STAT 8670 - Computational Methods in Statistics

Chi-Kuang Yeh

2025-07-04

Table of contents

Preface

Description

Topics included are optimization, numerical integration, bootstrapping, cross-validation and Jackknife, density estimation, smoothing, and use of the statistical computer package of S-plus/R.

Prerequisites

Math 4752/6752, and the ability to program in a high-level language.

Instructor

Chi-Kuang Yeh, I am a postdoctoral scholar at the Department of Statistics and Actuarial Science, McGill University.

- Office: M3–3102 Desk 10, but I hold office hour at M3 - 2101 Desk 1, 9:30 – 10:30 on Tuesday.
- Email: chi-kuang.yeh@mail.mcgill.ca

Office Hour

[Online link will be provided later]

Midterms

- ☐ Midterm 1: Date and topics TBA
- ☐ Midterm 2: Date and topics TBA
- ☐ Midterm 3: Date and topics TBA

Chapters and Associated Lectures

Those chapters are based on the lecture notes. This part will be updated frequently.

Chapter	Lecture Covered
1. TBA	L1

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.

Part I

Appendix

Appendix: Why R?

A.1 R and RStudio

For conducting analyses with data sets of hundreds to thousands of observations, calculating by hand is not feasible and you will need a statistical software. **R** is one of those. **R** can also be thought of as a high-level programming language. In fact, **R** is one of the top languages to be used by data analysts and data scientists. There are a lot of analysis packages in R that are currently developed and maintained by researchers around the world to deal with different data problems. And most importantly, **R** is free. In this book, we will learn how to use **R** to conduct basic statistical analyses.

A.2 RStudio Layout

RStudio consists of several panes: - **Source**: Where you write scripts and markdown documents. - **Console**: Where you type and execute R commands. - **Environment/History**: Shows your variables and command history. - **Files/Plots/Packages/Help/Viewer**: For file management, viewing plots, managing packages, accessing help, and viewing web content.

A.3 R Scripts

R scripts are plain text files containing R code. You can create a new script in RStudio by clicking `File > New File > R Script`.

A.4 R Help

Use `?function_name` or `help(function_name)` to access help for any R function. For example:

```
?mean
help(mean)
```

A.5 R Packages

Packages extend R's functionality. Install a package with:

```
install.packages("package_name")
```

Load a package with:

```
library(package_name)
```

A.6 R Markdown

R Markdown allows you to combine text, code, and output in a single document. Create a new R Markdown file in RStudio via `File > New File > R Markdown...`.

A.7 Vectors

Vectors are the most basic data structure in R.

```
x <- c(1, 2, 3, 4, 5)
x
```

[1] 1 2 3 4 5

```
You can perform operations on vectors:

```r
x * 2

## [1]  2  4  6  8 10
```

A.8 Data Sets

Data frames are used for storing data tables. Create a data frame:
```

```r
df <- data.frame(Name = c("Alice", "Bob"), Score = c(90, 85))
df
```

```
##    Name Score
## 1 Alice    90
## 2   Bob    85
```

You can import data from files using `read.csv()` or `read.table()`.

This appendix is adapted from Why R?.