STAT 8670 - Computational Methods in Statistics

Chi-Kuang Yeh

2025-07-05

Table of contents

Preface **4**
Description . . . 4
Prerequisites . . . 4
Instructor . . . 4
Office Hour . . . 4
Assignment . . . 4
Midterm . . . 5
Topics and Corresponding Lectures . . . 5
Recommended Textbooks . . . 5
Side Readings . . . 5

1 Data Structure and R Programming **6**
1.1 Data type . . . 6
1.1.1 To change data type . . . 7
1.2 Operators . . . 7
1.2.1 Comparison Operator . . . 7
1.2.2 Logical Operator . . . 7
1.3 Indexing . . . 7
1.4 Naming . . . 7
1.5 Arrray and Matrix . . . 7
1.6 Key and Value Pair . . . 7
1.7 Data Frame . . . 7
1.8 Tidyverse . . . 7

2 Optimization **8**
2.1 Speed comparison . . . 8

3 Resampling, Jackknife and Bootstrap **10**
3.1 Introduction . . . 10
3.2 Jackknife . . . 10
3.3 Bootstrap . . . 10
3.4 Applications . . . 10

References **11**

I Appendix **12**

4 Appendix: Introduction to R? **13**
4.1 R . . . 13
4.2 IDE . . . 13
4.2.1 Rstudio . . . 13
4.2.2 Visual Studio Code (VS Code) . . . 13
4.2.3 Positron . . . 14
4.3 RStudio Layout . . . 14
4.4 R Scripts . . . 14
4.5 R Help . . . 14
4.6 R Packages . . . 14
4.7 R Markdown . . . 15
4.8 Vectors . . . 15
4.9 Data Sets . . . 15

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 – Mathematical Statistics II, 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: 1216 Burnside Hall.
- Email: chi-kuang.yeh@mail.mcgill.ca.

Office Hour

[By appointment and a online link will be provided later]

Assignment

- ☐ Assignment 1: Date and topics TBA

Midterm

☐ Midterm 1: Date and topics TBA

Topics and Corresponding Lectures

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

Topic	Lecture Covered
Optimization	TBA
Numerical integration	TBA
Jackknife	TBA
Bootstrap	TBA
Cross-validation	TBA
Smoothing	TBA
Density estimation	TBA
Monte Carlo Methods	TBA

Recommended Textbooks

- Givens, G.H. and Hoeting, J.A. (2012). *Computational Statistics*. Wiley, New York.
- Rizzo, M.L. (2007) *Statistical Computing with R*. CRC Press, Roca Baton.
- Hothorn, T. and Everitt, B.S. (2006). *A Handbook of Statistical Analyses Using R*. CRC Press, Boca Raton.

Side Readings

- Wickham, H., Çetinkaya-Rundel, M. and Grolemund, G. (2023). *R for Data Science*. O'Reilly.

1 Data Structure and R Programming

Data types, operators, variables

Two basic types of objects: (1) data & (2) functions

- Data: can be a number, a vector, a matrix, a dataframe, a list or other datatypes
- Function: a function is a set of instructions that takes input, processes it, and returns output. Functions can be built-in or user-defined.

1.1 Data type

- Boolean/Logical: Yes or No, Head or Tail, True or False
- Integers: Whole numbers $\mathbb{Z}$, e.g., 1, 2, 3, -1, -2, -3
- Characters: Text strings, e.g., "Hello", "World"
- Floats:
- Missing data

```
day <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
weather <- c("Raining", "Sunny", NA, "Windy", "Snowing")
data.frame(rbind(day, weather))
```

```
             X1      X2        X3       X4      X5
day      Monday Tuesday Wednesday Thursday  Friday
weather Raining   Sunny      <NA>    Windy Snowing
```

- Other more complex type

1.1.1 To change data type

1.2 Operators

- Unary: One argument
- Binary: Two arguments

1.2.1 Comparison Operator

1.2.2 Logical Operator

1.3 Indexing

1.4 Naming

1.5 Arrray and Matrix

1.6 Key and Value Pair

1.7 Data Frame

1.8 Tidyverse

Some of the materials are adapted from CMU Stat36-350.

2 Optimization

The optimization plays an important role in statistical computing, especially in the context of maximum likelihood estimation (MLE) and other statistical inference methods. This chapter will cover various optimization techniques used in statistical computing.

For instance, for a linear regression

$$y = X\beta + \varepsilon.$$

From regression class, we know that the (ordinary) least-squares estimation (OLE) for β is given by $\widehat{\beta} = (X^\top X)^{-1} X^\top y$. It is convenient as the solution is in the **closed-form**! However, in the most case, the closed-form solutions will not be available.

For GLMs or non-linear regression, we need to do this **iteratively**!

2.1 Speed comparison

```
set.seed(2017-07-13)
X <- matrix(rnorm(5000 * 100), 5000, 100)
y <- rnorm(5000)
library(microbenchmark)
microbenchmark(solve(t(X) %*% X) %*% t(X) %*% y)
```

```
Warning in microbenchmark(solve(t(X) %*% X) %*% t(X) %*% y): less accurate
nanosecond times to avoid potential integer overflows
```

```
Unit: milliseconds
                           expr      min       lq     mean   median       uq
 solve(t(X) %*% X) %*% t(X) %*% y 29.43037 29.99257 30.73073 30.29705 31.27164
      max neval
 37.86305   100
```

```
microbenchmark(solve(t(X) %*% X) %*% t(X) %*% y,
               solve(crossprod(X), crossprod(X, y)))
```

```
Unit: milliseconds
                                 expr      min       lq     mean   median
     solve(t(X) %*% X) %*% t(X) %*% y 29.09991 29.61127 30.53744 29.78757
 solve(crossprod(X), crossprod(X, y)) 24.96855 25.04071 25.37561 25.08290
       uq      max neval
 31.08536 52.62608   100
 25.29003 40.07361   100
```

Examples are borrowed from the following sources:

- Peng, R.D. Advanced Statistical Computing.

3 Resampling, Jackknife and Bootstrap

3.1 Introduction

This chapter covers resampling methods including the jackknife and bootstrap techniques.

3.2 Jackknife

The jackknife is a resampling technique used to estimate the bias and variance of a statistic.

3.3 Bootstrap

The bootstrap is a resampling method that allows estimation of the sampling distribution of almost any statistic using random sampling methods.

3.4 Applications

These methods are widely used in statistical inference and have applications in various fields.

References

Part I

Appendix

4 Appendix: Introduction to R?

4.1 R

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. Most importantly, **R** is free! In this section, we will learn how to use **R** to conduct basic statistical analyses.

4.2 IDE

4.2.1 Rstudio

RStudio is an integrated development environment (IDE) designed specifically for working with the **R** programming language. It provides a user-friendly interface that includes a source editor, console, environment pane, and tools for plotting, debugging, version control, and package management. RStudio supports both R and Python and is widely used for data analysis, statistical modeling, and reproducible research. It also integrates seamlessly with tools like R Markdown, Shiny, and Quarto, making it popular among data scientists, statisticians, and educators.

4.2.2 Visual Studio Code (VS Code)

VS Code is a versatile code editor that supports multiple programming languages, including R. With the R extension for VS Code, users can write and execute R code, access R's console, and utilize features like syntax highlighting, code completion, and debugging. While not as specialized as RStudio for R development, VS Code offers a lightweight alternative with extensive customization options and support for various programming tasks.

4.2.3 Positron

Positron IDE is the next-generation integrated development environment developed by Posit, the company behind RStudio. Designed to be a modern, extensible, and language-agnostic IDE, Positron builds on the strengths of RStudio while supporting a broader range of languages and workflows, including R, Python, and Quarto.

4.3 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.

4.4 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`.

4.5 R Help

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

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

4.6 R Packages

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

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

Load a package with:

```
library(package_name)
```

4.7 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...`.

Recently, the posit team has developed a new version of the R Markdown called quarto document, with the file extension `.qmd`. It is still under rapid development.

4.8 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:

```
x * 2
```

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

4.9 Data Sets

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

```
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?.