Introduction to Epidemiology Data Analysis with R

Introductions

- Dr. Elza Rechtman (Course co-director)

elza.rechtman@mssm.edu

- Dr. Joselyn Chávez-Fuentes (Course co-director)

joselyn.chavez-fuentes@mssm.edu

- Placidus Fernando (Teaching Assistant)

placidus.fernando@icahn.mssm.edu

Course details

- Course Blackboard: learn.mssm.edu
- Tuesdays 6:15 pm – 8:45 pm
- 12 sessions
- Hybrid format
  - In Person: Annenberg 5 - 212
  - Zoom: https://mssm.zoom.us/j/94841240630

Course schedule (1)

Week 1	01/7/2025	Get up and running with R and RStudio	• The basic data analysis cycle • Download and install R • Download and install RStudio • Install a set of R packages called the Tidyverse • Understand the environment interface • Where and how to get help • Building scripts • R studio project
Week 2	01/14/2025	Introduction to coding with R (I)	• Create and name a variable • Create and name a vector • Converting a vector from one class to another • Access vector elements
Week 3	01/21/2025	Introduction to coding with R (II)	• Modify a vector • Create and name matrices • Operations with matrices
Week 4	01/28/2025	Introduction to coding with R (III)	• Create, name, and subset lists • Create and name a data frame
Week 5	02/04/2025	Data transformation	• Tidyverse packages • Tibbles • Arrange, Filter, Select, Mutate • Count, Summarise, Group • The pipe
Week 6	02/11/2025	Data Wrangling using the Tidyverse (I)	• Combining multiple operations with the pipe • Pivoting • Seperating • Other useful functions • Joins

Course schedule (2)

Week 7	02/18/2025	Data Visualization (I)	• Geometries and mappings • The layered grammar of graphics • Create visualizations using the x, y, color, size, alpha, and shape properties.
Week 8	02/25/2025	Data Visualization (II)	• Facets • Statistical transformations • Position adjustments
Week 9	03/04/2025	Data Visualization (III)	• Coordinate systems • Themes • Arranging plots • Jitter • Other geometrics • Limits
Week 10	03/11/2025	Modeling basics for Epidemiology research studies	• Introduction • Linear regression, t-test, ANOVA • Build simple linear regression models in R
Week 11	03/18/2025	Final Project Presentations and Course Wrap-up	• Present and discuss the final student epidemiology project using R
Week 12	TBD	Creating reports with Rmarkdown	

Grading

- Pass/Fail Grading System
- To pass, you need to:
  - Attend 80% of classes
  - Submit 80% of the assignments
  - Present a final project

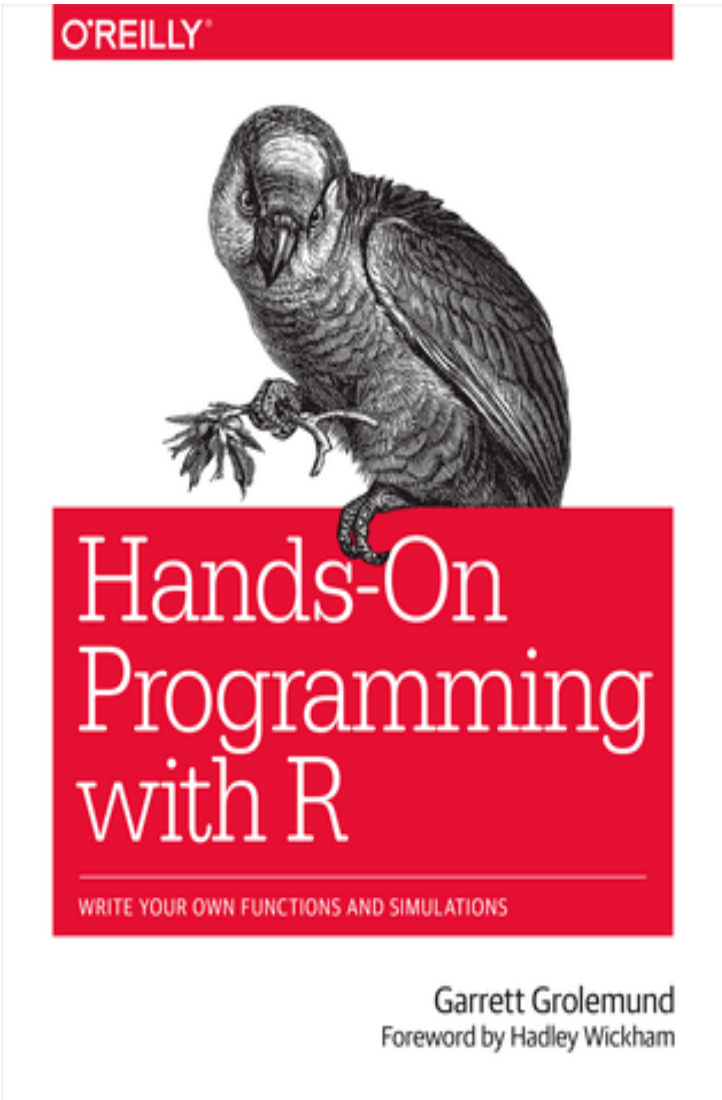
Questions

Week 1

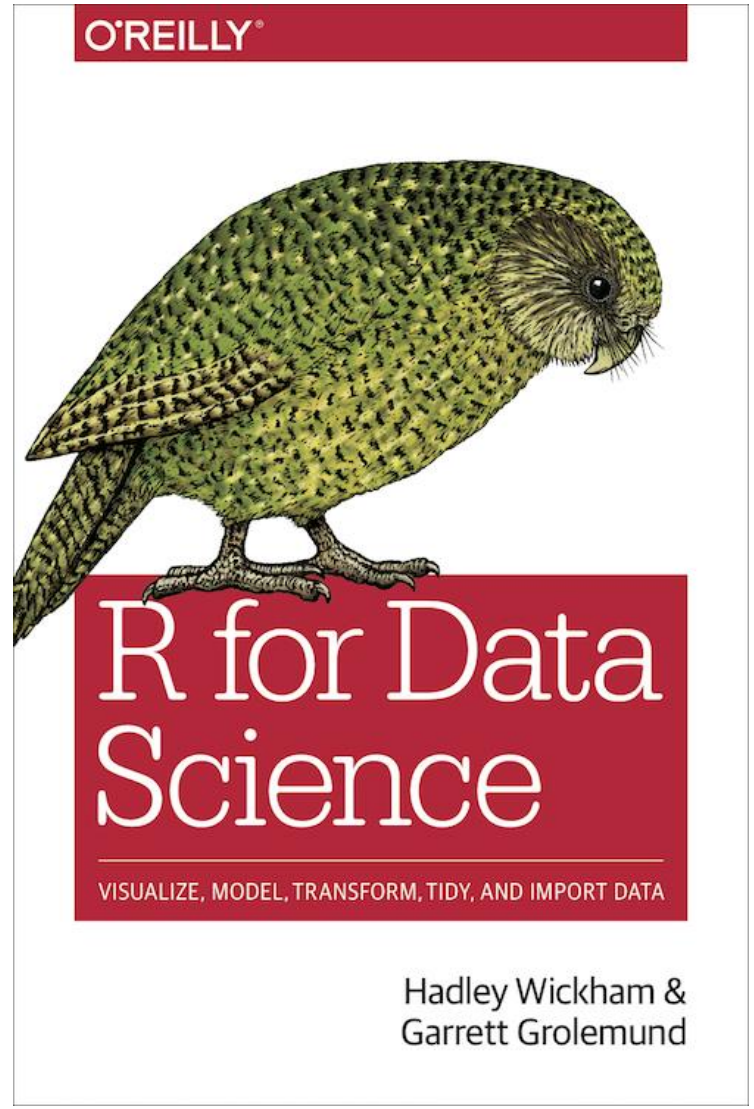
Get up and running with R and RStudio

Getting started with R

- What is R?
  - R is an open-source language widely used as a statistical software and data analysis tool to:
    - Manage and clean data
    - Carry out statistical analyses
    - Produce high-quality figures for research communications
- Why R?
  - Free & open source
  - Great community
  - 9000+ free packages
- What is RStudio?
  - An integrated development environment (IDE) for R

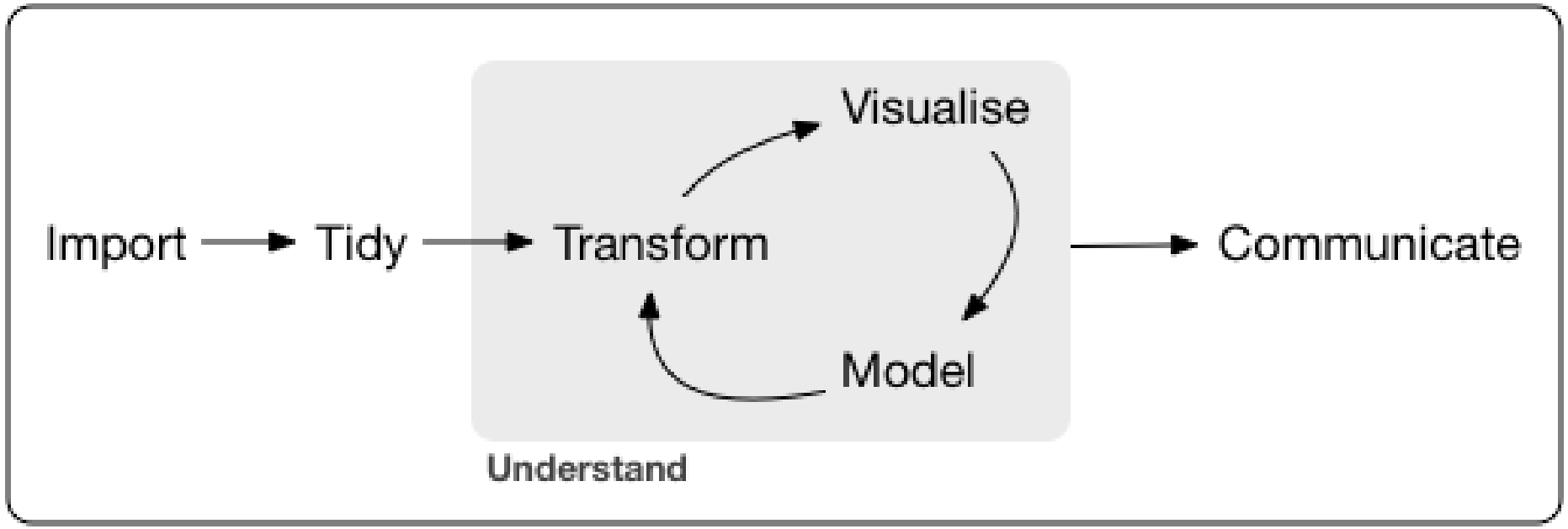


https://rstudio-education.github.io/hopr/



https://r4ds.had.co.nz/

A typical data science project :



Downloading and installing R

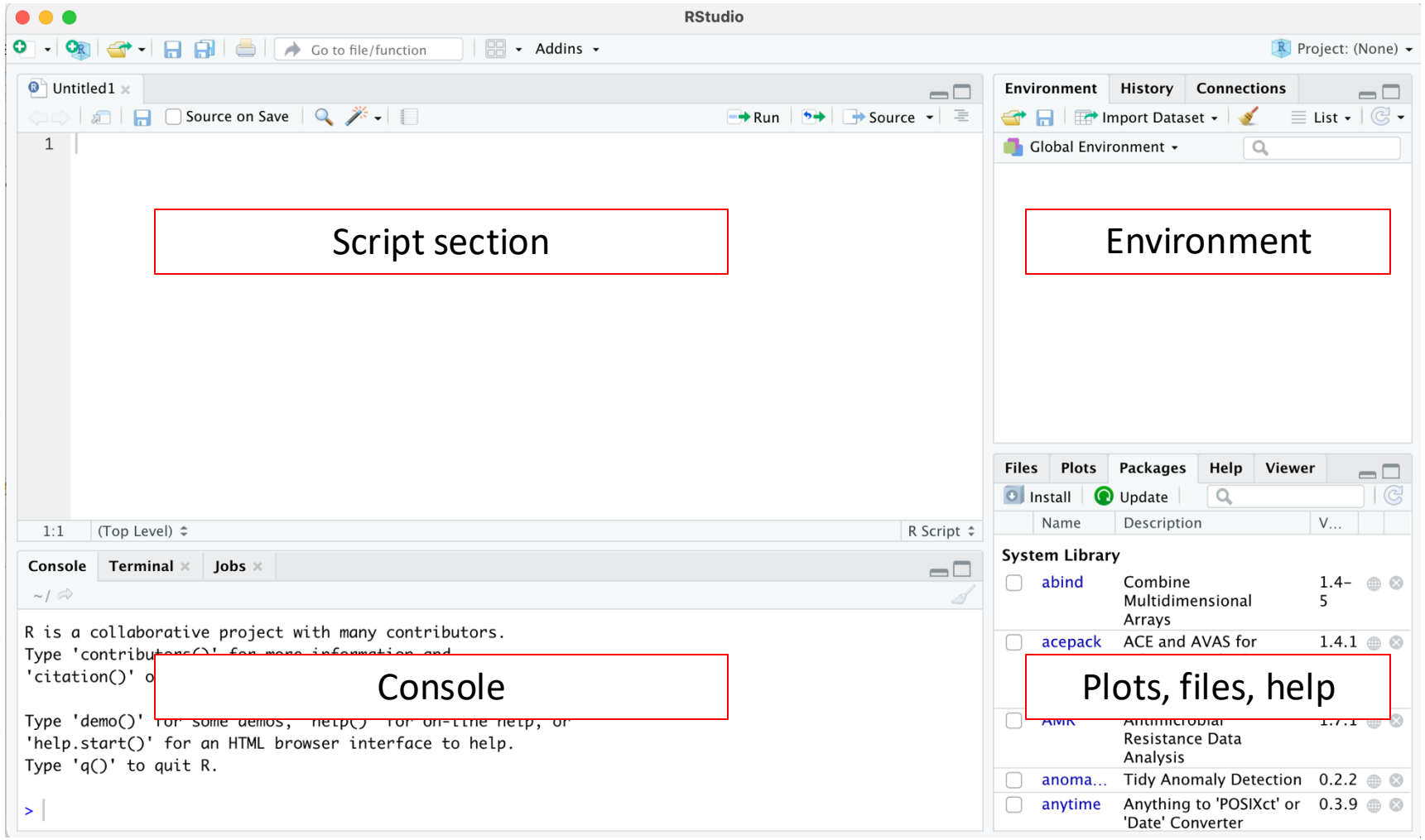
- Download and install R: http://lib.stat.cmu.edu/R/CRAN/
  - Windows
    - Download R for Windows
    - Base
    - Download R 4.4.2 for Windows
    - Run this program and step through the installation wizard that appears.
    - The wizard will install R into your program files folders and place a shortcut in your Start menu.
  - Mac
    - Download R for MacOS
    - R-4.4.2.pkg
    - An installer will download to guide you through the installation process

Downloading and installing RStudio

- Download and install RStudio: https://www.rstudio.com/products/rstudio/
- Download Rstudio Desktop
  - Windows
    - Download RSTUDIO-2024.12.0-467.EXE
    - Install through the installation wizard
  - Mac
    - Download RSTUDIO-2024.12.0-467.DMG
    - An installer will download to guide you through the installation process

RStudio Desktop

1. Open Rstudio
2. Start a new script



RStudio settings you can adjust

- **Restore workspace:** By default, R saves your workspace, which is no longer considered best practice.
  - To change:
    - Tools > Global Options >
      - **Uncheck** “Restore .RData into Workspace on startup”
      - Set “Save .RData on exit” to **Never**

- **Appearance:** By default, RStudio comes with a white background and black text.
  - To change:
    - Tools > Global Options > Appearance > Editor theme

Your turn!

In the script pane, write and run:

- 5 + 2 =
- 6 X 3 =
- 7^2 =
- $\sqrt{9}$ =

Using variables

- Type in the console a <- 5
- Check your environment
- Type in the console a*2

- Type in the console b <- 1:30
- b^2

Your turn!

- Choose any number
- Add 2 to it
- Multiply the result by 3
- Subtract 6 from the answer
- Divide what you get by 3

What did you get?

What if you would choose a different number and run the same steps?

Can you think of a way to write a script for this process?

Writing a script

- A script is a sequence of commands stored in an .R file
- R execute scripts sequentially line-by-line
- Comments start with #

example.R

```r
# load packages
library(tidyverse)
library(janitor)

# import data
data <- read_csv("COVID_19_0421.csv")

# Clean data
data_clean = data %>%
  filter(INFECTION_STATUS == "COVID-19") %>%
  select(ID, AGE, SEX, RACE, ETHNICITY, FACILITY, INFECTION_STATUS) %>%
  drop_na(c(ID:SEX)) %>%
  slice(c(1:500)) %>%
  clean_names()

# Plot data
p1 <- ggplot(data_clean, aes(x = age, y = bmi)) +
  geom_point()

#Statistical test
mod <- glm(bmi ~ age + sex, data = data_clean)
summary(mod)
```

24:1 (Top Level) R Script

Tips for writing a script

- Keyboard Shortcuts to run your script
  - Windows: Control + Enter
  - Mac: Command + Enter
- Don’t spend time memorizing functions that can easily be looked up and copied (you will be mostly copy-pasting and adapting existing R code)
- Don’t worry about making mistakes - you can’t do anything wrong!

Packages

- A Package is a collection of functions that are not included in the standard R installation (base-R)
- Install the tidyverse package using **install.packages()**
- Load the tidyverse package using **library()**

Getting help

- Error messages:
  - Google the error message.
  - copy-paste solutions into your R script and then modify it.
- RStudio’s built in Help - type ? and the command (for example ?read_csv).
- Help drop-down menu at the top of the RStudio window

RStudio Projects

- Keep data and scripts in the same folder
- Keep files from each project separated
- Set the working directory
- Scripts and output files will be automatically saved in your Rstudio project folder

Let’s create an R project

- 2 Methods
  - New Directory
  - Existing Directory

Your turn!

1. Create an R project called “Week_1”
2. Open the project
3. Open a script
4. Write a script in which you:
    1. Assign the number **81** to a variable called **a**
    2. Create a new variable **b** that equals the square root of **a**
    3. Save the script under the name “week1_script.R”
5. Close the R project
6. Reopen the R project and run your script

Import data into R

- Download “Sinai_covid.csv” from Blackboard and save it in your R project folder.
- In your Script Pane type:

```
library(tidyverse)

Sinai_covid <- read_csv("Sinai_covid.csv")

View(Sinai_covid)
```

Assignment 1

1. Download R script "Assignment1.R"
2. Save the script in the R project folder we created (“Week_1”)
3. Open R project “Week_1”
4. Open the script "Assignment1.R"
5. Complete all the questions
6. Save the script under a new name "Assignment1_FirstName_LastName.R”
   - For example, for me, it would be: "Assignment1_Elza_Rechtman.R"
7. Upload your assignment by next Monday, January 13th at 5 pm.