Chapter 1 Setup

1.1 What you need

To work through this book, you will need:

- Access to R/RStudio
- An internet connection (for installing packages)
- A basic familiarity with saving and opening files on your computer

1.2 R and RStudio (quick overview)

R is the language that runs your analysis. RStudio is the interface you use to write code, run code, view plots, and manage files. You have three options for using R/RStudio:

1. Use OSC Classroom On Demand
   Using the OSC Classroom on Demand is by far the easiest way to use R/RStudio for this course. All packages (see below for more info on packages) should already be loaded and tested to ensure compatability.
2. Install on your own machine
   This option involves two steps. First, install "base" R from the Comprehensive R Archive Network (CRAN): https://cran.r-project.org/. Second, install RStudio: https://posit.co/download/rstudio-desktop/#download.
3. Use a computer in an on-campus computer lab.
   This option should be considered a last resort. I cannot guarantee all packages will be able to be installed on university machines, and you would need to install the packages every time you use the machine.

1.3 Navigating RStudio

RStudio's interface is typically organized into four main panes in a default layout: the Source Editor, the R Console, the Environment/History, and the Files/Plots/Packages/Help/Viewer pane.

- **Source** (*usually* top-left)
  This pane is where you write, edit, and save your R scripts or R Markdown files. It functions like a text editor with features like syntax highlighting and code completion. Code is not executed immediately upon typing. You must explicitly send lines or sections of code to the Console (commonly using the "Run" button or keyboard shortcuts like Ctrl+Enter on Windows/Linux or Cmd+Enter on Mac) to execute them. This pane only appears when you have a file open.
- **Console** (*usually* bottom-left)
  This pane is where commands are actually run and text-based output, warnings, or error messages are displayed. You can type R commands directly into the console for immediate execution. The console shows a `>` prompt when it is ready to accept a new command. Commands typed directly here are not saved automatically, which is why writing in the Source editor is recommended for complete analyses.
- **Environments** (*usually* top-right)
  This pane helps manage your current R session and the objects within it. It may contain several tabs, including:
  - Environment Tab: Displays all the active objects (e.g., data frames, variables, functions, vectors) you've created or loaded during your current R session. You can inspect brief summaries of these objects here.
  - History Tab: Provides a log of all the commands that have been successfully executed in the console, which can be useful for reviewing past work.
  - Other Tabs: May also include tabs for Connections, Build, or Tutorials, depending on your RStudio configuration.
- **Output** (*usually* top-right)
  This multi-purpose pane provides access to various tools for project management and output viewing Files: A file browser for navigating your computer's directory structure and managing files within your current working directory. Plots: Where all the visualizations and graphs you create with R code will be displayed. It also may contain several tab, including:
  - Packages: Lists all installed R packages and allows you to install new ones or load them into your current session.

- Help: The built-in documentation browser for R functions and packages. You can search for help directly here or by using a command like `?function_name` in the console.
- Viewer: Used for displaying local web content, such as interactive HTML outputs, Shiny apps, or interactive plots generated by certain packages.
- Plots: Use for displaying static plots generated by code.

The layout of these panes can be customized via the *Tools -> Global Options -> Pane Layout* menu. For more information on the panes in RStudio, please visit https://docs.posit.co/ide/user/ide/guide/ui/ui-panes.html.

1.4 Installing and loading packages

Packages extend R. You typically:

1. Install a package once per computer
2. Load the package each time you start a new R session

1.4.1 Install packages (one-time)

Packages are installed using the `install.packages("package_name")` function. For example, run the code below in the Console. If you see messages scrolling by, that is normal.

```
install.packages("tidyverse")
install.packages("devtools")
```

1.4.2 Load packages (each session)

Packages are loaded using the `library(package_name)` function. You may see messages and/or warnings scroll by when loading package, which is normal behavior.

```
library(tidyverse)
library(devtools)
```

1.5 Installing the MKT4320BGSU package

The MKT4320BGSU package contains the functions and datasets used throughout this course.

1.5.1 Install from GitHub

```
devtools::install_github("jdmeyer73/MKT4320BGSU")
```

1.5.2 Load the package

```
library(MKT4320BGSU)
```

1.5.3 Verify everything works

If the chunks below run without errors, you are set.

```
data("directmktg")
head(directmktg)
```

```
# A tibble: 6 × 5
    userid   age buy   gender salary
     <dbl> <dbl> <fct> <fct>   <dbl>
1 15624510    19 No    Male       19
2 15810944    35 No    Male       20
3 15668575    26 No    Female     43
4 15603246    27 No    Female     57
5 15804002    19 No    Male       76
6 15728773    27 No    Male       58
```

1.6 Getting help when you're stuck

These are the most useful help tools in R:

- `?function_name` opens help for a function
- `help.search("keyword")` searches help files
- If an error happens, read the last line carefully (it often tells you what to fix)

Examples:

```
?mean
help.search("logistic regression")
```

1.7 How you'll use this book

Most weeks, your workflow will look like this:

1. Read the relevant sections of the book
2. Open the lab file for the week
3. Run the example code
4. Modify the code to answer lab questions
5. Interpret results and write conclusions

Errors are normal—debugging is part of learning analytics.

1.8 What's next

In the next chapter, we'll cover basic R fundamentals you'll use constantly, including:

- objects and assignment
- vectors and data frames
- common functions
- importing data and basic data manipulation