Introduction to the Tidyverse SISMID 2024 Module 1

Artwork by @allison_horst

R Studio, R Markdown, and best practices for coding in the Tidyverse

Learning objectives

- Navigate around and work within the R Studio IDE
- Create R Markdown scripts that weave together text, code, and outputs
- Use common coding best practices to write streamlined and readable scripts

Outline

1. **R Studio: what is it, why should you use it, and how do you use it?**
2. R Markdown: what is it, why should you use it, and how do you use it?
3. Tidyverse style guide and coding best practices

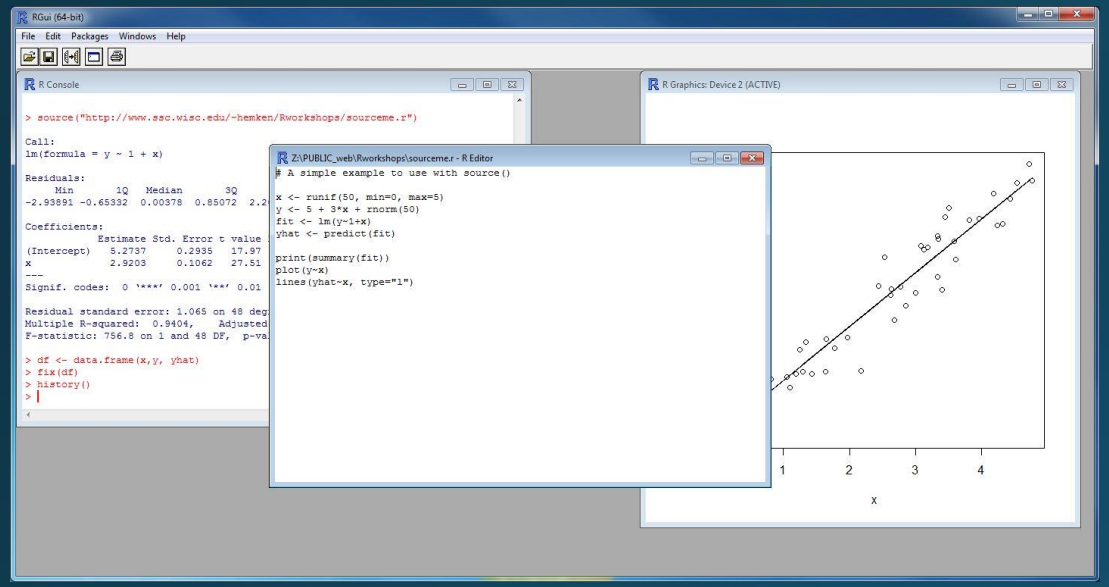
R Studio:

What is it?

Why should I use it?

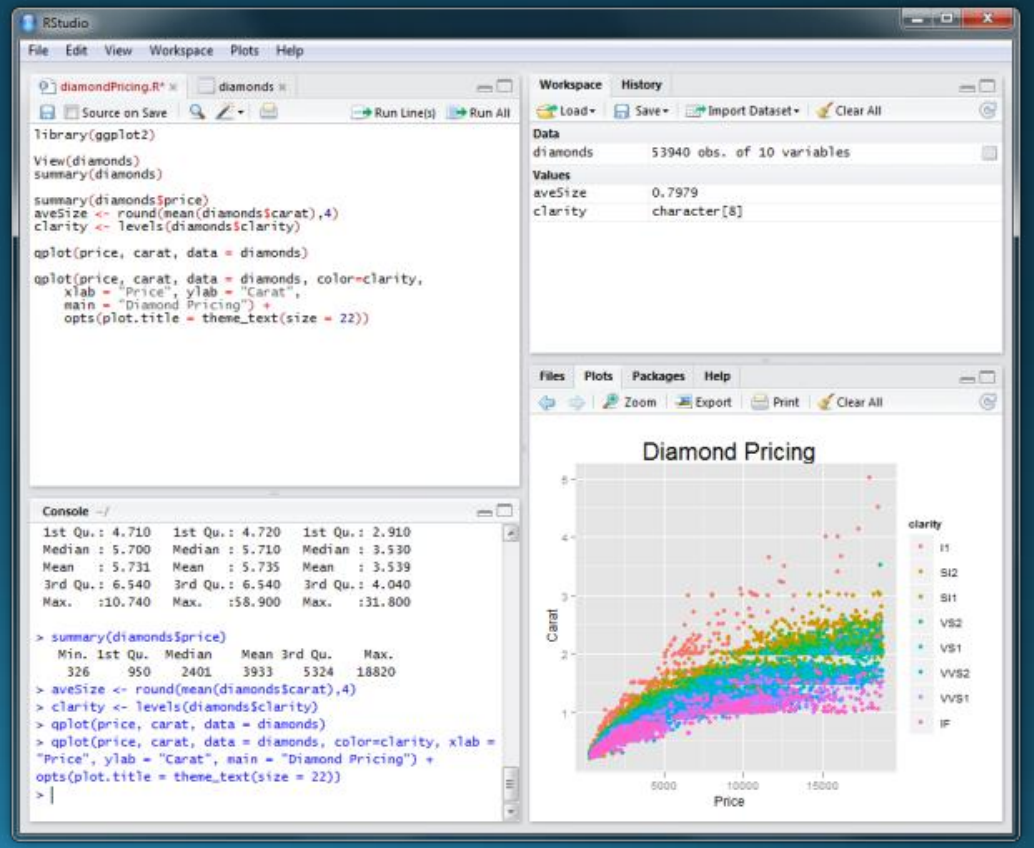
How do I use it?

R vs. RStudio®



R is a coding language

RStudio® is an IDE
(integrated development environment)



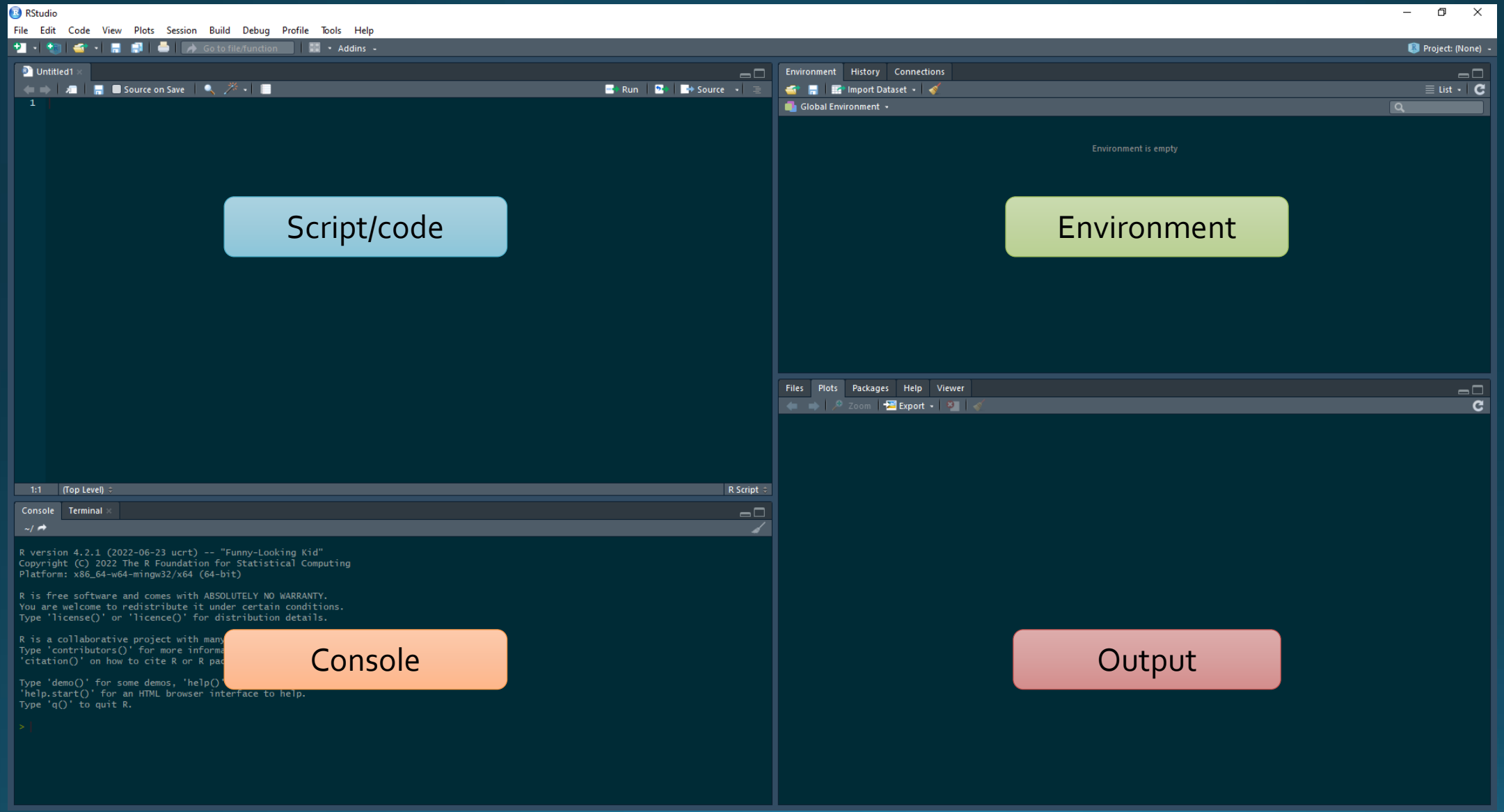
Why should I use RStudio®?

Provides one interface where all aspects development can be done

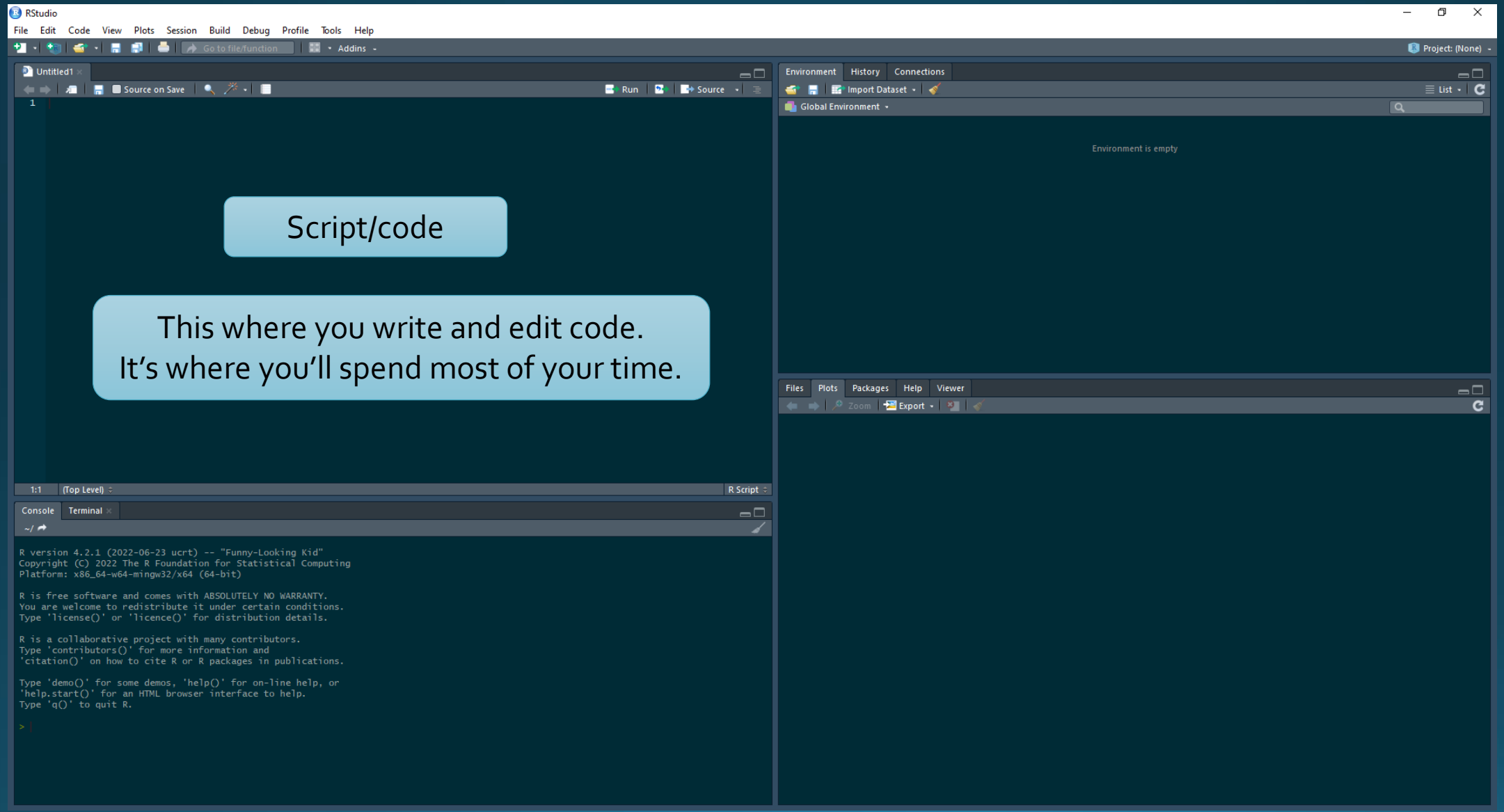
Authoring	Modifying	Compiling	Debugging	Deploying

Decreases start-up time and increases productivity

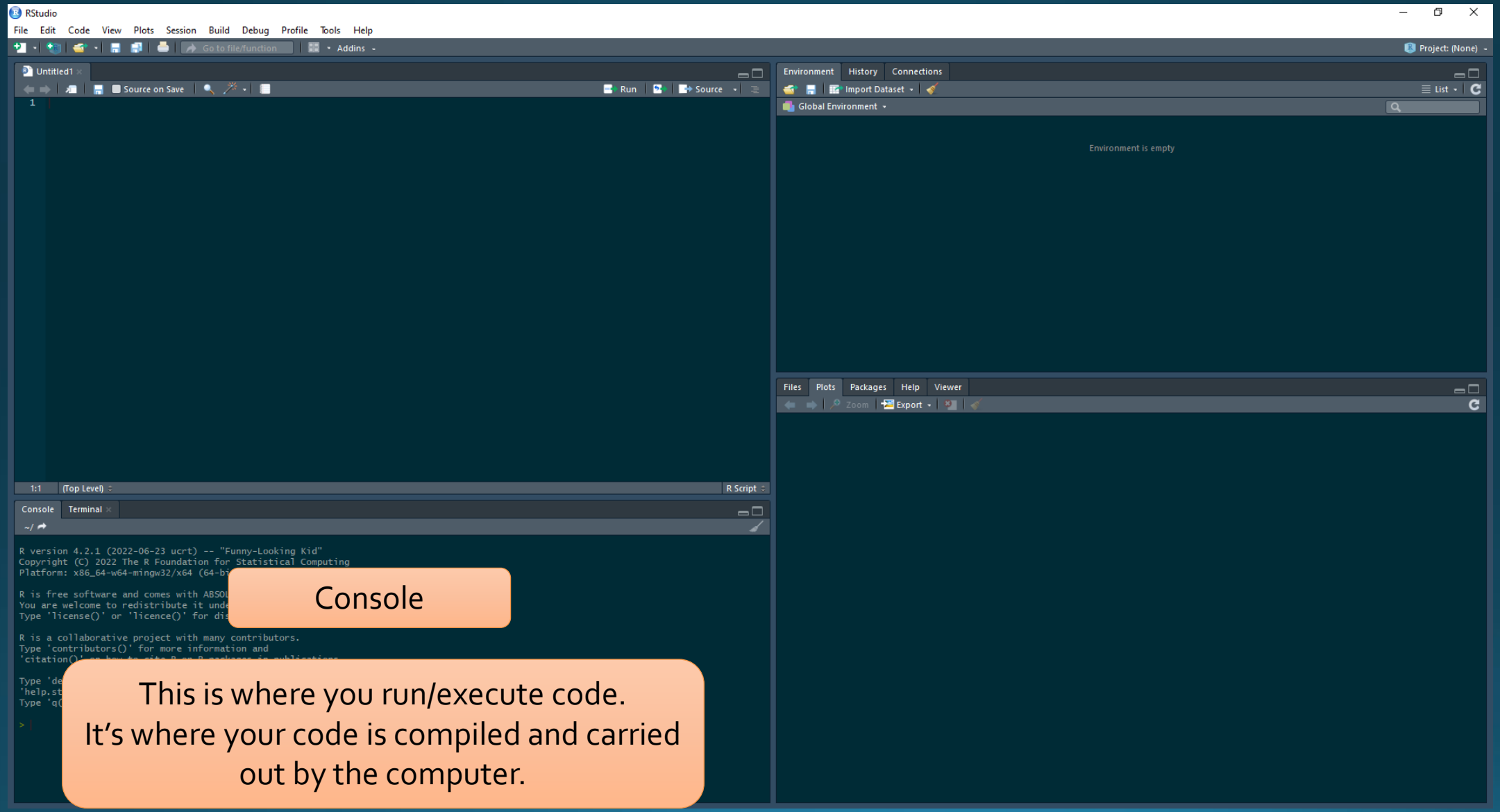
Components of RStudio® IDE



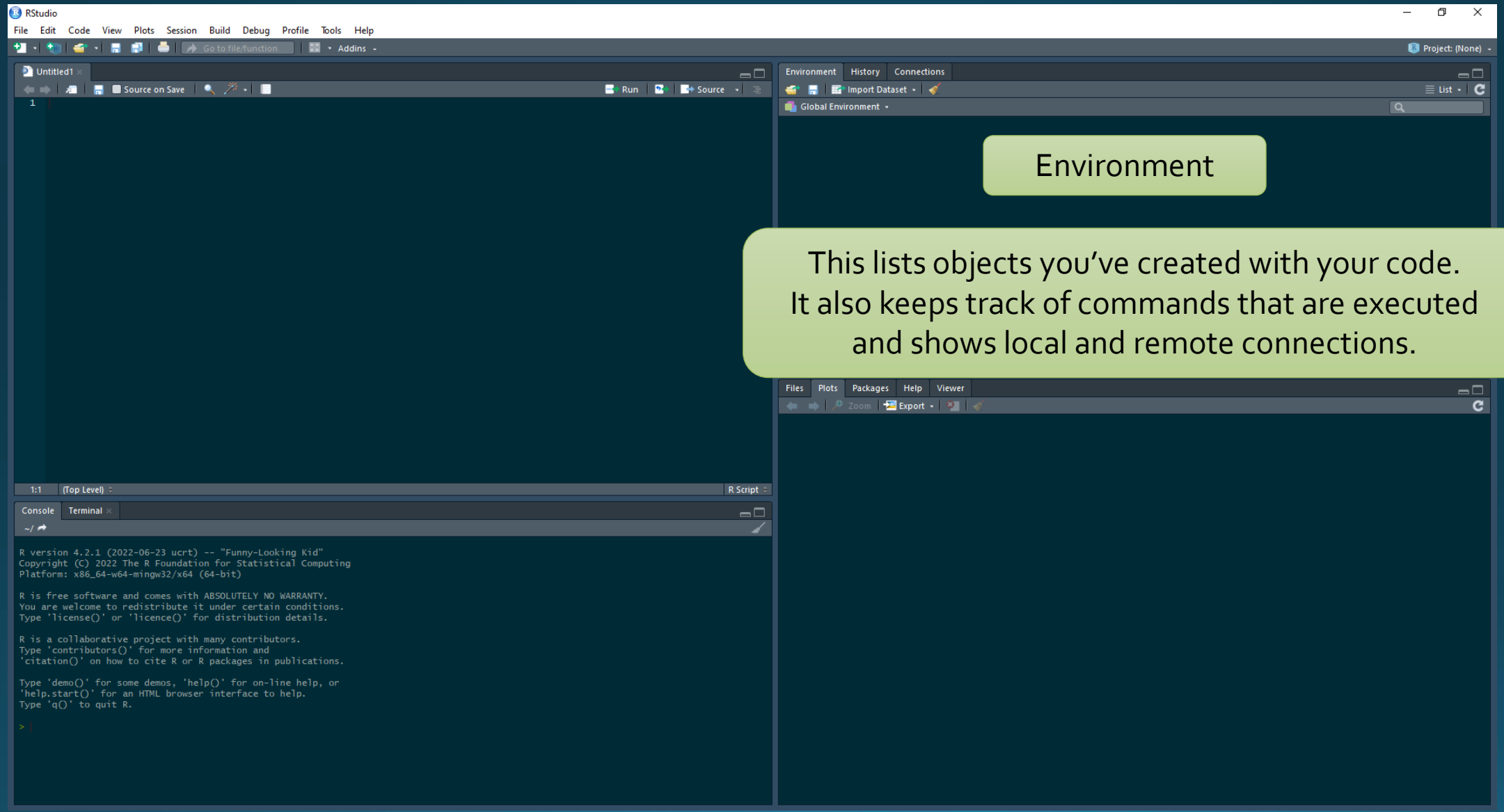
Components of RStudio® IDE



Components of RStudio® IDE



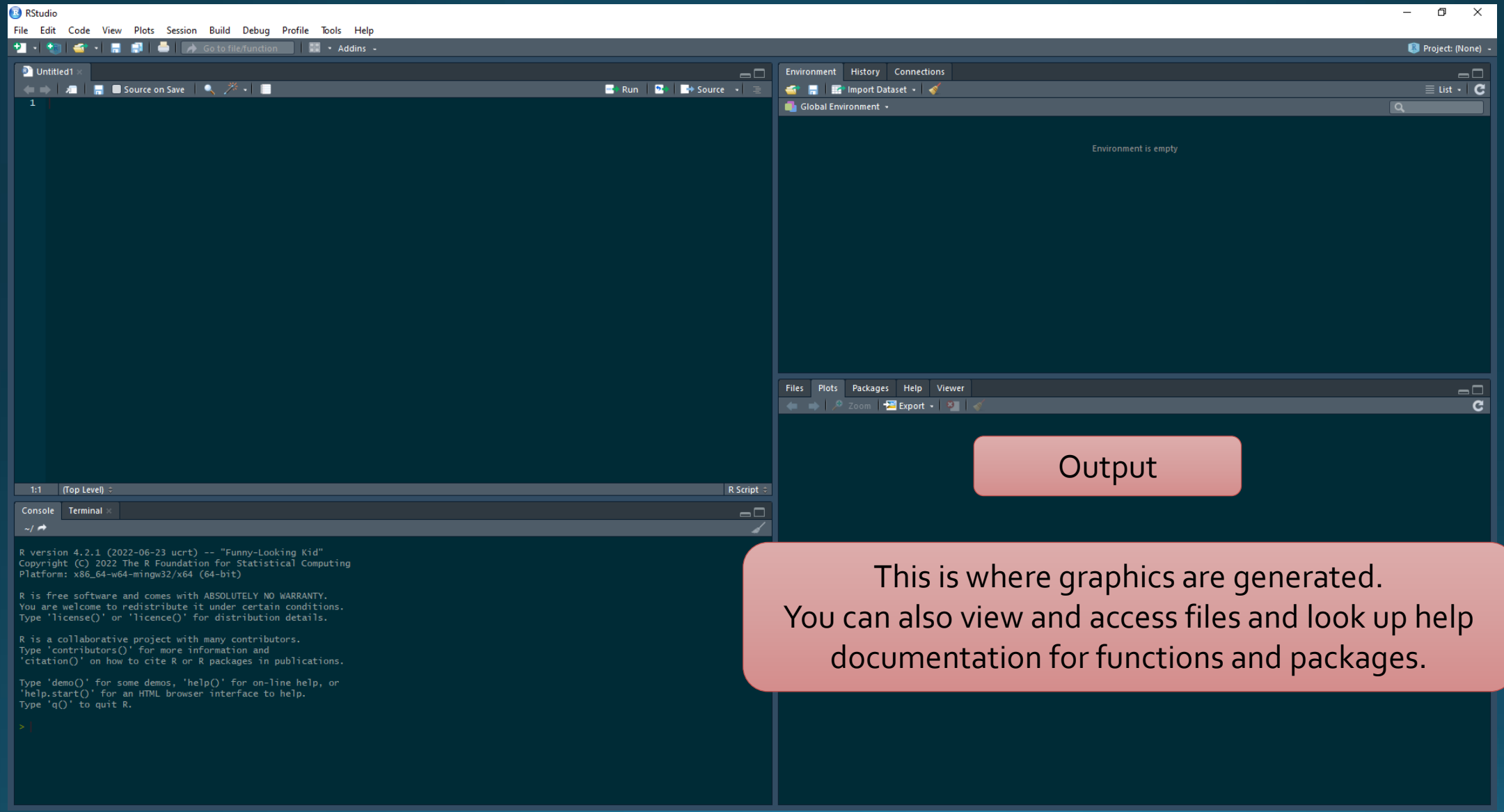
Components of RStudio® IDE



Environment

This lists objects you've created with your code.
It also keeps track of commands that are executed and shows local and remote connections.

Components of RStudio® IDE



Output

This is where graphics are generated.
You can also view and access files and look up help documentation for functions and packages.

Demos in R Studio

- Open + save a script
- Submit code to console
- Set working directory
- Global options
- Package and function help

Additional resources:

R Studio cheat sheet web page

https://rstudio.github.io/cheatsheets/rstudio-ide.pdf

Outline

1. R Studio: what is it, why should you use it, and how do you use it?
2. **R Markdown: what is it, why should you use it, and how do you use it?**
3. `Tidyverse` style guide and coding best practices

R Markdown:

What is it?
Why should I use it?
How do I use it?

Artwork by @allison_horst

What is R Markdown?

- **Markdown** is a formatting syntax used to create documents
- **R Markdown** is an extension of this syntax that allows us to integrate text, R code, and code outputs (e.g., numerical results, tables, plots, maps)

Illustrations by Alison Hill and Allison Horst, for RStudio.

How does R Markdown work?

- Using a single R Markdown file, we can:
  - Save and execute code (i.e., run analyses) **and**
  - Generate high-quality reports (i.e., communicate results)
- We do this by interspersing plain texts with "chunks" of code

chunks.Rmd

Knit HTML | Chunks

```
1  R Code Chunks
2  ==========
3
4  With R Markdown, you can insert R code
   chunks including plots:
5
6  ```{r qplot, fig.width=4, fig.height=3,
   message=FALSE}
7  # quick summary and plot
8  library(ggplot2)
9  summary(cars)
10 qplot(speed, dist, data=cars) +
11       geom_smooth()
12 ```
13
```

This code produces this output

R Code Chunks

Plain text

With R Markdown, you can insert R code chunks including plots:

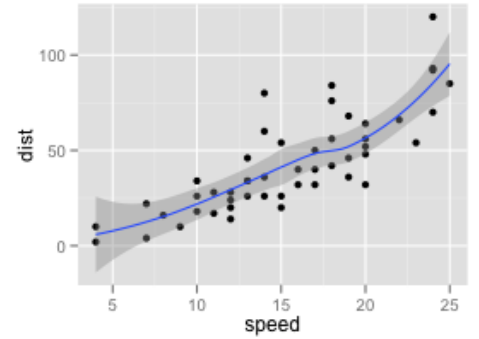
```
# quick summary and plot
library(ggplot2)
summary(cars)
```

Code chunk

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2
##  1st Qu.:12.0   1st Qu.: 26
##  Median :15.0   Median : 36
##  Mean   :15.4   Mean   : 43
##  3rd Qu.:19.0   3rd Qu.: 56
##  Max.   :25.0   Max.   :120
```

Numerical results

```
qplot(speed, dist, data = cars) + geom_smooth()
```



Plot

Reasons why R Markdown is awesome

Reproducibility

- I can give someone my code file + data file, and they can recreate everything I've done with a single click

Analysis_code.Rmd

Raw_data.txt
OR
Raw_data.csv
OR
Raw_data.xlsx

Final_report.PDF

Reasons why R Markdown is awesome

Supports multiple document types

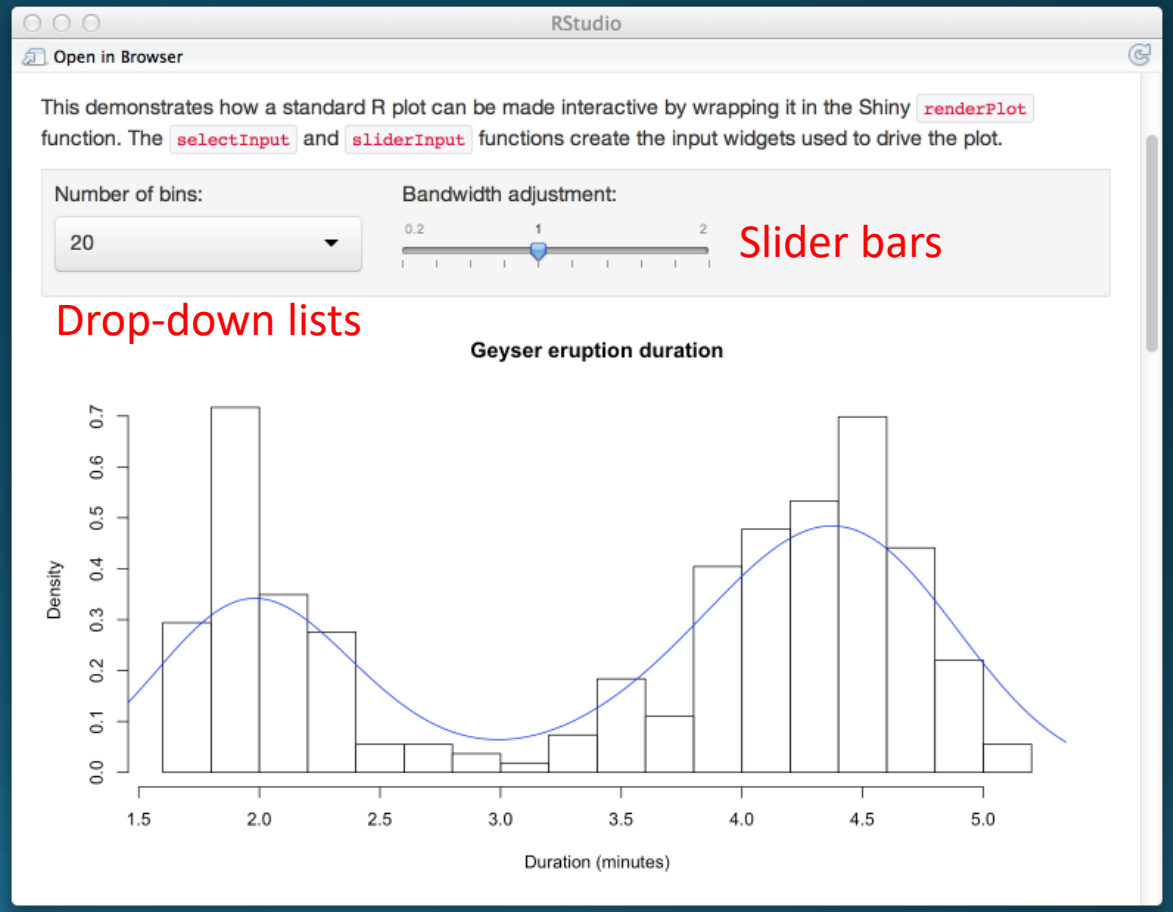
- Reports/manuscripts
  - Word
  - PDF
  - HTML
- Presentations
  - HTML
  - Beamer
  - ioslides
  - Powerpoint
- Dashboards
- Websites

Illustrations by Alison Hill and Allison Horst, for RStudio.

Reasons why R Markdown is awesome

Static or dynamic (interactive) output

Output changes within the document when parameters are changed



Reasons why R Markdown is awesome

Supports multiple coding languages

Helpful when you know how to perform a specialized task in a language other than R

Bash

Python

SQL

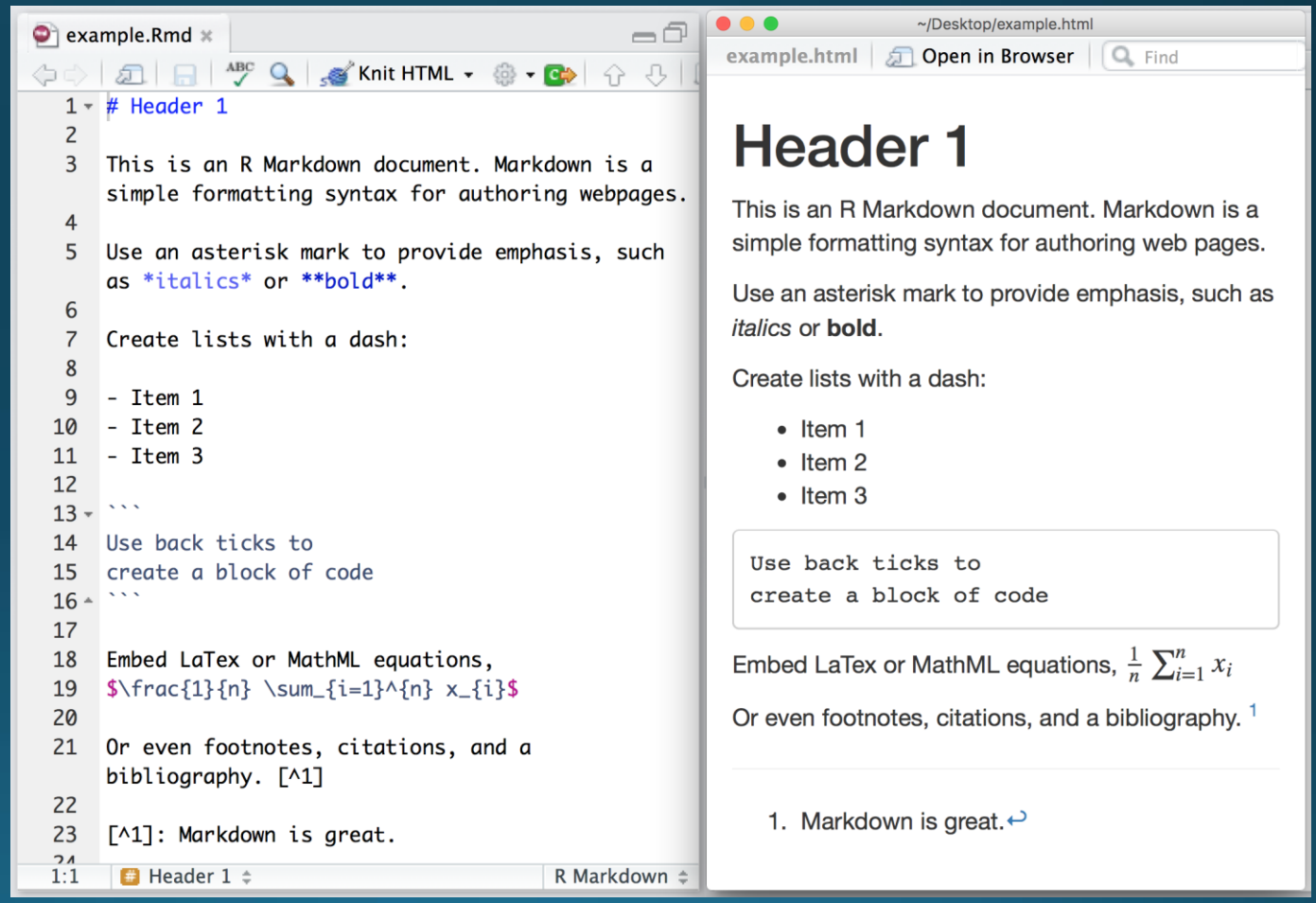
C++

JavaScript

CSS

Reasons why R Markdown is awesome

Document formatting is hard-coded instead of point-and-click



How I've used R Markdown

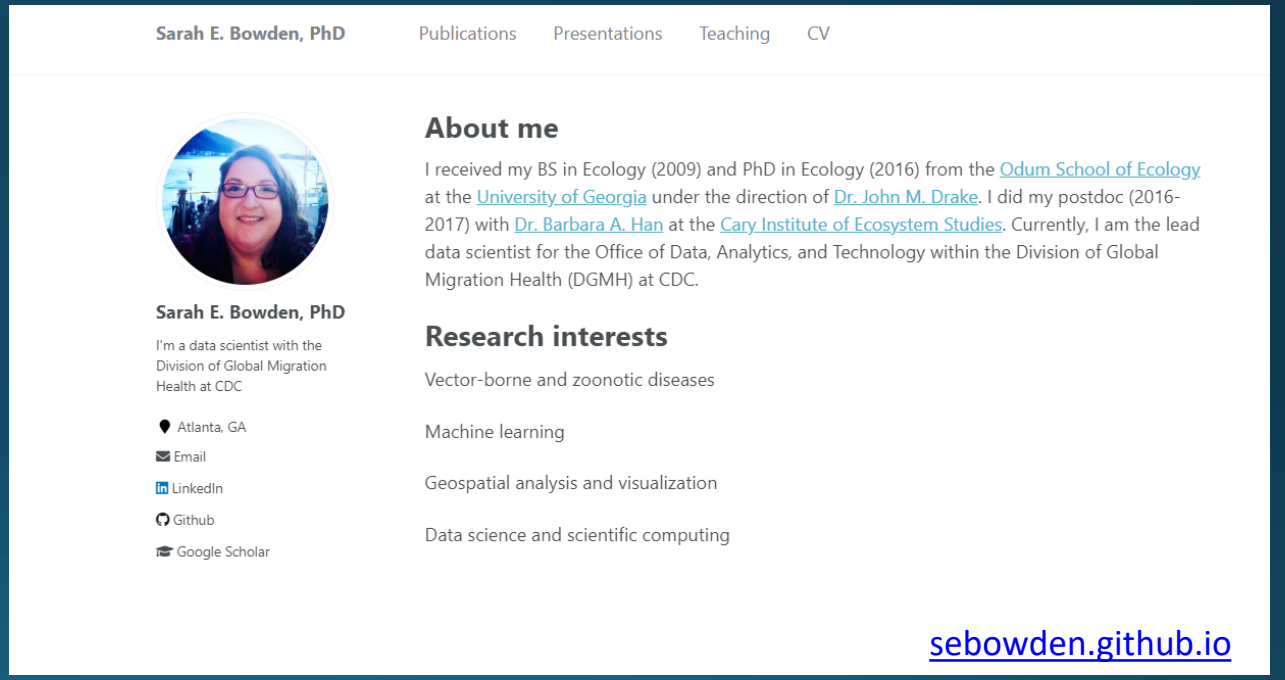
Formatting my CV

Creating coding tutorials

Creating presentations

Writing full scientific manuscripts

Creating a professional web page



R Markdown demos

- Open+ save .Rmd file
- Knit example doc
- Make a change/addition to example doc and re-knit
- New code chunk
- Chunk options

Additional resources:

R Markdown cheat sheet web page

R Markdown cheat sheet PDF

R Markdown gallery

Outline

1. R Studio: what is it, why should you use it, and how do you use it?
2. R Markdown: what is it, why should you use it, and how do you use it?
3. **`Tidyverse` style guide and coding best practices**

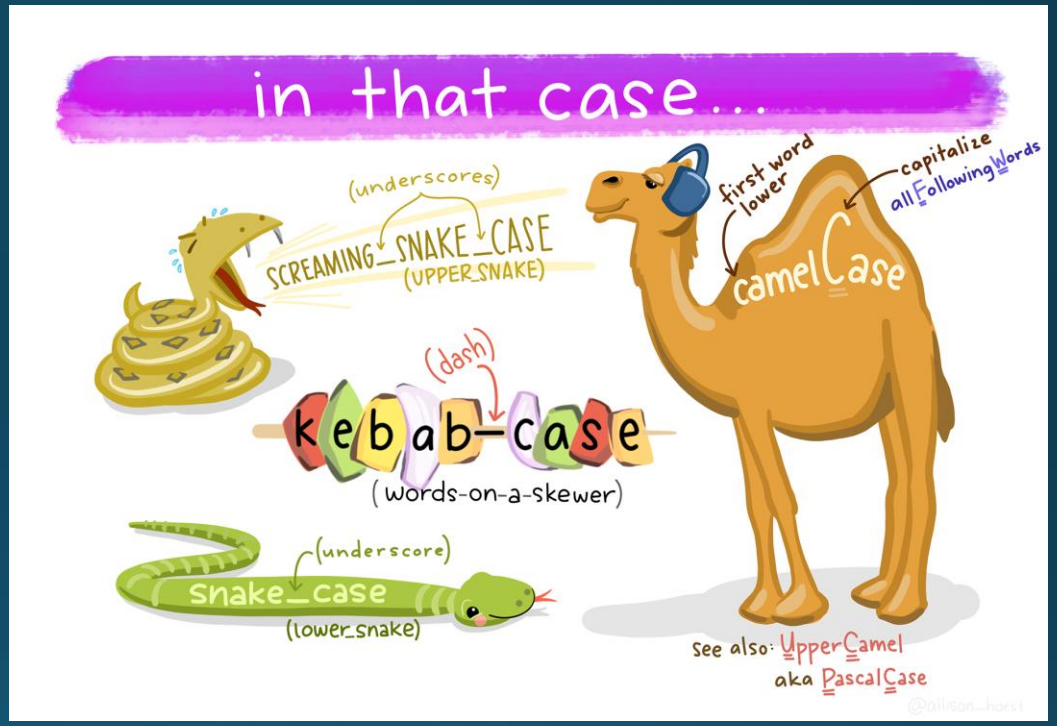
Tidyverse:

Style guide +

Coding best practices

Naming: files, objects, and functions

- Snake case preferred
  - my_variable ✓
  - my_file.R ✓
  - not.dot.case ✗
  - OrCamelCase ✗



- File names should be human and machine readable
  - Id30583_January_from_dave.csv ✗
  - drc_international_travel_volume_2022.csv ✓

Artwork by @allison_horst

Syntax

- Spacing
  - Space after a comma, not before
    - `X[ , 1]` ✓
    - `X[ , 1]` ✗
  - No spaces around parentheses for function calls
    - `sum(x, na.rm = TRUE)` ✓
    - `sum ( x, na.rm = TRUE )` ✗
  - Most operators (+, -, =, etc.) should have a space before and after
    - `y <- (x + 3) * z` ✓
    - `y <- (x+3)*z` ✗
- Indentation and alignment for readability
- Arrow assignment: `<-` is preferred over `=`

Styler package

- Reformats according to Tidyverse style guide
- R Studio add-in
- Install and demo

Other resources:

Tidyverse style guide

Styler package

Revisiting our learning objectives

- Navigate around and work within the R Studio IDE
- Create R Markdown scripts that weave together text, code, and outputs
- Use common coding best practices to write streamlined and readable scripts