Think Stats: Probability and Statistics for Programmers

Version 1.5.9

Think Stats

Probability and Statistics for Programmers

Version 1.5.9

Allen B. Downey

Green Tea Press

Needham, Massachusetts

Green Tea Press
9 Washburn Ave
Needham MA 02492

The original form of this book is LaTeX source code. Compiling this code has the effect of generating a device-independent representation of a textbook, which can be converted to other formats and printed.

The LaTeX source for this book is available from http://thinkstats.com.

The cover for this book is based on a photo by Paul Friel (http://flickr.com/people/frielp/), who made it available under the Creative Commons Attribution license. The original photo is at http://flickr.com/photos/frielp/11999738/.

Preface

Why I wrote this book

Think Stats: Probability and Statistics for Programmers is a textbook for a new kind of introductory prob-stat class. It emphasizes the use of statistics to explore large datasets. It takes a computational approach, which has several advantages:

- Students write programs as a way of developing and testing their understanding. For example, they write functions to compute a least squares fit, residuals, and the coefficient of determination. Writing and testing this code requires them to understand the concepts and implicitly corrects misunderstandings.

- Students run experiments to test statistical behavior. For example, they explore the Central Limit Theorem (CLT) by generating samples from several distributions. When they see that the sum of values from a Pareto distribution doesn't converge to normal, they remember the assumptions the CLT is based on.

- Some ideas that are hard to grasp mathematically are easy to understand by simulation. For example, we approximate p-values by running Monte Carlo simulations, which reinforces the meaning of the p-value.

- Using discrete distributions and computation makes it possible to present topics like Bayesian estimation that are not usually covered in an introductory class. For example, one exercise asks students to compute the posterior distribution for the "German tank problem," which is difficult analytically but surprisingly easy computationally.

- Because students work in a general-purpose programming language (Python), they are able to import data from almost any source. They are not limited to data that has been cleaned and formatted for a particular statistics tool.

The book lends itself to a project-based approach. In my class, students work on a semester-long project that requires them to pose a statistical question, find a dataset that can address it, and apply each of the techniques they learn to their own data.

To demonstrate the kind of analysis I want students to do, the book presents a case study that runs through all of the chapters. It uses data from two sources:

- The National Survey of Family Growth (NSFG), conducted by the U.S. Centers for Disease Control and Prevention (CDC) to gather "information on family life, marriage and divorce, pregnancy, infertility, use of contraception, and men's and women's health." (See `http://cdc.gov/nchs/nsfg.htm`.)

- The Behavioral Risk Factor Surveillance System (BRFSS), conducted by the National Center for Chronic Disease Prevention and Health Promotion to "track health conditions and risk behaviors in the United States." (See `http://cdc.gov/BRFSS/`.)

Other examples use data from the IRS, the U.S. Census, and the Boston Marathon.

How I wrote this book

When people write a new textbook, they usually start by reading a stack of old textbooks. As a result, most books contain the same material in pretty much the same order. Often there are phrases, and errors, that propagate from one book to the next; Stephen Jay Gould pointed out an example in his essay, "The Case of the Creeping Fox Terrier[1]."

I did not do that. In fact, I used almost no printed material while I was writing this book, for several reasons:

- My goal was to explore a new approach to this material, so I didn't want much exposure to existing approaches.

- Since I am making this book available under a free license, I wanted to make sure that no part of it was encumbered by copyright restrictions.

[1] A breed of dog that is about half the size of a Hyracotherium (see `http://wikipedia.org/wiki/Hyracotherium`).

- Many readers of my books don't have access to libraries of printed material, so I tried to make references to resources that are freely available on the Internet.
- Proponents of old media think that the exclusive use of electronic resources is lazy and unreliable. They might be right about the first part, but I think they are wrong about the second, so I wanted to test my theory.

The resource I used more than any other is Wikipedia, the bugbear of librarians everywhere. In general, the articles I read on statistical topics were very good (although I made a few small changes along the way). I include references to Wikipedia pages throughout the book and I encourage you to follow those links; in many cases, the Wikipedia page picks up where my description leaves off. The vocabulary and notation in this book are generally consistent with Wikipedia, unless I had a good reason to deviate.

Other resources I found useful were Wolfram MathWorld and (of course) Google. I also used two books, David MacKay's *Information Theory, Inference, and Learning Algorithms*, which is the book that got me hooked on Bayesian statistics, and Press et al.'s *Numerical Recipes in C*. But both books are available online, so I don't feel too bad.

Allen B. Downey
Needham MA

Allen B. Downey is a Professor of Computer Science at the Franklin W. Olin College of Engineering.

Contributor List

If you have a suggestion or correction, please send email to `downey@allendowney.com`. If I make a change based on your feedback, I will add you to the contributor list (unless you ask to be omitted).

If you include at least part of the sentence the error appears in, that makes it easy for me to search. Page and section numbers are fine, too, but not quite as easy to work with. Thanks!

- Lisa Downey and June Downey read an early draft and made many corrections and suggestions.

- Steven Zhang found several errors.
- Andy Pethan and Molly Farison helped debug some of the solutions, and Molly spotted several typos.
- Andrew Heine found an error in my error function.
- Dr. Nikolas Akerblom knows how big a Hyracotherium is.
- Alex Morrow clarified one of the code examples.
- Jonathan Street caught an error in the nick of time.
- Gábor Lipták found a typo in the book and the relay race solution.
- Many thanks to Kevin Smith and Tim Arnold for their work on plasTeX, which I used to convert this book to DocBook.
- George Caplan sent several suggestions for improving clarity.
- Julian Ceipek found an error and a number of typos.
- Stijn Debrouwere, Leo Marihart III, Jonathan Hammler, and Kent Johnson found errors in the first print edition.
- Dan Kearney found a typo.
- Jeff Pickhardt found a broken link and a typo.
- Jörg Beyer found typos in the book and made many corrections in the docstrings of the accompanying code.
- Tommie Gannert sent a patch file with a number of corrections.

Contents

Chapter 1

Statistical thinking for programmers

This book is about turning data into knowledge. Data is cheap (at least relatively); knowledge is harder to come by.

I will present three related pieces:

Probability is the study of random events. Most people have an intuitive understanding of degrees of probability, which is why you can use words like "probably" and "unlikely" without special training, but we will talk about how to make quantitative claims about those degrees.

Statistics is the discipline of using data samples to support claims about populations. Most statistical analysis is based on probability, which is why these pieces are usually presented together.

Computation is a tool that is well-suited to quantitative analysis, and computers are commonly used to process statistics. Also, computational experiments are useful for exploring concepts in probability and statistics.

The thesis of this book is that if you know how to program, you can use that skill to help you understand probability and statistics. These topics are often presented from a mathematical perspective, and that approach works well for some people. But some important ideas in this area are hard to work with mathematically and relatively easy to approach computationally.

The rest of this chapter presents a case study motivated by a question I heard when my wife and I were expecting our first child: do first babies tend to arrive late?

1.1 Do first babies arrive late?

If you Google this question, you will find plenty of discussion. Some people claim it's true, others say it's a myth, and some people say it's the other way around: first babies come early.

In many of these discussions, people provide data to support their claims. I found many examples like these:

> "My two friends that have given birth recently to their first babies, BOTH went almost 2 weeks overdue before going into labour or being induced."
>
> "My first one came 2 weeks late and now I think the second one is going to come out two weeks early!!"
>
> "I don't think that can be true because my sister was my mother's first and she was early, as with many of my cousins."

Reports like these are called **anecdotal evidence** because they are based on data that is unpublished and usually personal. In casual conversation, there is nothing wrong with anecdotes, so I don't mean to pick on the people I quoted.

But we might want evidence that is more persuasive and an answer that is more reliable. By those standards, anecdotal evidence usually fails, because:

Small number of observations: If the gestation period is longer for first babies, the difference is probably small compared to the natural variation. In that case, we might have to compare a large number of pregnancies to be sure that a difference exists.

Selection bias: People who join a discussion of this question might be interested because their first babies were late. In that case the process of selecting data would bias the results.

Confirmation bias: People who believe the claim might be more likely to contribute examples that confirm it. People who doubt the claim are more likely to cite counterexamples.

Inaccuracy: Anecdotes are often personal stories, and often misremembered, misrepresented, repeated inaccurately, etc.

So how can we do better?

1.2 A statistical approach

To address the limitations of anecdotes, we will use the tools of statistics, which include:

Data collection: We will use data from a large national survey that was designed explicitly with the goal of generating statistically valid inferences about the U.S. population.

Descriptive statistics: We will generate statistics that summarize the data concisely, and evaluate different ways to visualize data.

Exploratory data analysis: We will look for patterns, differences, and other features that address the questions we are interested in. At the same time we will check for inconsistencies and identify limitations.

Hypothesis testing: Where we see apparent effects, like a difference between two groups, we will evaluate whether the effect is real, or whether it might have happened by chance.

Estimation: We will use data from a sample to estimate characteristics of the general population.

By performing these steps with care to avoid pitfalls, we can reach conclusions that are more justifiable and more likely to be correct.

1.3 The National Survey of Family Growth

Since 1973 the U.S. Centers for Disease Control and Prevention (CDC) have conducted the National Survey of Family Growth (NSFG), which is intended to gather "information on family life, marriage and divorce, pregnancy, infertility, use of contraception, and men's and women's health. The survey results are used ... to plan health services and health education programs, and to do statistical studies of families, fertility, and health."[1]

We will use data collected by this survey to investigate whether first babies tend to come late, and other questions. In order to use this data effectively, we have to understand the design of the study.

[1] See `http://cdc.gov/nchs/nsfg.htm`.

The NSFG is a **cross-sectional** study, which means that it captures a snapshot of a group at a point in time. The most common alternative is a **longitudinal** study, which observes a group repeatedly over a period of time.

The NSFG has been conducted seven times; each deployment is called a **cycle**. We will be using data from Cycle 6, which was conducted from January 2002 to March 2003.

The goal of the survey is to draw conclusions about a **population**; the target population of the NSFG is people in the United States aged 15-44.

The people who participate in a survey are called **respondents**; a group of respondents is called a **cohort**. In general, cross-sectional studies are meant to be **representative**, which means that every member of the target population has an equal chance of participating. Of course that ideal is hard to achieve in practice, but people who conduct surveys come as close as they can.

The NSFG is not representative; instead it is deliberately **oversampled**. The designers of the study recruited three groups—Hispanics, African-Americans and teenagers—at rates higher than their representation in the U.S. population. The reason for oversampling is to make sure that the number of respondents in each of these groups is large enough to draw valid statistical inferences.

Of course, the drawback of oversampling is that it is not as easy to draw conclusions about the general population based on statistics from the survey. We will come back to this point later.

Exercise 1.1 Although the NSFG has been conducted seven times, it is not a longitudinal study. Read the Wikipedia pages `http://wikipedia.org/wiki/Cross-sectional_study` and `http://wikipedia.org/wiki/Longitudinal_study` to make sure you understand why not.

Exercise 1.2 In this exercise, you will download data from the NSFG; we will use this data throughout the book.

1. Go to `http://thinkstats.com/nsfg.html`. Read the terms of use for this data and click "I accept these terms" (assuming that you do).

2. Download the files named `2002FemResp.dat.gz` and `2002FemPreg.dat.gz`. The first is the respondent file, which contains one line for each of the 7,643 female respondents. The second file contains one line for each pregnancy reported by a respondent.

3. Online documentation of the survey is at `http://www.icpsr.umich.edu/nsfg6`. Browse the sections in the left navigation bar to get a sense of what data are included. You can also read the questionnaires at `http://cdc.gov/nchs/data/nsfg/nsfg_2002_questionnaires.htm`.

4. The web page for this book provides code to process the data files from the NSFG. Download `http://thinkstats.com/survey.py` and run it in the same directory you put the data files in. It should read the data files and print the number of lines in each:

```
Number of respondents 7643
Number of pregnancies 13593
```

5. Browse the code to get a sense of what it does. The next section explains how it works.

1.4 Tables and records

The poet-philosopher Steve Martin once said:

> "Oeuf" means egg, "chapeau" means hat. It's like those French have a different word for everything.

Like the French, database programmers speak a slightly different language, and since we're working with a database we need to learn some vocabulary.

Each line in the respondents file contains information about one respondent. This information is called a **record**. The variables that make up a record are called **fields**. A collection of records is called a **table**.

If you read `survey.py` you will see class definitions for `Record`, which is an object that represents a record, and `Table`, which represents a table.

There are two subclasses of `Record`—`Respondent` and `Pregnancy`—which contain records from the respondent and pregnancy tables. For the time being, these classes are empty; in particular, there is no init method to initialize their attributes. Instead we will use `Table.MakeRecord` to convert a line of text into a `Record` object.

There are also two subclasses of `Table`: `Respondents` and `Pregnancies`. The init method in each class specifies the default name of the data file and the