Principles and Techniques of Data Science

By Sam Lau, Joey Gonzalez, and Deb Nolan.

This is the textbook for Data 100, the Principles and Techniques of Data Science course at UC Berkeley.

Data 100 is the upper-division, semester-long data science course that follows Data 8, the Foundations of Data Science. The reader's assumed background is detailed in the About This Book page.

To set up the textbook for local development, see the the setup guide.

About This Book

In this book, we will proceed as though the reader is comfortable with the knowledge presented in Data 8 or some equivalent. In particular, we will assume that the reader is familiar with the following topics (links to pages from the Data 8 textbook are given in parentheses).

- Tabular data manipulation: selection, filtering, grouping, joining (link)
- Basic probability concepts (link)
- Sampling, empirical distributions of statistics (link)
- Hypothesis testing using bootstrap resampling (link)
- Least squares regression and regression inference (link)
- Classification (link)

In addition, we assume that the reader has taken a course in computer programming in Python, such as CS61A or some equivalent. We will not explain Python syntax except in special cases.

Finally, we assume that the reader has basic familiarity with partial derivatives, gradients, vector algebra, and matrix algebra.

Notation

This book covers topics from multiple disciplines. Unfortunately, some of these disciplines use the same notation to describe different concepts. In order to prevent headaches, we have devised notation that may differ slightly from the notation used in your discipline.

A population parameter is denoted by . The model parameter that minimizes a specified loss function is denoted by . Typically, we desire

. We use the plain variable to denote a model parameter that does not minimize a particular loss function. For example, we may arbitrarily set in order to calculate a model's loss at that choice of

. When using gradient descent to minimize a loss function, we use to represent the intermediate values of .

We will always use bold lowercase letters for vectors. For example, we represent a vector of population parameters using $
\boldsymbol{\theta^*} = [\theta^**1, \theta^**2, \ldots, \theta^**n]*

*\boldsymbol{\hat{\theta}} = [\hat{\theta*1}, \hat{\theta*2}, \ldots, \hat{\theta*n}
] $.

We will always use bold uppercase letters for matrices. For example, we commonly represent a data matrix using .

We will always use non-bolded uppercase letters for random variables, such as or .

When discussing the bootstrap, we use to denote the population parameter, to denote the sample test statistic, and to denote a bootstrapped test statistic.

The Data Science Lifecycle¶

In data science, we use large and diverse data sets to make conclusions about the world. In this book we discuss principles and techniques of data science through the dual lens of computational and inferential thinking. Practically speaking, this involves the following process:

1. Formulating a question or problem
2. Acquiring and cleaning data
3. Conducting exploratory data analysis
4. Using prediction and inference to draw conclusions

It is quite common for more questions and problems to emerge after the last step of this process, and we can thus repeatedly engage in this procedure to discover new characteristics of our world. This positive feedback loop is so central to our work that we call it the **data science lifecycle**.

If the data science lifecycle were as easy to conduct as it is to state, there would be no need for textbooks about the the subject. Fortunately, each of the steps in the lifecycle contain numerous challenges that reveal powerful and often surprising insights that form the foundation of making thoughtful decisions using data.

As in Data 8, we will begin with an example.

The Students of Data 100¶

The data science lifecycle involves the following general steps:

1. **Question/Problem Formulation:**
    1. What do we want to know or what problems are we trying to solve?
    2. What are our hypotheses?
    3. What are our metrics of success?

2. **Data Acquisition and Cleaning:**
    1. What data do we have and what data do we need?
    2. How will we collect more data?
    3. How do we organize the data for analysis?

3. **Exploratory Data Analysis:**
    1. Do we already have relevant data?
    2. What are the biases, anomalies, or other issues with the data?
    3. How do we transform the data to enable effective analysis?

4. **Prediction and Inference:**
    1. What does the data say about the world?
    2. Does it answer our questions or accurately solve the problem?
    3. How robust are our conclusions?

We now demonstrate this process applied to a dataset of student first names from a previous offering of Data 100. In this chapter, we proceed quickly in order to give the reader a general sense of a complete iteration through the lifecycle. In later chapters, we expand on each step in this process to develop a repertoire of skills and principles.

Question Formulation¶

We would like to figure out if the student first names give us additional information about the students themselves. Although this is a vague question to ask, it is enough to get us working with our data and we can make the question more precise as we go.

Data Acquisition and Cleaning¶

Let's begin by looking at our data, the roster of student first names that we've downloaded from a previous offering of Data 100.

Don't worry if you don't understand the code for now; we introduce the libraries in more depth soon. Instead, focus on the process and the charts that we create.

```
import pandas as pd

students = pd.read_csv('roster.csv')
students
```

	Name	Role
0	Keeley	Student
1	John	Student

2	BRYAN	Student
...	...	...
276	Ernesto	Waitlist Student
277	Athan	Waitlist Student
278	Michael	Waitlist Student

279 rows × 2 columns

We can quickly see that there are some quirks in the data. For example, one of the student's names is all uppercase letters. In addition, it is not obvious what the Role column is for.

In Data 100, we will study how to identify anomalies in data and apply corrections. The differences in capitalization will cause our programs to think that `'BRYAN'` and `'Bryan'` are different names when they are identical for our purposes. Let's convert all names to lower case to avoid this.

```
students['Name'] = students['Name'].str.lower()
students
```

	Name	Role
0	keeley	Student
1	john	Student
2	bryan	Student
...	...	...
276	ernesto	Waitlist Student
277	athan	Waitlist Student
278	michael	Waitlist Student

279 rows × 2 columns

Now that our data are in a more useful format, we proceed to exploratory data analysis.

Exploratory Data Analysis¶

The term Exploratory Data Analysis (EDA for short) refers to the process of discovering traits about our data that inform future analysis.

Here's the `students` table from the previous page:

```
students
```

	Name	Role
0	keeley	Student
1	john	Student
2	bryan	Student
...	...	...
276	ernesto	Waitlist Student
277	athan	Waitlist Student
278	michael	Waitlist Student

279 rows × 2 columns

We are left with a number of questions. How many students are in this roster? What does the `Role` column mean? We conduct EDA in order to understand our data more thoroughly.

Oftentimes, we explore the data by repeatedly posing questions as we uncover more information.

How many students are in our dataset?

```
print("There are", len(students), "students on the roster.")
```

```
There are 279 students on the roster.
```

A natural follow-up question: does this dataset contain the complete list of students? In this case, this table contains all students in one semester's offering of Data 100.

What is the meaning of the `Role` field?

We often example the field's data in order to understand the field itself.

```
students['Role'].value_counts().to_frame()
```

	Role
Student	237
Waitlist Student	42

We can see here that our data contain not only students enrolled in the class at the time but also the students on the waitlist. The `Role` column tells us whether each student is enrolled.

What about the names? How can we summarize this field?

In Data 100 we will work with many different kinds of data, including numerical, categorical, and text data. Each type of data has its own set of tools and techniques.

A quick way to start understanding the names is to examine the lengths of the names.

```
sns.distplot(students['Name'].str.len(),
             rug=True,
             bins=np.arange(12),
             axlabel="Number of Characters")
plt.xlim(0, 12)
plt.xticks(np.arange(12))
plt.ylabel('Proportion per character');
```

This visualization shows us that most names are between 3 and 9 characters long. This gives us a chance to check whether our data seem reasonable — if there were many names that were 1 character long we'd have good reason to re-examine our data.

What's in a Name?¶

Although this dataset is rather simple, we will soon see that first names alone can reveal quite a bit about our group of students.