

Foundations of Data Science for Biologists

Class Orientation and Introduction to R

BIO 724D

2024-AUG-27

Instructors: Greg Wray and Paul Magwene

Orientation to BIO 724D

What is BIO 724D?

A class that introduces how to work with biological data

- Practical skills for wrangling, processing, filtering, and visualizing data

- Best practices for analysis, interpretation, reproducibility, and reusability of data

Assumes no prior programming experience

A class that builds community

- Working with some data sets generated by Biology grad students

- Discussions with trainees, staff, and faculty in Biology and Duke more broadly

Not

- An introduction to statistics: though we will often discuss statistical methods

- An introduction to computer science: though we will introduce some concepts

Course structure

Lecture / lab: Tuesdays 3:05-5:35pm, 154 BioSci

Semi-flipped format: complete reading / video assignments *before* class

Class sessions a mix of traditional lecture and (mostly) hands-on engagement

Hands-on component involves individual follow-along as well as group exercises

Data lunch: Thursdays 12:00-1:00pm, 144 BioSci

Presentations by trainees, staff, and faculty over lunch

Discussions / questions / interaction encouraged



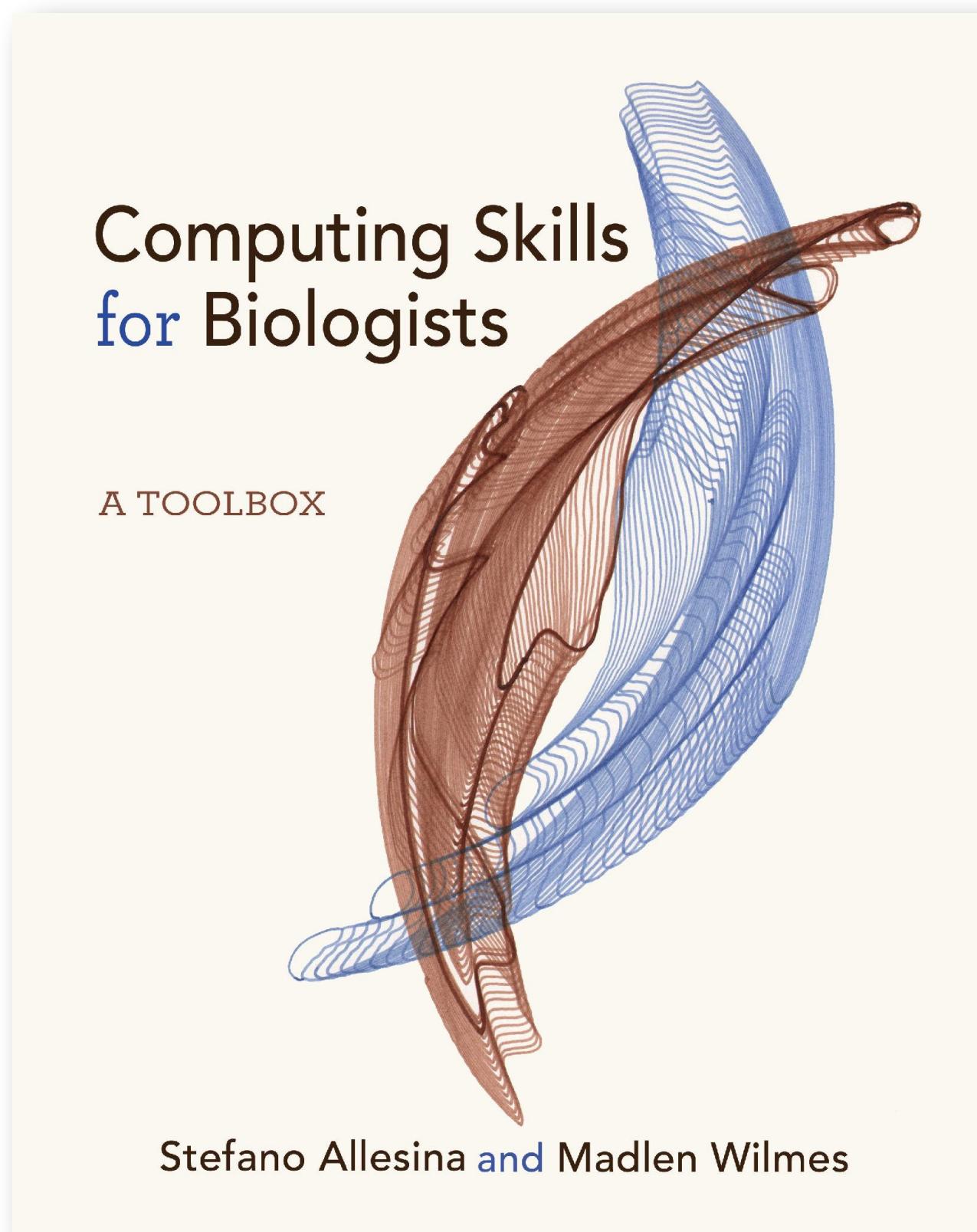
be respectful



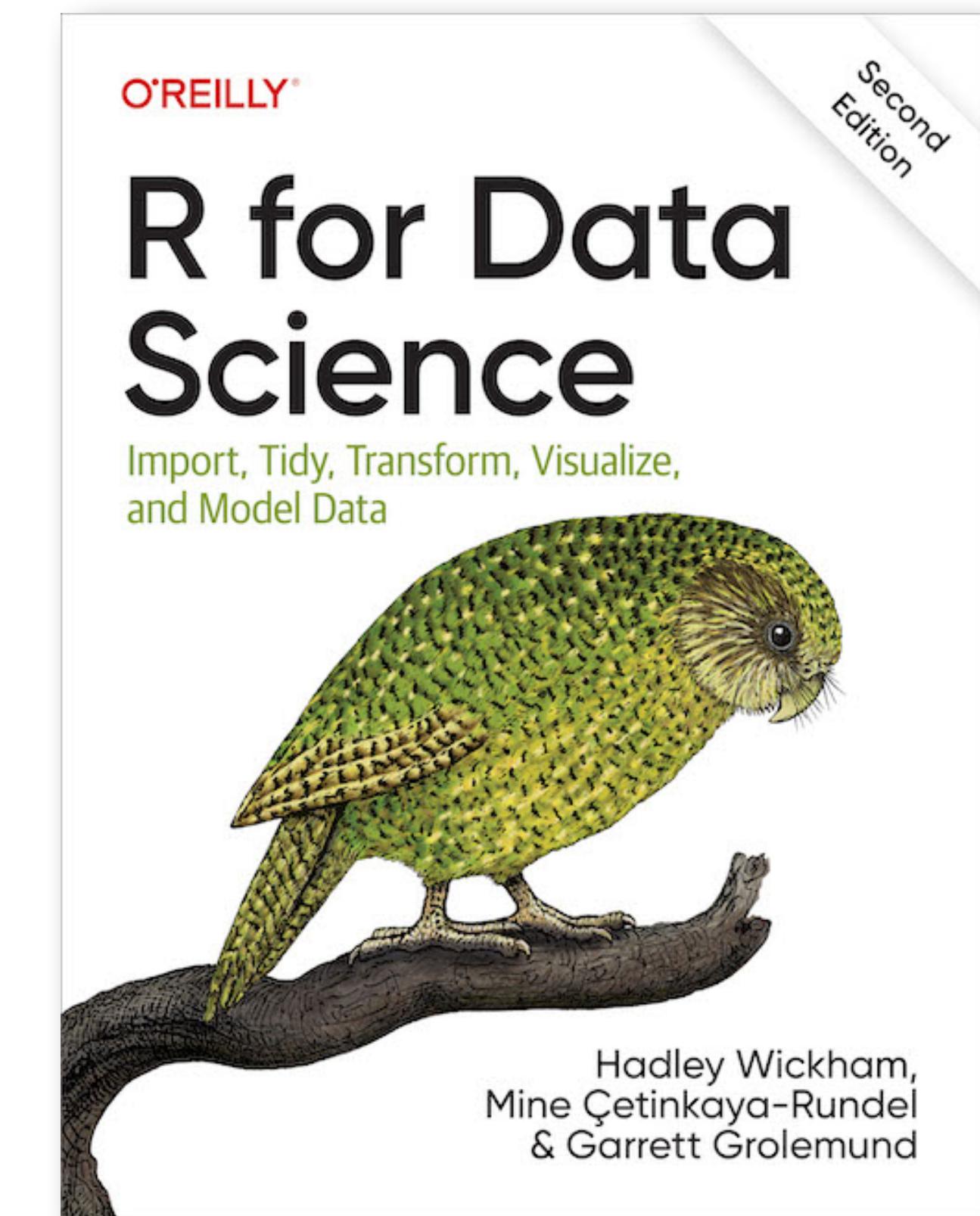
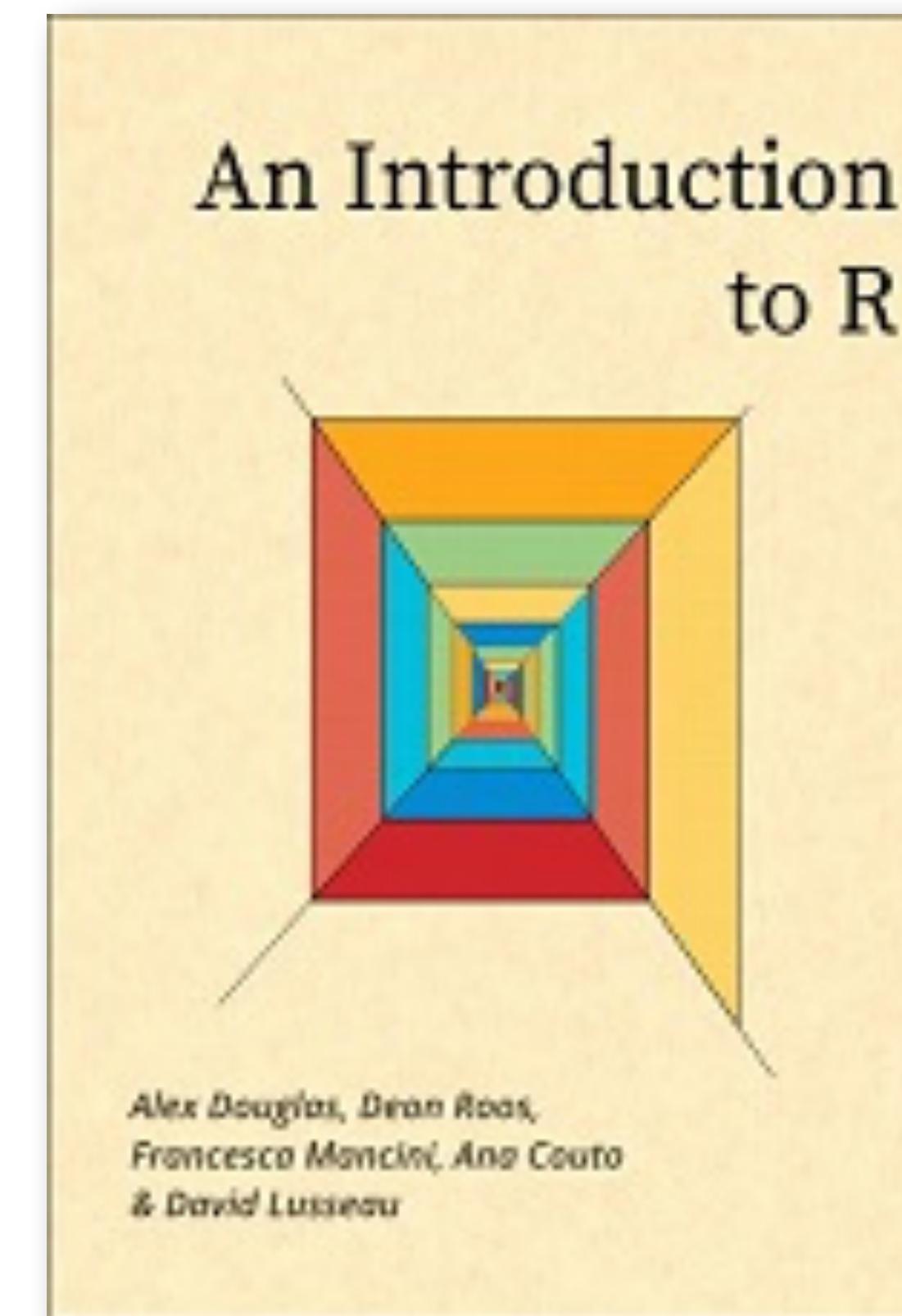
be polite

Course materials

Throughout the year:



For the R portion of the course:



All are available electronically for free: see the course wiki

Practicalities

Wiki: *the hub for most things related to the course*

- Schedule of lecture and lunch topics

- Information about course logistics, grading, and policies

- Materials for Tuesday sessions: readings, videos, data sets, notebooks, slides

- Assignments: problem sets and projects

- Links to resources: help with learning, reference material, deeper dives

GitHub: *where you turn in assignments*

- We will show you how to upload files

- Note the file naming convention in assignment instructions

Expectations

Adhere to the Duke Community Standard: collaboration and web resources are okay

Attend every class session and engage: follow along by actively writing your own code

Come to Tuesday sessions prepared: complete reading / video assignments *before* class

Submit homework and projects on time: late submissions will be scored lower

Stay on top of the material and ask questions: you will learn more if you engage and ask!!!

Familiarize yourself with the information on the course wiki

Assignments

Problem sets + project (90%)

Problem sets posted on Tuesday; due the following Monday at midnight

Graded for completion: code must run or include an explanation

Project posted on 12 Nov, due 3 Dec at midnight (worth 2 problem sets)

Data lunch reflections (5%)

Short description of new concept/method/application you learned

Turn in with your problem set; graded for completion

Learning notebook (5%)

File containing notes, code snippets, links, etc.; due 3 Dec at midnight

Resources for problem-solving

Be proactive: ask questions right away in class

Don't be shy: there are no "dumb" questions!

Your classmates will thank you!

Many sources can help you when you are stuck with a homework problem

Textbooks, Google, YouTube, LinkedIn Learning (free!)

Built-in help resources, R vignettes, code snippets from official documentation

Classmates (class Slack channel), friends, lab-mates, other grad students

On-line forums: StackOverflow and others

AI resources: ChatGPT, BingGPT, MS Copilot, etc.

Cite your sources and explain how they helped you

Code that isn't working

There may be occasions when you can't get your code to work properly: no problem!!

You will still get **full** credit if:

- You turn your homework in on time

- Identify specifically what is and is not working correctly

- Explain how you tried to fix the problem (be concise)

You will **not** get full credit if:

- You turn your assignment in late without prior arrangement

- You neglect to reveal and explain what isn't working properly (see above)

Test your code carefully and be transparent about any issues you can't fix!

What can you expect from BIO 724D?

You will learn concepts and skills that will:

- Be useful throughout your research career
- Save you lots of time and frustration
- Reduce errors in your work
- Make it easier for you to collaborate

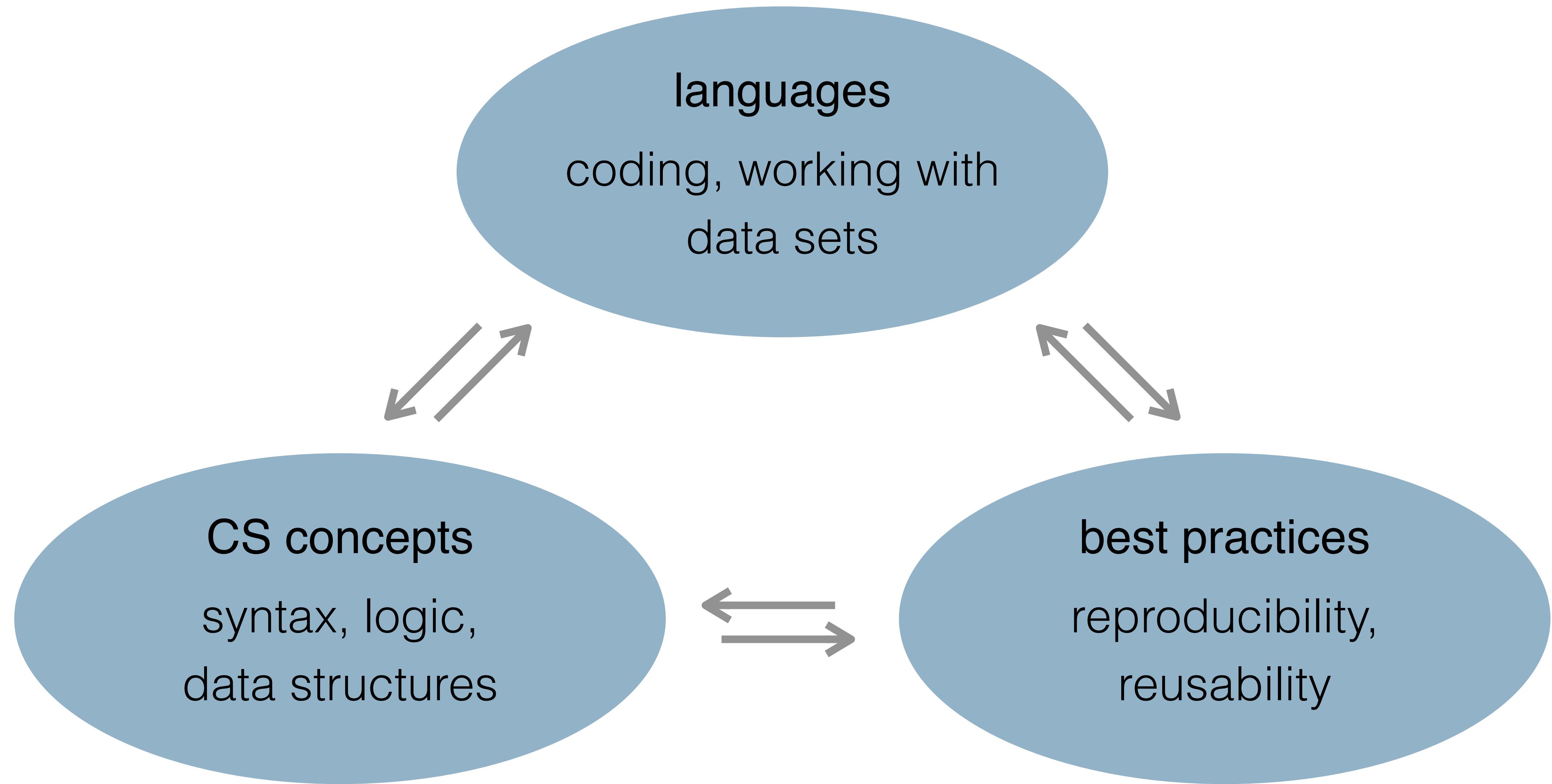
This comes at a cost

- You will need to practice, practice, practice
- You may feel frustrated at times, but persevere and *it will be worth it!*

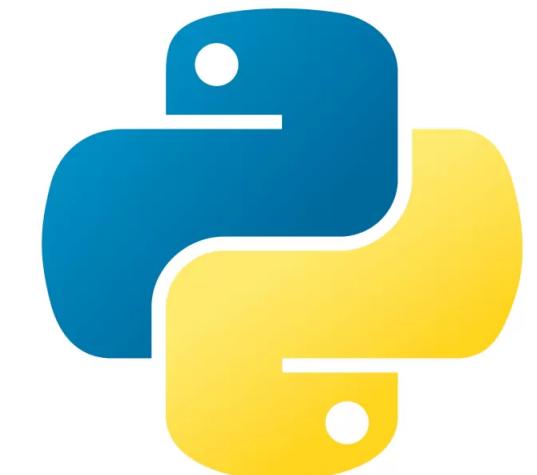
Your instructors will strive to make this process as smooth as possible

- Maintain a safe, respectful, and constructive learning environment
- Encourage questions and foster curiosity (there are no “dumb” questions)
- Provide support and resources to help you thrive and learn
- 2-semester format is intended to allow you to absorb information and reduce stress

Learning objectives



Languages we will work with



	<i>strengths</i>	<i>weaknesses</i>
	<p>Designed for statistics and visualization Good for numerical modeling Widely used in biological research</p>	<p>Awkward for tasks outside its niche Size of data sets limited by RAM Slow, limited graphics capabilities</p>
	<p>Designed for relational databases Extraordinarily powerful in this role Dominant database environment</p>	<p>Nearly useless outside its niche Procedural programming is awkward Not as flexible as some other QLs</p>
	<p>Nearly universal computing environment Extremely powerful, uniquely capable The most future-proof way to code</p>	<p>Terse syntax, not easily readable Does not provide much feedback Designed to work primarily with text</p>
	<p>General-purpose high-level language Readable code, consistent syntax Among most widely used languages</p>	<p>Not specialized: 2nd best at everything Slower than some other GPLs</p>

Learning objectives: the fine print

How to work with data:

Format data so that it is easy to work with

Reproducibly identify missing data, improperly formatted data, and outliers

How to write code:

Works as intended

Makes sense to future-you and others

Saves time and effort, reduces errors, and improves reproducibility

Can be re-purposed for other tasks, even unrelated ones

How to produce visuals:

Explore and understand complex data sets with ease

Portray your results to others with integrity, clarity, and accuracy

Customized to your precise specifications

Preview of topics

session	topics
1 - 10	R : importing, transforming, querying, and displaying data; Quarto
11 - 12	SQL : designing and querying relational databases
13	R : factors, date/time, geospatial
	winter break
14 - 19	UNIX : working with the command line and with text and tabular files
20 - 27	Python : programming; working with data in Python; Jupyter

Introduction to programming with R

Why learn about R?

Designed specifically for data analysis, modeling, and visualization

Designed by statisticians for statisticians: rigorous, tested, and respected

Free, open-source, non-proprietary, cross-platform, actively maintained

Encourages best practices in reproducibility, visualization, and sharing

Huge set of tested libraries for specialized tasks

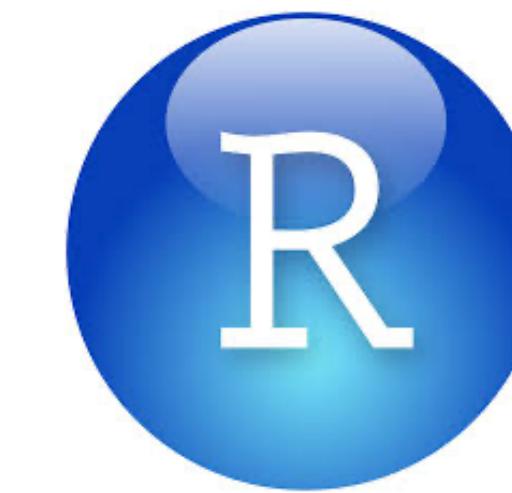
Used by biologists in many areas of study, familiar to many collaborators

Numerous libraries specific for biological research (e.g., GIS, modeling, genomics)

What is the difference between R and RStudio?



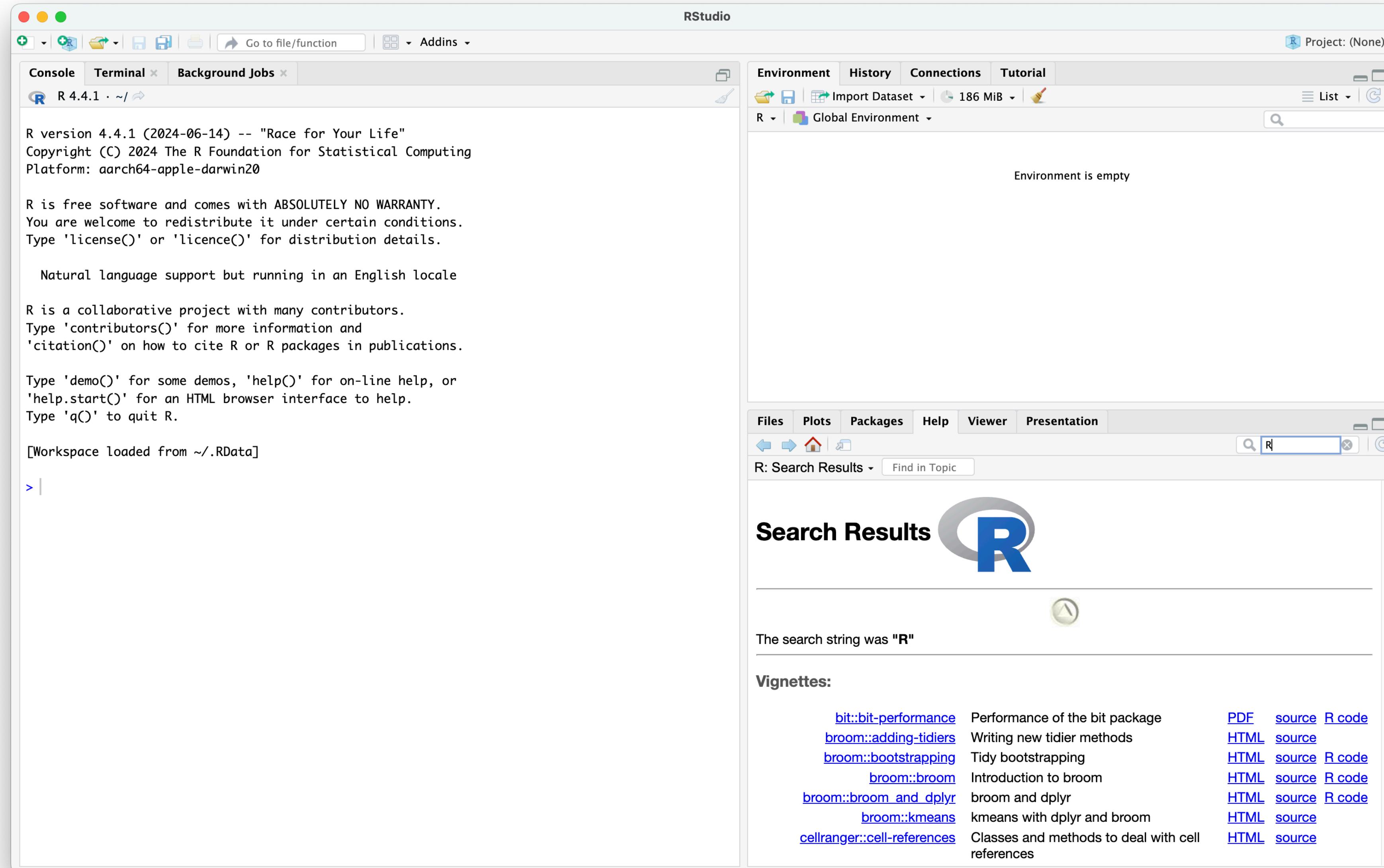
programming language
does the computing
“the engine”



programming environment
provides an interface to R
“the dashboard”



The RStudio interface



Typing at the console

If you enter an expression at the console, R will return the result:

```
> 6 * 7                      # multiplication  
[1] 42  
  
> sqrt(9)                    # using a function to find the square root  
[1] 3
```

To store a result, assign it to a variable name:

```
> my_result <- 6 * 7          # <- stores the result of an expression  
> print(my_result)           # displays the value stored in my_result  
[1] 42
```

When you quit RStudio, any stored values will be lost! The solution is to store code.

Creating a script

click on the left-most button and select R Script →

The screenshot shows the RStudio interface. On the left, a file browser sidebar is open, showing various file types like R Script, Quarto Document, and R Notebook. The 'R Script' option is highlighted with a red arrow pointing to it. The main workspace shows a script named 'ground Jobs.R' with the following content:

```
14) -- "Race for Your Life"
Foundation for Statistical Computing
darwin20

comes with ABSOLUTELY NO WARRANTY.
tribute it under certain conditions.
ence()' for distribution details.

but running in an English locale

ject with many contributors.
r more information and
ite R or R packages in publications.

emos, 'help()' for on-line help, or
ML browser interface to help.

/.RData]
```

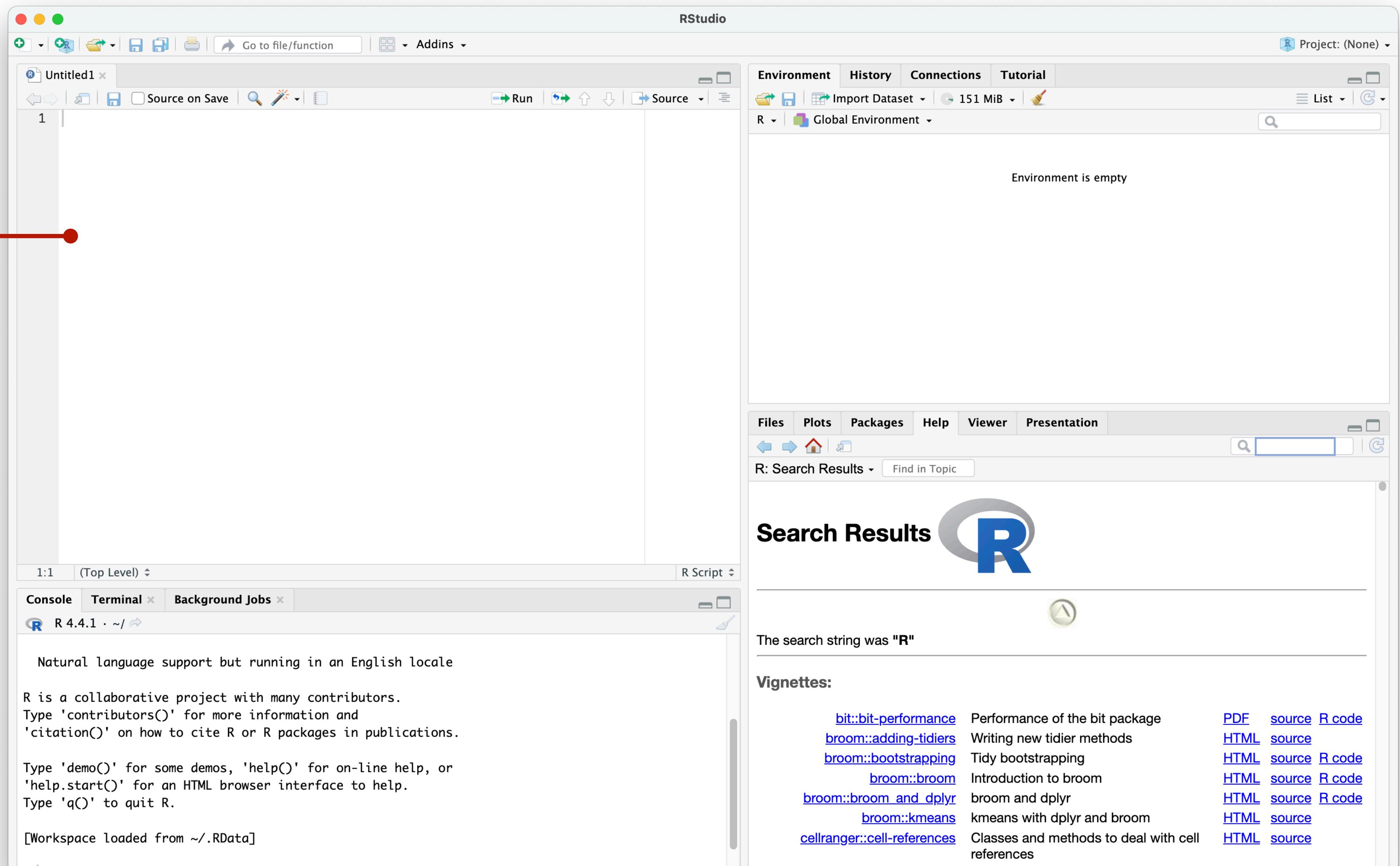
Below the script, the R console window shows the following session:

```
> 6 * 7
[1] 42
> my_result <- 6 * 7
> print(my_result)
[1] 42
>
```

The top right corner shows the RStudio environment tab bar with tabs for Environment, History, Connections, and Tutorial. The Environment tab is active, showing a value 'my_result' with the value '42'. The bottom right corner shows a search results panel for the term 'R', listing vignettes such as bit::bit-performance, broom::adding-tidiers, broom::bootstrapping, broom::broom, broom::broom_and_dplyr, broom::kmeans, and cellranger::cell-references, along with links to PDF, source, and R code.

The RStudio interface

type code here and
save for later use



Store code in your script

Let's write some code:

```
# my script to learn about R and RStudio  
  
my_result <- 6 * 7  
print(my_result)  
  
my_message <- 'hello world'  
print(my_message)
```

Save the file to preserve your work — you can retrieve it later by opening the script
Scripts are very useful for: saving time, record keeping, reproducibility, reusability

Some tips for writing scripts

Use comments at the beginning of the script to indicate overall purpose, date, author, etc.

Use comments throughout to explain what your code does

Use descriptive names for variables

Use whitespace for readability (blank lines and extra spaces are generally ignored)

Save time!

Use **tab** to autocomplete names of functions and variables

Use arrow keys to select the desired value when there are multiple options

Use keyboard shortcuts:

<- **opt + -** on Mac, **alt + -** on Windows (note: minus key)

Full list under Help menu, Keyboard Shortcuts Help

The RStudio interface

The screenshot displays the RStudio interface with several key components visible:

- Environment pane:** Shows the global environment with objects like `df` (344 obs. of 8 variables) and `p` (List of 11). The `df` object is expanded, showing columns such as `species`, `island`, `bill_length_mm`, etc.
- History pane:** Shows the command history with the following entries:

```
R 4.4.1 · ~/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/
> df <- read.csv('penguins.csv')
> library('tidyverse')
> p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
> p
Warning message:
Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
> |
```
- Console pane:** Shows the command history and any output or messages generated by the R code.
- Script pane:** Displays the R script file `RStudio_demo.R` with the following content:

```
3 # the console and environment
4 result <- 22 / 7
5 print(result)
6 print(pi)
7
8 message <- "hello world"
9 print(message)
10
11 # R has many built-in functions with help and examples
12 my_data <- c(2, 5, 14, -93)
13 the_average <- mean(my_data)
14 print(the_average)
15 help("mean")
16
17 # viewing and setting the working directory
18 getwd()
19 setwd("/Users/gwray/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/")
20
21 # import, view, and plot data
22 df <- read.csv('penguins.csv')
23 library('tidyverse')
24 p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
25 p
26
```
- Help pane:** Shows the documentation for the `mean` function, including the title "Arithmetic Mean", description, usage, and arguments.

An orange box highlights the **Console** pane, and an orange arrow points to it from the label "console" on the left.

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

Arguments

- x** an R object. Currently there are methods for numeric/logical vectors and [date](#), [date-time](#) and [time interval](#) objects. Complex vectors are allowed for `trim = 0`, only.

The RStudio interface

program —

The screenshot displays the RStudio interface with the following components:

- Script Editor (Top Left):** Shows the R script `RStudio_demo.R` containing code related to arithmetic mean calculations and ggplot2 usage.
- Console (Bottom Left):** Shows the R session output, including the execution of the script and a warning message about removed rows.
- Environment Browser (Top Right):** Displays the global environment with objects `df` and `p`.
- Documentation Viewer (Bottom Right):** Provides detailed information about the `mean` function, including its description, usage, arguments, and examples.

The RStudio interface

A screenshot of the RStudio interface. The left side shows an R script file named "RStudio_demo.R" with code demonstrating basic R operations like arithmetic, printing, and data manipulation. The right side shows the RStudio environment, which includes the Environment pane (highlighted with a green border), the History pane, the Connections pane, and the Tutorial pane. The Environment pane displays the global environment with objects like "df" (a data frame with 344 observations and 8 variables) and "p" (a list of 11 items). Below the environment is the R Help pane, which is currently displaying the documentation for the "mean" function. A green arrow points from the word "environment" in the title to the Environment pane.

environment

```
3 # the console and environment
4 result <- 22 / 7
5 print(result)
6 print(pi)
7
8 message <- "hello world"
9 print(message)
10
11 # R has many built-in functions with help and examples
12 my_data <- c(2, 5, 14, -93)
13 the_average <- mean(my_data)
14 print(the_average)
15 help("mean")
16
17 # viewing and setting the working directory
18 getwd()
19 setwd("/Users/gwray/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/")
20
21 # import, view, and plot data
22 df <- read.csv('penguins.csv')
23 library('tidyverse')
24 p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
25 p
26
```

21:30 (Top Level) R Script

Console Terminal × Background Jobs ×

```
R 4.4.1 · ~/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/
> df <- read.csv('penguins.csv')
> library('tidyverse')
> p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
> p
Warning message:
Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
> |
```

Environment History Connections Tutorial

R Global Environment

Data

- df 344 obs. of 8 variables
 - \$ species : chr "Adelie" "Adelie" "Adelie" ...
 - \$ island : chr "Torgersen" "Torgersen" "Torgersen" ...
 - \$ bill_length_mm : num 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
 - \$ bill_depth_mm : num 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
 - \$ flipper_length_mm: int 181 186 195 NA 193 190 181 195 193 190 ...
 - \$ body_mass_g : int 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
 - \$ sex : chr "male" "female" "female" NA ...
 - \$ year : int 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
- p List of 11

Values

- result 3.14285714285714

Files Plots Packages Help Viewer Presentation

R: Arithmetic Mean Find in Topic

mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

```
## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

x an R object. Currently there are methods for numeric/logical vectors and [date](#), [date-time](#) and [time interval](#) objects. Complex vectors are allowed for trim = 0, only.

trim the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed.

The RStudio interface

The screenshot displays the RStudio interface with the following components:

- Script Editor (Left):** Shows an R script named "RStudio_demo.R" containing code related to arithmetic mean calculations and data visualization.
- Environment Browser (Top Right):** Shows the global environment with objects like "df" (a data frame) and "p" (a list).
- Help Viewer (Bottom Right):** A detailed help page for the "mean" function, including sections for Description, Usage, Arguments, and Examples. A red box highlights the "mean" function entry in the search results.
- Console (Bottom Left):** Shows the R command-line interface with the same code executed, including a warning message about removed rows.

A pink arrow points from the text "help" to the help viewer window.

```
3 # the console and environment
4 result <- 22 / 7
5 print(result)
6 print(pi)
7
8 message <- "hello world"
9 print(message)
10
11 # R has many built-in functions with help and examples
12 my_data <- c(2, 5, 14, -93)
13 the_average <- mean(my_data)
14 print(the_average)
15 help("mean")
16
17 # viewing and setting the working directory
18 getwd()
19 setwd("/Users/gwray/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/")
20
21 # import, view, and plot data
22 df <- read.csv('penguins.csv')
23 library('tidyverse')
24 p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
25 p
26
```

```
21:30 (Top Level) R Script
```

```
R 4.4.1 · ~/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/
> df <- read.csv('penguins.csv')
> library('tidyverse')
> p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
> p
Warning message:
Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
>
```

Environment: Project: (None)

Environment History Connections Tutorial

Import Dataset 264 MB List C

Global Environment

Data

- df 344 obs. of 8 variables
 - \$ species : chr "Adelie" "Adelie" "Adelie" ...
 - \$ island : chr "Torgersen" "Torgersen" "Torgersen" ...
 - \$ bill_length_mm : num 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
 - \$ bill_depth_mm : num 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
 - \$ flipper_length_mm: int 181 186 195 NA 193 190 181 195 193 190 ...
 - \$ body_mass_g : int 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
 - \$ sex : chr "male" "female" "female" NA ...
 - \$ year : int 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
- p List of 11

Values

- result 3.14285714285714

Files Plots Packages Help Viewer Presentation

R: Arithmetic Mean Find in Topic

mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

```
## Default S3 method:  
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

x an R object. Currently there are methods for numeric/logical vectors and [date](#), [date-time](#) and [time interval](#) objects. Complex vectors are allowed for trim = 0, only.

trim the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed.

help

The RStudio interface

The screenshot displays the RStudio interface with the following components:

- Script Editor:** Shows an R script named "RStudio_demo.R" with code demonstrating basic R operations like arithmetic, printing, and using built-in functions.
- Console:** Displays the R command-line interface showing the execution of the script and the resulting output, including a warning message about removed rows.
- Environment Browser:** Shows the global environment with objects like "df" (a data frame with 344 observations and 8 variables) and "p" (a list of 11 items).
- Plots:** A scatter plot titled "body_mass_g" vs "flipper_length_mm". The x-axis ranges from 170 to 230, and the y-axis ranges from 3000 to 6000. The plot shows a positive correlation between flipper length and body mass.

A pink rectangular box highlights the "Plots" tab in the bottom navigation bar of the plot viewer, and a pink arrow points from this box to the word "plots" in pink text on the right side of the interface.

```
3 # the console and environment
4 result <- 22 / 7
5 print(result)
6 print(pi)
7
8 message <- "hello world"
9 print(message)
10
11 # R has many built-in functions with help and examples
12 my_data <- c(2, 5, 14, -93)
13 the_average <- mean(my_data)
14 print(the_average)
15 help("mean")
16
17 # viewing and setting the working directory
18 getwd()
19 setwd("/Users/gwray/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/")
20
21 # import, view, and plot data
22 df <- read.csv('penguins.csv')
23 library('tidyverse')
24 p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
25 p
26
```

```
R 4.4.1 · ~/Documents/Courses/Bio_724D_24/Slide_decks/01_intro/
> df <- read.csv('penguins.csv')
> library('tidyverse')
> p <- ggplot(df, aes(flipper_length_mm, body_mass_g)) + geom_point()
> p
Warning message:
Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
>
```

Environment | History | Connections | Tutorial | Import Dataset | 264 MB | Run | Source | Addins | Project: (None)

Global Environment | Data | Values | result: 3.14285714285714

File | Plots | Packages | Help | Viewer | Presentation | Zoom | Export | Publish

body_mass_g

flipper_length_mm

plots

