BST 219
Core Principles of Data Science

Lecture 4: R Basics
September 12, 2024

Recipe of the Day!

Spaghetti Carbonara
(Credit to Taylor)

Agenda

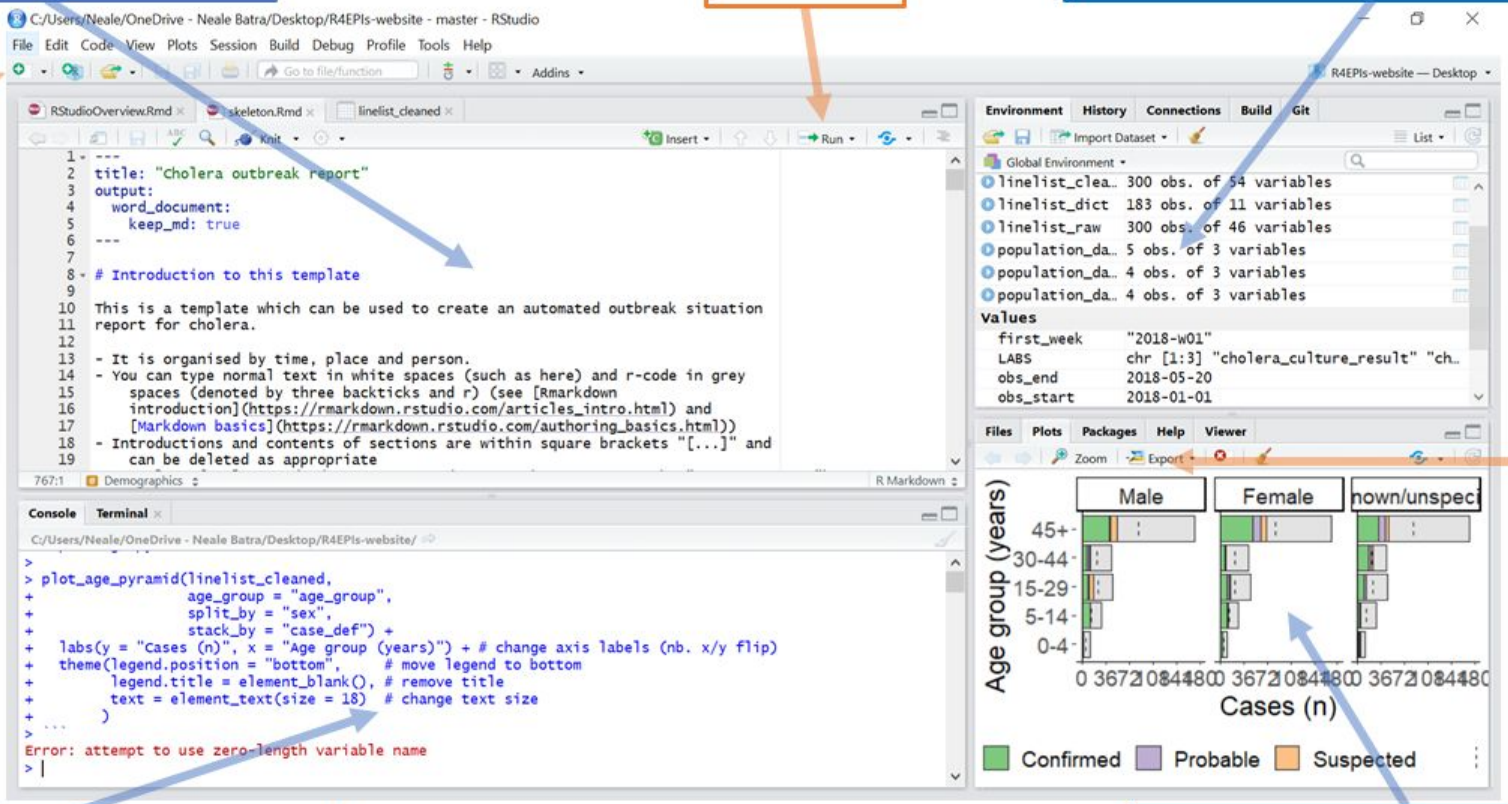
- Introduction to RMarkdown
- Coding basics
- Coding good practices

Source Pane
Edit and run scripts (e.g. Rmarkdown templates), and view datasets

Tip: Run script

Environment Pane
Overview of objects (datasets, parameters, lists, etc.) you have imported or created.

Tip:
Start new script



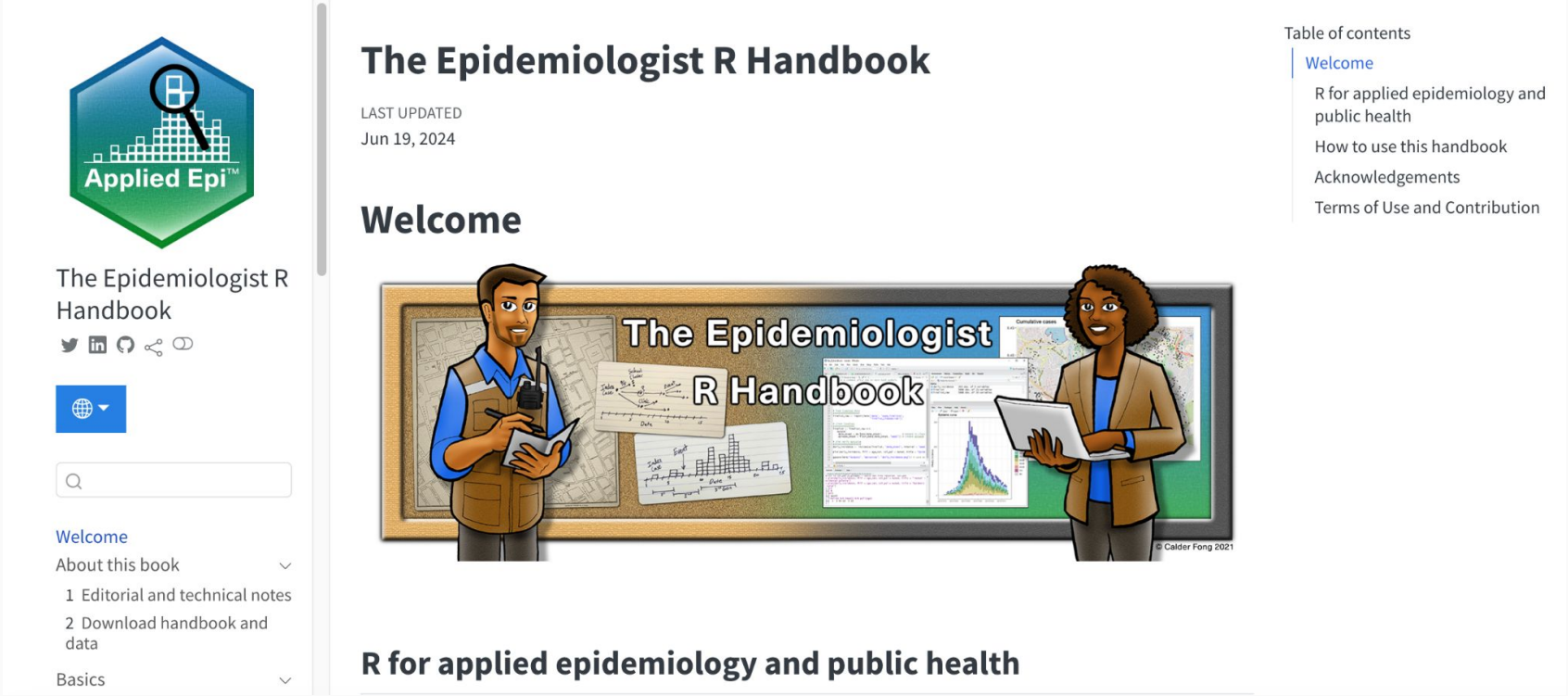
Tip: Zoom and export plots

R Console Pane
R commands run are shown here, and non-graphic output and errors are displayed

Plots, Packages, and Help Pane
Commonly used to view graphics, install packages, and view help

New R Resource!

The Epidemiologist R Handbook



Good Coding Practices

- Commenting your code (use the # sign)
  - What the code does
  - Why you are doing what you are doing
  - The inputs and outputs of a function
  - Break your code into logical sections

```
# A comment can be on a line by itself
# import data
linelist <- import("linelist_raw.xlsx") %>% # a comment can also come after code
# filter(age > 50)                          # It can also be used to deactivate / remove a line of code
  count()
```

Note: we expect you to comment your code for all assignments! It can be concise, but get into the habit of commenting your code now - future you will thank you.

Good Coding Practices

- Style (style I use)
  - Make your code as readable as possible
  - When naming objects, use only lowercase letters, numbers, and underscores (e.g., my_data, country, etc.)
  - Use frequent spaces, especially around operators (e.g., x = 17, age_new <- age_old + 1)

```
# Note that this code is easier to read ...
x_list <- c(3, 6, 9, 12, 15)

# ... than this code.
xlist<-c(3,6,9,12,15)
```

Good Coding Practices

- Naming convention
  - Objects like variables and data frames should be given informative names (e.g., age, sex, weight).
  - Bad examples include “c” or any other letter if it’s difficult to figure out what that letter represents