WORKBOOK

ONLINE • GUIDE • RSTUDIO •

RSTUDIO FOR BEGINNERS

A Quick Start Guide to Creating your First Graph for Mac Users

THE STATISTICAL CONSULTING FIRM OF SOUTH CAROLINA

RStudio for Beginners: A Quick Start Guide to Creating your First Graph *for Mac Users*

Table of Contents

Introduction

This workbook is designed for beginners to RStudio that want to quickly learn how to get started using built-in R datasets. For this purpose, examples are intentionally more simplistic in nature. For more advanced topics as you progress in your knowledge of R and RStudio be sure to subscribe to our webpage at https://statisticalconsultingfirmofsc.com.

Module 1: Downloading R and RStudio

1.1: Module Goals

In this module, the goal is to successfully download R AND RStudio. Yes. You read that right - you need to install both!

Why do you need both? You need both, because R is the actual program that you will be working with through the RStudio application. The two, in essence, form a pipeline where R is the engine under the hood of RStudio.

To install each the process will be slightly different based on the computing system being used. For the purposes of this guide, a Mac computing system is used.

By the end of this module, you should have:

- R installed on your Mac
- RStudio installed on your Mac

1.2: Downloading R

1. Go to https://www.r-project.org/
2. Select "Download R" under the Getting Started header (see Figure 1).

Figure 1: R Homepage (R Core Team, 2019)

3. Choose the mirror closest to your physical location on the Cran Mirrors Page (e.g., if you're in North Carolina, use the Duke University option)
4. Select the "Download R for macOS" option in the "Download and Install R" section (see Figure 2).

Figure 2: R CRAN Page (R Core Team, 2019)

5. Choose the link for the latest release corresponding to your Mac's chip* (see Figure 3).

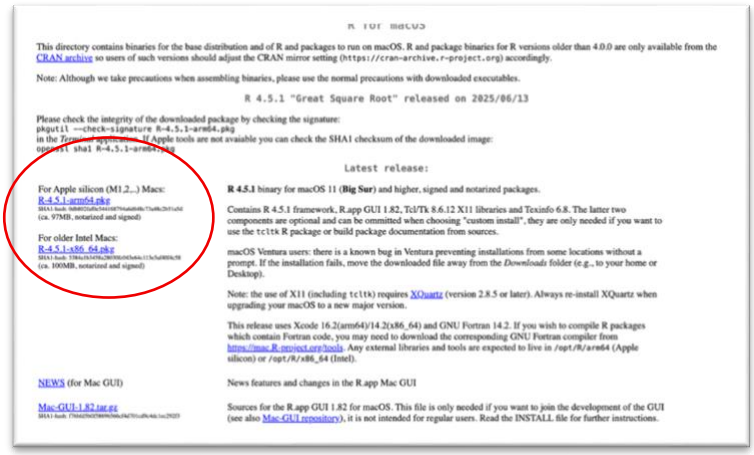

Figure 3: R for MacOS page (R Core Team, 2019)

5.1. *If you're unsure which chip you have, right-click the Apple icon located on the top left of your computer's toolbar

5.2.Right-click "About this Mac" on the drop-down menu.

- If beside chip, you have M[*anything*], use the link under "For Apple Silicon (M1,2,..) Macs".
- If you have an Intel chip, instead, use the link under "For Older Intel Macs".

6. Open the download from your "Downloads" folder.
7. Follow the prompts from the installer to complete installation.
8. Once installed, you can access R from the applications menu.

1.3: Downloading RStudio

1. Go to https://posit.co/downloads/ .
2. Select "Download R" under the "Download" header (see Figure 4).

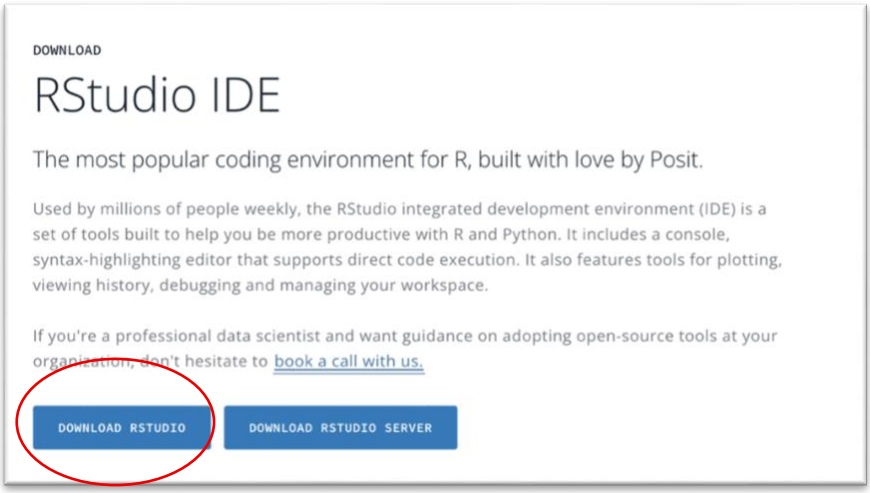

Figure 4: RStudio Homepage (Posit team, 2025)

3. Select the download option under number 2. If you have a previous version of Mac operating system (one other than listed), use the highlighted "download a previous version" to find the version that corresponds to your operating system (see Figure 5). *Note this may be a drop-down menu on some Macs – you'd still select the version that matches your Mac).*

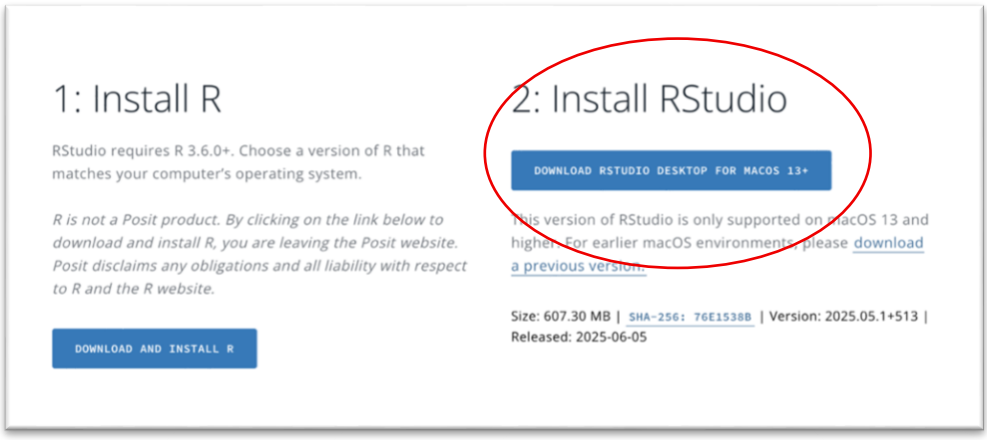

Figure 5: RStudio Install Options Screengrab (Posit team, 2025)

4. Once the download finishes, double-click to open it. (Look inside your "Downloads" folder).
5. On the popup, move the RStudio icon into the Applications folder (this will allow you to find RStudio on the Applications lists).

6. Once done installing, an RStudio icon will be found on the applications menu like that shown in Figure 6.

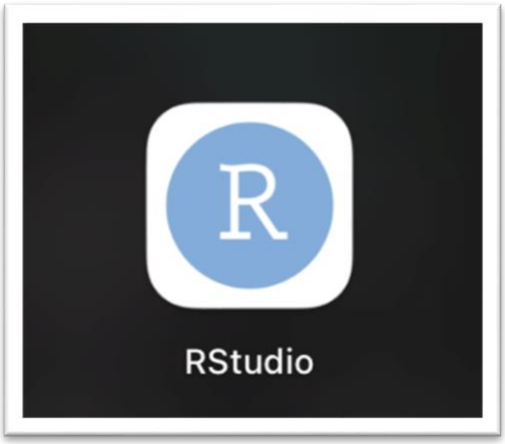

Figure 6: RStudio Icon (Posit team, 2025)

1.4: Module Recap

⚠To be able to move further you'll need to ensure both R and RStudio have been installed correctly. In working with RStudio remember:

- R is the engine behind RStudio, so you'll need both installed.
- RStudio is an integrated development environment (IDE) designed to make running R code more efficient and user-friendly.
- If you want to open the application, double-click on its icon on your computer.

In the next module we'll review the windowpanes you'll see in RStudio when you first install and how/when to use each.

Module 2: Understanding RStudio's Default Panes

2.1: Module Goals

In this module the goal is to describe the default panes you'll encounter on RStudio ***upon initial downloading*** and to identify when each pane will be useful.

By the end of this module, you should be able to

- Identify the 3 default panes shown on RStudio when initially downloaded
- Understand the purpose of the 3 default panes

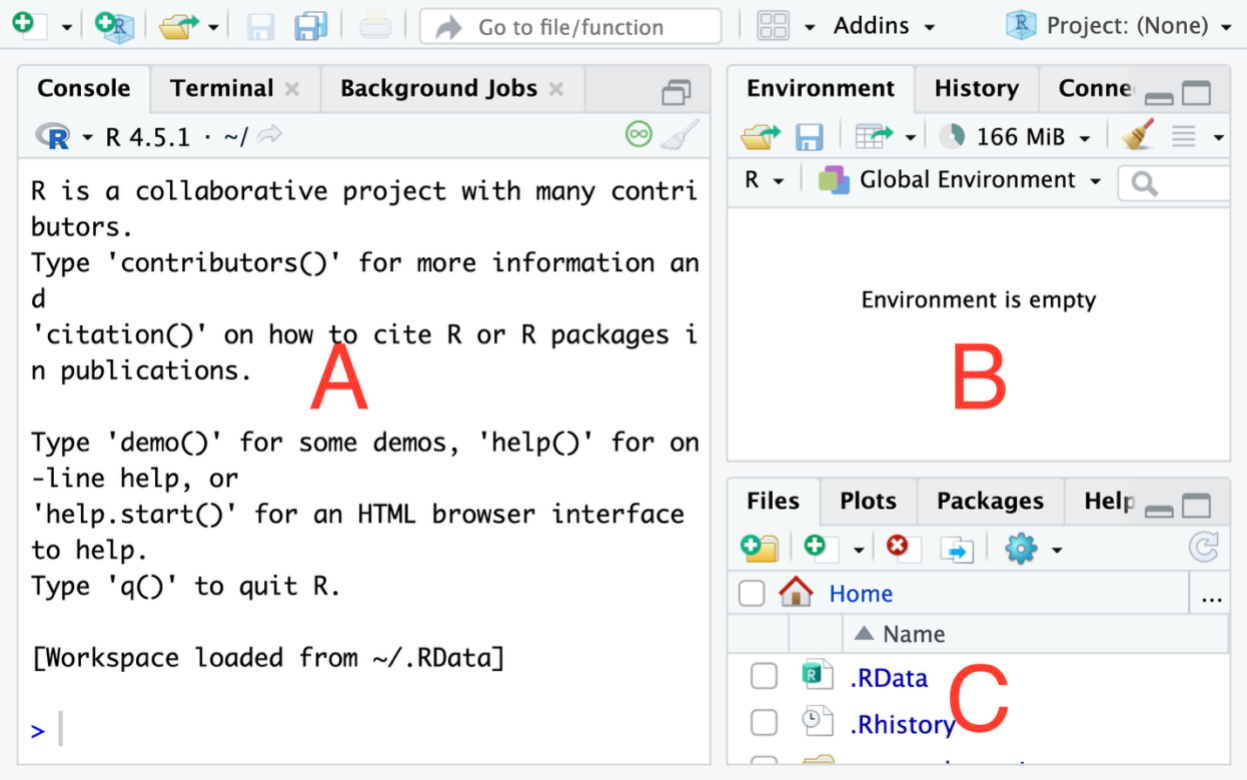

Figure 7: Default Panes in RStudio Console (Posit team, 2025)

2.2: The Console

Upon initial download, the window on the left side (denoted by A in Figure 7) is the *console*. This is the location where code can be entered and run. When there is an error or warning, it will also show in the console window. You can also use this window to install new packages and to access packages and/or built-in datasets.

2.3: The Global Environment

The window on the top left of RStudio (denoted by B in Figure 7) is the *global environment*. This is the location where globally defined variables and functions will be listed along with a preview of each's attributes.

2.4: The File Directory

The window on the bottom right of RStudio (denoted with a C in Figure 7) is the *files* window, you can use this to browse the files stored on your local computer.

Note: For the purposes of this guide, we'll see when we produce the graphs later, they will appear in this same location (bottom right) but instead the *Plots* tab will be highlighted.

2.5: Module Recap

When you first install RStudio, you'll see 3 default panes – the console, global environment, and the file directory. In general:

- Use the *console* to run code, install packages, access/call built in datasets, see errors or warnings, and to call R's built-in help function.
- Use the *global environment* window to see the names and a preview of any variable and/or functions defined.
- Use the File directory to quickly access files

Module 3: Accessing a Built-In R Dataset

3.1: Module Goals

In this module, the goal is to access our first built-in dataset in RStudio. By the end of the module, you should be able to:

- Identify all built-in datasets available with your installed packages
- Call a specific built-in dataset
- Preview a snippet of a built-in dataset

3.2: Finding the R Library of Built in Datasets

To call the list of built-in datasets in R, type the command "`data()`" in your console and press "enter" or "return" on your keyboard. A window like Figure 8 will appear in a script window (usually located above the console).

```
Data sets in package 'datasets':

AirPassengers     Monthly Airline Passenger Numbers
                  1949-1960
BJsales           Sales Data with Leading Indicator
BJsales.lead (BJsales)
                  Sales Data with Leading Indicator
BOD               Biochemical Oxygen Demand
CO2               Carbon Dioxide Uptake in Grass
                  Plants
```

Figure 8: Built-in Datasets in RStudio (Posit team, 2025)

In the resulting script, you'll note the name of the dataset and a brief description of it. When you identify the dataset with which you want to work, take note of its name. Be sure you're careful with the name's case as R and RStudio are case sensitive.

3.3: Accessing a Built-In Dataset

Whenever you want to work with a built-in dataset in R, you'll use its name (see Module 3.2) to call it. This is similar to how someone may call you a specific name to get your attention. The name of the dataset tells R with which dataset you want to work.

For this module (and the next) we'll use the `Loblolly` (Kung, 1986) dataset.

3.3.1: Previewing the First Few Rows of a Dataset:

To look at the first few rows of a dataset, use the command '`head([dataset_name])`'. For our example, call the first few rows of the Loblolly dataset with the command `head(Loblolly)` as shown in Figure 9.

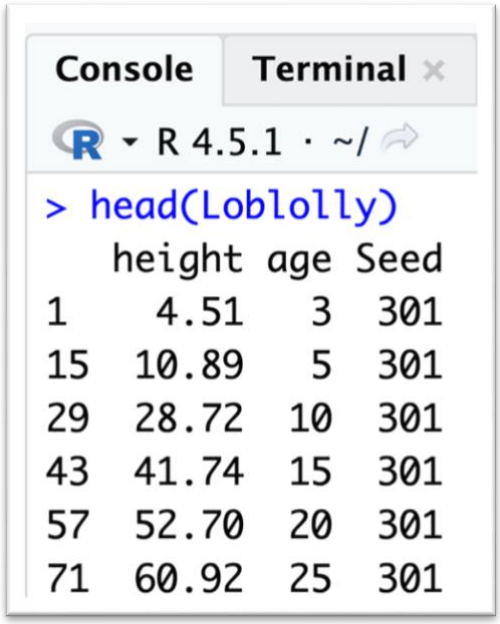

Figure 9: Loblolly (Kung, 1986) dataset preview in RStudio Console (Posit Team, 2025)

3.3.2: Previewing the Last Few Rows of a Dataset:

To preview the last few rows of a dataset, use the command '`tail([dataset_name])`'. For our example, use the command `tail(Loblolly)`. The output should look like Figure 10.

```
> tail(Loblolly)
   height age Seed
14   3.46   3  331
28   9.05   5  331
42  25.85  10  331
56  39.15  15  331
70  49.12  20  331
84  59.49  25  331
```

Figure 10: Loblolly (Kung, 1986) dataset preview on RStudio Console (Posit team, 2025)

If you want to see the entire dataset (not recommended for large datasets), type just the dataset's name in the console and press enter/return on your keyboard.

3.4: Module Recap

RStudio contains several built-in datasets to allow users to begin working with data. If you want to access a specific dataset, you reference it by its name being careful with case. Adding the `head()` or `tail()` commands will give you a preview of the first few or last few rows, respectively, for that dataset.

Module 4: Creating Graphs

4.1: Module Goals

In this module, the goal is to use a built-in R dataset to create a boxplot and histogram in RStudio. By the end of the module, you should be able to:

- Subset to a single column of data in a dataset
- Create a well-labeled boxplot and histogram
- Add color to graphs

4.2: Subsetting to a Single Column

Because this document is for beginners, we will only work with one column of data from the `Loblolly` dataset. To do this, we'll use the command `new_data = Loblolly[,1]`.

This line of code creates a new dataset with the name "new_data" that contains only the first column of information (height) from the Loblolly dataset.

To create our graphs in sections 4.3 and 4.4, we'll use the new_data dataset.

4.3: Creating a Histogram in RStudio

Histograms are a graphical display that can be used to show the distribution of quantitative (numeric) data. As the height column is a quantitative variable, we'll first create a histogram to see its distribution.

To create a basic histogram with RStudio use the command: `hist(new_data)`. This produces a basic histogram as seen in Figure 11 but notice how it's not informative.

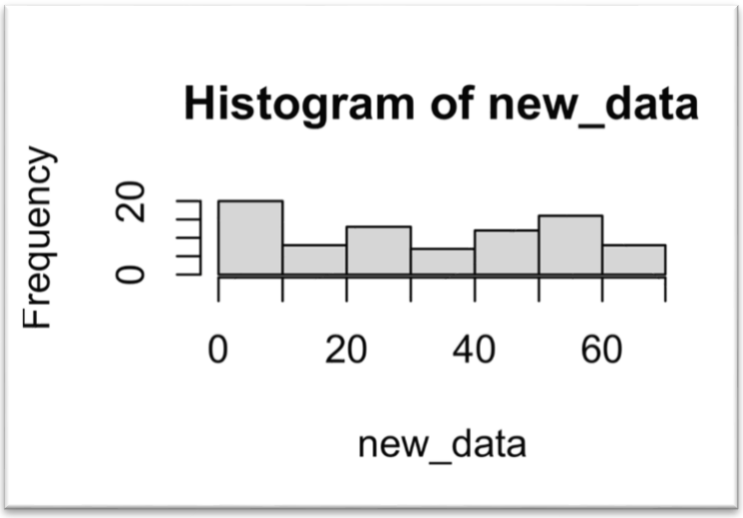

Figure 11:Histogram of Data in RStudio Console (Posit team, 2025)

If you want to add labels and color, we'll update this code to: `hist(new_data, xlab = "Height", ylab= "Frequency", main= "Height of Loblollies", col = "blue")`

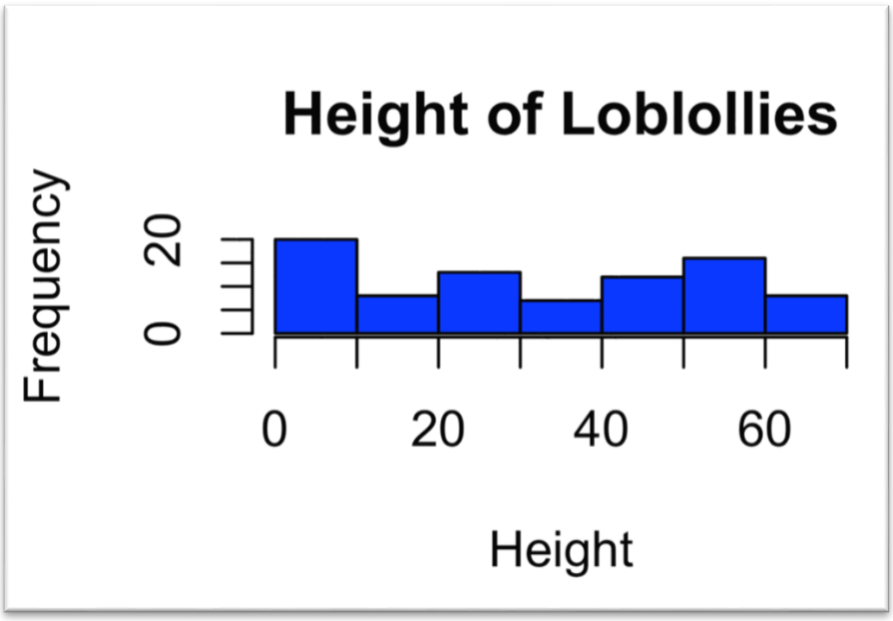

Figure 12: Updated Histogram of Data in RStudio Console (Posit team, 2025)

This expanded code produces the histogram shown in Figure 12. The additional entries add:

- `xlab` – Gives a title to the x-axis of the graph
- `ylab` – Gives a title to the y-axis of the graph
- `main` – Gives a heading title to the graph
- `col` – Gives the graph a color – in this case blue

4.4: Creating a Boxplot in RStudio

Boxplots are another graphical display that can be used for quantitative data.

To create a boxplot, we'll use the command: `boxplot(new_data)`. This produces the boxplot shown in Figure 13.

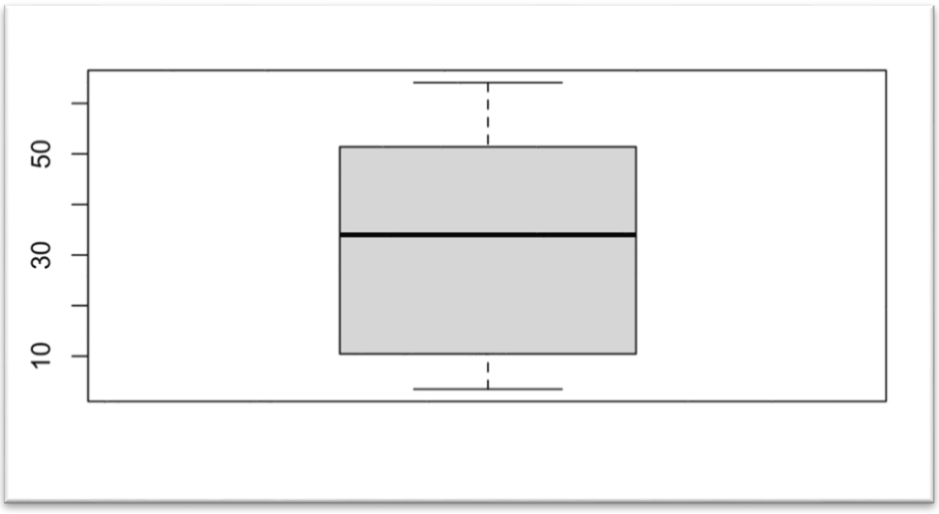

Figure 13:Default Boxplot in RStudio Console (Posit team, 2025)

To make it more informative, add the commands `main = "Height of Loblollies"`, `ylab = "Height"`, `col= "blue"`.

The final line of code: `boxplot(new_data, main = "Height of Loblollies", ylab= "Height", col = "blue")` is then used to produce the output we see in Figure 14.

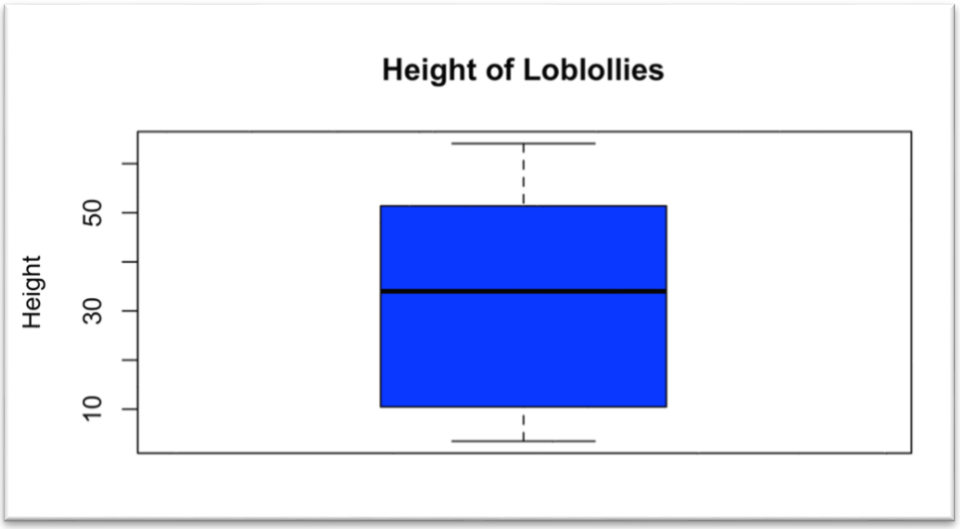

Figure 14: Updated Boxplot of Data in RStudio Console (Posit team, 2025)

4.5: Module Recap

RStudio can be used to create various graphical displays including the histogram and boxplot as demonstrated in this module.

Summary

Congratulations! You've gone from installing R and RStudio to creating not 1, but 2 graphical displays! More specifically, if you look at all the modules' objectives you now know how to:

- Install R on a Mac
- Install RStudio on a Mac
- Identify the 3 default panes shown on RStudio when its initially downloaded
- Describe the purpose of the 3 default panes
- Identify all built-in datasets available to you in RStudio
- Call and access a specific built-in dataset
- Preview a snippet of a dataset
- Subset to a single column of data in a dataset
- Create well-labeled histograms and boxplots
- Add color to graphs

Phew! Go you!

To learn more advanced topics as you progress in your knowledge of R and RStudio be sure to subscribe to our webpage at https://statisticalconsultingfirmofsc.com.

References:

Kung, F. H. (1986). "Fitting logistic growth curve with predetermined carrying capacity". Proceedings of the Statistical Computing Section. American Statistical Association, pp. 340–343.

Posit team (2025). *RStudio: Integrated Development Environment for R*. Posit Software, PBC, Boston, MA. URL. http://www.posit.co/.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.