

# **1: Introduction to Introduction to Statistics**

Frank Edwards

# Syllabus / schedule

[https://f-edwards.github.io/intro\\_stats](https://f-edwards.github.io/intro_stats)

# Course communication on Slack

**<https://introstatsfall2024.slack.com>**

## Books for class

- Diez, Rudel, and Barr: *Open Intro to Statistics* <https://leanpub.com/os>
- Healy: *Data Visualization* <https://socviz.co/>
- Alexander: *Telling Stories with Data* <https://tellingstorieswithdata.com/>

# Navigating the schedule

[https://f-edwards.github.io/intro\\_stats](https://f-edwards.github.io/intro_stats)

## Software installation

- Install R <https://cran.r-project.org/>
- Install RStudio <https://posit.co/download/rstudio-desktop/>

## **Open science basic principles**

- Science should be reproducible
- Scientific decision-making should be transparent
- Free and open source software are the best foundations for reproducible and open research

# RStudio walkthrough



# The script and the console

# Let's work with R

## Using the console

- Try some simple math operations with: + - / \*
- Now try some operations that use: ^ ()
- Try sqrt(); try log()

## Objects in R

For the following, please work in your RStudio Console

- In R, we use the assignment operator `<-` to assign values to objects
- Try to create an object named  $x$  and assign the value 3 to the object
- Confirm that  $x$  is equal to three using the `==` operator
- Compute  $\frac{x+3}{2x^2}$

## Objects in R

- Create a new object named  $y$  that is equal to  $3x$
- Confirm that  $y$  is equal to  $3x$  using the `==` operator
- Print both  $y$  and  $3x$  to the console

# The Environment

- List all objects in your environment with `ls()`
- Compare the results to your 'Environment' tab in RStudio
- Next, let's restart our session of R and see what happens to our Environment
- (Session Menu -> Restart R or Shift + Cmd + 0)

**X is gone, so is y!**

## **Reproducible research: the environment is a sandbox, treat it that way**

- The environment should always be treated as temporary
- It disappears when you close R
- For now, you never need to save the environment
- You can use the environment for quick checks on objects or operations



## **Reproducible research: the importance of scripts**

- Scripts provide line-by-line instructions for everything you do in your analysis.
- ALWAYS DO YOUR WORK IN SCRIPTS!
- Scripts allow your entire analysis to be reproduced by simply running the program
- To make a new script: Cmd+Shift+N, or File -> New File -> R Script

## File paths and the command line

We need to choose a good place to save this cool new script we are making

- Click the 'terminal' tab and let's check out how file paths work from the command line
- Basic unix command line tools: `cd`; `ls`; `mkdir`
- Navigate to your root directory with `cd ~`
- Use `ls` to identify a good place for you to store your stats course work
- create a new directory called `intro_stats` using `mkdir`

## Your class folder and path

- Now save this script as `lab1.R` in `./intro_stats/`
- Make note of its exact file path (the details before the cursor on your terminal)
- Create a new directory inside `intro_stats` called `hw`
- Use these directories all semester!

## Back to our script

Let's repeat the operations we did before in the console, but now in the script - On line 1: create an object named  $x$  and assign the value 3 to the object - On line 2: Confirm that  $x$  is equal to three using the `==` operator - On line 3: Create a new object named  $y$  that is equal to  $3x$  - On line 4: print the object  $y$  by simply calling  $y$

## Sourcing and confirming output

- Now run your program! Click 'source' or use cmd+shift+return to run the script
- Check out your environment tab in RStudio, what do you see?
- Restart R again: cmd+shift+0
- Now source your script again. Reproducibility!

# Congrats you wrote a program!

## Let's document your program

- We use `#` to write code comments
- Before each line of code in your script, write a brief comment explaining what the line of code will do in plain english

## Packages in R

- Packages extend the functionality of R (dramatically!)
- Thousands of FOSS packages available through CRAN
- Try it:
  - On the console: ``install.packages("tidyverse")`
  - you only have to install once!



## Loading packages in R

- Use the `library("packagename")` format to import a package
- Try it with tidyverse
- Package loading will usually be the first few lines of your scripts

# Homework 1

**[https://github.com/f-edwards/intro\\_stats/blob/master/hw/HW1.Rmd](https://github.com/f-edwards/intro_stats/blob/master/hw/HW1.Rmd)**