Lecture 1.4 – Intro to R and RStudio

Specific Learning Objectives:

1.1.1 – Understand how to use the command line.

1.1.2 – Understand how to use the help function of R.

1.1.3 – Understand the basic syntax of the R language.

1.1.4 – Execute inbuilt mathematical functions to perform calculations in R.

1.1.5 – Learn how to assign variables.

1.1.6 – Understand the basic syntax of functions in R.

1.1.7 – Open, edit, and save a script in RStudio's editor.

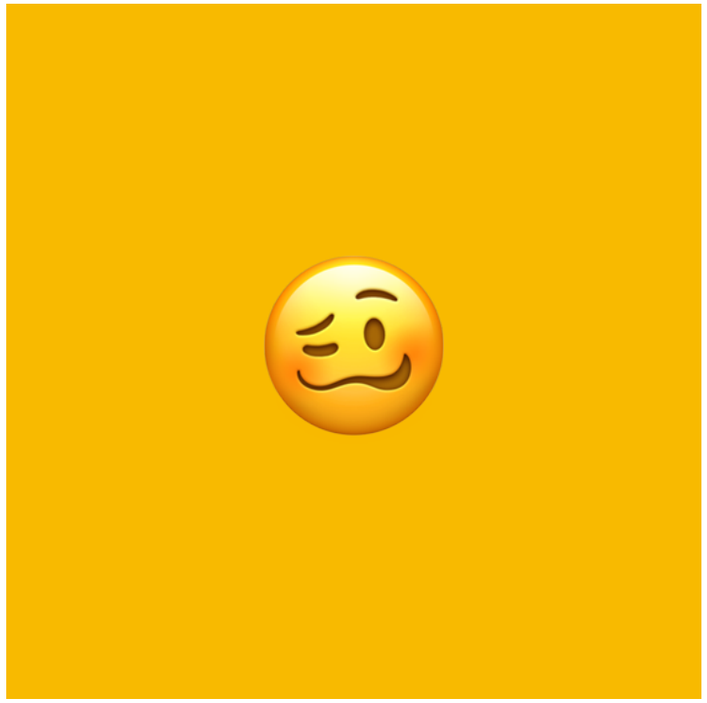
2.2.1 – Create reproducible scripts in R.

How Class will Work

Use the sticky notes to indicate:

I’m doing fine!

I need a hand.



Downloading *R* and RStudio

Download *R*: https://www.r-project.org/

1. **Go to https://cloud.r-project.org/**
2. **Select your operating system.**
3. **Select the latest release that is "notarized and signed."**
4. **Save and open the file, follow the instructions to install.**

Download RStuido: https://rstudio.com/products/rstudio/download/

1. **Select the RStudio Desktop version.**
2. **Download, open, and follow instructions to install.**
3. **Open RStudio to get started!**

Check Your Understanding

Have R calculate the square root of 164956 using the `sqrt()` function. What is the correct output?

Correct answer

a) `[1] 406.1478`

b) `[1] 12.8841`

c) `Error: unexpected input in "√"`

d) `Error in sqrt(164956, 2) : 2 arguments passed to 'sqrt' which requires 1`

Error in c: I used the sign √164956, even though I understand what this means, R doesn't understand so returns an error.

Error in d: I put too many numbers (or arguments) in the `sqrt()` function so it's telling you you need one argument only!

Check Your Understanding

Syntax is important here! Using a space between the < and - will change the meaning of the command.

Try these: do they all have the same result? Why or why not?

a) `x<-2` b) `x <- 2` c) `x< - 2`

Does it matter which way the arrow points?

Try these: which work and which produce errors?

d) `7 -> j` b) `7 <- j` c) `j -> 7`

Functions in R

Check your understanding!

What happens if you enter the numbers into mean without the **c( )**? Or like this:

```
> mean(1, 9, 8, 2)
```

Why does it do this?

Why would the mean not calculate correctly based on the help documentation for **mean( )**?

About R assignments

- **You will be turning in assignments in R as R scripts**
  - Scripts must have the extension ".R"
  - Please include the assignment number in the script name
  - Include your name, course number, section number, date, and assignment number as a comment at the top of the script.
  - For each question, please start the answer with a comment:
    ```
    # question a:
    answer code
    ```
  - Any written text should be included as a comment in the script.
- **How your scripts will be scored:**
  - Grader will download your script and hit "source" on top, then evaluate output based on the assignment question.
  - If there is an error on source, it will be scored 0 (unless that's part of the assignment!) Please CHECK FIRST that it will run by starting a new R session and then sourcing your script!

For the rest of the class…

- **In-class exercises:**

1. Write a short explanation about how to install a package using:
   a) RStudio's GUI package manager
   b) command line in R

2. Exercise 2.2 in Davies (End of Chapter 2 section 2).

- **Then, start on Assignment 1.5**

Action Items

1. **Complete Assignment 1.5**

2. **Read Davies Chapter 4 for next time.**