Text as Data

An introduction to quantitative analysis and reproducible research in R

Jerid C. Francom

April 30, 2019 (latest version)

Contents

Welcome

This is the website for *Text as Data: An introduction to quantitative analysis and reproducible research in R*

Author

Jerid Francom is Associate Professor of Spanish and Linguistics at Wake Forest University. His research interests are focused around quantitative approaches to language variation.

Preface

0.1 About

0.1.1 Aims

In recent years there has been a growing buzz around the term 'Data Science', and related terms, data analytics, data mining, *etc.* In a nutshell data science is the process by which an investigator leverages statistical methods and computational power to uncover insight from large datasets. Driven in large part by the increase in computing power available to the average individual and the increasing amount of electronic data that is now available through the internet, interest in data science has expanded to virtually all fields in academia and areas in the public sector. Data scientists are in high demand and this trend is expected to continue into the foreseeable future, which means that undergraduate and post-graduate students will be increasingly seeking out resources and training in the area. This book is an introduction to the fundamental concepts and practical programming skills from Data Science that are increasingly employed in a variety of language-centered fields and sub-fields. It is geared towards advanced undergraduates and graduate students of linguistics and related fields. As quantitative research skills are quickly becoming a core aspect of many language programs, this textbook aims to provide a fundamental understanding of theoretical concepts, programming skills, and statistical methods for doing quantitative linguistic analysis. Currently no textbook integrates these components (theoretical concepts, programming skills, and statistical methods) into one resource. The goal, then, is to integrate these principles into one textbook and to teach core concepts and methods from key language science fields, corpus and computational linguistics, through hands-on application of core concepts through R programming. Through these skills we will explore topics/ and replicate previous research in a variety of areas in language research (psycholinguistics, sociolinguistics, translation studies, *etc.*) using common methods and authentic data sources.

0.1.2 Approach

In linguistics (my primary area of expertise) as well in other fields, doing good data science and training good data scientists requires more than a computer and knowledge of statistics. It is an approach that relies heavily on intimate domain-specific knowledge and strong technical skills. Textbook resources for data science, in general and quantitative linguistic research, in particular, however, often treat data science concepts (research question design, sampling practices, statistical concepts, *etc.*) separate from code (programming implementation). In some cases there may be good reasons to keep concepts and code separate, but in practice concepts and code are inseparable. The practicing data scientist works fluidly within concepts and code implementation.

Teaching concepts or coding in an agnostic way breaks from common practice and tends to obscure the close relationship between each of these components in the data scientist's workflow. On the other hand, teaching concepts and code together enhances learning. Turning to a practical teaching example, say we are learning in class about a peculiar characteristic of word frequency distributions, the Zipf distribution. This distribution, in essence, predicts that the most frequent words in a language sample will comprise a relatively small number of the unique words; however, the combined frequency of these relatively small number of words makes up a large portion of all the words used in the sample. Now, I can describe the distribution, as I just did, but seeing the distribution is better (Figure 1).

Even better than seeing a graphic is knowing how to manipulate the language data and generate the graphic. Below is the code in the programming language R that created the previous graphic. [1]

```
scan(file = "data/Brown/browncorpus.txt", what = character()) %>% # read the corpus data
  tolower() %>% # change text to lowercase
  table() %>%  # create a word frequency table
  sort(decreasing = TRUE) %>% # sort the word frequency table
  head(25) %>% # select the top 100 words
  plot() # plot the top 25 words
```

This is a practical application of text analytics. With this knowledge in hand students gain a more enhanced appreciation of the Zipf distribution as well as tangible skills that can be re-purposed and used in subsequent text processing tasks.

0.1.3 Target audience

- Who is this book for?

[1] The text after the # on each line is a human-facing description of the command on that line.

0.2 R and RStudio

0.2.1 Why program? Why R?

... need to add good intro.

- What is programming? What (dis)advantages are there to programming?
- What is R? Why use R?
- How to learn R ... pRactice, pRactice, pRactice.

There will be many opportunities to see and read descriptions of R code in this textbook. Programming, however, is a skill that can only be acquired through hands-on practice.

0.2.1.1 Interactive learning

Swirl

To this end I have included a set of interactive R programming tutorials for you to complete. The tutorials are created with an R package called `swirl`. To access these tutorials you will need to open R in an RStudio session[2]. In RStudio type and run the following code:

```
install.packages("swirl")  # install the `swirl` package
library(swirl)  # load the `swirl` package into the current session
install_course_github("francojc", "dsfl-swirl")  # install the course modules
```

After installing `swirl` and the course tutorials you will be able access the tutorials by simply entering:

```
swirl()  #
```

This command will bring up a dialog in the R console that will allow you to pick from a set of tutorials available. Your progress will be logged and at the end of each module you will be given the opportunity to submit your work to your instructor.

Recipes

Each chapter will include a programming demonstration, or what will be called a 'Recipe' that aims to model and describe in detail the programming strategies that address the main concepts of the chapter.

Activities

[2] As part of this course R and RStudio have been set up for you. You can access the RStudio software here[3]. Note, you will need to be on campus to access this site or be running the WFU VPN[4]. For more on how to set up R and RStudio on your own computer see the R and RStudio chapter in the Appendix of this textbook

To build on your experience with coding each chapter will include a set of programming activities for the student to complete. These activities integrate the techniques learned in the interactive Swirl tutorials and the Recipe demonstration to tackle a practical subtask included in the data scientists workflow.

- Resources
  - Online resources: stackoverflow, etc.
  - Guided resources: swirl, learnr, DataCamp
  - Data resources

0.2.1.2 Text conventions

This book is about the concepts for understanding and the techniques for doing quantitative language science. Therefore there will be an intermingling of prose and code presented. As such, an attempt to establish consistent conventions throughout the text has been made to signal reader's attention as appropriate. As we explore concepts, R code itself will be incorporated into the text. This may be a unique textbook compared to others you have seen. It has been created using R itself –specifically using an R language package called `bookdown` (Xie, 2018). This R package makes it possible to write, execute ('run'), and display code and results within the text.

For example, the following text block shows actual R code and the results that are generated when running this code. Note that the hashtag `#` signals a **code comment**. The code follows within the same text block and a subsequent text block displays the output of the code.

```
# Add 1 plus 1
1 + 1
```

```
## [1] 2
```

Inline code will be used when code blocks are short and the results are not needed for display. For example, the same code as above will sometimes appear as `1 + 1`.

In terms of prose, key concepts will be signaled using ***bold italics***. Terms that appear in this typeface will also appear in the [glossary] at the end of the text. Furthermore, there are four pose text blocks that will be used to signal the reader's attention: *key points*, *notes*, *tips*, and *warnings*.

Key points summarize the main points to be covered in a chapter or a subsection of the text.

In this chapter you will learn:

- the goals of this textbook

- the reasoning for using the R programming language
- important text conventions employed in this textbook

Notes provide a bit more information on the topic or where to find more information.

R is more than a powerful statistical programming language, it also can be used to perform all the necessary steps in a data science project; including reporting. A relatively new addition to the reporting capabilities of R is the `bookdown` package (this textbook was created using this very package). You can find out more here[5].

Tips are used to signal helpful hints that might otherwise be overlooked.

During a the course of an exploratory work session, many R objects are often created to test ideas. At some point inspecting the workspace becomes difficult due to the number of objects displayed using `ls()`.

To remove all objects from the workspace, use `rm(list = ls())`.

Errors will be an inevitable part of learning, but some errors can be avoided. The text will used the warning text block to highlight typical pitfalls and errors.

Hello world!
This is a warning.

Although this is not intended to be a in-depth introduction to statistical techniques, mathematical formulas will be included in the text. These formulas will appear either inline $1 + 1 = 2$ or as figures:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \ \hat{P}(c) \prod_i \hat{P}(w_i|c) \tag{1}$$

0.2.2 RStudio work environment

Introduction to the popular IDE for R

Setup

0.3 Acknowledgements

Chapter 1

Introduction

1.1 Background

1.2 Structure

Bibliography

Xie, Y. (2018). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.9.