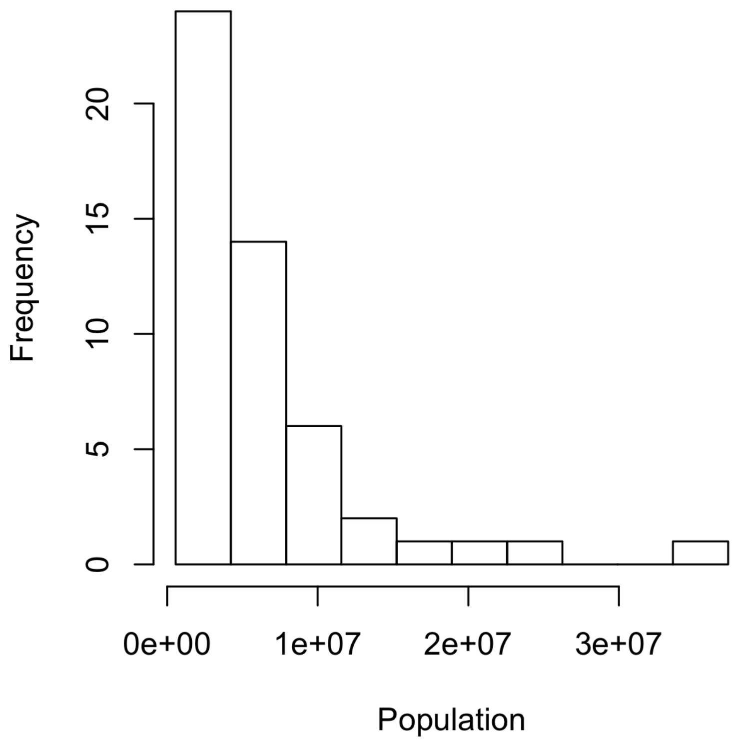


Figure 1-3. Histogram of state populations

A histogram is a way to visualize a frequency table, with bins on the x-axis and data count on the y-axis. To create a histogram corresponding to Table 1-5 in R, use the `hist` function with the `breaks` argument:

```
hist(state[["Population"]], breaks=breaks)
```

The histogram is shown in Figure 1-3. In general, histograms are plotted such that:

- Empty bins are included in the graph.
- Bins are equal width.
- Number of bins (or, equivalently, bin size) is up to the user.
- Bars are contiguous—no empty space shows between bars, unless there is an empty bin.

Statistical Moments

In statistical theory, location and variability are referred to as the first and second *moments* of a distribution. The third and fourth moments are called *skewness* and *kurtosis*. Skewness refers to whether the data is skewed to larger or smaller values and kurtosis indicates the propensity of the data to have extreme values. Generally, metrics are not used to measure skewness and kurtosis; instead, these are discovered through visual displays such as Figures 1-2 and 1-3.

Density Estimates

Related to the histogram is a density plot, which shows the distribution of data values as a continuous line. A density plot can be thought of as a smoothed histogram, although it is typically computed directly from the data through a *kernal density estimate* (see [Duong-2001] for a short tutorial). Figure 1-4 displays a density estimate superposed on a histogram. In R, you can compute a density estimate using the `density` function:

```
hist(state[["Murder.Rate"]], freq=FALSE)
lines(density(state[["Murder.Rate"]]), lwd=3, col="blue")
```

A key distinction from the histogram plotted in Figure 1-3 is the scale of the y-axis: a density plot corresponds to plotting the histogram as a proportion rather than counts (you specify this in R using the argument `freq=FALSE`).

Density Estimation

Density estimation is a rich topic with a long history in statistical literature. In fact, over 20 R packages have been published that offer functions for density estimation. [Deng-Wickham-2011] give a comprehesive review of R packages, with a particular recommendation for `ASH` or `KernSmooth`. For many data science problems, there is no need to worry about the various types of density estimates; it suffices to use the base functions.

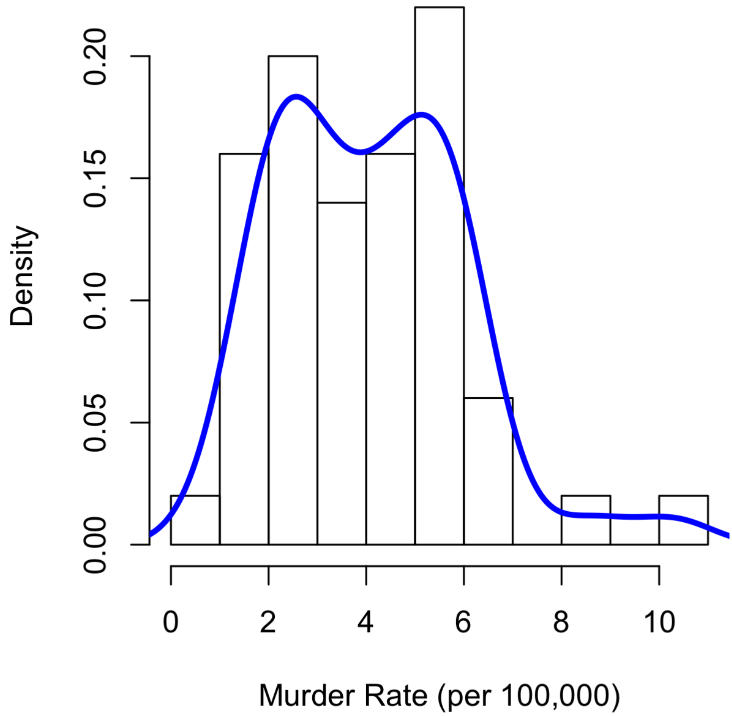


Figure 1-4. Density of state murder rates

Key Ideas

- A frequency histogram plots frequency counts on the y-axis and variable values on the x-axis; it gives a sense of the distribution of the data at a glance.
- A frequency table is a tabular version of the frequency counts found in a histogram.
- A boxplot—with the top and bottom of the box at the 75th and 25th percentiles, respectively—also gives a quick sense of the distribution of the data; it is often used in side-by-side displays to compare distributions.
- A density plot is a smoothed version of a histogram; it requires a function to estimate a plot based on the data (multiple estimates are possible, of course).

Further Reading

- A SUNY Oswego professor provides a step-by-step guide to creating a boxplot.
- Density estimation in R is covered in Henry Deng and Hadley Wickham's paper of the same name.
- R-Bloggers has a useful post on histograms in R, including customization elements, such as binning (breaks)
- R-Bloggers also has similar post on boxplots in R.

Exploring Binary and Categorical Data

For categorical data, simple proportions or percentages tell the story of the data.

Key Terms for Exploring Categorical Data

Mode
: The most commonly occurring category or value in a data set.

Expected value
: When the categories can be associated with a numeric value, this gives an average value based on a category's probability of occurrence.

Bar charts
: The frequency or proportion for each category plotted as bars.

Pie charts
: The frequency or proportion for each category plotted as wedges in a pie.

Getting a summary of a binary variable or a categorical variable with a few categories is a fairly easy matter: we just figure out the proportion of 1s, or of the important categories. For example, Table 1-6 shows the percentage of delayed flights by the cause of delay at Dallas/Fort Worth airport since 2010. Delays are categorized as being due to factors under carrier control, air traffic control (ATC) system delays, weather, security, or a late inbound aircraft.

Table 1-6. Percentage of delays by cause at Dallas-Fort Worth airport

Carrier	ATC	Weather	Security	Inbound
23.02	30.40	4.03	0.12	42.43

Bar charts are a common visual tool for displaying a single categorical variable, often seen in the popular press. Categories are listed on the x-axis, and frequencies or proportions on the y-axis. Figure 1-5 shows the airport delays per year by cause for Dallas/Fort Worth, and it is produced with the R function `barplot`:

```
barplot(as.matrix(dfw)/6, cex.axis=.5)
```

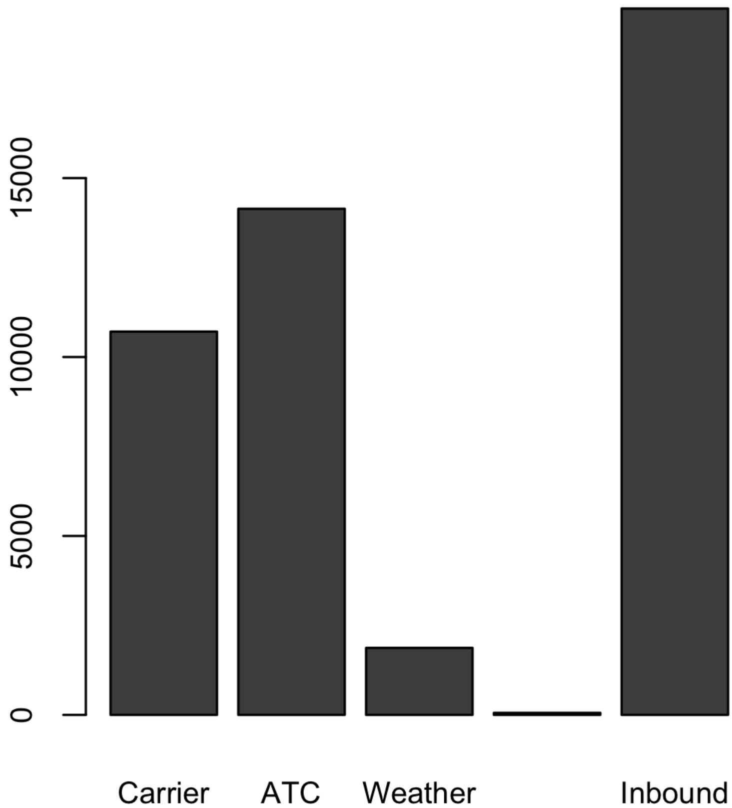


Figure 1-5. Bar plot airline delays at DFW by cause

Note that a bar chart resembles a histogram; in a bar chart the x-axis represents different categories of a factor variable, while in a histogram the x-axis represents values of a single variable on a numeric scale. In a histogram, the bars are typically shown touching each other, with gaps indicating values that did not occur in the data. In a bar chart, the bars are shown separate from one another.

Pie charts are an alternative to bar charts, although statisticians and data visualization experts generally eschew pie charts as less visually informative (see [Few-2007]).

> **Numerical Data as Categorical Data**
>
> In "Frequency Table and Histograms" on page 21, we looked at frequency tables based on binning the data. This implicitly converts the numeric data to an ordered factor. In this sense, histograms and bar charts are similar, except that the categories on the x-axis in the bar chart are not ordered. Converting numeric data to categorical data is an important and widely used step in data analysis since it reduces the complexity (and size) of the data. This aids in the discovery of relationships between features, particularly at the initial stages of an analysis.

Mode

The mode is the value—or values in case of a tie—that appears most often in the data. For example, the mode of the cause of delay at Dallas/Fort Worth airport is "Inbound." As another example, in most parts of the United States, the mode for religious preference would be Christian. The mode is a simple summary statistic for categorical data, and it is generally not used for numeric data.

Expected Value

A special type of categorical data is data in which the categories represent or can be mapped to discrete values on the same scale. A marketer for a new cloud technology, for example, offers two levels of service, one priced at $300/month and another at $50/month. The marketer offers free webinars to generate leads, and the firm figures that 5% of the attendees will sign up for the $300 service, 15% for the $50 service, and 80% will not sign up for anything. This data can be summed up, for financial purposes, in a single "expected value," which is a form of weighted mean in which the weights are probabilities.

The expected value is calculated as follows:

1. Multiply each outcome by its probability of occurring.
2. Sum these values.

In the cloud service example, the expected value of a webinar attendee is thus $22.50 per month, calculated as follows:

$$EV = (0.05)(300) + (0.15)(50) + (0.80)(0) = 22.5$$

The expected value is really a form of weighted mean: it adds the ideas of future expectations and probability weights, often based on subjective judgment. Expected value is a fundamental concept in business valuation and capital budgeting—for example, the expected value of five years of profits from a new acquisition, or the expected cost savings from new patient management software at a clinic.

Key Ideas

- Categorical data is typically summed up in proportions, and can be visualized in a bar chart.
- Categories might represent distinct things (apples and oranges, male and female), levels of a factor variable (low, medium, and high), or numeric data that has been binned.
- Expected value is the sum of values times their probability of occurrence, often used to sum up factor variable levels.

Further Reading

No statistics course is complete without a lesson on misleading graphs, which often involve bar charts and pie charts.

Correlation

Exploratory data analysis in many modeling projects (whether in data science or in research) involves examining correlation among predictors, and between predictors and a target variable. Variables X and Y (each with measured data) are said to be positively correlated if high values of X go with high values of Y, and low values of X go with low values of Y. If high values of X go with low values of Y, and vice versa, the variables are negatively correlated.

Key Terms for Correlation

Correlation coefficient
A metric that measures the extent to which numeric variables are associated with one another (ranges from –1 to +1).

Correlation matrix
A table where the variables are shown on both rows and columns, and the cell values are the correlations between the variables.

Scatterplot
A plot in which the x-axis is the value of one variable, and the y-axis the value of another.

Consider these two variables, perfectly correlated in the sense that each goes from low to high:

v1: {1, 2, 3}
v2: {4, 5, 6}

The vector sum of products is 4 + 10 + 18 = 32. Now try shuffling one of them and recalculating—the vector sum of products will never be higher than 32. So this sum of products could be used as a metric; that is, the observed sum of 32 could be compared to lots of random shufflings (in fact, this idea relates to a resampling-based estimate: see "Permutation Test" on page 88). Values produced by this metric, though, are not that meaningful, except by reference to the resampling distribution.

More useful is a standardized variant: the *correlation coefficient*, which gives an estimate of the correlation between two variables that always lies on the same scale. To compute *Pearson's correlation coefficient*, we multiply deviations from the mean for variable 1 times those for variable 2, and divide by the product of the standard deviations:

$$r = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{(N-1)s_x s_y}$$

Note that we divide by $n - 1$ instead of n; see "Degrees of Freedom, and n or $n - 1$?" on page 16 for more details. The correlation coefficient always lies between +1 (perfect positive correlation) and –1 (perfect negative correlation); 0 indicates no correlation.

Variables can have an association that is not linear, in which case the correlation coefficient may not be a useful metric. The relationship between tax rates and revenue raised is an example: as tax rates increase from 0, the revenue raised also increases. However, once tax rates reach a high level and approach 100%, tax avoidance increases and tax revenue actually declines.

Table 1-7, called a *correlation matrix*, shows the correlation between the daily returns for telecommunication stocks from July 2012 through June 2015. From the table, you can see that Verizon (VZ) and ATT (T) have the highest correlation. Level Three (LVLT), which is an infrastructure company, has the lowest correlation. Note the diagonal of 1s (the correlation of a stock with itself is 1), and the redundancy of the information above and below the diagonal.

Table 1-7. Correlation between telecommunication stock returns

	T	CTL	FTR	VZ	LVLT
T	1.000	0.475	0.328	0.678	0.279
CTL	0.475	1.000	0.420	0.417	0.287
FTR	0.328	0.420	1.000	0.287	0.260
VZ	0.678	0.417	0.287	1.000	0.242
LVLT	0.279	0.287	0.260	0.242	1.000

A table of correlations like Table 1-7 is commonly plotted to visually display the relationship between multiple variables. Figure 1-6 shows the correlation between the daily returns for major exchange traded funds (ETFs). In R, we can easily create this using the package `corrplot`:

```
etfs <- sp500_px[row.names(sp500_px)>"2012-07-01",
                 sp500_sym[sp500_sym$sector=="etf", 'symbol']]
library(corrplot)
corrplot(cor(etfs), method = "ellipse")
```

The ETFs for the S&P 500 (SPY) and the Dow Jones Index (DIA) have a high correlation. Similary, the QQQ and the XLK, composed mostly of technology companies, are postively correlated. Defensive ETFs, such as those tracking gold prices (GLD), oil prices (USO), or market volatility (VXX) tend to be negatively correlated with the other ETFs. The orientation of the ellipse indicates whether two variables are positively correlated (ellipse is pointed right) or negatively correlated (ellipse is pointed left). The shading and width of the ellipse indicate the strength of the association: thinner and darker ellipses correspond to stronger relationships.

Like the mean and standard deviation, the correlation coefficient is sensitive to outliers in the data. Software packages offer robust alternatives to the classical correlation coefficient. For example, the R function `cor` has a `trim` argument similar to that for computing a trimmed mean (see [R-base-2015]).

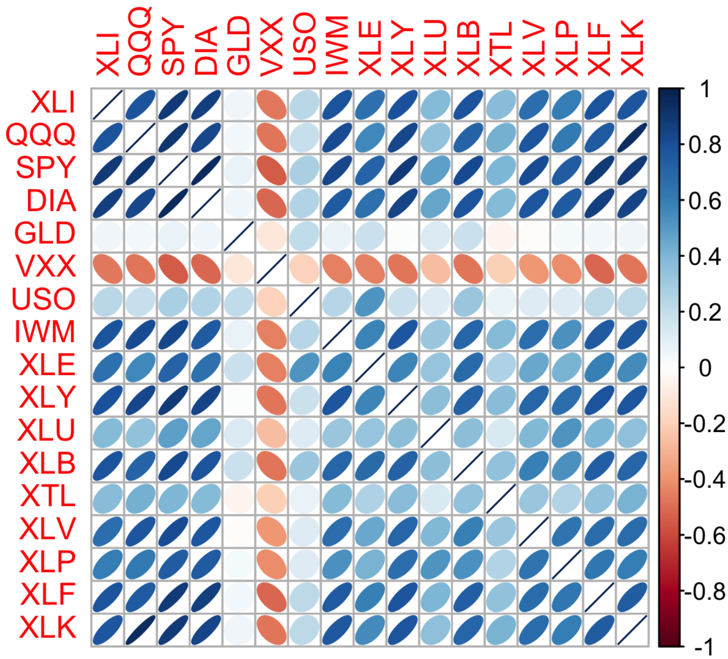


Figure 1-6. Correlation between ETF returns

Other Correlation Estimates

Statisticians have long ago proposed other types of correlation coefficients, such as *Spearman's rho* or *Kendall's tau*. These are correlation coefficients based on the rank of the data. Since they work with ranks rather than values, these estimates are robust to outliers and can handle certain types of nonlinearities. However, data scientists can generally stick to Pearson's correlation coefficient, and its robust alternatives, for exploratory analysis. The appeal of rank-based estimates is mostly for smaller data sets and specific hypothesis tests.

Scatterplots

The standard way to visualize the relationship between two measured data variables is with a scatterplot. The x-axis represents one variable, the y-axis another, and each point on the graph is a record. See Figure 1-7 for a plot between the daily returns for ATT and Verizon. This is produced in R with the command:

```
plot(telecom$T, telecom$VZ, xlab="T", ylab="VZ")
```

The returns have a strong positive relationship: on most days, both stocks go up or go down in tandem. There are very few days where one stock goes down significantly while the other stock goes up (and vice versa).

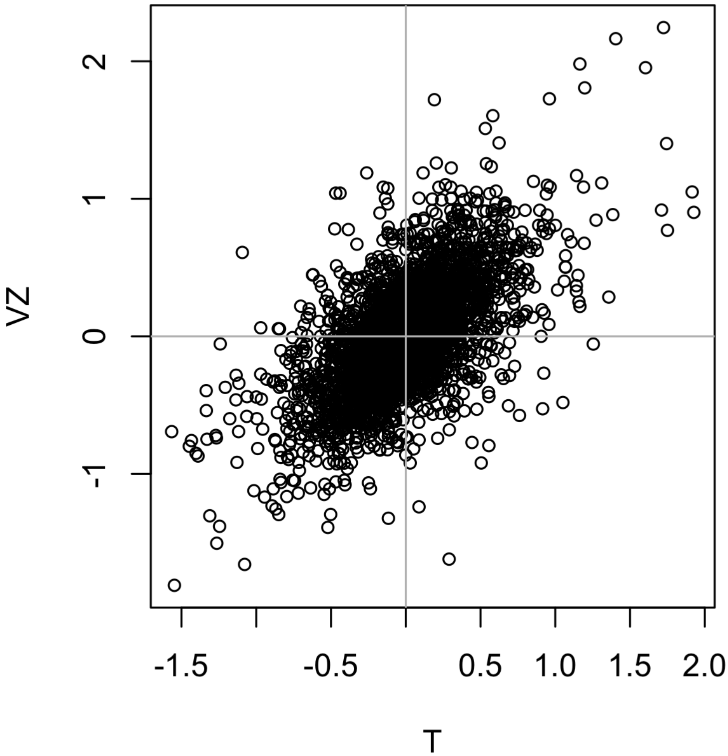


Figure 1-7. Scatterplot between returns for ATT and Verizon

Key Ideas for Correlation

- The correlation coefficient measures the extent to which two variables are associated with one another.
- When high values of v1 go with high values of v2, v1 and v2 are positively associated.
- When high values of v1 are associated with low values of v2, v1 and v2 are negatively associated.
- The correlation coefficient is a standardized metric so that it always ranges from –1 (perfect negative correlation) to +1 (perfect positive correlation).

- A correlation coefficient of 0 indicates no correlation, but be aware that random arrangements of data will produce both positive and negative values for the correlation coefficient just by chance.

Further Reading

Statistics, 4th ed., by David Freedman, Robert Pisani, and Roger Purves (W. W. Norton, 2007), has an excellent discussion of correlation.

Exploring Two or More Variables

Familiar estimators like mean and variance look at variables one at a time (*univariate analysis*). Correlation analysis (see "Correlation" on page 29) is an important method that compares two variables (*bivariate analysis*). In this section we look at additional estimates and plots, and at more than two variables (*multivariate analysis*).

Key Terms for Exploring Two or More Variables

Contingency tables
A tally of counts between two or more categorical variables.

Hexagonal binning
A plot of two numeric variables with the records binned into hexagons.

Contour plots
A plot showing the density of two numeric variables like a topographical map.

Violin plots
Similar to a boxplot but showing the density estimate.

Like univariate analysis, bivariate analysis involves both computing summary statistics and producing visual displays. The appropriate type of bivariate or multivariate analysis depends on the nature of the data: numeric versus categorical.

Hexagonal Binning and Contours (Plotting Numeric versus Numeric Data)

Scatterplots are fine when there is a relatively small number of data values. The plot of stock returns in Figure 1-7 involves only about 750 points. For data sets with hundreds of thousands or millions of records, a scatterplot will be too dense, so we need a different way to visualize the relationship. To illustrate, consider the data set `kc_tax`, which contains the tax-assessed values for residential properties in King County, Washington. In order to focus on the main part of the data, we strip out very expensive and very small or large residences using the `subset` function:

```
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 & SqFtTotLiving>100 &
                  SqFtTotLiving<3500)
nrow(kc_tax0)
[1] 432733
```

Figure 1-8 is a *hexagon binning* plot of the relationship between the finished square feet versus the tax-assessed value for homes in King County. Rather than plotting points, which would appear as a monolithic dark cloud, we grouped the records into hexagonal bins and plotted the hexagons with a color indicating the number of records in that bin. In this chart, the positive relationship between square feet and tax-assessed value is clear. An interesting feature is the hint of a second cloud above the main cloud, indicating homes that have the same square footage as those in the main cloud, but a higher tax-assessed value.

Figure 1-8 was generated by the powerful R package `ggplot2`, developed by Hadley Wickham [ggplot2]. `ggplot2` is one of several new software libraries for advanced exploratory visual analysis of data; see "Visualizing Multiple Variables" on page 40.

```
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient(low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```

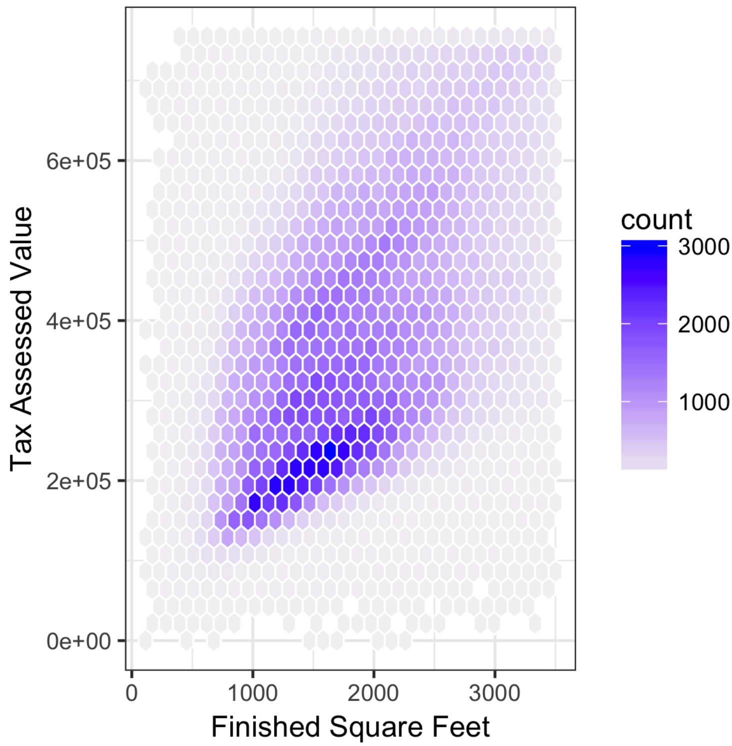


Figure 1-8. Hexagonal binning for tax-assessed value versus finished square feet

Figure 1-9 uses contours overlaid on a scatterplot to visualize the relationship between two numeric variables. The contours are essentially a topographical map to two variables; each contour band represents a specific density of points, increasing as one nears a "peak." This plot shows a similar story as Figure 1-8: there is a secondary peak "north" of the main peak. This chart was also created using `ggplot2` with the built-in `geom_density2d` function.

```
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point( alpha=0.1) +
  geom_density2d(colour="white") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```

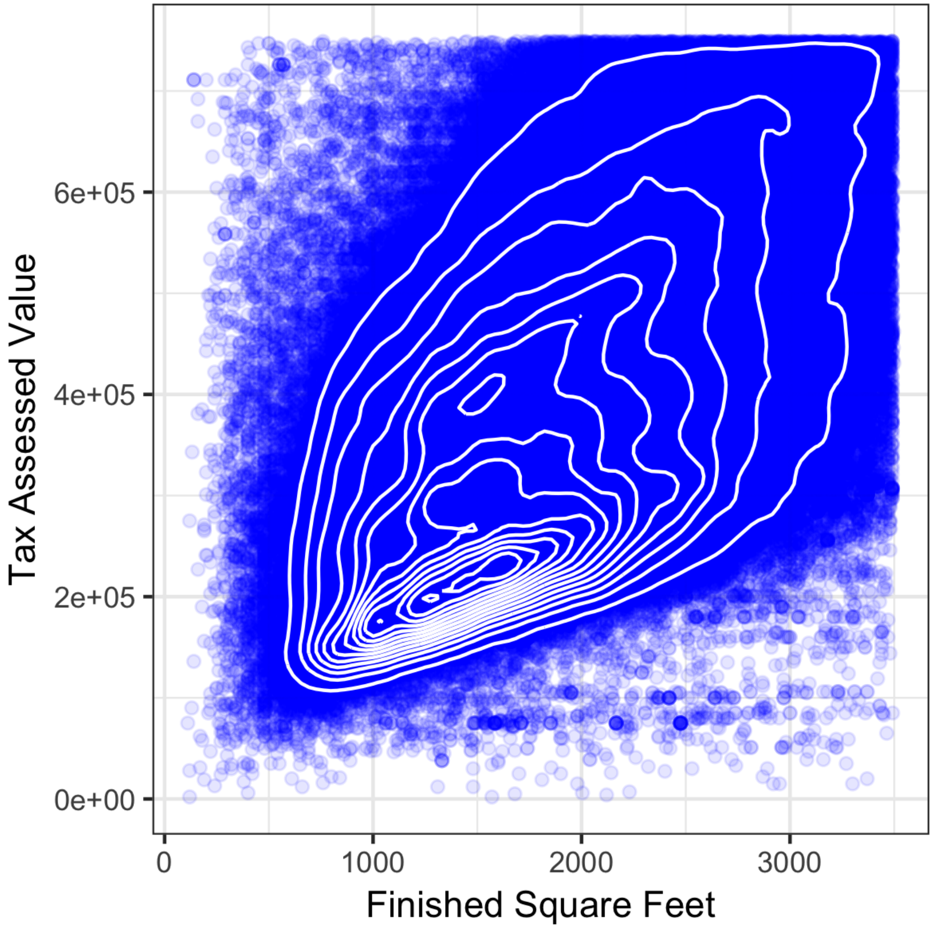


Figure 1-9. Contour plot for tax-assessed value versus finished square feet

Other types of charts are used to show the relationship between two numeric variables, including *heat maps*. Heat maps, hexagonal binning, and contour plots all give a visual representation of a two-dimensional density. In this way, they are natural analogs to histograms and density plots.

Two Categorical Variables

A useful way to summarize two categorical variables is a contingency table—a table of counts by category. Table 1-8 shows the contingency table between the grade of a personal loan and the outcome of that loan. This is taken from data provided by Lending Club, a leader in the peer-to-peer lending business. The grade goes from A (high) to G (low). The outcome is either paid off, current, late, or charged off (the balance of the loan is not expected to be collected). This table shows the count and row percentages. High-grade loans have a very low late/charge-off percentage as compared with lower-grade loans. Contingency tables can look at just counts, or also include column and total percentages. Pivot tables in Excel are perhaps the most common tool used to create contingency tables. In R, the `CrossTable` function in the `descr` package produces contingency tables, and the following code was used to create Table 1-8:

```
library(descr)
x_tab <- CrossTable(lc_loans$grade, lc_loans$status,
                    prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
```

Table 1-8. Contingency table of loan grade and status

Grade	Fully paid	Current	Late	Charged off	Total
A	20715	52058	494	1588	74855
	0.277	0.695	0.007	0.021	0.161
B	31782	97601	2149	5384	136916
	0.232	0.713	0.016	0.039	0.294
C	23773	92444	2895	6163	125275
	0.190	0.738	0.023	0.049	0.269
D	14036	55287	2421	5131	76875
	0.183	0.719	0.031	0.067	0.165
E	6089	25344	1421	2898	35752
	0.170	0.709	0.040	0.081	0.077
F	2376	8675	621	1556	13228
	0.180	0.656	0.047	0.118	0.028
G	655	2042	206	419	3322
	0.197	0.615	0.062	0.126	0.007
Total	99426	333451	10207	23139	466223

Categorical and Numeric Data

Boxplots (see "Percentiles and Boxplots" on page 20) are a simple way to visually compare the distributions of a numeric variable grouped according to a categorical variable. For example, we might want to compare how the percentage of flight delays varies across airlines. Figure 1-10 shows the percentage of flights in a month that were delayed where the delay was within the carrier's control.

```
boxplot(pct_delay ~ airline, data=airline_stats, ylim=c(0, 50))
```

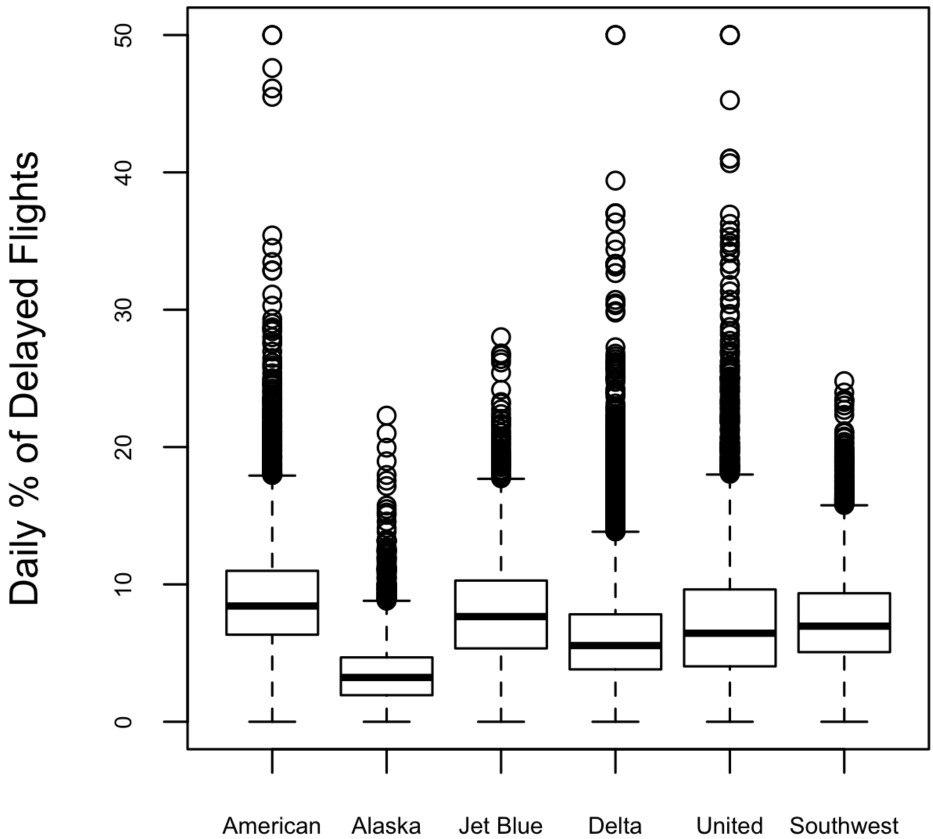


Figure 1-10. Boxplot of percent of airline delays by carrier

Alaska stands out as having the fewest delays, while American has the most delays: the lower quartile for American is higher than the upper quartile for Alaska.

A *violin plot*, introduced by [Hintze-Nelson-1998], is an enhancement to the boxplot and plots the density estimate with the density on the y-axis. The density is mirrored and flipped over and the resulting shape is filled in, creating an image resembling a violin. The advantage of a violin plot is that it can show nuances in the distribution that aren't perceptible in a boxplot. On the other hand, the boxplot more clearly shows the outliers in the data. In `ggplot2`, the function `geom_violin` can be used to create a violin plot as follows:

```
ggplot(data=airline_stats, aes(airline, pct_carrier_delay)) +
  ylim(0, 50) +
  geom_violin() +
  labs(x="", y="Daily % of Delayed Flights")
```

The corresponding plot is shown in Figure 1-11. The violin plot shows a concentration in the distribution near zero for Alaska, and to a lesser extent, Delta. This phenomenon is not as obvious in the boxplot. You can combine a violin plot with a boxplot by adding `geom_boxplot` to the plot (although this is best when colors are used).

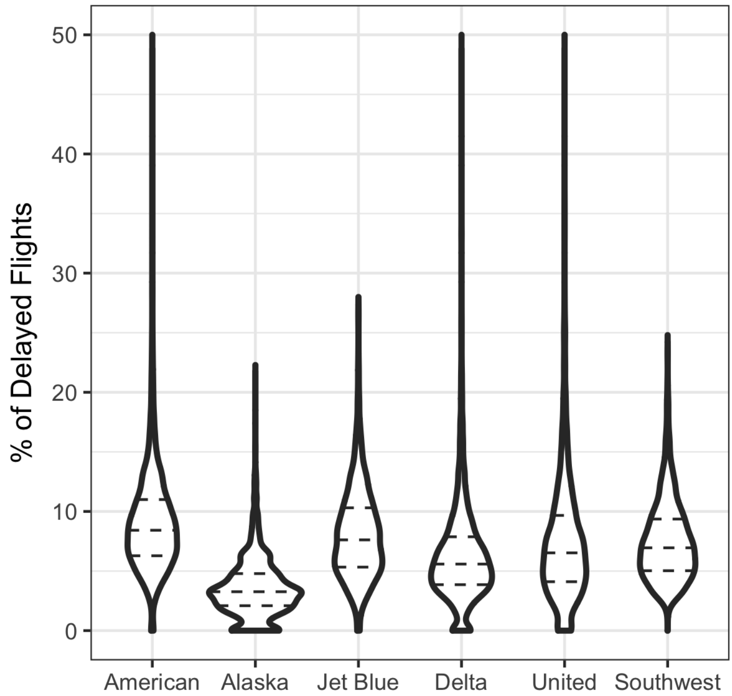


Figure 1-11. Combination of boxplot and violin plot of percent of airline delays by carrier

Visualizing Multiple Variables

The types of charts used to compare two variables—scatterplots, hexagonal binning, and boxplots—are readily extended to more variables through the notion of *conditioning*. As an example, look back at Figure 1-8, which showed the relationship between homes' finished square feet and tax-assessed values. We observed that there appears to be a cluster of homes that have higher tax-assessed value per square foot. Diving deeper, Figure 1-12 accounts for the effect of location by plotting the data for a set of zip codes. Now the picture is much clearer: tax-assessed value is much higher in some zip codes (98112, 98105) than in others (98108, 98057). This disparity gives rise to the clusters observed in Figure 1-8.

We created Figure 1-12 using `ggplot2` and the idea of *facets*, or a conditioning variable (in this case zip code):

```
ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
         aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="white", high="blue") +
  labs(x="Finished Square Feet", y="Tax Assessed Value") +
  facet_wrap("ZipCode")
```

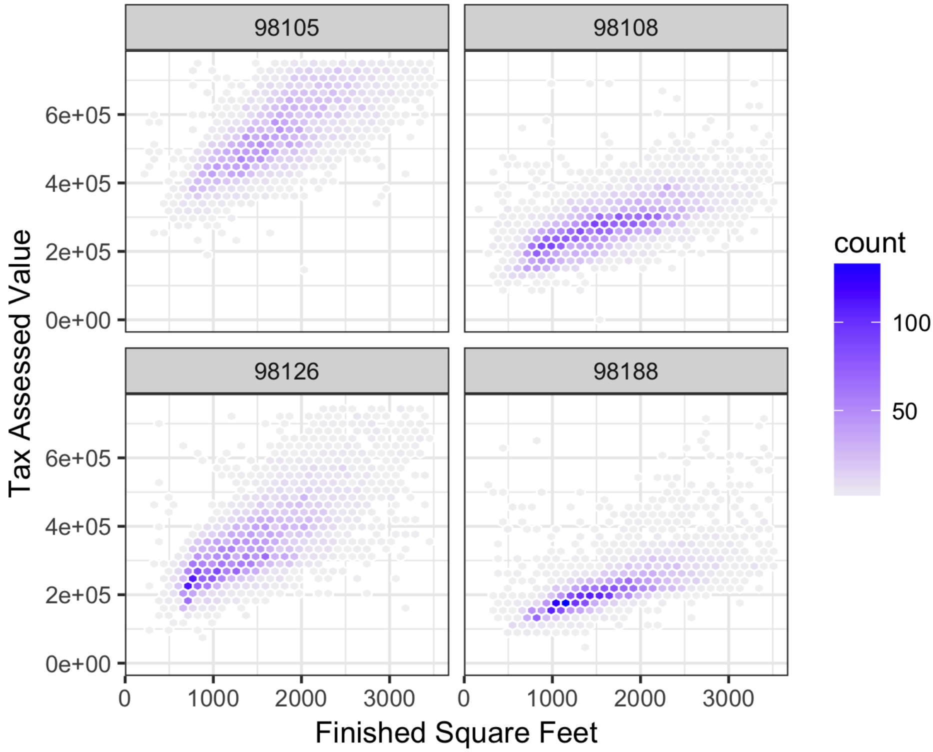


Figure 1-12. Tax-assessed value versus finished square feet by zip code

The concept of conditioning variables in a graphics system was pioneered with *Trellis graphics*, developed by Rick Becker, Bill Cleveland, and others at Bell Labs [Trellis-Graphics]. This idea has propogated to various modern graphics systems, such as the `lattice` ([lattice]) and `ggplot2` packages in R and the `Seaborn` ([seaborne]) and `Bokeh` ([bokeh]) modules in Python. Conditioning variables are also integral to business intelligence platforms such as Tableau and Spotfire. With the advent of vast computing power, modern visualization platforms have moved well beyond the humble beginnings of exploratory data analysis. However, key concepts and tools developed over the years still form a foundation for these systems.

Key Ideas

- Hexagonal binning and contour plots are useful tools that permit graphical examination of two numeric variables at a time, without being overwhelmed by huge amounts of data.
- Contingency tables are the standard tool for looking at the counts of two categorical variables.
- Boxplots and violin plots allow you to plot a numeric variable against a categorical variable.

Further Reading

- *Modern Data Science with R*, by Benjamin Baumer, Daniel Kaplan, and Nicholas Horton (CRC Press, 2017), has an excellent presentation of "a grammar for graphics" (the "gg" in `ggplot`).
- *Ggplot2: Elegant Graphics for Data Analysis*, by Hadley Wickham, is an excellent resource from the creator of `ggplot2` (Springer, 2009).
- Josef Fruehwald has a web-based tutorial on `ggplot2`.

Summary

With the development of exploratory data analysis (EDA), pioneered by John Tukey, statistics set a foundation that was a precursor to the field of data science. The key idea of EDA is that the first and most important step in any project based on data is to *look at the data*. By summarizing and visualizing the data, you can gain valuable intuition and understanding of the project.

This chapter has reviewed concepts ranging from simple metrics, such as estimates of location and variability, to rich visual displays to explore the relationships between multiple variables, as in Figure 1-12. The diverse set of tools and techniques being developed by the open source community, combined with the expressiveness of the R and Python languages, has created a plethora of ways to explore and analyze data. Exploratory analysis should be a cornerstone of any data science project.

CHAPTER 2

Data and Sampling Distributions

A popular misconception holds that the era of big data means the end of a need for sampling. In fact, the proliferation of data of varying quality and relevance reinforces the need for sampling as a tool to work efficiently with a variety of data and to minimize bias. Even in a big data project, predictive models are typically developed and piloted with samples. Samples are also used in tests of various sorts (e.g., pricing, web treatments).

Figure 2-1 shows a schematic that underpins the concepts in this chapter. The lefthand side represents a population that, in statistics, is assumed to follow an underlying but *unknown* distribution. The only thing available is the *sample* data and its empirical distribution, shown on the righthand side. To get from the lefthand side to the righthand side, a *sampling* procedure is used (represented by dashed arrows). Traditional statistics focused very much on the lefthand side, using theory based on strong assumptions about the population. Modern statistics has moved to the righthand side, where such assumptions are not needed.

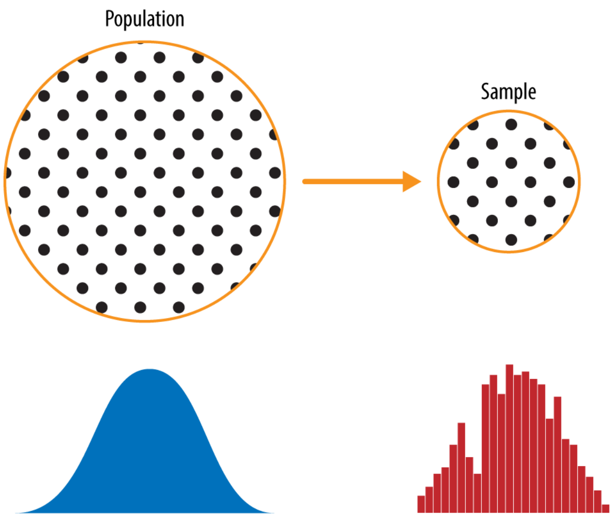


Figure 2-1. Population versus sample

In general, data scientists need not worry about the theoretical nature of the lefthand side, and instead should focus on the sampling procedures and the data at hand. There are some notable exceptions. Sometimes data is generated from a physical process that can be modeled. The simplest example is flipping a coin: this follows a binomial distribution. Any real-life binomial situation (buy or don't buy, fraud or no fraud, click or don't click) can be modeled effectively by a coin (with modified probability of landing heads, of course). In these cases, we can gain additional insight by using our understanding of the population.

Random Sampling and Sample Bias

A *sample* is a subset of data from a larger data set; statisticians call this larger data set the *population*. A population in statistics is not the same thing as in biology—it is a large, defined but sometimes theoretical or imaginary, set of data.

Key Terms for Random Sampling

Sample
A subset from a larger data set.

Population
The larger data set or idea of a data set.

N (n)
The size of the population (sample).

Random sampling
Drawing elements into a sample at random.

Stratified sampling
Dividing the population into strata and randomly sampling from each strata.

Simple random sample
The sample that results from random sampling without stratifying the population.

Sample bias
A sample that misrepresents the population.

Random sampling is a process in which each available member of the population being sampled has an equal chance of being chosen for the sample at each draw. The sample that results is called a *simple random sample*. Sampling can be done *with replacement*, in which observations are put back in the population after each draw for possible future reselection. Or it can be done *without replacement*, in which case observations, once selected, are unavailable for future draws.

Data quality often matters more than data quantity when making an estimate or a model based on a sample. Data quality in data science involves completeness, consistency of format, cleanliness, and accuracy of individual data points. Statistics adds the notion of *representativeness*.

The classic example is the *Literary Digest* poll of 1936 that predicted a victory of Al Landon against Franklin Roosevelt. The *Literary Digest*, a leading periodical of the day, polled its entire subscriber base, plus additional lists of individuals, a total of over 10 million, and predicted a landslide victory for Landon. George Gallup, founder of the Gallup Poll, conducted biweekly polls of just 2,000, and accurately predicted a Roosevelt victory. The difference lay in the selection of those polled.

The *Literary Digest* opted for quantity, paying little attention to the method of selection. They ended up polling those with relatively high socioeconomic status (their own subscribers, plus those who, by virtue of owning luxuries like telephones and automobiles, appeared in marketers' lists). The result was *sample bias*; that is, the sample was different in some meaningful nonrandom way from the larger population it was meant to represent. The term *nonrandom* is important—hardly any sample, including random samples, will be exactly representative of the population. Sample bias occurs when the difference is meaningful, and can be expected to continue for other samples drawn in the same way as the first.

Self-Selection Sampling Bias

The reviews of restaurants, hotels, cafes, and so on that you read on social media sites like Yelp are prone to bias because the people submitting them are not randomly selected; rather, they themselves have taken the initiative to write. This leads to self-selection bias—the people motivated to write reviews may be those who had poor experiences, may have an association with the establishment, or may simply be a different type of person from those who do not write reviews. Note that while self-selection samples can be unreliable indicators of the true state of affairs, they may be more reliable in simply comparing one establishment to a similar one; the same self-selection bias might apply to each.

Bias

Statistical bias refers to measurement or sampling errors that are systematic and produced by the measurement or sampling process. An important distinction should be made between errors due to random chance, and errors due to bias. Consider the physical process of a gun shooting at a target. It will not hit the absolute center of the target every time, or even much at all. An unbiased process will produce error, but it is random and does not tend strongly in any direction (see Figure 2-2). The results shown in Figure 2-3 show a biased process—there is still random error in both the x and y direction, but there is also a bias. Shots tend to fall in the upper-right quadrant.

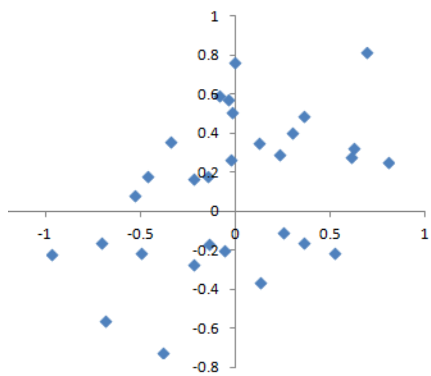


Figure 2-2. Scatterplot of shots from a gun with true aim

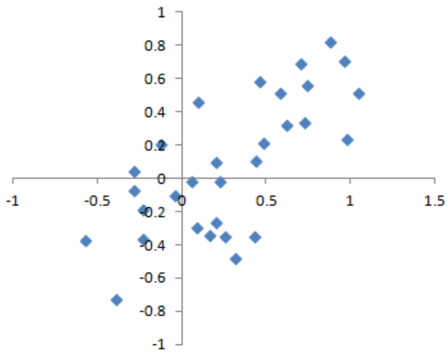


Figure 2-3. Scatterplot of shots from a gun with biased aim

Bias comes in different forms, and may be observable or invisible. When a result does suggest bias (e.g., by reference to a benchmark or actual values), it is often an indicator that a statistical or machine learning model has been misspecified, or an important variable left out.

Random Selection

To avoid the problem of sample bias that led the *Literary Digest* to predict Landon over Roosevelt, George Gallup (shown in Figure 2-4) opted for more scientifically chosen methods to achieve a sample that was representative of the US voter. There are now a variety of methods to achieve representativeness, but at the heart of all of them lies *random sampling*.

Figure 2-4. George Gallup, catapulted to fame by the Literary Digest's "big data" failure

Random sampling is not always easy. Proper definition of an accessible population is key. Suppose we want to generate a representative profile of customers and we need to conduct a pilot customer survey. The survey needs to be representative but is labor intensive.

First we need to define who a customer is. We might select all customer records where purchase amount > 0. Do we include all past customers? Do we include refunds? Internal test purchases? Resellers? Both billing agent and customer?

Next we need to specify a sampling procedure. It might be "select 100 customers at random." Where a sampling from a flow is involved (e.g., real-time customer transactions or web visitors), timing considerations may be important (e.g., a web visitor at 10 a.m. on a weekday may be different from a web visitor at 10 p.m. on a weekend).

In *stratified sampling*, the population is divided up into *strata*, and random samples are taken from each stratum. Political pollsters might seek to learn the electoral preferences of whites, blacks, and Hispanics. A simple random sample taken from the population would yield too few blacks and Hispanics, so those strata could be overweighted in stratified sampling to yield equivalent sample sizes.

Size versus Quality: When Does Size Matter?

In the era of big data, it is sometimes surprising that smaller is better. Time and effort spent on random sampling not only reduce bias, but also allow greater attention to data exploration and data quality. For example, missing data and outliers may contain useful information. It might be prohibitively expensive to track down missing values or evaluate outliers in millions of records, but doing so in a sample of several thousand records may be feasible. Data plotting and manual inspection bog down if there is too much data.

So when *are* massive amounts of data needed?

The classic scenario for the value of big data is when the data is not only big, but sparse as well. Consider the search queries received by Google, where columns are terms, rows are individual search queries, and cell values are either 0 or 1, depending on whether a query contains a term. The goal is to determine the best predicted search destination for a given query. There are over 150,000 words in the English language, and Google processes over 1 trillion queries per year. This yields a huge matrix, the vast majority of whose entries are "0."

This is a true big data problem—only when such enormous quantities of data are accumulated can effective search results be returned for most queries. And the more data accumulates, the better the results. For popular search terms this is not such a problem—effective data can be found fairly quickly for the handful of extremely popular topics trending at a particular time. The real value of modern search technology lies in the ability to return detailed and useful results for a huge variety of search queries, including those that occur only with a frequency, say, of one in a million.

Consider the search phrase "Ricky Ricardo and Little Red Riding Hood." In the early days of the internet, this query would probably have returned results on Ricky Ricardo the band leader, the television show *I Love Lucy* in which he starred, and the children's story *Little Red Riding Hood*. Later, now that trillions of search queries have been accumulated, this search query returns the exact *I Love Lucy* episode in which Ricky narrates, in dramatic fashion, the Little Red Riding Hood story to his infant son in a comic mix of English and Spanish.

Keep in mind that the number of actual *pertinent* records—ones in which this exact search query, or something very similar, appears (together with information on what link people ultimately clicked on)—might need only be in the thousands to be effective. However, many trillions of data points are needed in order to obtain these pertinent records (and random sampling, of course, will not help). See also "Long-Tailed Distributions" on page 67.

Sample Mean versus Population Mean

The symbol $\bar{x}$ (pronounced x-bar) is used to represent the mean of a sample from a population, whereas μ is used to represent the mean of a population. Why make the distinction? Information about samples is observed, and information about large populations is often inferred from smaller samples. Statisticians like to keep the two things separate in the symbology.

Key Ideas

- Even in the era of big data, random sampling remains an important arrow in the data scientist's quiver.
- Bias occurs when measurements or observations are systematically in error because they are not representative of the full population.
- Data quality is often more important than data quantity, and random sampling can reduce bias and facilitate quality improvement that would be prohibitively expensive.

Further Reading

- A useful review of sampling procedures can be found in Ronald Fricker's chapter "Sampling Methods for Web and E-mail Surveys," found in the *Sage Handbook of Online Research Methods*. This chapter includes a review of the modifications to random sampling that are often used for practical reasons of cost or feasibility.
- The story of the *Literary Digest* poll failure can be found on the Capital Century website.

Selection Bias

To paraphrase Yogi Berra, "If you don't know what you're looking for, look hard enough and you'll find it."

Selection bias refers to the practice of selectively choosing data—consciously or unconsciously—in a way that that leads to a conclusion that is misleading or ephemeral.

Key Terms

Bias
: Systematic error.

Data snooping
: Extensive hunting through data in search of something interesting.

Vast search effect
: Bias or nonreproducibility resulting from repeated data modeling, or modeling data with large numbers of predictor variables.

If you specify a hypothesis and conduct a well-designed experiment to test it, you can have high confidence in the conclusion. Such is often not the case, however. Often, one looks at available data and tries to discern patterns. But is the pattern for real, or just the product of *data snooping*—that is, extensive hunting through the data until something interesting emerges? There is a saying among statisticians: "If you torture the data long enough, sooner or later it will confess."

The difference between a phenomenon that you verify when you test a hypothesis using an experiment, versus a phenomenon that you discover by perusing available data, can be illuminated with the following thought experiment.

Imagine that someone tells you she can flip a coin and have it land heads on the next 10 tosses. You challenge her (the equivalent of an experiment), and she proceeds to toss it 10 times, all landing heads. Clearly you ascribe some special talent to her—the probability that 10 coin tosses will land heads just by chance is 1 in 1,000.

Now imagine that the announcer at a sports stadium asks the 20,000 people in attendance each to toss a coin 10 times, and report to an usher if they get 10 heads in a row. The chance that *somebody* in the stadium will get 10 heads is extremely high (more than 99%—it's 1 minus the probability that nobody gets 10 heads). Clearly, selecting, after the fact, the person (or persons) who gets 10 heads at the stadium does not indicate they have any special talent—it's most likely luck.

Since repeated review of large data sets is a key value proposition in data science, selection bias is something to worry about. A form of selection bias of particular concern to data scientists is what John Elder (founder of Elder Research, a respected data mining consultancy) calls the *vast search effect*. If you repeatedly run different models and ask different questions with a large data set, you are bound to find something interesting. Is the result you found truly something interesting, or is it the chance outlier?

We can guard against this by using a holdout set, and sometimes more than one holdout set, against which to validate performance. Elder also advocates the use of what he calls *target shuffling* (a permutation test, in essence) to test the validity of predictive associations that a data mining model suggests.

Typical forms of selection bias in statistics, in addition to the vast search effect, include nonrandom sampling (see *sampling bias*), cherry-picking data, selection of time intervals that accentuate a partiular statistical effect, and stopping an experiment when the results look "interesting."

Regression to the Mean

Regression to the mean refers to a phenomenon involving successive measurements on a given variable: extreme observations tend to be followed by more central ones. Attaching special focus and meaning to the extreme value can lead to a form of selection bias.

Sports fans are familiar with the "rookie of the year, sophomore slump" phenomenon. Among the athletes who begin their career in a given season (the rookie class), there is always one who performs better than all the rest. Generally, this "rookie of the year" does not do as well in his second year. Why not?

In nearly all major sports, at least those played with a ball or puck, there are two elements that play a role in overall performance:

- Skill
- Luck

Regression to the mean is a consequence of a particular form of selection bias. When we select the rookie with the best performance, skill and good luck are probably contributing. In his next season, the skill will still be there but, in most cases, the luck will not, so his performance will decline—it will regress. The phenomenon was first identified by Francis Galton in 1886 [Galton-1886], who wrote of it in connection with genetic tendencies; for example, the children of extremely tall men tend not to be as tall as their father (see Figure 2-5).

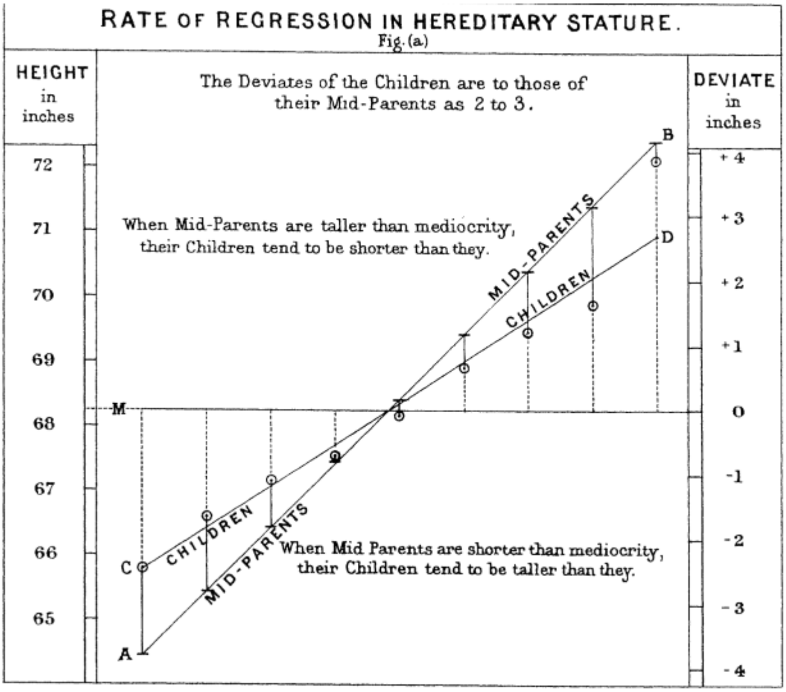


Figure 2-5. Galton's study that identified the phenomenon of regression to the mean

Regression to the mean, meaning to "go back," is distinct from the statistical modeling method of linear regression, in which a linear relationship is estimated between predictor variables and an outcome variable.

Key Ideas

- Specifying a hypothesis, then collecting data following randomization and random sampling principles, ensures against bias.
- All other forms of data analysis run the risk of bias resulting from the data collection/analysis process (repeated running of models in data mining, data snooping in research, and after-the-fact selection of interesting events).

Further Reading

- Christopher J. Pannucci and Edwin G. Wilkins' article "Identifying and Avoiding Bias in Research" in (surprisingly) *Plastic and Reconstructive Surgery* (August 2010) has an excellent review of various types of bias that can enter into research, including selection bias.
- Michael Harris's article "Fooled by Randomness Through Selection Bias" provides an interesting review of selection bias considerations in stock market trading schemes, from the perspective of traders.

Sampling Distribution of a Statistic

The term *sampling distribution* of a statistic refers to the distribution of some sample statistic, over many samples drawn from the same population. Much of classical statistics is concerned with making inferences from (small) samples to (very large) populations.

Key Terms

Sample statistic
: A metric calculated for a sample of data drawn from a larger population.

Data distribution
: The frequency distribution of individual *values* in a data set.

Sampling distribution
: The frequency distribution of a *sample statistic* over many samples or resamples.

Central limit theorem
: The tendency of the sampling distribution to take on a normal shape as sample size rises.

Standard error
: The variability (standard deviation) of a sample *statistic* over many samples (not to be confused with *standard deviation*, which, by itself, refers to variability of individual data *values*).

Typically, a sample is drawn with the goal of measuring something (with a *sample statistic*) or modeling something (with a statistical or machine learning model). Since our estimate or model is based on a sample, it might be in error; it might be different if we were to draw a different sample. We are therefore interested in how different it might be—a key concern is *sampling variability*. If we had lots of data, we could draw additional samples and observe the distribution of a sample statistic directly. Typically, we will calculate our estimate or model using as much data as is easily available, so the option of drawing additional samples from the population is not readily available.

> It is important to distinguish between the distribution of the individual data points, known as *the data distribution*, and the distribution of a sample statistic, known as the *sampling distribution*.

The distribution of a sample statistic such as the mean is likely to be more regular and bell-shaped than the distribution of the data itself. The larger the sample that the statistic is based on, the more this is true. Also, the larger the sample, the narrower the distribution of the sample statistic.

This is illustrated in an example using annual income for loan applicants to Lending Club (see "A Small Example: Predicting Loan Default" on page 211 for a description of the data). Take three samples from this data: a sample of 1,000 values, a sample of 1,000 means of 5 values, and a sample of 1,000 means of 20 values. Then plot a histogram of each sample to produce Figure 2-6.

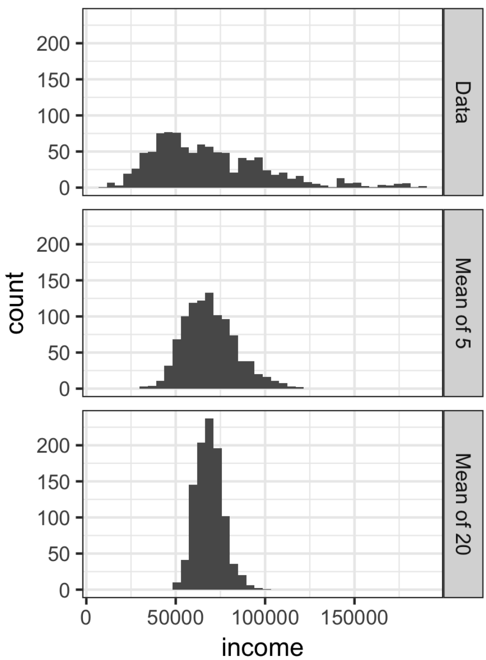


Figure 2-6. Histogram of annual incomes of 1,000 loan applicants (top), then 1000 means of n=5 applicants (middle), and n=20 (bottom)

The histogram of the individual data values is broadly spread out and skewed toward higher values as is to be expected with income data. The histograms of the means of 5 and 20 are increasingly compact and more bell-shaped. Here is the R code to generate these histograms, using the visualization package `ggplot2`.

```
library(ggplot2)
# take a simple random sample
samp_data <- data.frame(income=sample(loans_income, 1000),
                        type='data_dist')
# take a sample of means of 5 values
samp_mean_05 <- data.frame(
  income = tapply(sample(loans_income, 1000*5),
                  rep(1:1000, rep(5, 1000)), FUN=mean),
  type = 'mean_of_5')
# take a sample of means of 20 values
samp_mean_20 <- data.frame(
  income = tapply(sample(loans_income, 1000*20),
                  rep(1:1000, rep(20, 1000)), FUN=mean),
  type = 'mean_of_20')
# bind the data.frames and convert type to a factor
income <- rbind(samp_data, samp_mean_05, samp_mean_20)
income$type = factor(income$type,
                     levels=c('data_dist', 'mean_of_5', 'mean_of_20'),
                     labels=c('Data', 'Mean of 5', 'Mean of 20'))
# plot the histograms
ggplot(income, aes(x=income)) +
  geom_histogram(bins=40) +
  facet_grid(type ~ .)
```

Central Limit Theorem

This phenomenon is termed the *central limit theorem*. It says that the means drawn from multiple samples will resemble the familiar bell-shaped normal curve (see "Normal Distribution" on page 64), even if the source population is not normally distributed, provided that the sample size is large enough and the departure of the data from normality is not too great. The central limit theorem allows normal-approximation formulas like the t-distribution to be used in calculating sampling distributions for inference—that is, confidence intervals and hypothesis tests.

The central limit theorem receives a lot of attention in traditional statistics texts because it underlies the machinery of hypothesis tests and confidence intervals, which themselves consume half the space in such texts. Data scientists should be aware of this role, but, since formal hypothesis tests and confidence intervals play a small role in data science, and the bootstrap is available in any case, the central limit theorem is not so central in the practice of data science.

Standard Error

The *standard error* is a single metric that sums up the variability in the sampling distribution for a statistic. The standard error can be estimated using a statistic based on the standard deviation s of the sample values, and the sample size n:

$$\text{Standard error} = SE = \frac{s}{\sqrt{n}}$$

As the sample size increases, the standard error decreases, corresponding to what was observed in Figure 2-6. The relationship between standard error and sample size is sometimes referred to as the *square-root of n* rule: in order to reduce the standard error by a factor of 2, the sample size must be increased by a factor of 4.

The validity of the standard error formula arises from the central limit theorem (see "Central Limit Theorem" on page 55). In fact, you don't need to rely on the central limit theorem to understand standard error. Consider the following approach to measure standard error:

1. Collect a number of brand new samples from the population.
2. For each new sample, calculate the statistic (e.g., mean).
3. Calculate the standard deviation of the statistics computed in step 2; use this as your estimate of standard error.

In practice, this approach of collecting new samples to estimate the standard error is typically not feasible (and statistically very wasteful). Fortunately, it turns out that it is not necessary to draw brand new samples; instead, you can use *bootstrap* resamples (see "The Bootstrap" on page 57). In modern statistics, the bootstrap has become the standard way to to estimate standard error. It can be used for virtually any statistic and does not rely on the central limit theorem or other distributional assumptions.

Standard Deviation versus Standard Error

Do not confuse standard deviation (which measures the variability of individual data points) with standard error (which measures the variability of a sample metric).

Key Ideas

- The frequency distribution of a sample statistic tells us how that metric would turn out differently from sample to sample.
- This sampling distribution can be estimated via the bootstrap, or via formulas that rely on the central limit theorem.
- A key metric that sums up the variability of a sample statistic is its standard error.

Further Reading

David Lane's online multimedia resource in statistics has a useful simulation that allows you to select a sample statistic, a sample size and number of iterations and visualize a histogram of the resulting frequency distribution.

The Bootstrap

One easy and effective way to estimate the sampling distribution of a statistic, or of model parameters, is to draw additional samples, with replacement, from the sample itself and recalculate the statistic or model for each resample. This procedure is called the *bootstrap*, and it does not necessarily involve any assumptions about the data or the sample statistic being normally distributed.

Key Terms

Bootstrap sample
A sample taken with replacement from an observed data set.

Resampling
The process of taking repeated samples from observed data; includes both bootstrap and permutation (shuffling) procedures.

Conceptually, you can imagine the bootstrap as replicating the original sample thousands or millions of times so that you have a hypothetical population that embodies all the knowledge from your original sample (it's just larger). You can then draw samples from this hypothetical population for the purpose of estimating a sampling distribution. See Figure 2-7.

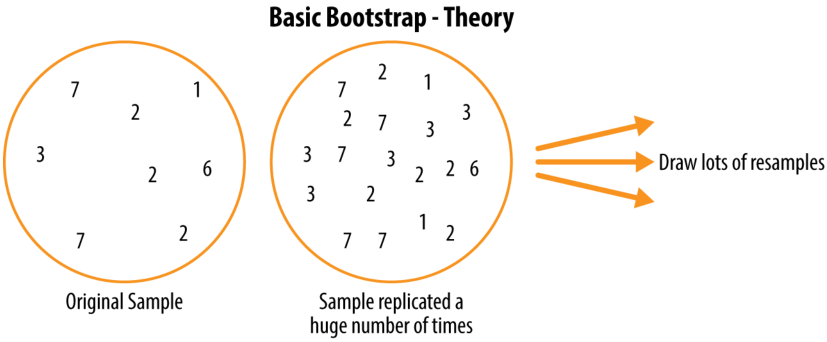


Figure 2-7. The idea of the bootstrap

In practice, it is not necessary to actually replicate the sample a huge number of times. We simply replace each observation after each draw; that is, we *sample with replacement*. In this way we effectively create an infinite population in which the probability of an element being drawn remains unchanged from draw to draw. The algorithm for a bootstrap resampling of the mean is as follows, for a sample of size *n*:

1. Draw a sample value, record, replace it.
2. Repeat *n* times.
3. Record the mean of the *n* resampled values.
4. Repeat steps 1–3 *R* times.
5. Use the *R* results to:
    a. Calculate their standard deviation (this estimates sample mean standard error).
    b. Produce a histogram or boxplot.
    c. Find a confidence interval.

R, the number of iterations of the bootstrap, is set somewhat arbitrarily. The more iterations you do, the more accurate the estimate of the standard error, or the confidence interval. The result from this procedure is a bootstrap set of sample statistics or estimated model parameters, which you can then examine to see how variable they are.

The R package `boot` combines these steps in one function. For example, the following applies the bootstrap to the incomes of people taking out loans:

```
library(boot)
stat_fun <- function(x, idx) median(x[idx])
boot_obj <- boot(loans_income, R = 1000, statistic=stat_fun)
```

The function `stat_fun` computes the median for a given sample specified by the index `idx`. The result is as follows:

```
Bootstrap Statistics :
    original   bias    std. error
t1*    62000 -70.5595    209.1515
```

The original estimate of the median is $62,000. The bootstrap distribution indicates that the estimate has a *bias* of about –$70 and a standard error of $209.

The bootstrap can be used with multivariate data, where the rows are sampled as units (see Figure 2-8). A model might then be run on the bootstrapped data, for example, to estimate the stability (variability) of model parameters, or to improve predictive power. With classification and regression trees (also called *decision trees*), running multiple trees on bootstrap samples and then averaging their predictions (or, with classification, taking a majority vote) generally performs better than using a single tree. This process is called *bagging* (short for "bootstrap aggregating": see "Bagging and the Random Forest" on page 228).

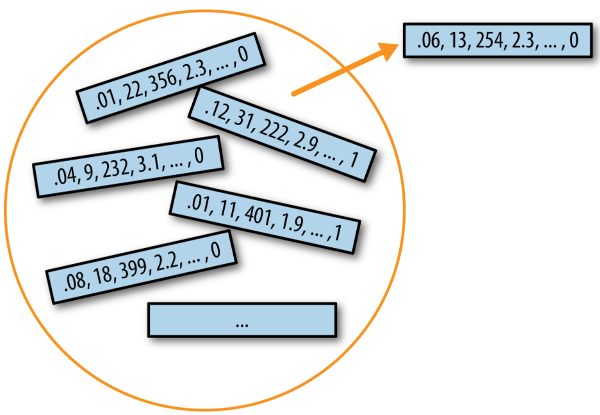


Figure 2-8. Multivariate bootstrap sampling

The repeated resampling of the bootstrap is conceptually simple, and Julian Simon, an economist and demographer, published a compendium of resampling examples, including the bootstrap, in his 1969 text *Basic Research Methods in Social Science* (Random House). However, it is also computationally intensive, and was not a feasible option before the widespread availability of computing power. The technique gained its name and took off with the publication of several journal articles and a book by Stanford statistician Bradley Efron in the late 1970s and early 1980s. It was particularly popular among researchers who use statistics but are not statisticians, and for use with metrics or models where mathematical approximations are not readily available. The sampling distribution of the mean has been well established since 1908; the sampling distribution of many other metrics has not. The bootstrap can be used for sample size determination; experiment with different values for n to see how the sampling distribution is affected.

The bootstrap met with considerable skepticism when it was first introduced; it had the aura to many of spinning gold from straw. This skepticism stemmed from a misunderstanding of the bootstrap's purpose.

> The bootstrap does not compensate for a small sample size; it does not create new data, nor does it fill in holes in an existing data set. It merely informs us about how lots of additional samples would behave when drawn from a population like our original sample.

Resampling versus Bootstrapping

Sometimes the term *resampling* is used synonymously with the term *bootstrapping*, as just outlined. More often, the term *resampling* also includes permutation procedures (see "Permutation Test" on page 88), where multiple samples are combined and the sampling may be done without replacement. In any case, the term *bootstrap* always implies sampling with replacement from an observed data set.

Key Ideas

- The bootstrap (sampling with replacement from a data set) is a powerful tool for assessing the variability of a sample statistic.
- The bootstrap can be applied in similar fashion in a wide variety of circumstances, without extensive study of mathematical approximations to sampling distributions.
- It also allows us to estimate sampling distributions for statistics where no mathematical approximation has been developed.
- When applied to predictive models, aggregating multiple bootstrap sample predictions (bagging) outperforms the use of a single model.

Further Reading

- *An Introduction to the Bootstrap* by Bradley Efron and Robert Tibshirani (Chapman Hall, 1993) was the first book-length treatment of the bootstrap. It is still widely read.

- The retrospective on the bootstrap in the May 2003 issue of *Statistical Science*, (vol. 18, no. 2), discusses (among other antecedents, in Peter Hall's "Prehistory") Julian Simon's first publication of the bootstrap in 1969.
- See *An Introduction to Statistical Learning* by Gareth James et al. (Springer, 2013) for sections on the bootstrap and, in particular, bagging.

Confidence Intervals

Frequency tables, histograms, boxplots, and standard errors are all ways to understand the potential error in a sample estimate. Confidence intervals are another.

Key Terms

Confidence level
: The percentage of confidence intervals, constructed in the same way from the same population, expected to contain the statistic of interest.

Interval endpoints
: The top and bottom of the confidence interval.

There is a natural human aversion to uncertainty; people (especially experts) say, "I don't know" far too rarely. Analysts and managers, while acknowledging uncertainty, nonetheless place undue faith in an estimate when it is presented as a single number (a *point estimate*). Presenting an estimate not as a single number but as a range is one way to counteract this tendency. Confidence intervals do this in a manner grounded in statistical sampling principles.

Confidence intervals always come with a coverage level, expressed as a (high) percentage, say 90% or 95%. One way to think of a 90% confidence interval is as follows: it is the interval that encloses the central 90% of the bootstrap sampling distribution of a sample statistic (see "The Bootstrap" on page 57). More generally, an *x*% confidence interval around a sample estimate should, on average, contain similar sample estimates *x*% of the time (when a similar sampling procedure is followed).

Given a sample of size *n*, and a sample statistic of interest, the algorithm for a bootstrap confidence interval is as follows:

1. Draw a random sample of size *n* with replacement from the data (a resample).
2. Record the statistic of interest for the resample.
3. Repeat steps 1–2 many (*R*) times.

4. For an x% confidence interval, trim [(1 – [x/100]) / 2]% of the R resample results from either end of the distribution.
5. The trim points are the endpoints of an x% bootstrap confidence interval.

Figure 2-9 shows a a 90% confidence interval for the mean annual income of loan applicants, based on a sample of 20 for which the mean was $57,573.

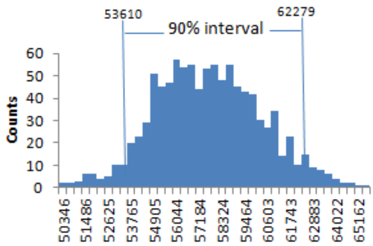


Figure 2-9. Bootstrap confidence interval for the annual income of loan applicants, based on a sample of 20

The bootstrap is a general tool that can be used to generate confidence intervals for most statistics, or model parameters. Statistical textbooks and software, with roots in over a half-century of computerless statistical analysis, will also reference confidence intervals generated by formulas, especially the t-distribution (see "Student's t-Distribution" on page 69).

> Of course, what we are really interested in when we have a sample result is "what is the probability that the true value lies within a certain interval?" This is not really the question that a confidence interval answers, but it ends up being how most people interpret the answer.
>
> The probability question associated with a confidence interval starts out with the phrase "Given a sampling procedure and a population, what is the probability that…" To go in the opposite direction, "Given a sample result, what is the probability that (something is true about the population)," involves more complex calculations and deeper imponderables.

The percentage associated with the confidence interval is termed the *level of confidence*. The higher the level of confidence, the wider the interval. Also, the smaller the sample, the wider the interval (i.e., the more uncertainty). Both make sense: the more confident you want to be, and the less data you have, the wider you must make the confidence interval to be sufficiently assured of capturing the true value.

> For a data scientist, a confidence interval is a tool to get an idea of how variable a sample result might be. Data scientists would use this information not to publish a scholarly paper or submit a result to a regulatory agency (as a researcher might), but most likely to communicate the potential error in an estimate, and, perhaps, learn whether a larger sample is needed.

Key Ideas

- Confidence intervals are the typical way to present estimates as an interval range.
- The more data you have, the less variable a sample estimate will be.
- The lower the level of confidence you can tolerate, the narrower the confidence interval will be.
- The bootstrap is an effective way to construct confidence intervals.

Further Reading

- For a bootstrap approach to confidence intervals, see *Introductory Statistics and Analytics: A Resampling Perspective* by Peter Bruce (Wiley, 2014) or *Statistics* by Robin Lock and four other Lock family members (Wiley, 2012).
- Engineers, who have a need to understand the precision of their measurements, use confidence intervals perhaps more than most disciplines, and *Modern Engineering Statistics* by Tom Ryan (Wiley, 2007) discusses confidence intervals. It also reviews a tool that is just as useful and gets less attention: prediction intervals (intervals around a single value, as opposed to a mean or other summary statistic).

Normal Distribution

The bell-shaped normal distribution is iconic in traditional statistics.[1] The fact that distributions of sample statistics are often normally shaped has made it a powerful tool in the development of mathematical formulas that approximate those distributions.

Key Terms

Error
: The difference between a data point and a predicted or average value.

Standardize
: Subtract the mean and divide by the standard deviation.

z-score
: The result of standardizing an individual data point.

Standard normal
: A normal distribution with mean = 0 and standard deviation = 1.

QQ-Plot
: A plot to visualize how close a sample distribution is to a normal distribution.

In a normal distribution (Figure 2-10), 68% of the data lies within one standard deviation of the mean, and 95% lies within two standard deviations.

> It is a common misconception that the normal distribution is called that because most data follows a normal distribution—that is, it is the normal thing. Most of the variables used in a typical data science project—in fact most raw data as a whole—are *not* normally distributed: see "Long-Tailed Distributions" on page 67. The utility of the normal distribution derives from the fact that many statistics *are* normally distributed in their sampling distribution. Even so, assumptions of normality are generally a last resort, used when empirical probability distributions, or bootstrap distributions, are not available.

1 The bell curve is iconic but perhaps overrated. George W. Cobb, the Mount Holyoke statistician noted for his contribution to the philosophy of teaching introductory statistics, argued in a November 2015 editorial in the *American Statistician* that the "standard introductory course, which puts the normal distribution at its center, had outlived the usefulness of its centrality."

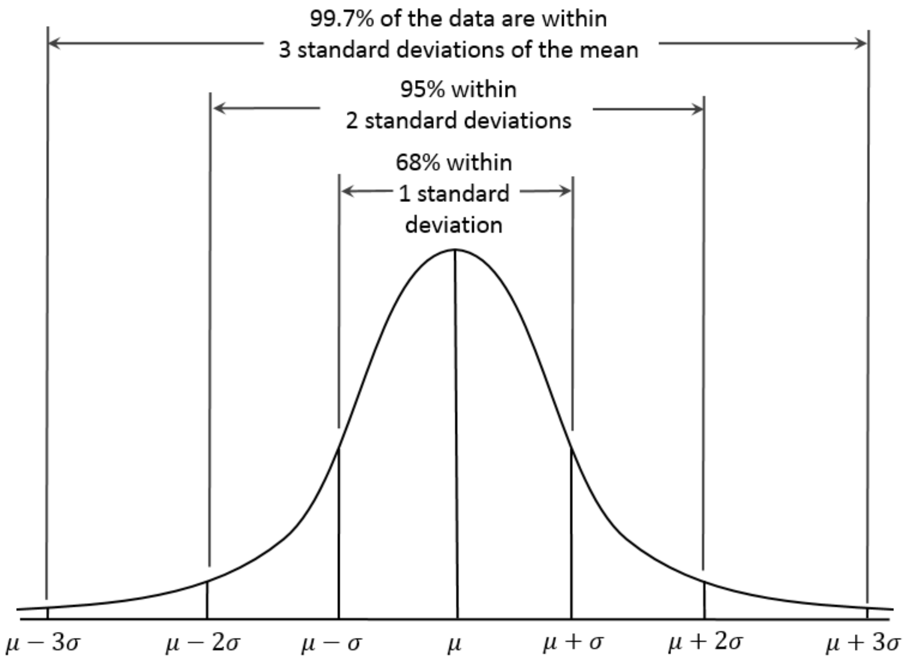


Figure 2-10. Normal curve

> The normal distribution is also referred to as a *Gaussian* distribution after Carl Friedrich Gauss, a prodigous German mathematician from the late 18th and early 19th century. Another name previously used for the normal distribution was the "error" distribution. Statistically speaking, an *error* is the difference between an actual value and a statistical estimate like the sample mean. For example, the standard deviation (see "Estimates of Variability" on page 13) is based on the errors from the mean of the data. Gauss's development of the normal distribution came from his study of the errors of astronomical measurements that were found to be normally distributed.

Standard Normal and QQ-Plots

A *standard normal* distribution is one in which the units on the x-axis are expressed in terms of standard deviations away from the mean. To compare data to a standard normal distribution, you subtract the mean then divide by the standard deviation; this is also called *normalization* or *standardization* (see "Standardization (Normalization, Z-Scores)" on page 215). Note that "standardization" in this sense is unrelated to database record standardization (conversion to a common format). The transformed value is termed a *z-score*, and the normal distribution is sometimes called the *z-distribution*.

A QQ-Plot is used to visually determine how close a sample is to the normal distribution. The QQ-Plot orders the *z*-scores from low to high, and plots each value's *z*-score on the y-axis; the x-axis is the corresponding quantile of a normal distribution for that value's rank. Since the data is normalized, the units correspond to the number of standard deviations away of the data from the mean. If the points roughly fall on the diagonal line, then the sample distribution can be considered close to normal. Figure 2-11 shows a QQ-Plot for a sample of 100 values randomly generated from a normal distribution; as expected, the points closely follow the line. This figure can be produced in R with the `qqnorm` function:

```
norm_samp <- rnorm(100)
qqnorm(norm_samp)
abline(a=0, b=1, col='grey')
```

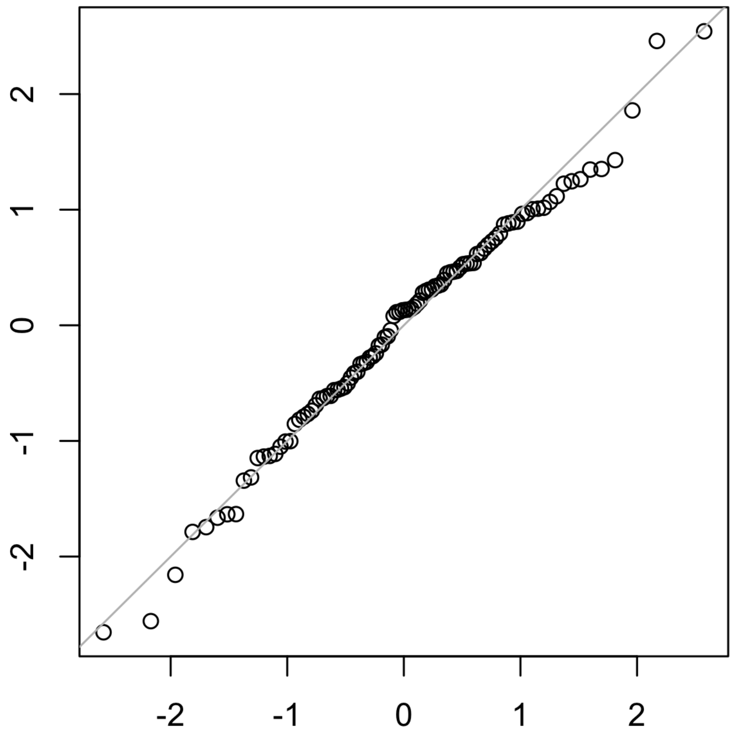


Figure 2-11. QQ-Plot of a sample of 100 values drawn from a normal distribution

Converting data to *z*-scores (i.e., standardizing or normalizing the data) does *not* make the data normally distributed. It just puts the data on the same scale as the standard normal distribution, often for comparison purposes.

Key Ideas

- The normal distribution was essential to the historical development of statistics, as it permitted mathematical approximation of uncertainty and variability.
- While raw data is typically not normally distributed, errors often are, as are averages and totals in large samples.
- To convert data to *z*-scores, you subtract the mean of the data and divide by the standard deviation; you can then compare the data to a normal distribution.

Long-Tailed Distributions

Despite the importance of the normal distribution historically in statistics, and in contrast to what the name would suggest, data is generally not normally distributed.

Key Terms for Long-Tail Distribution

Tail
: The long narrow portion of a frequency distribution, where relatively extreme values occur at low frequency.

Skew
: Where one tail of a distribution is longer than the other.

While the normal distribution is often appropriate and useful with respect to the distribution of errors and sample statistics, it typically does not characterize the distribution of raw data. Sometimes, the distribution is highly *skewed* (asymmetric), such as with income data, or the distribution can be discrete, as with binomial data. Both symmetric and asymmetric distributions may have *long tails*. The tails of a distribution correspond to the extreme values (small and large). Long tails, and guarding against them, are widely recognized in practical work. Nassim Taleb has proposed the *black swan* theory, which predicts that anamolous events, such as a stock market crash, are much more likely to occur than would be predicted by the normal distribution.

A good example to illustrate the long-tailed nature of data is stock returns. Figure 2-12 shows the QQ-Plot for the daily stock returns for Netflix (NFLX). This is generated in R by:

```
nflx <- sp500_px[,'NFLX']
nflx <- diff(log(nflx[nflx>0]))
qqnorm(nflx)
abline(a=0, b=1, col='grey')
```

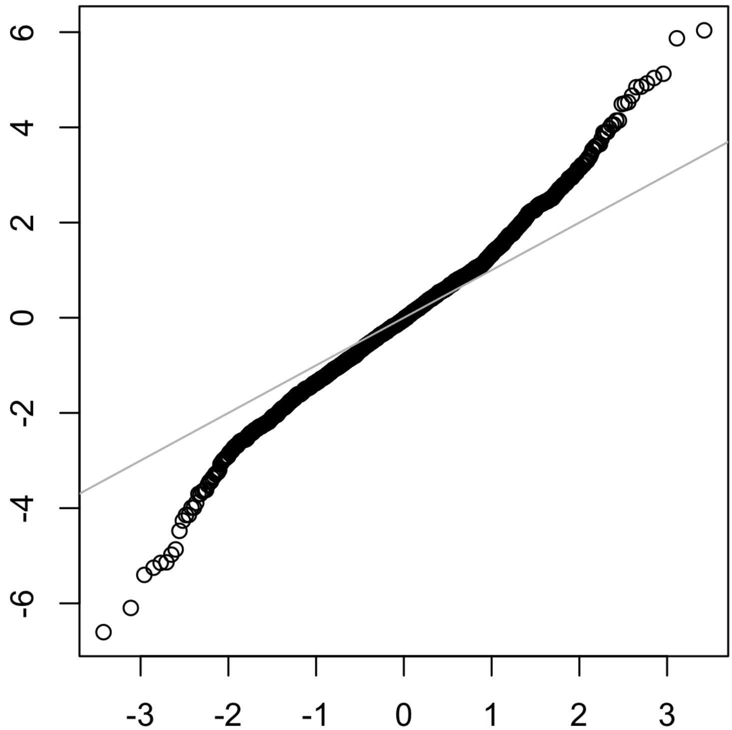


Figure 2-12. QQ-Plot of the returns for NFLX

In contrast to Figure 2-11, the points are far below the line for low values and far above the line for high values. This means that we are much more likely to observe extreme values than would be expected if the data had a normal distribution. Figure 2-12 shows another common phenomena: the points are close to the line for the data within one standard deviation of the mean. Tukey refers to this phenomenon as data being "normal in the middle," but having much longer tails (see [Tukey-1987]).

There is much statistical literature about the task of fitting statistical distributions to observed data. Beware an excessively data-centric approach to this job, which is as much art as science. Data is variable, and often consistent, on its face, with more than one shape and type of distribution. It is typically the case that domain and statistical knowledge must be brought to bear to determine what type of distribution is appropriate to model a given situation. For example, we might have data on the level of internet traffic on a server over many consecutive 5-second periods. It is useful to know that the best distribution to model "events per time period" is the Poisson (see "Poisson Distributions" on page 75).

Key Ideas for Long-Tail Distribution

- Most data is not normally distributed.
- Assuming a normal distribution can lead to underestimation of extreme events ("black swans").

Further Reading

- *The Black Swan*, 2nd ed., by Nassim Taleb (Random House, 2010).
- *Handbook of Statistical Distributions with Applications*, 2nd ed., by K. Krishnamoorthy (CRC Press, 2016)

Student's t-Distribution

The *t-distribution* is a normally shaped distribution, but a bit thicker and longer on the tails. It is used extensively in depicting distributions of sample statistics. Distributions of sample means are typically shaped like a t-distribution, and there is a family of t-distributions that differ depending on how large the sample is. The larger the sample, the more normally shaped the t-distribution becomes.

Key Terms for Student's t-Distribution

n
: Sample size.

Degrees of freedom
: A parameter that allows the t-distribution to adjust to different sample sizes, statistics, and number of groups.

The t-distribution is often called *Student's t* because it was published in 1908 in *Biometrika* by W. S. Gossett under the name "Student." Gossett's employer, the Guinness brewery, did not want competitors to know that it was using statistical methods, so insisted that Gossett not use his name on the article.

Gossett wanted to answer the question "What is the sampling distribution of the mean of a sample, drawn from a larger population?" He started out with a resampling experiment—drawing random samples of 4 from a data set of 3,000 measurements of criminals' height and left-middle-finger lengths. (This being the era of eugenics, there was much interest in data on criminals, and in discovering correlations between criminal tendencies and physical or psychological attributes.) He plotted the standardized results (the *z*-scores) on the x-axis and the frequency on the y-axis. Separately, he had derived a function, now known as *Student's t*, and he fit this function over the sample results, plotting the comparison (see Figure 2-13).

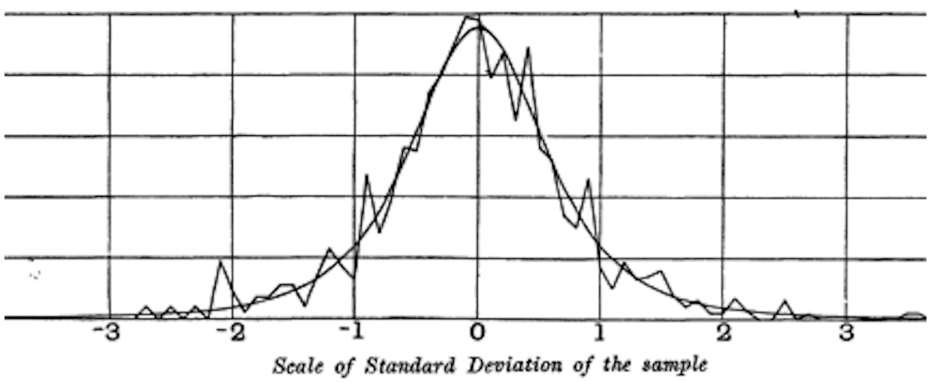


Figure 2-13. Gossett's resampling experiment results and fitted t-curve (from his 1908 Biometrika paper)

A number of different statistics can be compared, after standardization, to the t-distribution, to estimate confidence intervals in light of sampling variation. Consider a sample of size n for which the sample mean $\bar{x}$ has been calculated. If s is the sample standard deviation, a 90% confidence interval around the sample mean is given by:

$$\bar{x} \pm t_{n-1}(.05) \times \frac{s}{n}$$

where $t_{n-1}(.05)$ is the value of the t-statistic, with $(n - 1)$ degrees of freedom (see "Degrees of Freedom" on page 104), that "chops off" 5% of the t-distribution at either end. The t-distribution has been used as a reference for the distribution of a sample mean, the difference between two sample means, regression parameters, and other statistics.

Had computing power been widely available in 1908, statistics would no doubt have relied much more heavily on computationally intensive resampling methods from the start. Lacking computers, statisticians turned to mathematics and functions such as the t-distribution to approximate sampling distributions. Computer power enabled practical resampling experiments in the 1980s, but by then, use of the t-distribution and similar distributions had become deeply embedded in textbooks and software.

The t-distribution's accuracy in depicting the behavior of a sample statistic requires that the distribution of that statistic for that sample be shaped like a normal distribution. It turns out that sample statistics *are* often normally distributed, even when the underlying population data is not (a fact which led to widespread application of the t-distribution). This phenomenon is termed the *central limit theorem* (see "Central Limit Theorem" on page 55).

> What do data scientists need to know about the t-distribution and the central limit theorem? Not a whole lot. These distributions are used in classical statistical inference, but are not as central to the purposes of data science. Understanding and quantifying uncertainty and variation are important to data scientists, but empirical bootstrap sampling can answer most questions about sampling error. However, data scientists will routinely encounter t-statistics in output from statistical software and statistical procedures in R, for example in A-B tests and regressions, so familiarity with its purpose is helpful.

Key Ideas

- The t-distribution is actually a family of distributions resembling the normal distribution, but with thicker tails.
- It is widely used as a reference basis for the distribution of sample means, differences between two sample means, regression parameters, and more.

Further Reading

- The original Gossett paper in *Biometrica* from 1908 is available as a PDF.
- A standard treatment of the t-distribution can be found in David Lane's online resource.

Binomial Distribution

Key Terms for Binomial Distribution

Trial
: An event with a discrete outcome (e.g., a coin flip).

Success
: The outcome of interest for a trial.

  Synonyms
  : "1" (as opposed to "0")

Binomial
: Having two outcomes.

  Synonyms
  : yes/no, 0/1, binary

Binomial trial
: A trial with two outcomes.

  Synonym
  : Bernoulli trial

Binomial distribution
: Distribution of number of successes in x trials.

Synonym

Bernoulli distribution

Yes/no (binomial) outcomes lie at the heart of analytics since they are often the culmination of a decision or other process; buy/don't buy, click/don't click, survive/die, and so on. Central to understanding the binomial distribution is the idea of a set of *trials*, each trial having two possible outcomes with definite probabilities.

For example, flipping a coin 10 times is a binomial experiment with 10 trials, each trial having two possible outcomes (heads or tails); see Figure 2-14. Such yes/no or 0/1 outcomes are termed *binary* outcomes, and they need not have 50/50 probabilities. Any probabilities that sum to 1.0 are possible. It is conventional in statistics to term the "1" outcome the *success* outcome; it is also common practice to assign "1" to the more rare outcome. Use of the term *success* does not imply that the outcome is desirable or beneficial, but it does tend to indicate the outcome of interest. For example, loan defaults or fraudulent transactions are relatively uncommon events that we may be interested in predicting, so they are termed "1s" or "successes."

Figure 2-14. The tails side of a buffalo nickel

The binomial distribution is the frequency distribution of the number of successes (x) in a given number of trials (n) with specified probability (p) of success in each trial. There is a family of binomial distributions, depending on the values of x, n, and p. The binomial distribution would answer a question like:

> If the probability of a click converting to a sale is 0.02, what is the probability of observing 0 sales in 200 clicks?

The R function `dbinom` calculates binomial probabilities. For example:

```
dbinom(x=2, n=5, p=0.1)
```

would return 0.0729, the probability of observing exactly $x = 2$ successes in $n = 5$ trials, where the probability of success for each trial is $p = 0.1$.

Often we are interested in determining the probability of x or fewer successes in n trials. In this case, we use the function `pbinom`:

```
pbinom(2, 5, 0.1)
```

This would return 0.9914, the probability of observing two or fewer successes in five trials, where the probability of success for each trial is 0.1.

The mean of a binomial distribution is $n \times p$; you can also think of this as the expected number of successes in n trials, for success probability = p.

The variance is $n \times p(1 - p)$. With a large enough number of trials (particularly when p is close to 0.50), the binomial distribution is virtually indistinguishable from the normal distribution. In fact, calculating binomial probabilities with large sample sizes is computationally demanding, and most statistical procedures use the normal distribution, with mean and variance, as an approximation.

Key Ideas

- Binomial outcomes are important to model, since they represent, among other things, fundamental decisions (buy or don't buy, click or don't click, survive or die, etc.).
- A binomial trial is an experiment with two possible outcomes: one with probability p and the other with probability $1 - p$.
- With large n, and provided p is not too close to 0 or 1, the binomial distribution can be approximated by the normal distribution.

Further Reading

- Read about the "quincunx", a pinball-like simulation device for illustrating the binomial distribution.
- The binomial distribution is a staple of introductory statistics, and all introductory statistics texts will have a chapter or two on it.

Poisson and Related Distributions

Many processes produce events randomly at a given overall rate—visitors arriving at a website, cars arriving at a toll plaza (events spread over time), imperfections in a square meter of fabric, or typos per 100 lines of code (events spread over space).

Key Terms for Poisson and Related Distributions

Lambda
: The rate (per unit of time or space) at which events occur.

Poisson distribution
: The frequency distribution of the number of events in sampled units of time or space.

Exponential distribution
: The frequency distribution of the time or distance from one event to the next event.

Weibull distribution
: A generalized version of the exponential, in which the event rate is allowed to shift over time.

Poisson Distributions

From prior data we can estimate the average number of events per unit of time or space, but we might also want to know how different this might be from one unit of time/space to another. The Poisson distribution tells us the distribution of events per unit of time or space when we sample many such units. It is useful when addressing queuing questions like "How much capacity do we need to be 95% sure of fully processing the internet traffic that arrives on a server in any 5-second period?"

The key parameter in a Poisson distribution is λ, or lambda. This is the mean number of events that occurs in a specified interval of time or space. The variance for a Poisson distribution is also λ.

A common technique is to generate random numbers from a Poisson distribution as part of a queuing simulation. The `rpois` function in R does this, taking only two arguments—the quantity of random numbers sought, and lambda:

```
rpois(100, lambda = 2)
```

This code will generate 100 random numbers from a Poisson distribution with $\lambda = 2$. For example, if incoming customer service calls average 2 per minute, this code will simulate 100 minutes, returning the number of calls in each of those 100 minutes.

Exponential Distribution

Using the same parameter λ that we used in the Poisson distribution, we can also model the distribution of the time between events: time between visits to a website or between cars arriving at a toll plaza. It is also used in engineering to model time to failure, and in process management to model, for example, the time required per service call. The R code to generate random numbers from an exponential distribution takes two arguments, *n* (the quantity of numbers to be generated), and *rate*, the number of events per time period. For example:

```
rexp(n = 100, rate = .2)
```

This code would generate 100 random numbers from an exponential distribution where the mean number of events per time period is 2. So you could use it to simulate 100 intervals, in minutes, between service calls, where the average rate of incoming calls is 0.2 per minute.

A key assumption in any simulation study for either the Poisson or exponential distribution is that the rate, λ, remains constant over the period being considered. This is rarely reasonable in a global sense; for example, traffic on roads or data networks varies by time of day and day of week. However, the time periods, or areas of space, can usually be divided into segments that are sufficiently homogeneous so that analysis or simulation within those periods is valid.

Estimating the Failure Rate

In many applications, the event rate, λ, is known or can be estimated from prior data. However, for rare events, this is not necessarily so. Aircraft engine failure, for example, is sufficiently rare (thankfully) that, for a given engine type, there may be little data on which to base an estimate of time between failures. With no data at all, there is little basis on which to estimate an event rate. However, you can make some guesses: if no events have been seen after 20 hours, you can be pretty sure that the rate is not 1 per hour. Via simulation, or direct calculation of probabilities, you can assess different hypothetical event rates and estimate threshold values below which the rate is very unlikely to fall. If there is some data but not enough to provide a precise, reliable estimate of the rate, a goodness-of-fit test (see "Chi-Square Test" on page 111) can be applied to various rates to determine how well they fit the observed data.

Weibull Distribution

In many cases, the event rate does not remain constant over time. If the period over which it changes is much longer than the typical interval between events, there is no problem; you just subdivide the analysis into the segments where rates are relatively constant, as mentioned before. If, however, the event rate changes over the time of the interval, the exponential (or Poisson) distributions are no longer useful. This is likely to be the case in mechanical failure—the risk of failure increases as time goes by. The *Weibull* distribution is an extension of the exponential distribution, in which the event rate is allowed to change, as specified by a *shape parameter*, β. If $\beta > 1$, the probability of an event increases over time, if $\beta < 1$, it decreases. Because the Weibull distribution is used with time-to-failure analysis instead of event rate, the second parameter is expressed in terms of characteristic life, rather than in terms of the rate of events per interval. The symbol used is η, the Greek letter eta. It is also called the *scale* parameter.

With the Weibull, the estimation task now includes estimation of both parameters, β and η. Software is used to model the data and yield an estimate of the best-fitting Weibull distribution.

The R code to generate random numbers from a Weibull distribution takes three arguments, `n` (the quantity of numbers to be generated), `shape`, and `scale`. For example, the following code would generate 100 random numbers (lifetimes) from a Weibull distribution with shape of 1.5 and characteristic life of 5,000:

```
rweibull(100,1.5,5000)
```

Key Ideas

- For events that occur at a constant rate, the number of events per unit of time or space can be modeled as a Poisson distribution.
- In this scenario, you can also model the time or distance between one event and the next as an exponential distribution.
- A changing event rate over time (e.g., an increasing probability of device failure) can be modeled with the Weibull distribution.

Further Reading

- *Modern Engineering Statistics* by Tom Ryan (Wiley, 2007) has a chapter devoted to the probability distributions used in engineering applications.
- Read an engineering-based perspective on the use of the Weibull distribution (mainly from an engineering perspective) here and here.

Summary

In the era of big data, the principles of random sampling remain important when accurate estimates are needed. Random selection of data can reduce bias and yield a higher quality data set than would result from just using the conveniently available data. Knowledge of various sampling and data generating distributions allows us to quantify potential errors in an estimate that might be due to random variation. At the same time, the bootstrap (sampling with replacement from an observed data set) is an attractive "one size fits all" method to determine possible error in sample estimates.

CHAPTER 3
Statistical Experiments and Significance Testing

Design of experiments is a cornerstone of the practice of statistics, with applications in virtually all areas of research. The goal is to design an experiment in order to confirm or reject a hypothesis. Data scientists are faced with the need to conduct continual experiments, particularly regarding user interface and product marketing. This chapter reviews traditional experimental design and discusses some common challenges in data science. It also covers some oft-cited concepts in statistical inference and explains their meaning and relevance (or lack of relevance) to data science.

Whenever you see references to statistical significance, t-tests, or p-values, it is typically in the context of the classical statistical inference "pipeline" (see Figure 3-1). This process starts with a hypothesis ("drug A is better than the existing standard drug," "price A is more profitable than the existing price B"). An experiment (it might be an A/B test) is designed to test the hypothesis—designed in such a way that, hopefully, will deliver conclusive results. The data is collected and analyzed, and then a conclusion is drawn. The term *inference* reflects the intention to apply the experiment results, which involve a limited set of data, to a larger process or population.

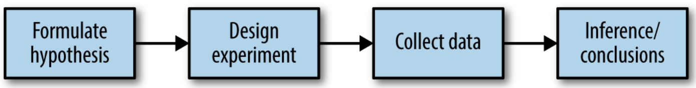


Figure 3-1. The classical statistical inference pipeline

A/B Testing

An A/B test is an experiment with two groups to establish which of two treatments, products, procedures, or the like is superior. Often one of the two treatments is the standard existing treatment, or no treatment. If a standard (or no) treatment is used, it is called the *control*. A typical hypothesis is that treatment is better than control.

Key Terms for A/B Testing

Treatment
: Something (drug, price, web headline) to which a subject is exposed.

Treatment group
: A group of subjects exposed to a specific treatment.

Control group
: A group of subjects exposed to no (or standard) treatment.

Randomization
: The process of randomly assigning subjects to treatments.

Subjects
: The items (web visitors, patients, etc.) that are exposed to treatments.

Test statistic
: The metric used to measure the effect of the treatment.

A/B tests are common in web design and marketing, since results are so readily measured. Some examples of A/B testing include:

- Testing two soil treatments to determine which produces better seed germination
- Testing two therapies to determine which suppresses cancer more effectively
- Testing two prices to determine which yields more net profit
- Testing two web headlines to determine which produces more clicks (Figure 3-2)
- Testing two web ads to determine which generates more conversions

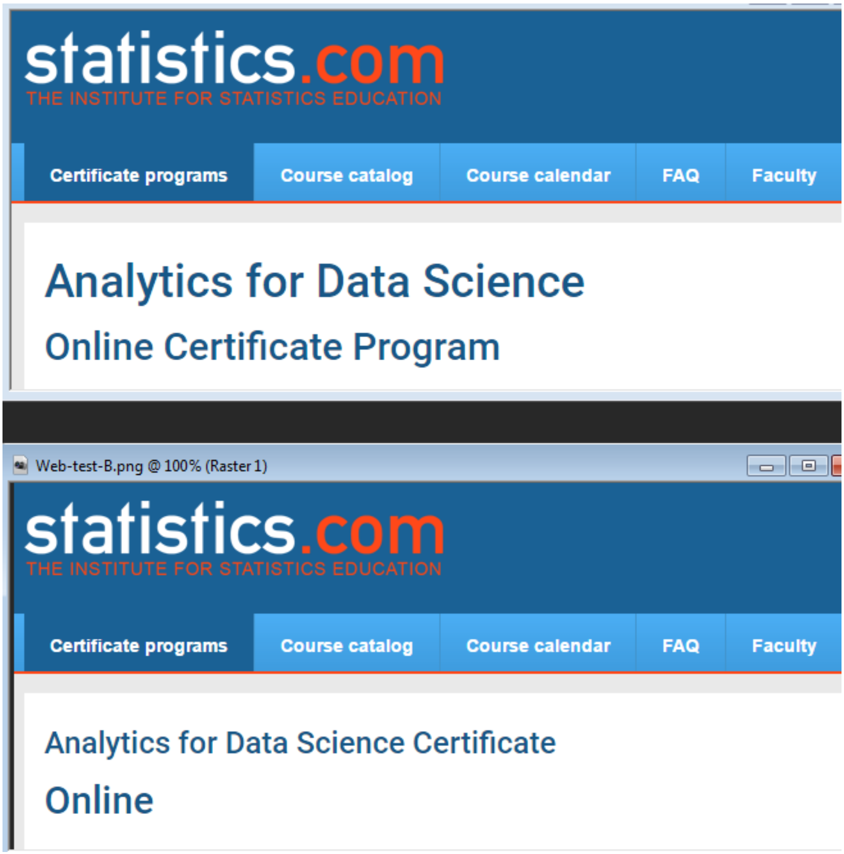


Figure 3-2. Marketers continually test one web presentation against another

A proper A/B test has *subjects* that can be assigned to one treatment or another. The subject might be a person, a plant seed, a web visitor; the key is that the subject is exposed to the treatment. Ideally, subjects are *randomized* (assigned randomly) to treatments. In this way, you know that any difference between the treatment groups is due to one of two things:

- The effect of the different treatments
- Luck of the draw in which subjects are assigned to which treatments (i.e., the random assignment may have resulted in the naturally better-performing subjects being concentrated in A or B)

You also need to pay attention to the *test statistic* or metric you use to compare group A to group B. Perhaps the most common metric in data science is a binary variable:

click or no-click, buy or don't buy, fraud or no fraud, and so on. Those results would be summed up in a 2×2 table. Table 3-1 is a 2×2 table for an actual price test.

Table 3-1. 2×2 table for ecommerce experiment results

Outcome	Price A	Price B
Conversion	200	182
No conversion	23,539	22,406

If the metric is a continuous variable (purchase amount, profit, etc.), or a count (e.g., days in hospital, pages visited) the result might be displayed differently. If one were interested not in conversion, but in revenue per page view, the results of the price test in Table 3-1 might look like this in typical default software output:

> Revenue/page-view with price A: mean = 3.87, SD = 51.10
>
> Revenue/page-view with price B: mean = 4.11, SD = 62.98

"SD" refers to the standard deviation of the values within each group.

> Just because statistical software—including R—generates output by default does not mean that all the output is useful or relevant. You can see that the preceding standard deviations are not that useful; on their face they suggest that numerous values might be negative, when negative revenue is not feasible. This data consists of a small set of relatively high values (page views with conversions) and a huge number of 0-values (page views with no conversion). It is difficult to sum up the variability of such data with a single number, though the mean absolute deviation from the mean (7.68 for A and 8.15 for B) is more reasonable than the standard deviation.

Why Have a Control Group?

Why not skip the control group and just run an experiment applying the treatment of interest to only one group, and compare the outcome to prior experience?

Without a control group, there is no assurance that "other things are equal" and that any difference is really due to the treatment (or to chance). When you have a control group, it is subject to the same conditions (except for the treatment of interest) as the treatment group. If you simply make a comparison to "baseline" or prior experience, other factors, besides the treatment, might differ.

Blinding in studies

> A *blind study* is one in which the subjects are unaware of whether they are getting treatment A or treatment B. Awareness of receiving a particular treatment can affect response. A *double blind* study is one in which the investigators and facilitators (e.g., doctors and nurses in a medical study) are unaware which subjects are getting which treatment. Blinding is not possible when the nature of the treatment is transparent—for example, cognitive therapy from a computer versus a psychologist.

The use of A/B testing in data science is typically in a web context. Treatments might be the design of a web page, the price of a product, the wording of a headline, or some other item. Some thought is required to preserve the principles of randomization. Typically the subject in the experiment is the web visitor, and the outcomes we are interested in measuring are clicks, purchases, visit duration, number of pages visited, whether a particular page is visited, and the like. In a standard A/B experiment, you need to decide on one metric ahead of time. Multiple behavior metrics might be collected and be of interest, but if the experiment is expected to lead to a decision between treatment A and treatment B, a single metric, or *test statistic*, needs to be established beforehand. Selecting a test statistic *after* the experiment is conducted opens the door to researcher bias.

Why Just A/B? Why Not C, D…?

A/B tests are popular in the marketing and ecommerce worlds, but are far from the only type of statistical experiment. Additional treatments can be included. Subjects might have repeated measurements taken. Pharmaceutical trials where subjects are scarce, expensive, and acquired over time are sometimes designed with multiple opportunities to stop the experiment and reach a conclusion.

Traditional statistical experimental designs focus on answering a static question about the efficacy of specified treatments. Data scientists are less interested in the question:

> Is the difference between price A and price B statistically significant?

than in the question:

> Which, out of multiple possible prices, is best?

For this, a relatively new type of experimental design is used: the *multi-arm bandit* (see “Multi-Arm Bandit Algorithm” on page 119).

Getting Permission

In scientific and medical research involving human subjects, it is typically necessary to get their permission, as well as obtain the approval of an institutional review board. Experiments in business that are done as a part of ongoing operations almost never do this. In most cases (e.g., pricing experiments, or experiments about which headline to show or which offer should be made), this practice is widely accepted. Facebook, however, ran afoul of this general acceptance in 2014 when it experimented with the emotional tone in users' newsfeeds. Facebook used sentiment analysis to classify newsfeed posts as positive or negative, then altered the positive/negative balance in what it showed users. Some randomly selected users experienced more positive posts, while others experienced more negative posts. Facebook found that the users who experienced a more positive newsfeed were more likely to post positively themselves, and vice versa. The magnitude of the effect was small, however, and Facebook faced much criticism for conducting the experiment without users' knowledge. Some users speculated that Facebook might have pushed some extremely depressed users over the edge, if they got the negative version of their feed.

Key Ideas

- Subjects are assigned to two (or more) groups that are treated exactly alike, except that the treatment under study differs from one to another.
- Ideally, subjects are assigned randomly to the groups.

For Further Reading

- Two-group comparisons (A/B tests) are a staple of traditional statistics, and just about any introductory statistics text will have extensive coverage of design principles and inference procedures. For a discussion that places A/B tests in more of a data science context and uses resampling, see *Introductory Statistics and Analytics: A Resampling Perspective* by Peter Bruce (Wiley, 2014).
- For web testing, the logistical aspects of testing can be just as challenging as the statistical ones. A good place to start is the Google Analytics help section on Experiments.
- Beware advice found in the ubiquitous guides to A/B testing that you see on the web, such as these words in one such guide: "Wait for about 1,000 total visitors and make sure you run the test for a week." Such general rules of thumb are not

statistically meaningful; see "Power and Sample Size" on page 122 for more detail.

Hypothesis Tests

Hypothesis tests, also called *significance tests*, are ubiquitous in the traditional statistical analysis of published research. Their purpose is to help you learn whether random chance might be responsible for an observed effect.

Key Terms

Null hypothesis
: The hypothesis that chance is to blame.

Alternative hypothesis
: Counterpoint to the null (what you hope to prove).

One-way test
: Hypothesis test that counts chance results only in one direction.

Two-way test
: Hypothesis test that counts chance results in two directions.

An A/B test (see "A/B Testing" on page 80) is typically constructed with a hypothesis in mind. For example, the hypothesis might be that price B produces higher profit. Why do we need a hypothesis? Why not just look at the outcome of the experiment and go with whichever treatment does better?

The answer lies in the tendency of the human mind to underestimate the scope of natural random behavior. One manifestation of this is the failure to anticipate extreme events, or so-called "black swans" (see "Long-Tailed Distributions" on page 67). Another manifestation is the tendency to misinterpret random events as having patterns of some significance. Statistical hypothesis testing was invented as a way to protect researchers from being fooled by random chance.

Misinterpreting Randomness

You can observe the human tendency to underestimate randomness in this experiment. Ask several friends to invent a series of 50 coin flips: have them write down a series of random Hs and Ts. Then ask them to actually flip a coin 50 times and write down the results. Have them put the real coin flip results in one pile, and the made-up results in another. It is easy to tell which results are real: the real ones will have longer runs of Hs or Ts. In a set of 50 *real* coin flips, it is not at all unusual to see five or six Hs or Ts in a row. However, when most of us are inventing random coin flips and we have gotten three or four Hs in a row, we tell ourselves that, for the series to look random, we had better switch to T.

The other side of this coin, so to speak, is that when we *do* see the real-world equivalent of six Hs in a row (e.g., when one headline outperforms another by 10%), we are inclined to attribute it to something real, not just chance.

In a properly designed A/B test, you collect data on treatments A and B in such a way that any observed difference between A and B must be due to either:

- Random chance in assignment of subjects
- A true difference between A and B

A statistical hypothesis test is further analysis of an A/B test, or any randomized experiment, to assess whether random chance is a reasonable explanation for the observed difference between groups A and B.

The Null Hypothesis

Hypothesis tests use the following logic: "Given the human tendency to react to unusual but random behavior and interpret it as something meaningful and real, in our experiments we will require proof that the difference between groups is more extreme than what chance might reasonably produce." This involves a baseline assumption that the treatments are equivalent, and any difference between the groups is due to chance. This baseline assumption is termed the *null hypothesis*. Our hope is then that we can, in fact, prove the null hypothesis *wrong*, and show that the outcomes for groups A and B are more different than what chance might produce.

One way to do this is via a resampling permutation procedure, in which we shuffle together the results from groups A and B and then repeatedly deal out the data in groups of similar sizes, then observe how often we get a difference as extreme as the observed difference. See "Resampling" on page 88 for more detail.

Alternative Hypothesis

Hypothesis tests by their nature involve not just a null hypothesis, but also an offsetting alternative hypothesis. Here are some examples:

- Null = "no difference between the means of group A and group B," alternative = "A is different from B" (could be bigger or smaller)
- Null = "A ≤ B," alternative = "B > A"

- Null = "B is not X% greater than A," alternative = "B is X% greater than A"

Taken together, the null and alternative hypotheses must account for all possibilities. The nature of the null hypothesis determines the structure of the hypothesis test.

One-Way, Two-Way Hypothesis Test

Often, in an A/B test, you are testing a new option (say B), against an established default option (A) and the presumption is that you will stick with the default option unless the new option proves itself definitively better. In such a case, you want a hypothesis test to protect you from being fooled by chance in the direction favoring B. You don't care about being fooled by chance in the other direction, because you would be sticking with A unless B proves definitively better. So you want a *directional* alternative hypothesis (B is better than A). In such a case, you use a *one-way* (or one-tail) hypothesis test. This means that extreme chance results in only one direction direction count toward the p-value.

If you want a hypothesis test to protect you from being fooled by chance in either direction, the alternative hypothesis is *bidirectional* (A is different from B; could be bigger or smaller). In such a case, you use a *two-way* (or two-tail) hypothesis. This means that extreme chance results in either direction count toward the p-value.

A one-tail hypothesis test often fits the nature of A/B decision making, in which a decision is required and one option is typically assigned "default" status unless the other proves better. Software, however, including R, typically provides a two-tail test in its default output, and many statisticians opt for the more conservative two-tail test just to avoid argument. One-tail versus two-tail is a confusing subject, and not that relevant for data science, where the precision of p-value calculations is not terribly important.

Key Ideas

- A *null hypothesis* is a logical construct embodying the notion that nothing special has happened, and any effect you observe is due to random chance.
- The *hypothesis test* assumes that the null hypothesis is true, creates a "null model" (a probability model), and tests whether the effect you observe is a reasonable outcome of that model.

Further Reading

- *The Drunkard's Walk* by Leonard Mlodinow (Vintage Books, 2008) is a readable survey of the ways in which "randomness rules our lives."
- David Freedman, Robert Pisani, and Roger Purves's classic statistics text *Statistics*, 4th ed. (W. W. Norton, 2007) has excellent nonmathematical treatments of most statistics topics, including hypothesis testing.
- *Introductory Statistics and Analytics: A Resampling Perspective* by Peter Bruce (Wiley, 2014) develops hypothesis testing concepts using resampling.

Resampling

Resampling in statistics means to repeatedly sample values from observed data, with a general goal of assessing random variability in a statistic. It can also be used to assess and improve the accuracy of some machine-learning models (e.g., the predictions from decision tree models built on multiple bootstrapped data sets can be averaged in a process known as *bagging*: see "Bagging and the Random Forest" on page 228).

There are two main types of resampling procedures: the *bootstrap* and *permutation* tests. The bootstrap is used to assess the reliability of an estimate; it was discussed in the previous chapter (see "The Bootstrap" on page 57). Permutation tests are used to test hypotheses, typically involving two or more groups, and we discuss those in this section.

Key Terms

Permutation test
: The procedure of combining two or more samples together, and randomly (or exhaustively) reallocating the observations to resamples.

  Synonyms
  : Randomization test, random permutation test, exact test.

With or without replacement
: In sampling, whether or not an item is returned to the sample before the next draw.

Permutation Test

In a *permutation* procedure, two or more samples are involved, typically the groups in an A/B or other hypothesis test. *Permute* means to change the order of a set of values. The first step in a *permutation test* of a hypothesis is to combine the results from groups A and B (and, if used, C, D, …) together. This is the logical embodiment of the null hypothesis that the treatments to which the groups were exposed do not differ. We then test that hypothesis by randomly drawing groups from this combined set, and seeing how much they differ from one another. The permutation procedure is as follows:

1. Combine the results from the different groups in a single data set.
2. Shuffle the combined data, then randomly draw (without replacing) a resample of the same size as group A.
3. From the remaining data, randomly draw (without replacing) a resample of the same size as group B.
4. Do the same for groups C, D, and so on.
5. Whatever statistic or estimate was calculated for the original samples (e.g., difference in group proportions), calculate it now for the resamples, and record; this constitutes one permutation iteration.
6. Repeat the previous steps *R* times to yield a permutation distribution of the test statistic.

Now go back to the observed difference between groups and compare it to the set of permuted differences. If the observed difference lies well within the set of permuted differences, then we have not proven anything—the observed difference is within the range of what chance might produce. However, if the observed difference lies outside most of the permutation distribution, then we conclude that chance is *not* responsible. In technical terms, the difference is *statistically significant*. (See “Statistical Significance and P-Values” on page 93.)

Example: Web Stickiness

A company selling a relatively high-value service wants to test which of two web presentations does a better selling job. Due to the high value of the service being sold, sales are infrequent and the sales cycle is lengthy; it would take too long to accumulate enough sales to know which presentation is superior. So the company decides to measure the results with a proxy variable, using the detailed interior page that describes the service.

> A *proxy* variable is one that stands in for the true variable of interest, which may be unavailable, too costly, or too time-consuming to measure. In climate research, for example, the oxygen content of ancient ice cores is used as a proxy for temperature. It is useful to have at least *some* data on the true variable of interest, so the strength of its association with the proxy can be assessed.

One potential proxy variable for our company is the number of clicks on the detailed landing page. A better one is how long people spend on the page. It is reasonable to think that a web presentation (page) that holds people's attention longer will lead to more sales. Hence, our metric is average session time, comparing page A to page B.

Due to the fact that this is an interior, special-purpose page, it does not receive a huge number of visitors. Also note that Google Analytics, which is how we measure session time, cannot measure session time for the last session a person visits. Instead of deleting that session from the data, though, GA records it as a zero, so the data requires additional processing to remove those sessions. The result is a total of 36 sessions for the two different presentations, 21 for page A and 15 for page B. Using `ggplot`, we can visually compare the session times using side-by-side boxplots:

```
ggplot(session_times, aes(x=Page, y=Time)) +
  geom_boxplot()
```

The boxplot, shown in Figure 3-3, indicates that page B leads to longer sessions than page A. The means for each group can be computed as follows:

```
mean_a <- mean(session_times[session_times['Page']=='Page A', 'Time'])
mean_b <- mean(session_times[session_times['Page']=='Page B', 'Time'])
mean_b - mean_a
[1] 21.4
```

Page B has session times greater, on average, by 21.4 seconds versus page A. The question is whether this difference is within the range of what random chance might produce, or, alternatively, is statistically significant. One way to answer this is to apply a permutation test—combine all the session times together, then repeatedly shuffle and divide them into groups of 21 (recall that n = 21 for page A) and 15 (n = 15 for B).

To apply a permutation test, we need a function to randomly assign the 36 session times to a group of 21 (page A) and a group of 15 (page B):

```
perm_fun <- function(x, n1, n2)
{
  n <- n1 + n2
  idx_b <- sample(1:n, n1)
  idx_a <- setdiff(1:n, idx_b)
  mean_diff <- mean(x[idx_b]) - mean(x[idx_a])
  return(mean_diff)
}
```

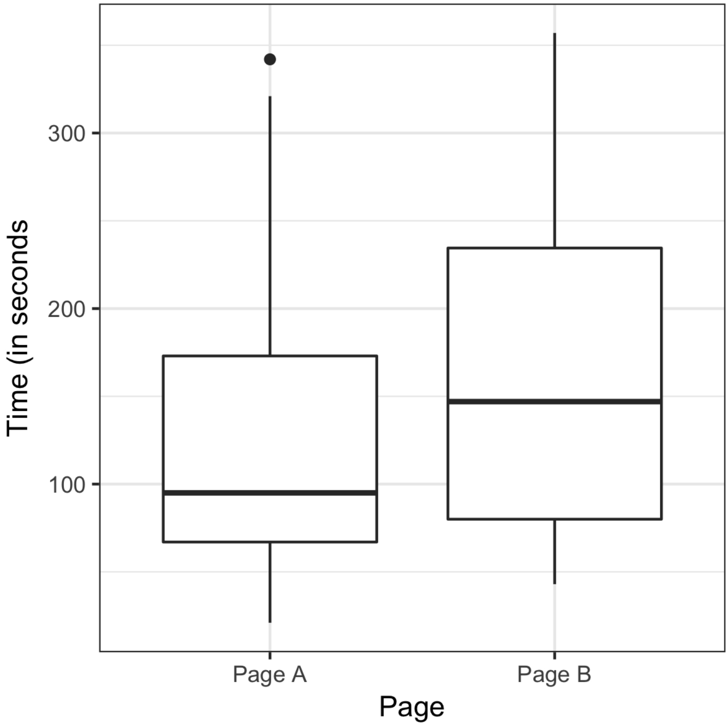


Figure 3-3. Session times for web pages A and B

This function works by sampling without replacement n2 indices and assigning them to the B group; the remaining n1 indices are assigned to group A. The difference between the two means is returned. Calling this function *R* = 1,000 times and specifying n2 = 15 and n1 = 21 leads to a distribution of differences in the session times that can be plotted as a histogram.

```
perm_diffs <- rep(0, 1000)
for(i in 1:1000)
  perm_diffs[i] = perm_fun(session_times[,'Time'], 21, 15)
hist(perm_diffs, xlab='Session time differences (in seconds)')
abline(v = mean_b - mean_a)
```

The histogram, shown in Figure 3-4 shows that mean difference of random permutations often exceeds the observed difference in session times (the vertical line). This suggests that the oberved difference in session time between page A and page B is well within the range of chance variation, thus is not statistically significant.

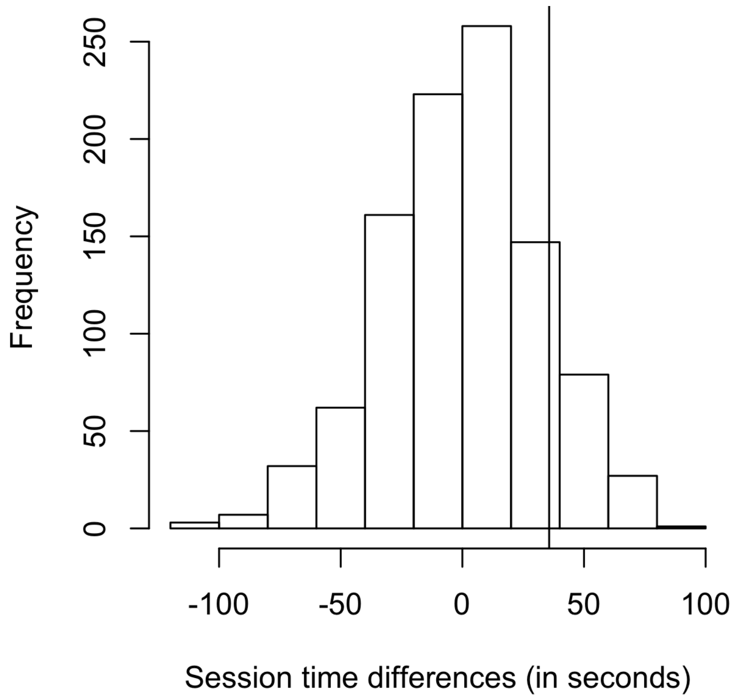


Figure 3-4. Frequency distribution for session time differences between pages A and B

Exhaustive and Bootstrap Permutation Test

In addition to the preceding random shuffling procedure, also called a *random permutation test* or a *randomization test*, there are two variants of the permutation test:

- An *exhaustive permutation test*
- A *bootstrap permutation test*

In an exhaustive permutation test, instead of just randomly shuffling and dividing the data, we actually figure out all the possible ways it could be divided. This is practical only for relatively small sample sizes. With a large number of repeated shufflings, the random permutation test results approximate those of the exhaustive permutation test, and approach them in the limit. Exhaustive permutation tests are also sometimes called *exact tests*, due to their statistical property of guaranteeing that the null model will not test as "significant" more than the alpha level of the test (see "Statistical Significance and P-Values" on page 93).

In a bootstrap permutation test, the draws outlined in steps 2 and 3 of the random permutation test are made *with replacement* instead of without replacement. In this way the resampling procedure models not just the random element in the assignment of treatment to subject, but also the random element in the selection of subjects from a population. Both procedures are encountered in statistics, and the distinction between them is somewhat convoluted and not of consequence in the practice of data science.

Permutation Tests: The Bottom Line for Data Science

Permutation tests are useful heuristic procedures for exploring the role of random variation. They are relatively easy to code, interpret and explain, and they offer a useful detour around the formalism and "false determinism" of formula-based statistics.

One virtue of resampling, in contrast to formula approaches, is that it comes much closer to a "one size fits all" approach to inference. Data can be numeric or binary. Sample sizes can be the same or different. Assumptions about normally-distributed data are not needed.

Key Ideas

- In a permutation test, multiple samples are combined, then shuffled.
- The shuffled values are then divided into resamples, and the statistic of interest is calculated.
- This process is then repeated, and the resampled statistic is tabulated.
- Comparing the observed value of the statistic to the resampled distribution allows you to judge whether an observed difference between samples might occur by chance.

For Further Reading

- *Randomization Tests*, 4th ed., by Eugene Edgington and Patrick Onghena (Chapman Hall, 2007), but don't get too drawn into the thicket of nonrandom sampling.
- *Introductory Statistics and Analytics: A Resampling Perspective* by Peter Bruce (Wiley, 2015).

Statistical Significance and P-Values

Statistical significance is how statisticians measure whether an experiment (or even a study of existing data) yields a result more extreme than what chance might produce.

If the result is beyond the realm of chance variation, it is said to be statistically significant.

> ### Key Terms
>
> ***P-value***
> : Given a chance model that embodies the null hypothesis, the p-value is the probability of obtaining results as unusual or extreme as the observed results.
>
> ***Alpha***
> : The probability threshold of "unusualness" that chance results must surpass, for actual outcomes to be deemed statistically significant.
>
> ***Type 1 error***
> : Mistakenly concluding an effect is real (when it is due to chance).
>
> ***Type 2 error***
> : Mistakenly concluding an effect is due to chance (when it is real).

Consider in Table 3-2 the results of the web test shown earlier.

Table 3-2. 2×2 table for ecommerce experiment results

Outcome	Price A	Price B
Conversion	200	182
No conversion	23539	22406

Price A converts almost 5% better than price B (0.8425% versus 0.8057%—a difference of 0.0368 percentage points), big enough to be meaningful in a high-volume business. We have over 45,000 data points here, and it is tempting to consider this as "big data," not requiring tests of statistical significance (needed mainly to account for sampling variability in small samples). However, the conversion rates are so low (less than 1%) that the actual meaningful values—the conversions—are only in the 100s, and the sample size needed is really determined by these conversions. We can test whether the difference in conversions between prices A and B is within the range of chance variation, using a resampling procedure. By "chance variation," we mean the random variation produced by a probability model that embodies the null hypothesis that there is no difference between the rates (see "The Null Hypothesis" on page 86). The following permutation procedure asks "if the two prices share the same conversion rate, could chance variation produce a difference as big as 5%?"

1. Create an urn with all sample results: this represents the supposed shared conversion rate of 382 ones and 45,945 zeros = 0.008246 = 0.8246%.

2. Shuffle and draw out a resample of size 23,739 (same *n* as price A), and record how many 1s.
3. Record the number of 1s in the remaining 22,588 (same *n* as price B).
4. Record the difference in proportion 1s.
5. Repeat steps 2–4.
6. How often was the difference >= 0.0368?

Reusing the function `perm_fun` defined in "Example: Web Stickiness" on page 89, we can create a histogram of randomly permuted differences in conversion rate:

```
obs_pct_diff <- 100*(200/23739 - 182/22588)
conversion <- c(rep(0, 45945), rep(1, 382))
perm_diffs <- rep(0, 1000)
for(i in 1:1000)
  perm_diffs[i] = 100*perm_fun(conversion, 23739, 22588 )
hist(perm_diffs, xlab='Session time differences (in seconds)')
abline(v = obs_pct_diff)
```

See the histogram of 1,000 resampled results in Figure 3-5: as it happens, in this case the observed difference of 0.0368% is well within the range of chance variation.

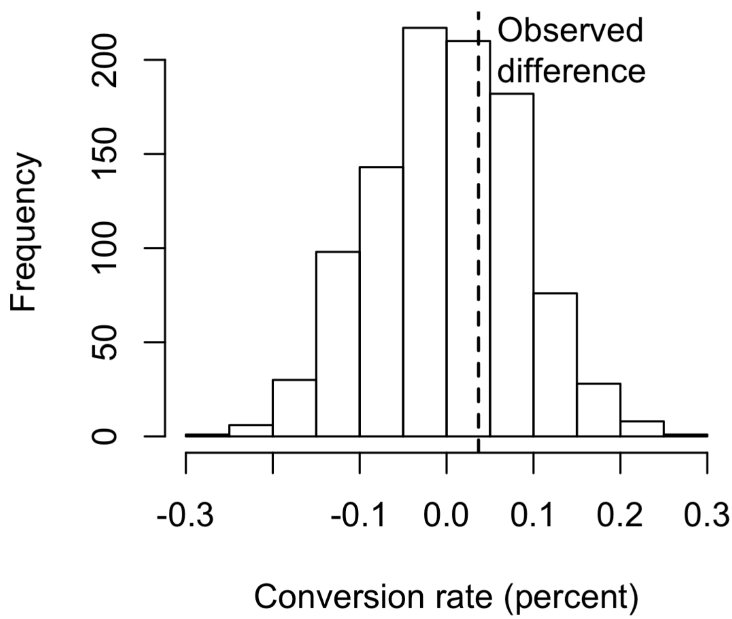


Figure 3-5. Frequency distribution for the difference in conversion rates between pages A and B

P-Value

Simply looking at the graph is not a very precise way to measure statistical significance, so of more interest is the *p-value*. This is the frequency with which the chance model produces a result more extreme than the observed result. We can estimate a p-value from our permutation test by taking the proportion of times that the permutation test produces a difference equal to or greater than the observed difference:

```
mean(perm_diffs > obs_pct_diff)
[1] 0.308
```

The p-value is 0.308, which means that we would expect to achieve a result as extreme as this, or more extreme, by random chance over 30% of the time.

In this case, we didn't need to use a permutation test to get a p-value. Since we have a binomial distribution, we can approximate the p-value using the normal distribution. In R code, we do this using the function `prop.test`:

```
> prop.test(x=c(200,182), n=c(23739,22588), alternative="greater")

	2-sample test for equality of proportions with continuity correction

data:  c(200, 182) out of c(23739, 22588)
X-squared = 0.14893, df = 1, p-value = 0.3498
alternative hypothesis: greater
95 percent confidence interval:
 -0.001057439  1.000000000
sample estimates:
     prop 1      prop 2
0.008424955 0.008057376
```

The argument `x` is the number of successes for each group and the argument `n` is the number of trials. The normal approximation yields a p-value of 0.3498, which is close to the p-value obtained from the permutation test.

Alpha

Statisticians frown on the practice of leaving it to the researcher's discretion to determine whether a result is "too unusual" to happen by chance. Rather, a threshold is specified in advance, as in "more extreme than 5% of the chance (null hypothesis) results"; this threshold is known as alpha. Typical alpha levels are 5% and 1%. Any chosen level is an arbitrary decision—there is nothing about the process that will guarantee correct decisions x% of the time. This is because the probability question being answered is *not* "what is the probability that this happened by chance?" but rather "given a chance model, what is the probability of a result this extreme?" We then deduce backward about the appropriateness of the chance model, but that judgment does not carry a probability. This point has been the subject of much confusion.

Value of the p-value

Considerable controversy has surrounded the use of the p-value in recent years. One psychology journal has gone so far as to "ban" the use of p-values in submitted papers on the grounds that publication decisions based solely on the p-value were resulting in the publication of poor research. Too many researchers, only dimly aware of what a p-value really means, root around in the data and among different possible hypotheses to test, until they find a combination that yields a significant p-value and, hence, a paper suitable for publication.

The real problem is that people want more meaning from the p-value than it contains. Here's what we would *like* the p-value to convey:

> The probability that the result is due to chance.

We hope for a low value, so we can conclude that we've proved something. This is how many journal editors were interpreting the p-value. But here's what the p-value *actually* represents:

> The probability that, *given a chance model*, results as extreme as the observed results could occur.

The difference is subtle, but real. A significant p-value does not carry you quite as far along the road to "proof" as it seems to promise. The logical foundation for the conclusion "statistically significant" is somewhat weaker when the real meaning of the p-value is understood.

In March 2016, the American Statistical Association, after much internal deliberation, revealed the extent of misunderstanding about p-values when it issued a cautionary statement regarding their use.

The ASA statement stressed six principles for researchers and journal editors:

1. P-values can indicate how incompatible the data are with a specified statistical model.
2. P-values do not measure the probability that the studied hypothesis is true, or the probability that the data were produced by random chance alone.
3. Scientific conclusions and business or policy decisions should not be based only on whether a p-value passes a specific threshold.
4. Proper inference requires full reporting and transparency.
5. A p-value, or statistical significance, does not measure the size of an effect or the importance of a result.
6. By itself, a p-value does not provide a good measure of evidence regarding a model or hypothesis.

Type 1 and Type 2 Errors

In assessing statistical significance, two types of error are possible:

- Type 1 error, in which you mistakenly conclude an effect is real, when it is really just due to chance
- Type 2 error, in which you mistakenly conclude that an effect is not real (i.e., due to chance), when it really is real

Actually, a Type 2 error is not so much an error as a judgment that the sample size is too small to detect the effect. When a p-value falls short of statistical significance (e.g., it exceeds 5%), what we are really saying is "effect not proven." It could be that a larger sample would yield a smaller p-value.

The basic function of significance tests (also called *hypothesis tests*) is to protect against being fooled by random chance; thus they are typically structured to minimize Type 1 errors.

Data Science and P-Values

The work that data scientists do is typically not destined for publication in scientific journals, so the debate over the value of a p-value is somewhat academic. For a data scientist, a p-value is a useful metric in situations where you want to know whether a model result that appears interesting and useful is within the range of normal chance variability. As a decision tool in an experiment, a p-value should not be considered controlling, but merely another point of information bearing on a decision. For example, p-values are sometimes used as intermediate inputs in some statistical or machine learning models—a feature night be included in or excluded from a model depending on its p-value.

Key Ideas

- Significance tests are used to determine whether an observed effect is within the range of chance variation for a null hypothesis model.
- The p-value is the probability that results as extreme as the observed results might occur, given a null hypothesis model.
- The alpha value is the threshold of "unusualness" in a null hypothesis chance model.
- Significance testing has been much more relevant for formal reporting of research than for data science (but has been fading recently, even for the former).

Further Reading

- Stephen Stigler, "Fisher and the 5% Level," *Chance* vol. 21, no. 4 (2008): 12. This article is a short commentary on Ronald Fisher's 1925 book *Statistical Methods for Research Workers*, and his emphasis on the 5% level of significance.
- See also "Hypothesis Tests" on page 85 and the further reading mentioned there.

t-Tests

There are numerous types of significance tests, depending on whether the data comprises count data or measured data, how many samples there are, and what's being measured. A very common one is the *t-test*, named after Student's t-distribution, originally developed by W. S. Gossett to approximate the distribution of a single sample mean (see "Student's t-Distribution" on page 69).

Key Terms

Test statistic
A metric for the difference or effect of interest.

t-statistic
A standardized version of the test statistic.

t-distribution
A reference distribution (in this case derived from the null hypothesis), to which the observed t-statistic can be compared.

All significance tests require that you specify a *test statistic* to measure the effect you are interested in, and help you determine whether that observed effect lies within the range of normal chance variation. In a resampling test (see the discussion of permutation in "Permutation Test" on page 88), the scale of the data does not matter. You create the reference (null hypothesis) distribution from the data itself, and use the test statistic as is.

In the 1920s and 30s, when statistical hypothesis testing was being developed, it was not feasible to randomly shuffle data thousands of times to do a resampling test. Statisticians found that a good approximation to the permutation (shuffled) distribution was the t-test, based on Gossett's t-distribution. It is used for the very common two-sample comparison—A/B test—in which the data is numeric. But in order for the t-distribution to be used without regard to scale, a standardized form of the test statistic must be used.

A classic statistics text would at this stage show various formulas that incorporate Gossett's distribution and demonstrate how to standardize your data to compare it to the standard t-distribution. These formulas are not shown here because all statistical software, as well as R and Python, include commands that embody the formula. In R, the function is `t.test`:

```
> t.test(Time ~ Page, data=session_times, alternative='less' )

	Welch Two Sample t-test

data:  Time by Page
t = -1.0983, df = 27.693, p-value = 0.1408
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf 19.59674
sample estimates:
mean in group Page A mean in group Page B
            126.3333             162.0000
```

The alternative hypothesis is that the session time mean for page A is less than for page B. This is fairly close to the permutation test p-value of 0.124 (see "Example: Web Stickiness" on page 89).

In a resampling mode, we structure the solution to reflect the observed data and the hypothesis to be tested, not worrying about whether the data is numeric or binary, sample sizes are balanced or not, sample variances, or a variety of other factors. In the formula world, many variations present themselves, and they can be bewildering. Statisticians need to navigate that world and learn its map, but data scientists do not—they are typically not in the business of sweating the details of hypothesis tests and confidence intervals the way a researcher preparing a paper for presentation might.

Key Ideas

- Before the advent of computers, resampling tests were not practical and statisticians used standard reference distributions.
- A test statistic could then be standardized and compared to the reference distribution.
- One such widely used standardized statistic is the t-statistic.

Further Reading

- Any introductory statistics text will have illustrations of the t-statistic and its uses; two good ones are *Statistics*, 4th ed., by David Freedman, Robert Pisani, and Roger Purves (W. W. Norton, 2007) and *The Basic Practice of Statistics* by David S. Moore (Palgrave Macmillan, 2010).
- For a treatment of both the t-test and resampling procedures in parallel, see *Introductory Statistics and Analytics: A Resampling Perspective* by Peter Bruce (Wiley, 2014) or *Statistics* by Robin Lock and four other Lock family members (Wiley, 2012).

Multiple Testing

As we've mentioned previously, there is a saying in statistics: "torture the data long enough, and it will confess." This means that if you look at the data through enough different perspectives, and ask enough questions, you can almost invariably find a statistically significant effect.

Key Terms

Type 1 error
Mistakenly concluding that an effect is statistically significant.

False discovery rate
Across multiple tests, the rate of making a Type 1 error.

Adjustment of p-values
Accounting for doing multiple tests on the same data.

Overfitting
Fitting the noise.

For example, if you have 20 predictor variables and one outcome variable, all *randomly* generated, the odds are pretty good that at least one predictor will (falsely) turn out to be statistically significant if you do a series of 20 significance tests at the alpha = 0.05 level. As previously discussed, this is called a *Type 1 error*. You can calculate this probability by first finding the probability that all will *correctly* test nonsignificant at the 0.05 level. The probability that *one* will correctly test nonsignificant is 0.95, so the probability that all 20 will correctly test nonsignificant is 0.95 × 0.95 × 0.95 … or

$0.95^{20} = 0.36$.[1] The probability that at least one predictor will (falsely) test significant is the flip side of this probability, or 1 – (*probability that all will be nonsignificant*) = 0.64.

This issue is related to the problem of overfitting in data mining, or "fitting the model to the noise." The more variables you add, or the more models you run, the greater the probability that something will emerge as "significant" just by chance.

In supervised learning tasks, a holdout set where models are assessed on data that the model has not seen before mitigates this risk. In statistical and machine learning tasks not involving a labeled holdout set, the risk of reaching conclusions based on statistical noise persists.

In statistics, there are some procedures intended to deal with this problem in very specific circumstances. For example, if you are comparing results across multiple treatment groups you might ask multiple questions. So, for treatments A–C, you might ask:

- Is A different from B?
- Is B different from C?
- Is A different from C?

Or, in a clinical trial, you might want to look at results from a therapy at multiple stages. In each case, you are asking multiple questions, and with each question, you are increasing the chance of being fooled by chance. Adjustment procedures in statistics can compensate for this by setting the bar for statistical significance more stringently than it would be set for a single hypothesis test. These adjustment procedures typically involve "dividing up the alpha" according to the number of tests. This results in a smaller alpha (i.e., a more stringent bar for statistical significance) for each test. One such procedure, the Bonferroni adjustment, simply divides the alpha by the number of observations n.

However, the problem of multiple comparisons goes beyond these highly structured cases and is related to the phenomenon of repeated data "dredging" that gives rise to the saying about torturing the data. Put another way, given sufficiently complex data, if you haven't found something interesting, you simply haven't looked long and hard enough. More data is available now than ever before, and the number of journal articles published nearly doubled between 2002 and 2010. This gives rise to lots of

1 The multiplication rule states that the probability of n independent events all happening is the product of the individual probabilities. For example, if you and I each flip a coin once, the probability that your coin and my coin will both land heads is $0.5 \times 0.5 = 0.25$.

opportunities to find something interesting in the data, including multiplicity issues such as:

- Checking for multiple pairwise differences across groups
- Looking at multiple subgroup results ("we found no significant treatment effect overall, but we did find an effect for unmarried women younger than 30")
- Trying lots of statistical models
- Including lots of variables in models
- Asking a number of different questions (i.e., different possible outcomes)

False Discovery Rate

The term *false discovery rate* was originally used to describe the rate at which a given set of hypothesis tests would falsely identify a significant effect. It became particularly useful with the advent of genomic research, in which massive numbers of statistical tests might be conducted as part of a gene sequencing project. In these cases, the term applies to the testing protocol, and a single false "discovery" refers to the outcome of a hypothesis test (e.g., between two samples). Researchers sought to set the parameters of the testing process to control the false discovery rate at a specified level. The term has also been used in the data mining community in a classification context, in which a false discovery is a mislabeling of a single record—in particular the mislabeling of 0s as 1s (see Chapter 5 and "The Rare Class Problem" on page 196).

For a variety of reasons, including especially this general issue of "multiplicity," more research does not necessarily mean better research. For example, the pharmaceutical company Bayer found in 2011 that when it tried to replicate 67 scientific studies, it could fully replicate only 14 of them. Nearly two-thirds could not be replicated at all.

In any case, the adjustment procedures for highly defined and structured statistical tests are too specific and inflexible to be of general use to data scientists. The bottom line for data scientists on multiplicity is:

- For predictive modeling, the risk of getting an illusory model whose apparent efficacy is largely a product of random chance is mitigated by cross-validation (see "Cross-Validation" on page 138), and use of a holdout sample.
- For other procedures without a labeled holdout set to check the model, you must rely on:

— Awareness that the more you query and manipulate the data, the greater the role that chance might play; and

— Resampling and simulation heuristics to provide random chance benchmarks against which observed results can be compared.

Key Ideas

- Multiplicity in a research study or data mining project (multiple comparisons, many variables, many models, etc.) increases the risk of concluding that something is significant just by chance.
- For situations involving multiple statistical comparisons (i.e., multiple tests of significance) there are statistical adjustment procedures.
- In a data mining situation, use of a holdout sample with labeled outcome variables can help avoid misleading results.

Further Reading

1. For a short exposition of one procedure (Dunnett's) to adjust for multiple comparisons, see David Lane's online statistics text.
2. Megan Goldman offers a slightly longer treatment of the Bonferroni adjustment procedure.
3. For an in-depth treatment of more flexible statistical procedures to adjust p-values, see *Resampling-Based Multiple Testing* by Peter Westfall and Stanley Young (Wiley, 1993).
4. For a discussion of data partitioning and the use of holdout samples in predictive modeling, see *Data Mining for Business Analytics*, Chapter 2, by Galit Shmueli, Peter Bruce, and Nitin Patel (Wiley, 2016).

Degrees of Freedom

In the documentation and settings to many statistical tests, you will see reference to "degrees of freedom." The concept is applied to statistics calculated from sample data, and refers to the number of values free to vary. For example, if you know the mean for a sample of 10 values, and you also know 9 of the values, you also know the 10th value. Only 9 are free to vary.

Key Terms

***n* or sample size**
: The number of observations (also called *rows* or *records*) in the data.

d.f.
: Degrees of freedom.

The number of degrees of freedom is an input to many statistical tests. For example, degrees of freedom is the name given to the $n - 1$ denominator seen in the calculations for variance and standard deviation. Why does it matter? When you use a sample to estimate the variance for a population, you will end up with an estimate that is slightly biased downward if you use n in the denominator. If you use $n - 1$ in the denominator, the estimate will be free of that bias.

A large share of a traditional statistics course or text is consumed by various standard tests of hypotheses (t-test, F-test, etc.). When sample statistics are standardized for use in traditional statistical formulas, degrees of freedom is part of the standardization calculation to ensure that your standardized data matches the appropriate reference distribution (t-distribution, F-distribution, etc.).

Is it important for data science? Not really, at least in the context of significance testing. For one thing, formal statistical tests are used only sparingly in data science. For another, the data size is usually large enough that it rarely makes a real difference for a data scientist whether, for example, the denominator has n or $n - 1$.

There is one context, though, in which it is relevant: the use of factored variables in regression (including logistic regression). Regression algorithms choke if exactly redundant predictor variables are present. This most commonly occurs when factoring categorical variables into binary indicators (dummies). Consider day of week. Although there are seven days of the week, there are only six degrees of freedom in specifying day of week. For example, once you know that day of week is not Monday through Saturday, you know it must be Sunday. Inclusion of the Mon–Sat indicators thus means that *also* including Sunday would cause the regression to fail, due to a *multicollinearity* error.

Key Ideas

- The number of degrees of freedom (d.f.) forms part of the calculation to standardize test statistics so they can be compared to reference distributions (t-distribution, F-distribution, etc.).
- The concept of degrees of freedom lies behind the factoring of categorical variables into $n - 1$ indicator or dummy variables when doing a regression (to avoid multicollinearity).

Further Reading

There are several web tutorials on degrees of freedom.

ANOVA

Suppose that, instead of an A/B test, we had a comparison of multiple groups, say A-B-C-D, each with numeric data. The statistical procedure that tests for a statistically significant difference among the groups is called *analysis of variance*, or *ANOVA*.

Key Terms for ANOVA

Pairwise comparison
A hypothesis test (e.g., of means) between two groups among multiple groups.

Omnibus test
A single hypothesis test of the overall variance among multiple group means.

Decomposition of variance
Separation of components. contributing to an individual value (e.g., from the overall average, from a treatment mean, and from a residual error).

F-statistic
A standardized statistic that measures the extent to which differences among group means exceeds what might be expected in a chance model.

SS
"Sum of squares," referring to deviations from some average value.

Table 3-3 shows the stickiness of four web pages, in numbers of seconds spent on the page. The four pages are randomly switched out so that each web visitor receives one at random. There are a total of five visitors for each page, and, in Table 3-3, each column is an independent set of data. The first viewer for page 1 has no connection to the first viewer for page 2. Note that in a web test like this, we cannot fully implement the classic randomized sampling design in which each visitor is selected at random from some huge population. We must take the visitors as they come. Visitors may systematically differ depending on time of day, time of week, season of the year, conditions of their internet, what device they are using, and so on. These factors should be considered as potential bias when the experiment results are reviewed.

Table 3-3. Stickiness (in seconds) for four web pages

	Page 1	Page 2	Page 3	Page 4
	164	178	175	155
	172	191	193	166
	177	182	171	164
	156	185	163	170
	195	177	176	168
Average	172	185	176	162
Grand average				173.75

Now, we have a conundrum (see Figure 3-6). When we were comparing just two groups, it was a simple matter; we merely looked at the difference between the means of each group. With four means, there are six possible comparisons between groups:

- Page 1 compared to page 2
- Page 1 compared to page 3
- Page 1 compared to page 4
- Page 2 compared to page 3
- Page 2 compared to page 4
- Page 3 compared to page 4

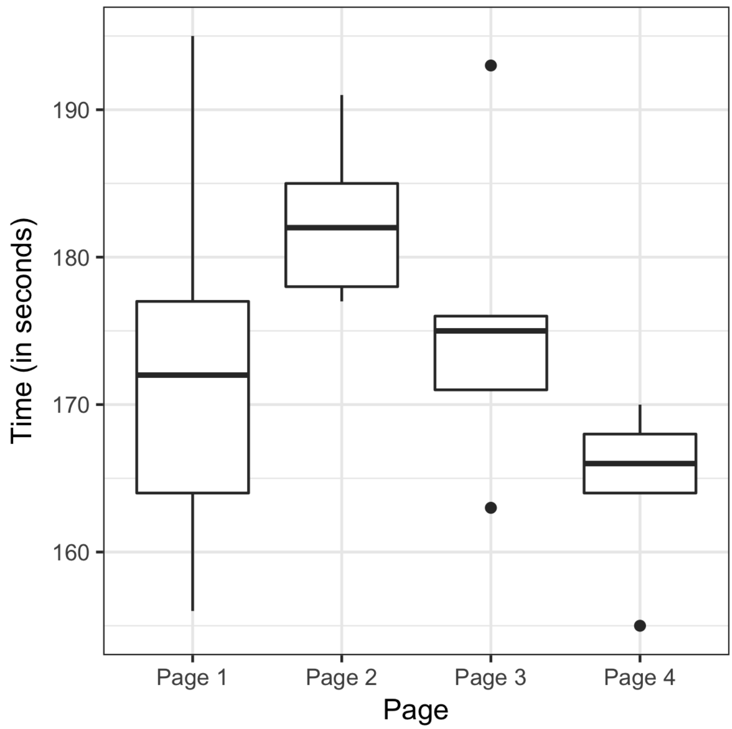


Figure 3-6. Boxplots of the four groups show considerable differences among them

The more such *pairwise* comparisons we make, the greater the potential for being fooled by random chance (see "Multiple Testing" on page 101). Instead of worrying about all the different comparisons between individual pages we could possibly make, we can do a single overall *omnibus* test that addresses the question, "Could all the pages have the same underlying stickiness, and the differences among them be due to the random way in which a common set of session times got allocated among the four pages?"

The procedure used to test this is ANOVA. The basis for it can be seen in the following resampling procedure (specified here for the A-B-C-D test of web page stickiness):

1. Combine all the data together in a single box
2. Shuffle and draw out four resamples of five values each
3. Record the mean of each of the four groups
4. Record the variance among the four group means

5. Repeat steps 2–4 many times (say 1,000)

What proportion of the time did the resampled variance exceed the observed variance? This is the p-value.

This type of permutation test is a bit more involved than the type used in "Permutation Test" on page 88. Fortunately, the `aovp` function in the `lmPerm` package computes a permutation test for this case:

```
> library(lmPerm)
> summary(aovp(Time ~ Page, data=four_sessions))
[1] "Settings:  unique SS "
Component 1 :
            Df R Sum Sq R Mean Sq Iter Pr(Prob)
Page         3    831.4    277.13 3104  0.09278 .
Residuals   16   1618.4    101.15
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value, given by `Pr(Prob)`, is 0.09278. The column `Iter` lists the number of iterations taken in the permutation test. The other columns correspond to a traditional ANOVA table and are described next.

F-Statistic

Just like the t-test can be used instead of a permutation test for comparing the mean of two groups, there is a statistical test for ANOVA based on the *F-statistic*. The F-statistic is based on the ratio of the variance across group means (i.e., the treatment effect) to the variance due to residual error. The higher this ratio, the more statistically significant the result. If the data follows a normal distribution, then statistical theory dictates that the statistic should have a certain distribution. Based on this, it is possible to compute a p-value.

In R, we can compute an *ANOVA table* using the `aov` function:

```
> summary(aov(Time ~ Page, data=four_sessions))
            Df Sum Sq Mean Sq F value Pr(>F)
Page         3  831.4   277.1    2.74 0.0776 .
Residuals   16 1618.4   101.2
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

`Df` is "degrees of freedom," `Sum Sq` is "sum of squares," `Mean Sq` is "mean squares" (short for mean-squared deviations), and `F value` is the F-statistic. For the grand average, sum of squares is the departure of the grand average from 0, squared, times 20 (the number of observations). The degrees of freedom for the grand average is 1, by definition. For the treatment means, the degrees of freedom is 3 (once three values are set, and then the grand average is set, the other treatment mean cannot vary). Sum of squares for the treatment means is the sum of squared departures between the treatment means and the grand average. For the residuals, degrees of freedom is 20 (all observations can vary), and SS is the sum of squared difference between the individual observations and the treatment means. Mean squares (MS) is the sum of squares divided by the degrees of freedom. The F-statistic is MS(treatment)/MS(error). The F value thus depends only on this ratio, and can be compared to a standard F distribution to determine whether the differences among treatment means is greater than would be expected in random chance variation.

Decomposition of Variance

Observed values in a data set can be considered sums of different components. For any observed data value within a data set, we can break it down into the grand average, the treatment effect, and the residual error. We call this a "decomposition of variance."

1. Start with grand average (173.75 for web page stickiness data).
2. Add treatment effect, which might be negative (independent variable = web page).
3. Add residual error, which might be negative.

Thus, the decomposition of the variance for the top-left value in the A-B-C-D test table is as follows:

1. Start with grand average: 173.75
2. Add treatment (group) effect: –1.75 (172 – 173.75).
3. Add residual: –8 (164 – 172).
4. Equals: 164.

Two-Way ANOVA

The A-B-C-D test just described is a "one-way" ANOVA, in which we have one factor (group) that is varying. We could have a second factor involved—say, "weekend versus weekday"—with data collected on each combination (group A weekend, group A weekday, group B weekend, etc.). This would be a "two-way ANOVA," and we would handle it in similar fashion to the one-way ANOVA by identifying the "interaction effect." After identifying the grand average effect, and the treatment effect, we then separate the weekend and the weekday observations for each group, and find the difference between the averages for those subsets and the treatment average.

You can see that ANOVA, then two-way ANOVA, are the first steps on the road toward a full statistical model, such as regression and logistic regression, in which multiple factors and their effects can be modeled (see Chapter 4).

Key Ideas

- ANOVA is a statistical proecdure for analyzing the results of an experiment with multiple groups.
- It is the extension of similar procedures for the A/B test, used to assess whether the overall variation among groups is within the range of chance variation.
- A useful outcome of an ANOVA is the identification of variance components associated with group treatments, interaction effects, and errors.

Further Reading

1. *Introductory Statistics: A Resampling Perspective* by Peter Bruce (Wiley, 2014) has a chapter on ANOVA.
2. *Introduction to Design and Analysis of Experiments* by George Cobb (Wiley, 2008) is a comprehensive and readable treatment of its subject.

Chi-Square Test

Web testing often goes beyond A/B testing and tests multiple treatments at once. The chi-square test is used with count data to test how well it fits some expected distribution. The most common use of the *chi-square* statistic in statistical practice is with $r \times c$ contingency tables, to assess whether the null hypothesis of independence among variables is reasonable.

The chi-square test was originally developed by Karl Pearson in 1900. The term "chi" comes from the greek letter ξ used by Pearson in the article.

Key Terms

Chi-square statistic
A measure of the extent to which some observed data departs from expectation.

Expectation or expected
How we would expect the data to turn out under some assumption, typically the null hypothesis.

d.f.
Degrees of freedom.

$r \times c$ means "rows by columns"—a 2×3 table has two rows and three columns.

Chi-Square Test: A Resampling Approach

Suppose you are testing three different headlines—A, B, and C—and you run them each on 1,000 visitors, with the results shown in Table 3-4.

Table 3-4. Web testing results of three different headlines

	Headline A	Headline B	Headline C
Click	14	8	12
No-click	986	992	988

The headlines certainly appear to differ. Headline A returns nearly twice the click rate of B. The actual numbers are small, though. A resampling procedure can test whether the click rates differ to an extent greater than chance might cause. For this test, we need to have the "expected" distribution of clicks, and, in this case, that would be under the null hypothesis assumption that all three headlines share the same click rate, for an overall click rate of 34/3,000. Under this assumption, our contingency table would look like Table 3-5.

Table 3-5. Expected if all three headlines have the same click rate (null hypothesis)

	Headline A	Headline B	Headline C
Click	11.33	11.33	11.33
No-click	988.67	988.67	988.67

The *Pearson residual* is defined as:

$$R = \frac{\text{Observed} - \text{Expected}}{\sqrt{\text{Expected}}}$$

R measures the extent to which the actual counts differ from these expected counts (see Table 3-6).

Table 3-6. Pearson residuals

	Headline A	Headline B	Headline C
Click	0.792	-0.990	0.198
No-click	-0.085	0.106	-0.021

The chi-squared statistic is defined as the sum of the squared Pearson residuals:

$$\xi = \sum_{i}^{r} \sum_{j}^{c} R^2$$

where r and c are the number of rows and columns, respectively. The chi-squared statistic for this example is 1.666. Is that more than could reasonably occur in a chance model?

We can test with this resampling algorithm:

1. Constitute a box with 34 ones (clicks) and 2,966 zeros (no clicks).
2. Shuffle, take three separate samples of 1,000, and count the clicks in each.
3. Find the squared differences between the shuffled counts and the expected counts, and sum them.
4. Repeat steps 2 and 3, say, 1,000 times.
5. How often does the resampled sum of squared deviations exceed the observed? That's the p-value.

The function `chisq.test` can be used to compute a resampled chi-square statistic. For the click data, the chi-square test is:

```
> chisq.test(clicks, simulate.p.value=TRUE)

        Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data:  clicks
X-squared = 1.6659, df = NA, p-value = 0.4853
```

The test shows that this result could easily have been obtained by randomness.

Chi-Squared Test: Statistical Theory

Asymptotic statistical theory shows that the distribution of the chi-squared statistic can be approximated by a *chi-square distribution*. The appropriate standard chi-square distribution is determined by the *degrees of freedom* (see "Degrees of Freedom" on page 104). For a contingency table, the degrees of freedom are related to the number of rows (r) and columns (s) as follows:

$$\text{degrees of freedom} = (r-1) \times (c-1)$$

The chi-square distribution is typically skewed, with a long tail to the right; see Figure 3-7 for the distribution with 1, 2, 5, and 10 degrees of freedom. The further out on the chi-square distribution the observed statistic is, the lower the p-value.

The function `chisq.test` can be used to compute the p-value using the chi-squared distribution as a reference:

```
> chisq.test(clicks, simulate.p.value=FALSE)

        Pearson's Chi-squared test

data:  clicks
X-squared = 1.6659, df = 2, p-value = 0.4348
```

The p-value is a little less than the resampling p-value: this is because the chi-square distribution is only an approximation of the actual distribution of the statistic.

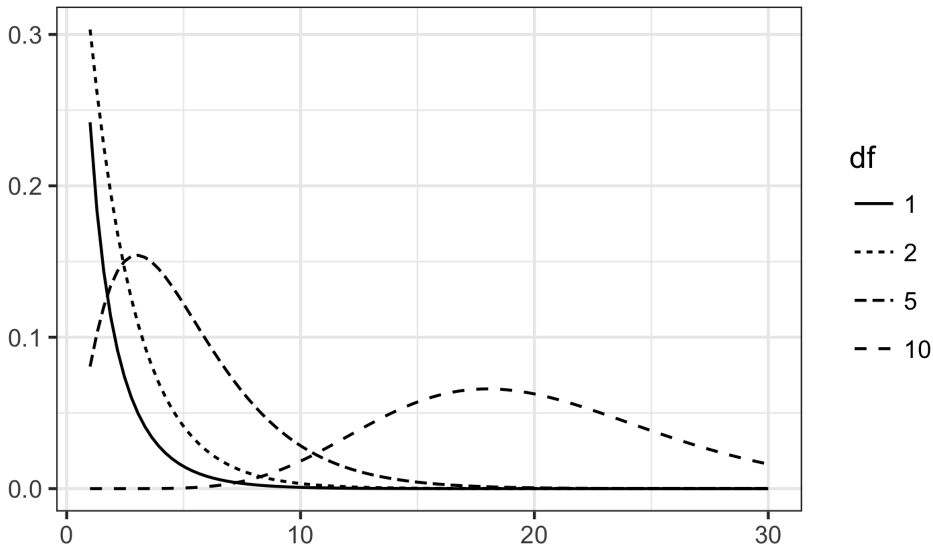


Figure 3-7. Chi-square distribution with various degrees of freedom (probability on y-axis, value of chi-square statistic on x-axis)

Fisher's Exact Test

The chi-square distribution is a good approximation of the shuffled resampling test just described, except when counts are extremely low (single digits, especially five or fewer). In such cases, the resampling procedure will yield more accurate p-values. In fact, most statistical software has a procedure to actually enumerate *all* the possible rearrangements (permutations) that can occur, tabulate their frequencies, and determine exactly how extreme the observed result is. This is called *Fisher's exact test* after the great statistician R. A. Fisher. R code for Fisher's exact test is simple in its basic form:

```
> fisher.test(clicks)

        Fisher's Exact Test for Count Data

data:  clicks
p-value = 0.4824
alternative hypothesis: two.sided
```

The p-value is very close to the p-value of 0.4853 obtained using the resampling method.

Where some counts are very low but others are quite high (e.g., the denominator in a conversion rate), it may be necessary to do a shuffled permutation test instead of a full exact test, due to the difficulty of calculating all possible permutations. The preceding R function has several arguments that control whether to use this approximation (`simulate.p.value=TRUE or FALSE`), how many iterations should be used (`B=...`), and a computational constraint (`workspace=...`) that limits how far calculations for the *exact* result should go.

Detecting Scientific Fraud

An interesting example is provided by Tufts University researcher Thereza Imanishi-Kari, who was accused in 1991 of fabricating data in her research. Congressman John Dingell became involved, and the case eventually led to the resignation of her colleague, David Baltimore, from the presidency of Rockefeller University.

Imanishi-Kari was ultimately exonerated after a lengthy proceeding. However, one element in the case rested on statistical evidence regarding the expected distribution of digits in her laboratory data, where each observation had many digits. Investigators focused on the *interior* digits, which would be expected to follow a *uniform random* distribution. That is, they would occur randomly, with each digit having equal probability of occurring (the lead digit might be predominantly one value, and the final digits might be affected by rounding). Table 3-7 lists the frequencies of interior digits from the actual data in the case.

Table 3-7. Central digit in laboratory data

Digit	Frequency
0	14
1	71
2	7
3	65
4	23
5	19
6	12
7	45
8	53
9	6

The distribution of the 315 digits, shown in Figure 3-8 certainly looks nonrandom:

Investigators calculated the departure from expectation (31.5—that's how often each digit would occur in a strictly uniform distribution) and used a chi-square test (a resampling procedure could equally have been used) to show that the actual distribution was well beyond the range of normal chance variation.

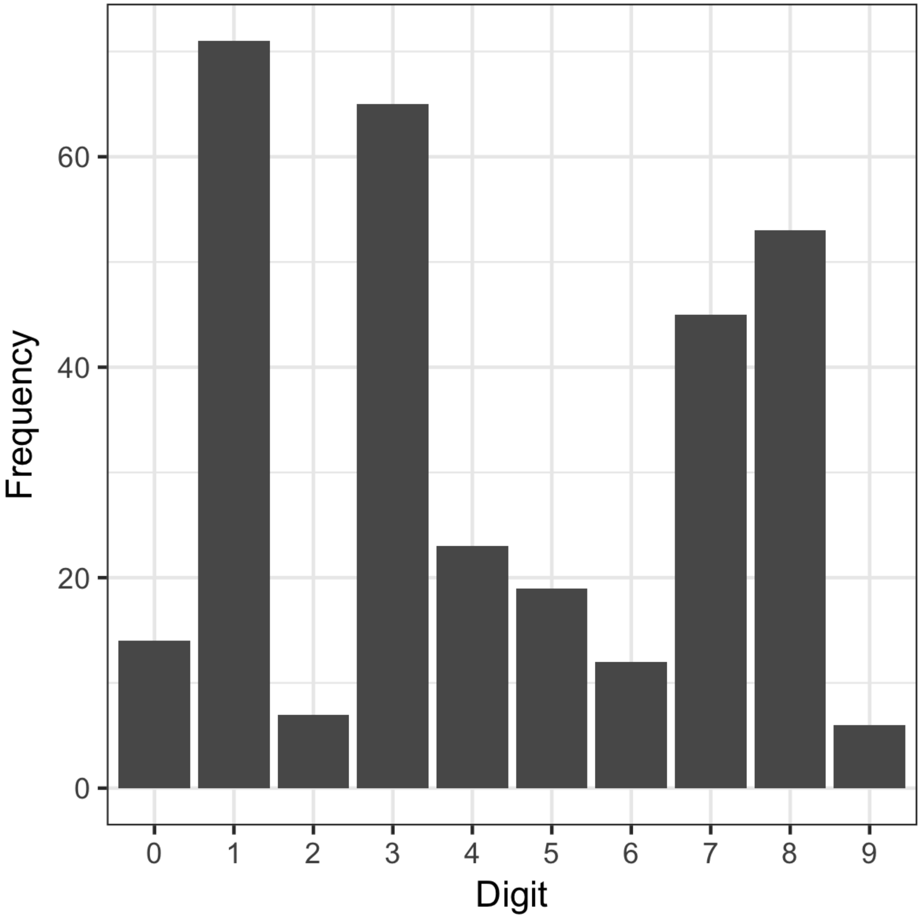


Figure 3-8. Frequency histogram for Imanishi-Kari lab data

Relevance for Data Science

Most standard uses of the chi-square test, or Fisher's exact test, are not terribly relevant for data science. In most experiments, whether A-B or A-B-C…, the goal is not simply to establish statistical significance, but rather to arive at the best treatment. For this purpose, multi-armed bandits (see "Multi-Arm Bandit Algorithm" on page 119) offer a more complete solution.

One data science application of the chi-square test, especially Fisher's exact version, is in determining appropriate sample sizes for web experiments. These experiments often have very low click rates and, despite thousands of exposures, count rates might be too small to yield definitive conclusions in an experiment. In such cases, Fisher's exact test, the chi-square test, and other tests can be useful as a component of power and sample size calculations (see "Power and Sample Size" on page 122).

Chi-square tests are used widely in research by investigators in search of the elusive statistically significant p-value that will allow publication. Chi-square tests, or similar resampling simulations, are used in data science applications more as a filter to determine whether an effect or feature is worthy of further consideration than as a formal test of significance. For example, they are used in spatial statistics and mapping to determine whether spatial data conforms to a specified null distribution (e.g., are crimes concentrated in a certain area to a greater degree than random chance would allow?). They can also be used in automated feature selection in machine learning, to assess class prevalence across features and identify features where the prevalence of a certain class is unusually high or low, in a way that is not compatible with random variation.

Key Ideas

- A common procedure in statistics is to test whether observed data counts are consistent with an assumption of independence (e.g., propensity to buy a particular item is independent of gender).
- The chi-square distribution is the reference distribution (which embodies the assumption of independence) to which the observed calculated chi-square statistic must be compared.

Further Reading

- R. A. Fisher's famous "Lady Tasting Tea" example from the beginning of the 20th century remains a simple and effective illustration of his exact test. Google "Lady Tasting Tea," and you will find a number of good writeups.
- Stat Trek offers a good tutorial on the chi-square test.

Multi-Arm Bandit Algorithm

Multi-arm bandits offer an approach to testing, especially web testing, that allows explicit optimization and more rapid decision making than the traditional statistical approach to designing experiments.

> ### Key Terms
>
> ***Multi-arm bandit***
> An imaginary slot machine with multiple arms for the customer to choose from, each with different payoffs, here taken to be an analogy for a multitreatment experiment.
>
> ***Arm***
> A treatment in an experiment (e.g., "headline A in a web test").
>
> ***Win***
> The experimental analog of a win at the slot machine (e.g., "customer clicks on the link").

A traditional A/B test involves data collected in an experiment, according to a specified design, to answer a specific question such as, "Which is better, treatment A or treatment B?" The presumption is that once we get an answer to that question, the experimenting is over and we proceed to act on the results.

You can probably perceive several difficulties with that approach. First, our answer may be inconclusive: "effect not proven." In other words, the results from the experiment may suggest an effect, but if there is an effect, we don't have a big enough sample to prove it (to the satisfaction of the traditional statistical standards). What decision do we take? Second, we might want to begin taking advantage of results that come in prior to the conclusion of the experiment. Third, we might want the right to change our minds or to try something different based on additional data that comes in after the experiment is over. The traditional approach to experiments and hypothesis tests dates from the 1920s, and is rather inflexible. The advent of computer power and software has enabled more powerful flexible approaches. Moreover, data science (and business in general) is not so worried about statistical significance, but more concerned with optimizing overall effort and results.

Bandit algorithms, which are very popular in web testing, allow you to test multiple treatments at once and reach conclusions faster than traditional statistical designs. They take their name from slot machines used in gambling, also termed one-armed bandits (since they are configured in such a way that they extract money from the gambler in a steady flow). If you imagine a slot machine with more than one arm, each arm paying out at a different rate, you would have a multi-armed bandit, which is the full name for this algorithm.

Your goal is to win as much money as possible, and more specifically, to identify and settle on the winning arm sooner rather than later. The challenge is that you don't know at what rate the arms pay out—you only know the results of pulling the arm. Suppose each "win" is for the same amount, no matter which arm. What differs is the probability of a win. Suppose further that you initially try each arm 50 times and get the following results:

> Arm A: 10 wins out of 50
> Arm B: 2 win out of 50
> Arm C: 4 wins out of 50

One extreme approach is to say, "Looks like arm A is a winner—let's quit trying the other arms and stick with A." This takes full advantage of the information from the initial trial. If A is truly superior, we get the benefit of that early on. On the other hand, if B or C is truly better, we lose any opportunity to discover that. Another extreme approach is to say, "This all looks to be within the realm of chance—let's keep pulling them all equally." This gives maximum opportunity for alternates to A to show themselves. However, in the process, we are deploying what seem to be inferior treatments. How long do we permit that? Bandit algorithms take a hybrid approach: we start pulling A more often, to take advantage of its apparent superiority, but we don't abandon B and C. We just pull them less often. If A continues to outperform, we continue to shift resources (pulls) away from B and C and pull A more often. If, on the other hand, C starts to do better, and A starts to do worse, we can shift pulls from A back to C. If one of them turns out to be superior to A and this was hidden in the initial trial due to chance, it now has an opportunity to emerge with further testing.

Now think of applying this to web testing. Instead of multiple slot machine arms, you might have multiple offers, headlines, colors, and so on, being tested on a website. Customers either click (a "win" for the merchant) or don't click. Initially, the offers are shown randomly and equally. If, however, one offer starts to outperform the others, it can be shown ("pulled") more often. But what should the parameters of the algorithm that modifies the pull rates be? What "pull rates" should we change to, and when should we change?

Here is one simple algorithm, the epsilon-greedy algorithm for an A/B test:

1. Generate a random number between 0 and 1.
2. If the number lies between 0 and epsilon (where epsilon is a number between 0 and 1, typically fairly small), flip a fair coin (50/50 probability), and:

   a. If the coin is heads, show offer A.
   b. If the coin is tails, show offer B.
3. If the number is ≥ epsilon, show whichever offer has had the highest response rate to date.

Epsilon is the single parameter that governs this algorithm. If epsilon is 1, we end up with a standard simple A/B experiment (random allocation between A and B for each subject). If epsilon is 0, we end up with a purely *greedy* algorithm—it seeks no further experimentation, simply assigning subjects (web visitors) to the best-performing treatment.

A more sophisticated algorithm uses "Thompson's sampling." This procedure "samples" (pulls a bandit arm) at each stage to maximize the probability of choosing the best arm. Of course you don't know which is the best arm—that's the whole problem! —but as you observe the payoff with each successive draw, you gain more information. Thompson's sampling uses a Bayesian approach: some prior distribution of rewards is assumed initially, using what is called a *beta distribution* (this is a common mechanism for specifying prior information in a Bayesian problem). As information accumulates from each draw, this information can be updated, allowing the selection of the next draw to be better optimized as far as choosing the right arm.

Bandit algorithms can efficiently handle 3+ treatments and move toward optimal selection of the "best." For traditional statistical testing procedures, the complexity of decision making for 3+ treatments far outstrips that of the traditional A/B test, and the advantage of bandit algorithms is much greater.

Key Ideas

- Traditional A/B tests envision a random sampling process, which can lead to excessive exposure to the inferior treatment.
- Multi-arm bandits, in contrast, alter the sampling process to incorporate information learned during the experiment and reduce the frequency of the inferior treatment.
- They also facilitate efficient treatment of more than two treatments.
- There are different algorithms for shifting sampling probability away from the inferior treatment(s) and to the (presumed) superior one.

Further Reading

- An excellent short treatment of multi-arm bandit algorithms is found in *Bandit Algorithms*, by John Myles White (O'Reilly, 2012). White includes Python code, as well as the results of simulations to assess the performance of bandits.
- For more (somewhat technical) information about Thompson sampling, see "Analysis of Thompson Sampling for the Multi-armed Bandit Problem" by Shipra Agrawal and Navin Goyal.

Power and Sample Size

If you run a web test, how do you decide how long it should run (i.e., how many impressions per treatment are needed)? Despite what you may read in many guides to web testing on the web, there is no good general guidance—it depends, mainly, on the frequency with which the desired goal is attained.

> ### Key Terms
>
> ***Effect size***
> The minimum size of the effect that you hope to be able to detect in a statistical test, such as "a 20% improvement in click rates".
>
> ***Power***
> The probability of detecting a given effect size with a given sample size.
>
> ***Significance level***
> The statistical significance level at which the test will be conducted.

One step in statistical calculations for sample size is to ask "Will a hypothesis test actually reveal a difference between treatments A and B?" The outcome of a hypothesis test—the p-value—depends on what the real difference is between treatment A and treatment B. It also depends on the luck of the draw—who gets selected for the groups in the experiment. But it makes sense that the bigger the actual difference between treatments A and B, the greater the probability that our experiment will reveal it; and the smaller the difference, the more data will be needed to detect it. To distinguish between a .350 hitter in baseball, and a .200 hitter, not that many at-bats are needed. To distinguish between a .300 hitter and a .280 hitter, a good many more at-bats will be needed.

Power is the probability of detecting a specified *effect size* with specified sample characteristics (size and variability). For example, we might say (hypothetically) that the probability of distinguishing between a .330 hitter and a .200 hitter in 25 at-bats is

0.75. The effect size here is a difference of .130. And “detecting” means that a hypothesis test will reject the null hypothesis of “no difference” and conclude there is a real effect. So the experiment of 25 at-bats (n = 25) for two hitters, with an effect size of 0.130, has (hypothetical) power of 0.75 or 75%.

You can see that there are several moving parts here, and it is easy to get tangled up with the numerous statistical assumptions and formulas that will be needed (to specify sample variability, effect size, sample size, alpha-level for the hypothesis test, etc., and to calculate power). Indeed, there is special-purpose statistical software to calculate power. Most data scientists will not need to go through all the formal steps needed to report power, for example, in a published paper. However, they may face occasions where they want to collect some data for an A/B test, and collecting or processing the data involves some cost. In that case, knowing approximately how much data to collect can help avoid the situation where you collect data at some effort, and the result ends up being inconclusive. Here’s a fairly intuitive alternative approach:

1. Start with some hypothetical data that represents your best guess about the data that will result (perhaps based on prior data)—for example, a box with 20 ones and 80 zeros to represent a .200 hitter, or a box with some observations of “time spent on website.”
2. Create a second sample simply by adding the desired effect size to the first sample—for example, a second box with 33 ones and 67 zeros, or a second box with 25 seconds added to each initial “time spent on website.”
3. Draw a bootstrap sample of size n from each box.
4. Conduct a permutation (or formula-based) hypothesis test on the two bootstrap samples and record whether the difference between them is statistically significant.
5. Repeat the preceding two steps many times and determine how often the difference was significant—that’s the estimated power.

Sample Size

The most common use of power calculations is to estimate how big a sample you will need.

For example, suppose you are looking at click-through rates (clicks as a percentage of exposures), and testing a new ad against an existing ad. How many clicks do you need to accumulate in the study? If you are only interested in results that show a huge difference (say a 50% difference), a relatively small sample might do the trick. If, on the other hand, even a minor difference would be of interest, then a much larger sample is needed. A standard approach is to establish a policy that a new ad must do better than an existing ad by some percentage, say 10%; otherwise, the existing ad will remain in place. This goal, the "effect size," then drives the sample size.

For example, suppose current click-through rates are about 1.1%, and you are seeking a 10% boost to 1.21%. So we have two boxes, box A with 1.1% ones (say 110 ones and 9,890 zeros), and box B with 1.21% ones (say 121 ones and 9,879 zeros). For starters, let's try 300 draws from each box (this would be like 300 "impressions" for each ad). Suppose our first draw yields the following:

Box A: 3 ones
Box B: 5 ones

Right away we can see that any hypothesis test would reveal this difference (5 versus 3) to be well within the range of chance variation. This combination of sample size (n = 300 in each group) and effect size (10% difference) is too small for any hypothesis test to reliably show a difference.

So we can try increasing the sample size (let's try 2,000 impressions), and require a larger improvement (30% instead of 10%).

For example, suppose current click-through rates are still 1.1%, but we are now seeking a 50% boost to 1.65%. So we have two boxes: box A still with 1.1% ones (say 110 ones and 9,890 zeros), and box B with 1.65% ones (say 165 ones and 9,868 zeros). Now we'll try 2,000 draws from each box. Suppose our first draw yields the following:

Box A: 19 ones
Box B: 34 ones

A significance test on this difference (34–19) shows it still registers as "not significant" (though much closer to significance than the earlier difference of 5–3). To calculate power, we would need to repeat the previous procedure many times, or use statistical software that can calculate power, but our initial draw suggests to us that even detecting a 50% improvement will require several thousand ad impressions.

In summary, for calculating power or required sample size, there are four moving parts:

- Sample size
- Effect size you want to detect
- Significance level (alpha) at which the test will be conducted
- Power

Specify any three of them, and the fourth can be calculated. Most commonly, you would want to calculate sample size, so you must specify the other three. Here is R code for a test involving two proportions, where both samples are the same size (this uses the pwr package):

```
pwr.2p.test(h = ..., n = ..., sig.level = ..., power = )

h= effect size (as a proportion)
n = sample size
sig.level = the significance level (alpha) at which the test will be conducted
power = power (probability of detecting the effect size)
```

Key Ideas

- Finding out how big a sample size you need requires thinking ahead to the statistical test you plan to conduct.
- You must specify the minimum size of the effect that you want to detect.
- You must also specify the required probability of detecting that effect size (power).
- Finally, you must specify the significance level (alpha) at which the test will be conducted.

Further Reading

1. *Sample Size Determination and Power*, by Tom Ryan (Wiley, 2013), is a comprehensive and readable review of this subject.
2. Steve Simon, a statistical consultant, has written a very engaging narrative-style post on the subject.

Summary

The principles of experimental design—randomization of subjects into two or more groups receiving different treatments—allow us to draw valid conclusions about how well the treatments work. It is best to include a control treatment of "making no change." The subject of formal statistical inference—hypothesis testing, p-values, t-tests, and much more along these lines—occupies much time and space in a traditional statistics course or text, and the formality is mostly unneeded from a data science perspective. However, it remains important to recognize the role that random variation can play in fooling the human brain. Intuitive resampling procedures (permutation and bootstrap) allow data scientists to gauge the extent to which chance variation can play a role in their data analysis.

CHAPTER 4
Regression and Prediction

Perhaps the most common goal in statistics is to answer the question: Is the variable X (or more likely, $X_1, ..., X_p$) associated with a variable Y, and, if so, what is the relationship and can we use it to predict Y?

Nowhere is the nexus between statistics and data science stronger than in the realm of prediction—specifically the prediction of an outcome (target) variable based on the values of other "predictor" variables. Another important connection is in the area of *anomaly detection*, where regression diagnostics originally intended for data analysis and improving the regression model can be used to detect unusual records. The antecedents of correlation and linear regression date back over a century.

Simple Linear Regression

Simple linear regression models the relationship between the magnitude of one variable and that of a second—for example, as X increases, Y also increases. Or as X increases, Y decreases.[1] Correlation is another way to measure how two variables are related: see the section "Correlation" on page 29. The difference is that while correlation measures the strength of an association between two variables, regression quantifies the nature of the relationship.

1 This and subsequent sections in this chapter © 2017 Datastats, LLC, Peter Bruce and Andrew Bruce, used by permission.

Key Terms for Simple Linear Regression

Response
: The variable we are trying to predict.

  Synonyms
  : dependent variable, Y-variable, target, outcome

Independent variable
: The variable used to predict the response.

  Synonyms
  : independent variable, X-variable, feature, attribute

Record
: The vector of predictor and outcome values for a specific individual or case.

  Synonyms
  : row, case, instance, example

Intercept
: The intercept of the regression line—that is, the predicted value when $X = 0$.

  Synonyms
  : b_0, β_0

Regression coefficient
: The slope of the regression line.

  Synonyms
  : slope, b_1, β_1, parameter estimates, weights

Fitted values
: The estimates $\hat{Y}_i$ obtained from the regression line.

  Synonyms
  : predicted values

Residuals
: The difference between the observed values and the fitted values.

  Synonyms
  : errors

Least squares
: The method of fitting a regression by minimizing the sum of squared residuals.

Synonyms
ordinary least squares

The Regression Equation

Simple linear regression estimates exactly how much Y will change when X changes by a certain amount. With the correlation coefficient, the variables X and Y are interchangable. With regression, we are trying to predict the Y variable from X using a linear relationship (i.e., a line):

$$Y = b_0 + b_1 X$$

We read this as "Y equals b_1 times X, plus a constant b_0." The symbol b_0 is known as the *intercept* (or constant), and the symbol b_1 as the *slope* for X. Both appear in R output as *coefficients*, though in general use the term *coefficient* is often reserved for b_1. The Y variable is known as the *response* or *dependent* variable since it depends on X. The X variable is known as the *predictor* or *independent* variable. The machine learning community tends to use other terms, calling Y the *target* and X a *feature* vector.

Consider the scatterplot in Figure 4-1 displaying the number of years a worker was exposed to cotton dust (`Exposure`) versus a measure of lung capacity (`PEFR` or "peak expiratory flow rate"). How is `PEFR` related to `Exposure`? It's hard to tell just based on the picture.

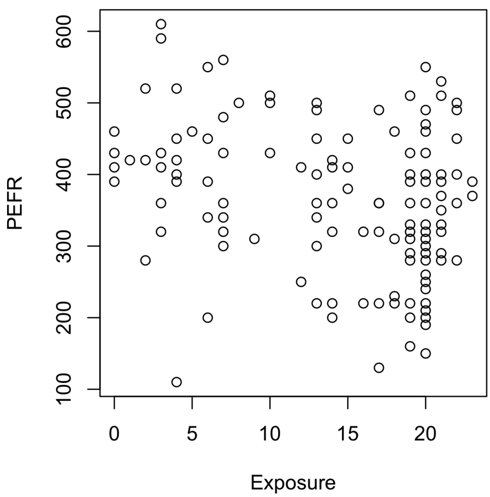


Figure 4-1. Cotton exposure versus lung capacity

Simple linear regression tries to find the "best" line to predict the response PEFR as a function of the predictor variable Exposure.

$$\text{PEFR} = b_0 + b_1\text{Exposure}$$

The lm function in R can be used to fit a linear regression.

```
model <- lm(PEFR ~ Exposure, data=lung)
```

lm standards for *linear model* and the ~ symbol denotes that PEFR is predicted by Exposure.

Printing the model object produces the following output:

```
Call:
lm(formula = PEFR ~ Exposure, data = lung)

Coefficients:
(Intercept)     Exposure
    424.583       -4.185
```

The intercept, or b_0, is 424.583 and can be interpreted as the predicted PEFR for a worker with zero years exposure. The regression coefficient, or b_1, can be interpreted as follows: for each additional year that a worker is exposed to cotton dust, the worker's PEFR measurement is reduced by –4.185.

The regression line from this model is displayed in Figure 4-2.

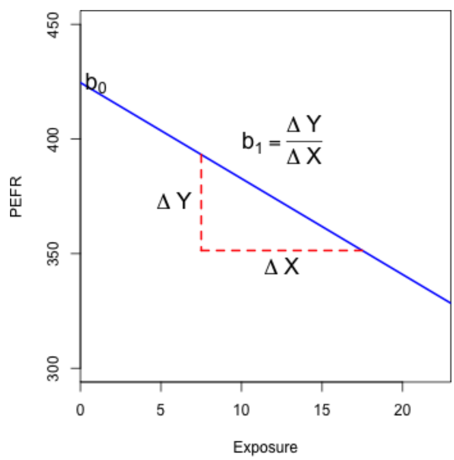


Figure 4-2. Slope and intercept for the regression fit to the lung data

Fitted Values and Residuals

Important concepts in regression analysis are the *fitted* values and *residuals*. In general, the data doesn't fall exactly on a line, so the regression equation should include an explicit error term e_i:

$$Y_i = b_0 + b_1 X_i + e_i$$

The *fitted* values, also referred to as the *predicted* values, are typically denoted by $\hat{Y}_i$ (Y-hat). These are given by:

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_i$$

The notation $\hat{b}_0$ and $\hat{b}_1$ indicates that the coefficients are estimated versus known.

Hat Notation: Estimates Versus Known Values

The "hat" notation is used to differentiate between estimates and known values. So the symbol $\hat{b}$ ("b-hat") is an estimate of the unknown parameter b. Why do statisticians differentiate between the estimate and the true value? The estimate has uncertainty, whereas the true value is fixed.[2]

We compute the residuals $\hat{e}_i$ by subtracting the *predicted* values from the original data:

$$\hat{e}_i = Y_i - \hat{Y}_i$$

In R, we can obtain the fitted values and residuals using the functions `predict` and `residuals`:

```
fitted <- predict(model)
resid <- residuals(model)
```

Figure 4-3 illustrates the residuals from the regression line fit to the lung data. The residuals are the length of the vertical dashed lines from the data to the line.

2 In Bayesian statistics, the true value is assumed to be a random variable with a specified distribution. In the Bayesian context, instead of estimates of unknown parameters, there are posterior and prior distributions.

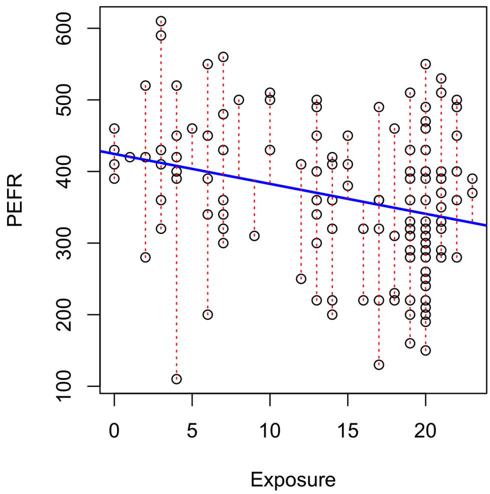


Figure 4-3. Residuals from a regression line (note the different y-axis scale from Figure 4-2, hence the apparently different slope)

Least Squares

How is the model fit to the data? When there is a clear relationship, you could imagine fitting the line by hand. In practice, the regression line is the estimate that minimizes the sum of squared residual values, also called the *residual sum of squares* or *RSS*:

$$\begin{aligned} RSS &= \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \\ &= \sum_{i=1}^{n} \left(Y_i - \hat{b}_0 - \hat{b}_1 X_i\right)^2 \end{aligned}$$

The estimates $\hat{b}_0$ and $\hat{b}_1$ are the values that minimize RSS.

The method of minimizing the sum of the squared residuals is termed *least squares* regression, or *ordinary least squares* (OLS) regression. It is often attributed to Carl Friedrich Gauss, the German mathmetician, but was first published by the French mathmetician Adrien-Marie Legendre in 1805. Least squares regression leads to a simple formula to compute the coefficients:

$$\hat{b}_1 = \frac{\sum_{i=1}^{n} (Y_i - \overline{Y})(X_i - \overline{X})}{\sum_{i=1}^{n} (X_i - \overline{X})^2}$$

$$\hat{b}_0 = \overline{Y} - \hat{b}_1 \overline{X}$$

Historically, computational convenience is one reason for the widespread use of least squares in regression. With the advent of big data, computational speed is still an important factor. Least squares, like the mean (see "Median and Robust Estimates" on page 10), are sensitive to outliers, although this tends to be a signicant problem only in small or moderate-sized problems. See "Outliers" on page 156 for a discussion of outliers in regression.

Regression Terminology

When analysts and researchers use the term *regression* by itself, they are typically referring to linear regression; the focus is usually on developing a linear model to explain the relationship between predictor variables and a numeric outcome variable. In its formal statistical sense, regression also includes nonlinear models that yield a functional relationship between predictors and outcome variables. In the machine learning community, the term is also occasionally used loosely to refer to the use of any predictive model that produces a predicted numeric outcome (standing in distinction from classification methods that predict a binary or categorical outcome).

Prediction versus Explanation (Profiling)

Historically, a primary use of regression was to illuminate a supposed linear relationship between predictor variables and an outcome variable. The goal has been to understand a relationship and explain it using the data that the regression was fit to. In this case, the primary focus is on the estimated slope of the regression equation, $\hat{b}$. Economists want to know the relationship between consumer spending and GDP growth. Public health officials might want to understand whether a public information campaign is effective in promoting safe sex practices. In such cases, the focus is not on predicting individual cases, but rather on understanding the overall relationship.

With the advent of big data, regression is widely used to form a model to predict individual outcomes for new data, rather than explain data in hand (i.e., a predictive model). In this instance, the main items of interest are the fitted values $\hat{Y}$. In marketing, regression can be used to predict the change in revenue in response to the size of an ad campaign. Universities use regression to predict students' GPA based on their SAT scores.

A regression model that fits the data well is set up such that changes in X lead to changes in Y. However, by itself, the regression equation does not prove the direction of causation. Conclusions about causation must come from a broader context of understanding about the relationship. For example, a regression equation might show a definite relationship between number of clicks on a web ad and number of conversions. It is our knowledge of the marketing process, not the regression equation, that leads us to the conclusion that clicks on the ad lead to sales, and not vice versa.

Key Ideas

- The regression equation models the relationship between a response variable Y and a predictor variable X as a line.
- A regression model yields fitted values and residuals—predictions of the response and the errors of the predictions.
- Regression models are typically fit by the method of least squares.
- Regression is used both for prediction and explanation.

Further Reading

For an in-depth treatment of prediction versus explanation, see Galit Shmueli's article "To Explain or to Predict".

Multiple Linear Regression

When there are multiple predictors, the equation is simply extended to accommodate them:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_p X_p + e$$

Instead of a line, we now have a linear model—the relationship between each coefficient and its variable (feature) is linear.

Key Terms for Multiple Linear Regression

Root mean squared error
: The square root of the average squared error of the regression (this is the most widely used metric to compare regression models).

Synonyms
RMSE

Residual standard error
The same as the root mean squared error, but adjusted for degrees of freedom.

Synonyms
RSE

R-squared
The proportion of variance explained by the model, from 0 to 1.

Synonyms
coefficient of determination, R^2

t-statistic
The coefficient for a predictor, divided by the standard error of the coefficient, giving a metric to compare the importance of variables in the model.

Weighted regression
Regression with the records having different weights.

All of the other concepts in simple linear regression, such as fitting by least squares and the definition of fitted values and residuals, extend to the multiple linear regression setting. For example, the fitted values are given by:

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_{1,i} + \hat{b}_2 X_{2,i} + \ldots + \hat{b}_p X_{p,i}$$

Example: King County Housing Data

An example of using regression is in estimating the value of houses. County assessors must estimate the value of a house for the purposes of assessing taxes. Real estate consumers and professionals consult popular websites such as Zillow to ascertain a fair price. Here are a few rows of housing data from King County (Seattle), Washington, from the `house data.frame`:

```
head(house[, c("AdjSalePrice", "SqFtTotLiving", "SqFtLot", "Bathrooms",
               "Bedrooms", "BldgGrade")])
Source: local data frame [6 x 6]

  AdjSalePrice SqFtTotLiving SqFtLot Bathrooms Bedrooms BldgGrade
         (dbl)         (int)   (int)     (dbl)    (int)     (int)
1       300805          2400    9373      3.00        6         7
2      1076162          3764   20156      3.75        4        10
3       761805          2060   26036      1.75        4         8
4       442065          3200    8618      3.75        5         7
5       297065          1720    8620      1.75        4         7
6       411781           930    1012      1.50        2         8
```

The goal is to predict the sales price from the other variables. The `lm` handles the multiple regression case simply by including more terms on the righthand side of the equation; the argument `na.action=na.omit` causes the model to drop records that have missing values:

```
house_lm <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, na.action=na.omit)
```

Printing `house_lm` object produces the following output:

```
house_lm

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade, data = house, na.action = na.omit)

Coefficients:
  (Intercept)  SqFtTotLiving        SqFtLot      Bathrooms
   -5.219e+05      2.288e+02     -6.051e-02     -1.944e+04
     Bedrooms      BldgGrade
   -4.778e+04      1.061e+05
```

The interpretation of the coefficients is as with simple linear regression: the predicted value $\hat{Y}$ changes by the coefficient b_j for each unit change in X_j assuming all the other variables, X_k for $k \neq j$, remain the same. For example, adding an extra finished square foot to a house increases the estimated value by roughly $229; adding 1,000 finished square feet implies the value will increase by $228,800.

Assessing the Model

The most important performance metric from a data science perspective is *root mean squared error*, or *RMSE*. RMSE is the square root of the average squared error in the predicted $\hat{y}_i$ values:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{n}}$$

This measures the overall accuracy of the model, and is a basis for comparing it to other models (including models fit using machine learning techniques). Similar to RMSE is the *residual standard error*, or *RSE*. In this case we have p predictors, and the RSE is given by:

$$RSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{(n - p - 1)}}$$

The only difference is that the denominator is the degrees of freedom, as opposed to number of records (see "Degrees of Freedom" on page 104). In practice, for linear regression, the difference between RMSE and RSE is very small, particularly for big data applications.

The `summary` function in R computes RSE as well as other metrics for a regression model:

```
summary(house_lm)

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade, data = house, na.action = na.omit)

Residuals:
     Min       1Q   Median       3Q      Max
-1199508  -118879   -20982    87414  9472982

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -5.219e+05  1.565e+04 -33.349  < 2e-16 ***
SqFtTotLiving  2.288e+02  3.898e+00  58.699  < 2e-16 ***
SqFtLot       -6.051e-02  6.118e-02  -0.989    0.323
Bathrooms     -1.944e+04  3.625e+03  -5.362 8.32e-08 ***
Bedrooms      -4.778e+04  2.489e+03 -19.194  < 2e-16 ***
BldgGrade      1.061e+05  2.396e+03  44.287  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 261200 on 22683 degrees of freedom
Multiple R-squared:  0.5407,    Adjusted R-squared:  0.5406
F-statistic:  5340 on 5 and 22683 DF,  p-value: < 2.2e-16
```

Another useful metric that you will see in software output is the *coefficient of determination*, also called the *R-squared* statistic or R^2. R-squared ranges from 0 to 1 and measures the proportion of variation in the data that is accounted for in the model. It is useful mainly in explanatory uses of regression where you want to assess how well the model fits the data. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

The denominator is proportional to the variance of Y. The output from R also reports an *adjusted R-squared*, which adjusts for the degrees of freedom; seldom is this significantly different in multiple regression.

Along with the estimated coefficients, R reports the standard error of the coefficients (SE) and a *t-statistic*:

$$t_b = \frac{\hat{b}}{\mathrm{SE}\left(\hat{b}\right)}$$

The t-statistic—and its mirror image, the p-value—measures the extent to which a coefficient is "statistically significant"—that is, outside the range of what a random chance arrangement of predictor and target variable might produce. The higher the t-statistic (and the lower the p-value), the more significant the predictor. Since parsimony is a valuable model feature, it is useful to have a tool like this to guide choice of variables to include as predictors (see "Model Selection and Stepwise Regression" on page 139).

> In addition to the t-statistic, R and other packages will often report a *p-value* (`Pr(>|t|)` in the R output) and *F-statistic*. Data scientists do not generally get too involved with the interpretation of these statistics, nor with the issue of statistical significance. Data scientists primarily focus on the t-statistic as a useful guide for whether to include a predictor in a model or not. High t-statistics (which go with p-values near 0) indicate a predictor should be retained in a model, while very low t-statistics indicate a predictor could be dropped. See "P-Value" on page 96 for more discussion.

Cross-Validation

Classic statistical regression metrics (R^2, F-statistics, and p-values) are all "in-sample" metrics—they are applied to the same data that was used to fit the model. Intuitively, you can see that it would make a lot of sense to set aside some of the original data, not use it to fit the model, and then apply the model to the set-aside (holdout) data to see how well it does. Normally, you would use a majority of the data to fit the model, and use a smaller portion to test the model.

This idea of "out-of-sample" validation is not new, but it did not really take hold until larger data sets became more prevalent; with a small data set, analysts typically want to use all the data and fit the best possible model.

Using a holdout sample, though, leaves you subject to some uncertainty that arises simply from variability in the small holdout sample. How different would the assessment be if you selected a different holdout sample?

Cross-validation extends the idea of a holdout sample to multiple sequential holdout samples. The algorithm for basic *k-fold cross-validation* is as follows:

1. Set aside *1/k* of the data as a holdout sample.
2. Train the model on the remaining data.

3. Apply (score) the model to the *1/k* holdout, and record needed model assessment metrics.
4. Restore the first *1/k* of the data, and set aside the next *1/k* (excluding any records that got picked the first time).
5. Repeat steps 2 and 3.
6. Repeat until each record has been used in the holdout portion.
7. Average or otherwise combine the model assessment metrics.

The division of the data into the training sample and the holdout sample is also called a *fold*.

Model Selection and Stepwise Regression

In some problems, many variables could be used as predictors in a regression. For example, to predict house value, additional variables such as the basement size or year built could be used. In R, these are easy to add to the regression equation:

```
house_full <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                   Bedrooms + BldgGrade + PropertyType + NbrLivingUnits +
                   SqFtFinBasement + YrBuilt + YrRenovated +
                   NewConstruction,
                 data=house, na.action=na.omit)
```

Adding more variables, however, does not necessarily mean we have a better model. Statisticians use the principle of *Occam's razor* to guide the choice of a model: all things being equal, a simpler model should be used in preference to a more complicated model.

Including additional variables always reduces RMSE and increases R^2. Hence, these are not appropriate to help guide the model choice. In the 1970s, Hirotugu Akaike, the eminent Japanese statistician, deveoped a metric called *AIC* (Akaike's Information Criteria) that penalizes adding terms to a model. In the case of regression, AIC has the form:

$$AIC = 2P + n \log(\mathtt{RSS}/n)$$

where p is the number of variables and n is the number of records. The goal is to find the model that minimizes AIC; models with k more extra variables are penalized by $2k$.

AIC, BIC and Mallows Cp

The formula for AIC may seem a bit mysterious, but in fact it is based on asymptotic results in information theory. There are several variants to AIC:

- AICc: a version of AIC corrected for small sample sizes.
- BIC or Bayesian information criteria: similar to AIC with a stronger penalty for including additional variables to the model.
- Mallows Cp: A variant of AIC developed by Colin Mallows.

Data scientists generally do not need to worry about the differences among these in-sample metrics or the underlying theory behind them.

How do we find the model that minimizes AIC? One approach is to search through all possible models, called *all subset regression*. This is computationally expensive and is not feasible for problems with large data and many variables. An attractive alternative is to use *stepwise regression*, which successively adds and drops predictors to find a model that lowers AIC. The `MASS` package by Venebles and Ripley offers a stepwise regression function called `stepAIC`:

```
library(MASS)
step <- stepAIC(house_full, direction="both")
step

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + Bathrooms + Bedrooms +
    BldgGrade + PropertyType + SqFtFinBasement + YrBuilt, data = house0,
    na.action = na.omit)

Coefficients:
              (Intercept)              SqFtTotLiving
               6227632.22                     186.50
                Bathrooms                   Bedrooms
                 44721.72                  -49807.18
                BldgGrade  PropertyTypeSingle Family
                139179.23                   23328.69
    PropertyTypeTownhouse            SqFtFinBasement
                 92216.25                       9.04
                  YrBuilt
                 -3592.47
```

The function chose a model in which several variables were dropped from `house_full`: `SqFtLot`, `NbrLivingUnits`, `YrRenovated`, and `NewConstruction`.

Simpler yet are *forward selection* and *backward selection*. In forward selection, you start with no predictors and add them one-by-one, at each step adding the predictor