

# **STAT 8670 - Computational Methods in Statistics**

Chi-Kuang Yeh

2025-07-04

# Table of contents

<b>Preface</b>	<b>3</b>
Description . . . . .	3
Prerequisites . . . . .	3
Instructor . . . . .	3
Office Hour . . . . .	3
Assignment . . . . .	3
Midterm . . . . .	4
Topics and Corresponding Lectures . . . . .	4
<b>1 Introduction</b>	<b>5</b>
<b>2 Summary</b>	<b>6</b>
<b>References</b>	<b>7</b>
 <b>I Appendix</b>	 <b>8</b>
<b>Appendix: Introduction to R?</b>	<b>9</b>
R . . . . .	9
IDE . . . . .	9
RStudio Layout . . . . .	10
R Scripts . . . . .	10
R Help . . . . .	10
R Packages . . . . .	10
R Markdown . . . . .	11
Vectors . . . . .	11
Data Sets . . . . .	11

# 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](mailto: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

# 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: Introduction to R?

## 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.

## IDE

### 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.

### 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.

## 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.

## 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.

## 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**.

## R Help

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

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

## R Packages

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

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

Load a package with:

```
library(package_name)
```

## 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...**

## 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
```

## 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?](#).