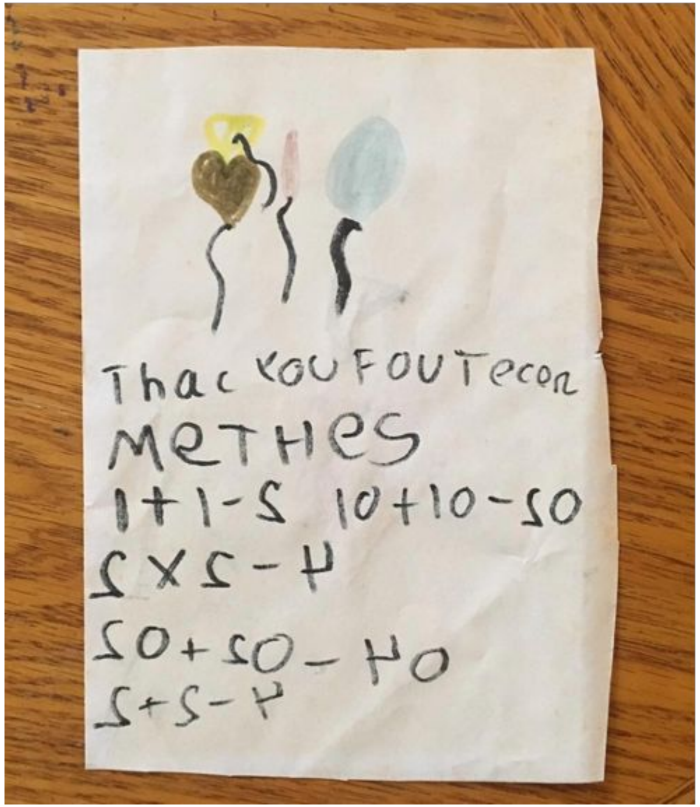
Data Science Basics in R

Day 1: Introduction to Statistical Programming

Introductions

- Your name
- What you do for school, work, and/or fun
- Why you signed up for this course
- Have you ever used R before?

Introductions



Thac YOU FOU Tecor
METHeS
1+1-2 10+10-20
2x2-4
4-2x2
20+20-40
2+2-4

Tools for Health Security
Nita Madhav
Stephanie Eaneff
Beth Cameron
Global Health Security 2019
Progress to date, Opportunities for the Future

Housekeeping notes

- Take a break whenever you need one, we will also have a few structured breaks as a larger group
- Outlet locations
- Trash cans
- Try to come with a charged laptop
- If you have questions, you can find me at *sde31@georgetown.edu*

Housekeeping notes

All course materials are available on github, and we’ll talk more about github in general later in this course.

https://github.com/seaneff/data-science-basics-2024

What to expect: Learning R

- Learning R is fun! And also frustrating.
- You won’t be an expert by the end of this week.
- But over time and as you practice, it gets easier!

What to expect: The next week

- We'll balance slides/demos with hands-on exercises.
- You decide how you learn best...
  - listening with your computer away
  - laptop out and typing along
  - taking notes with paper/pen
- All of these materials are publicly accessible on github.

Workshop goals

This workshop will build literacy and basic proficiency in statistical programming, with a focus on the skills needed to conduct data analyses in professional healthcare and public health workplaces.

We will cover the basics of data management, data cleaning, data visualization, and basic statistical calculations in R, and version control in github. Participants will leave with a small portfolio of relevant data visualizations and analyses completed using a real-world public health dataset.

Workshop goals

- Learning to program (in R) can be fun and creative, and doesn't have to be overwhelming or intimidating.
- Anyone can learn to write code.

Artwork by Allison Horst

Learning by doing

For some people, it’s easier to learn by doing, typing, and making mistakes. Others prefer to listen, think, and work through problems later on their own.

In this workshop, we’ll pause to do worked examples. Sometimes these will be confusing. This is the point! We will learn together by trial and error.

If you are more comfortable following along for now, feel free to just watch and try at home. But I really encourage you to try, the best way to learn R is to repeatedly do stuff wrong and then figure out the errors.

Artwork by Allison Horst

Goals for today

- Understand what statistical programming is
- Get acquainted with Rstudio
- Write your very first R code (at least, of this workshop)
    - vectors
    - functions
    - accessing documentation
- Explore github to access course materials

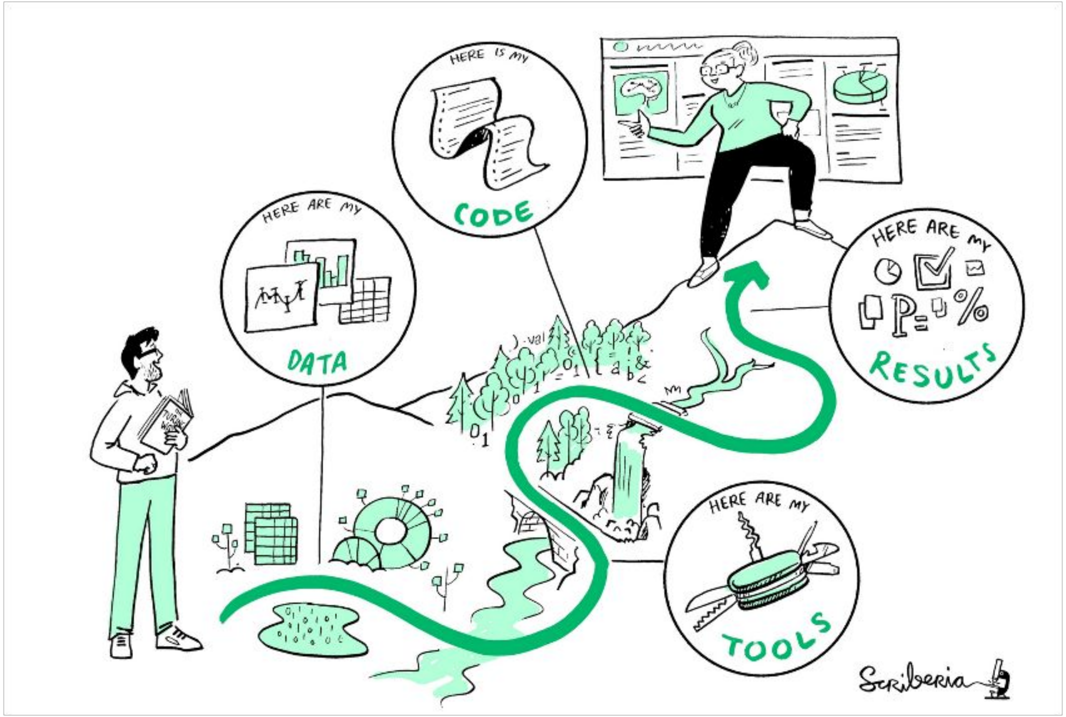
Goals for today

- Understand what statistical programming is
- Get acquainted with Rstudio
- Write your very first R code (at least, of this workshop)
    - vectors
    - functions
    - accessing documentation
- Explore github to access course materials

What is statistical programming, and why should I care?

What is statistical programming?

Statistical programming is using code to clean, analyze, visualize, and interpret data.



What is R?

- R is a programming language for statistical computing
- Created by Ross Ihaka and Robert Gentleman in 1996
- R is open-source and free
- Many people use R in different ways and for different purposes, but it's defined specifically for data analysis and visualization (unlike other open-source languages like python)

What is RStudio?

- R Studio is, put simply, a place to write and run R code
- It's an IDE (integrated development environment) and supports both R and python
- It's also free (with enterprise upgrades)

Why learn R?

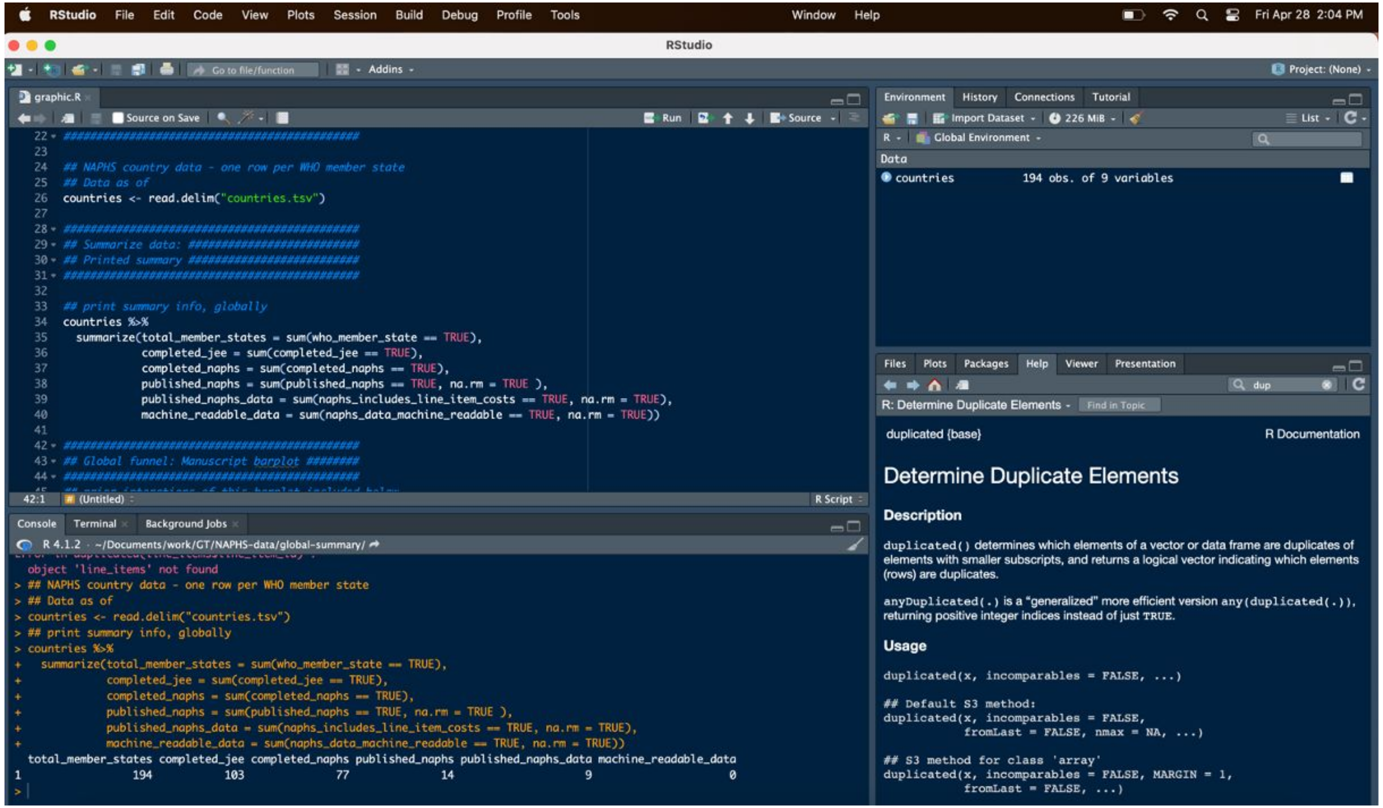
- Learning R helps you understand your data and understand how analysis works, whether you're a researcher, a data scientist, or someone who collaborates with folks who do analysis
- Coding helps you think rigorously about your questions
- It's free (vs. other more expensive tools like SAS or SPSS)
- Sharable, reproducible code and research
- Lots of academics/companies/agencies use it
- It's fun (honestly)

Goals for today

- Understand what statistical programming is
- Get acquainted with Rstudio
- Write your very first R code (at least, of this workshop)
  - vectors
  - functions
  - accessing documentation
- Explore github to access course materials

How do I use RStudio?

R Studio console



```r
##############################################

## NAPHS country data - one row per WHO member state
## Data as of
countries <- read.delim("countries.tsv")

##############################################
## Summarize data: ###########################
## Printed summary ###########################
##############################################

## print summary info, globally
countries %>%
  summarize(total_member_states = sum(who_member_state == TRUE),
            completed_jee = sum(completed_jee == TRUE),
            completed_naphs = sum(completed_naphs == TRUE),
            published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
            published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
            machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))

##############################################
## Global funnel: Manuscript barplot ########
##############################################
```

```
R 4.1.2 · ~/Documents/work/GT/NAPHS-data/global-summary/
  object 'line_items' not found
> ## NAPHS country data - one row per WHO member state
> ## Data as of
> countries <- read.delim("countries.tsv")
> ## print summary info, globally
> countries %>%
+   summarize(total_member_states = sum(who_member_state == TRUE),
+             completed_jee = sum(completed_jee == TRUE),
+             completed_naphs = sum(completed_naphs == TRUE),
+             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
+             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
+             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
  total_member_states completed_jee completed_naphs published_naphs published_naphs_data machine_readable_data
1                 194           103              77              14                    9                     0
> 
```

Data

countries   194 obs. of 9 variables

duplicated {base}   R Documentation

Determine Duplicate Elements

Description

`duplicated()` determines which elements of a vector or data frame are duplicates of elements with smaller subscripts, and returns a logical vector indicating which elements (rows) are duplicates.

`anyDuplicated(.)` is a "generalized" more efficient version `any(duplicated(.))`, returning positive integer indices instead of just `TRUE`.

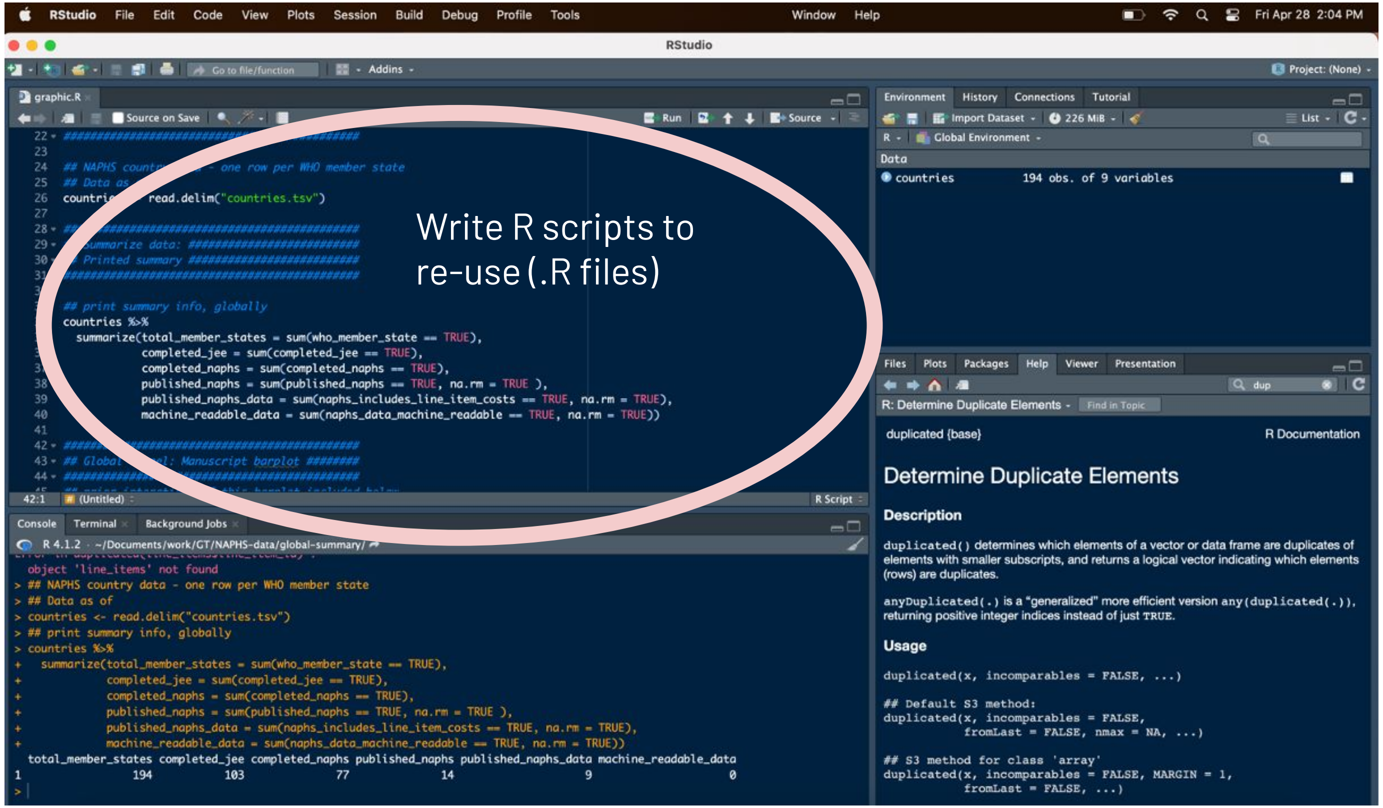
Usage

```r
duplicated(x, incomparables = FALSE, ...)

## Default S3 method:
duplicated(x, incomparables = FALSE,
           fromLast = FALSE, nmax = NA, ...)

## S3 method for class 'array'
duplicated(x, incomparables = FALSE, MARGIN = 1,
           fromLast = FALSE, ...)
```

R Studio console



Write R scripts to re-use (.R files)

```
##########################################
## NAPHS countr       - one row per WHO member state
## Data as
countrie    read.delim("countries.tsv")

##########################################
  Summarize data: ##########################
# Printed summary ##########################
##########################################

## print summary info, globally
countries %>%
  summarize(total_member_states = sum(who_member_state == TRUE),
            completed_jee = sum(completed_jee == TRUE),
            completed_naphs = sum(completed_naphs == TRUE),
            published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
            published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
            machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))

##########################################
## Global     el: Manuscript barplot ########
##########################################
```

```
R 4.1.2 · ~/Documents/work/GT/NAPHS-data/global-summary/
  object 'line_items' not found
> ## NAPHS country data - one row per WHO member state
> ## Data as of
> countries <- read.delim("countries.tsv")
> ## print summary info, globally
> countries %>%
+   summarize(total_member_states = sum(who_member_state == TRUE),
+             completed_jee = sum(completed_jee == TRUE),
+             completed_naphs = sum(completed_naphs == TRUE),
+             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
+             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
+             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
  total_member_states completed_jee completed_naphs published_naphs published_naphs_data machine_readable_data
1                 194           103              77              14                    9                     0
> 
```

```
Data
countries          194 obs. of 9 variables
```

R: Determine Duplicate Elements

duplicated {base} R Documentation

Determine Duplicate Elements

Description

`duplicated()` determines which elements of a vector or data frame are duplicates of elements with smaller subscripts, and returns a logical vector indicating which elements (rows) are duplicates.

`anyDuplicated(.)` is a "generalized" more efficient version `any(duplicated(.))`, returning positive integer indices instead of just `TRUE`.

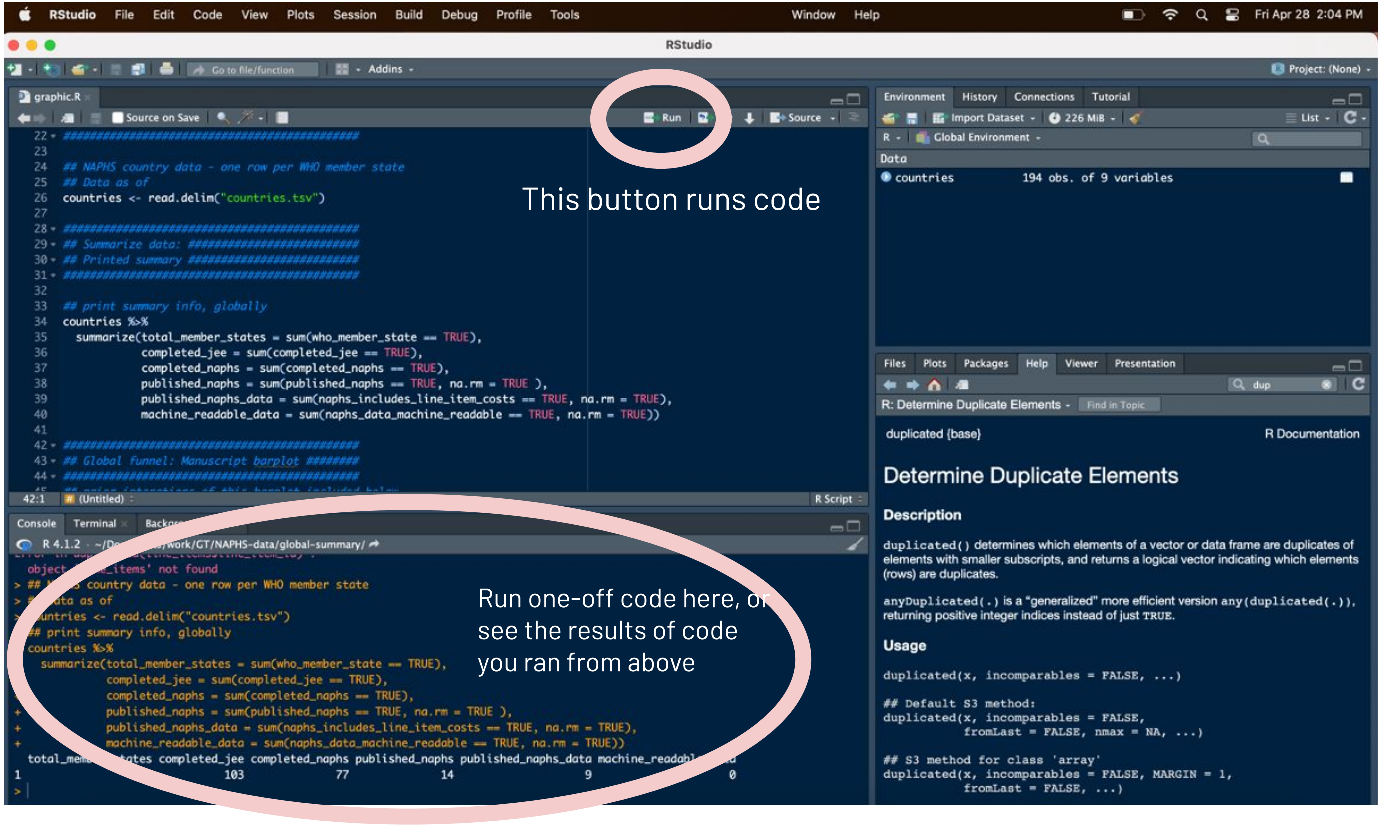
Usage

```
duplicated(x, incomparables = FALSE, ...)

## Default S3 method:
duplicated(x, incomparables = FALSE,
           fromLast = FALSE, nmax = NA, ...)

## S3 method for class 'array'
duplicated(x, incomparables = FALSE, MARGIN = 1,
           fromLast = FALSE, ...)
```

R Studio console



```r
###########################################

## NAPHS country data - one row per WHO member state
## Data as of
countries <- read.delim("countries.tsv")

###########################################
## Summarize data: ########################
## Printed summary ########################
###########################################

## print summary info, globally
countries %>%
  summarize(total_member_states = sum(who_member_state == TRUE),
            completed_jee = sum(completed_jee == TRUE),
            completed_naphs = sum(completed_naphs == TRUE),
            published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
            published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
            machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))

###########################################
## Global funnel: Manuscript barplot ########
###########################################
```

```
R 4.1.2 · ~/D[illegible]/work/GT/NAPHS-data/global-summary/
  object [illegible]e_items' not found
> ## [illegible] country data - one row per WHO member state
> #[illegible]ata as of
>[illegible]untries <- read.delim("countries.tsv")
  ## print summary info, globally
  countries %>%
    summarize(total_member_states = sum(who_member_state == TRUE),
              completed_jee = sum(completed_jee == TRUE),
              completed_naphs = sum(completed_naphs == TRUE),
+             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
+             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
+             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
  total_mem[illegible]tates completed_jee completed_naphs published_naphs published_naphs_data machine_readabl[illegible]
1                       103              77              14                    9                     0
>
```

```
Data
countries        194 obs. of 9 variables
```

R: Determine Duplicate Elements

duplicated {base} R Documentation

Determine Duplicate Elements

Description

`duplicated()` determines which elements of a vector or data frame are duplicates of elements with smaller subscripts, and returns a logical vector indicating which elements (rows) are duplicates.

`anyDuplicated(.)` is a "generalized" more efficient version `any(duplicated(.))`, returning positive integer indices instead of just `TRUE`.

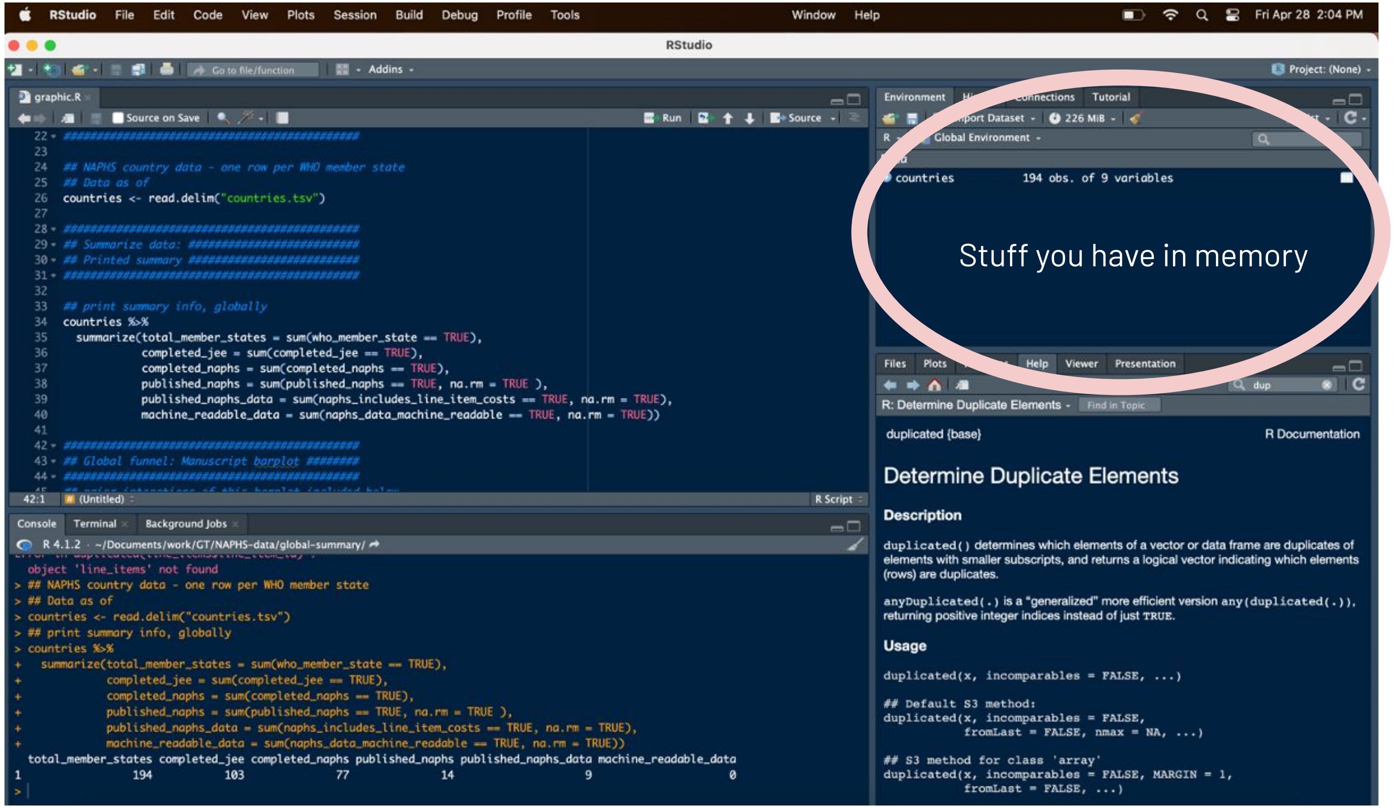
Usage

```r
duplicated(x, incomparables = FALSE, ...)

## Default S3 method:
duplicated(x, incomparables = FALSE,
           fromLast = FALSE, nmax = NA, ...)

## S3 method for class 'array'
duplicated(x, incomparables = FALSE, MARGIN = 1,
           fromLast = FALSE, ...)
```

R Studio console



Stuff you have in memory

```
countries          194 obs. of 9 variables
```

```r
############################################
## NAPHS country data - one row per WHO member state
## Data as of
countries <- read.delim("countries.tsv")

############################################
## Summarize data: #########################
## Printed summary #########################
############################################

## print summary info, globally
countries %>%
  summarize(total_member_states = sum(who_member_state == TRUE),
            completed_jee = sum(completed_jee == TRUE),
            completed_naphs = sum(completed_naphs == TRUE),
            published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
            published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
            machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))

############################################
## Global funnel: Manuscript barplot #######
############################################
```

```
  object 'line_items' not found
> ## NAPHS country data - one row per WHO member state
> ## Data as of
> countries <- read.delim("countries.tsv")
> ## print summary info, globally
> countries %>%
+   summarize(total_member_states = sum(who_member_state == TRUE),
+             completed_jee = sum(completed_jee == TRUE),
+             completed_naphs = sum(completed_naphs == TRUE),
+             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
+             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
+             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
  total_member_states completed_jee completed_naphs published_naphs published_naphs_data machine_readable_data
1                 194           103              77              14                    9                     0
> 
```

duplicated {base} R Documentation

Determine Duplicate Elements

Description

`duplicated()` determines which elements of a vector or data frame are duplicates of elements with smaller subscripts, and returns a logical vector indicating which elements (rows) are duplicates.

`anyDuplicated(.)` is a "generalized" more efficient version `any(duplicated(.))`, returning positive integer indices instead of just `TRUE`.

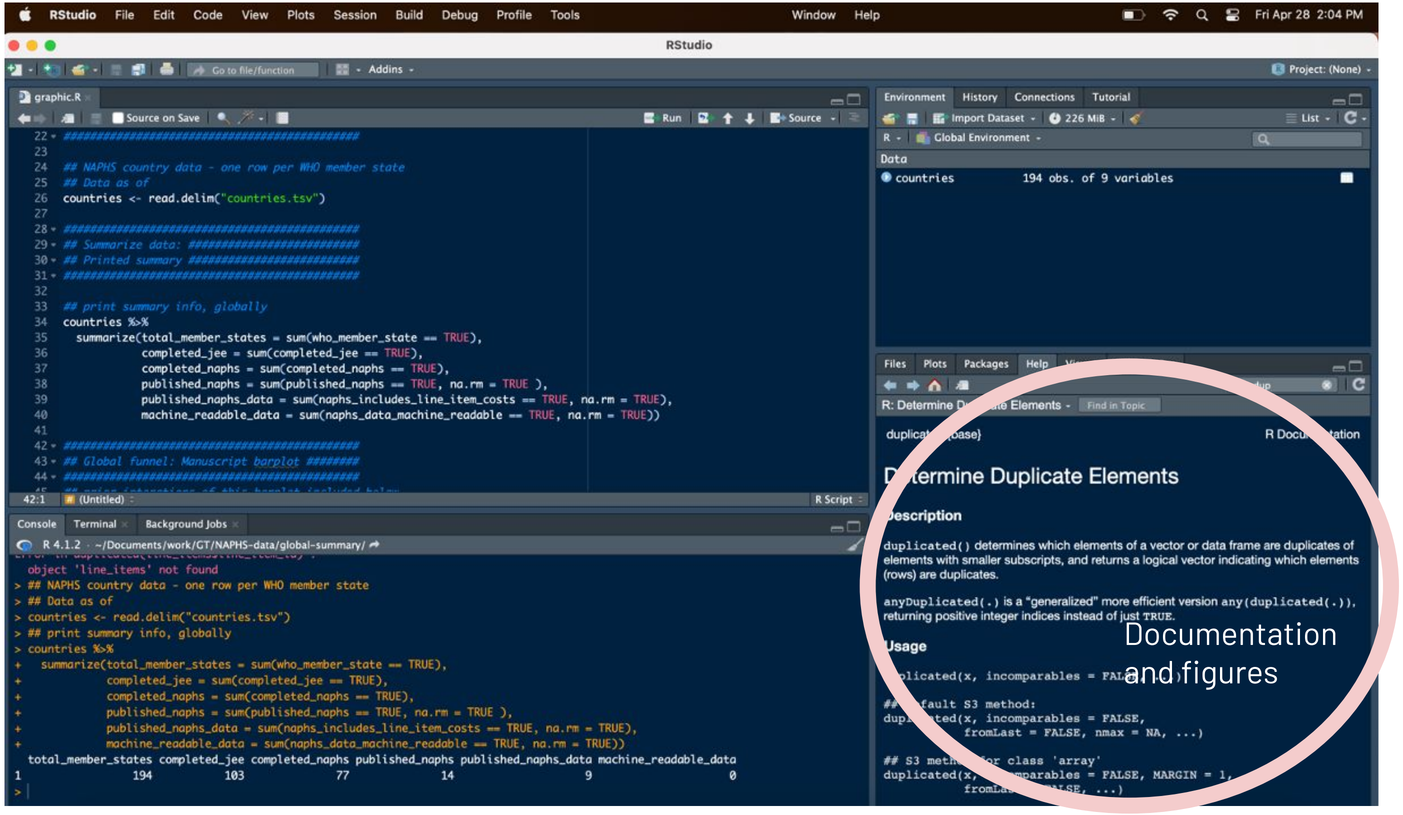
Usage

```
duplicated(x, incomparables = FALSE, ...)

## Default S3 method:
duplicated(x, incomparables = FALSE,
           fromLast = FALSE, nmax = NA, ...)

## S3 method for class 'array'
duplicated(x, incomparables = FALSE, MARGIN = 1,
           fromLast = FALSE, ...)
```

R Studio console



```r
############################################

## NAPHS country data - one row per WHO member state
## Data as of
countries <- read.delim("countries.tsv")

############################################
## Summarize data: ##########################
## Printed summary ##########################
############################################

## print summary info, globally
countries %>%
  summarize(total_member_states = sum(who_member_state == TRUE),
            completed_jee = sum(completed_jee == TRUE),
            completed_naphs = sum(completed_naphs == TRUE),
            published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
            published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
            machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))

############################################
## Global funnel: Manuscript barplot ########
############################################
```

```
R 4.1.2 · ~/Documents/work/GT/NAPHS-data/global-summary/
  object 'line_items' not found
> ## NAPHS country data - one row per WHO member state
> ## Data as of
> countries <- read.delim("countries.tsv")
> ## print summary info, globally
> countries %>%
+   summarize(total_member_states = sum(who_member_state == TRUE),
+             completed_jee = sum(completed_jee == TRUE),
+             completed_naphs = sum(completed_naphs == TRUE),
+             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE ),
+             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
+             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
  total_member_states completed_jee completed_naphs published_naphs published_naphs_data machine_readable_data
1                 194           103              77              14                    9                     0
> 
```

```
countries            194 obs. of 9 variables
```

duplicat {base}   R Docu tation

D termine Duplicate Elements

Description

`duplicated()` determines which elements of a vector or data frame are duplicates of elements with smaller subscripts, and returns a logical vector indicating which elements (rows) are duplicates.

`anyDuplicated(.)` is a "generalized" more efficient version `any(duplicated(.))`, returning positive integer indices instead of just `TRUE`.

Usage

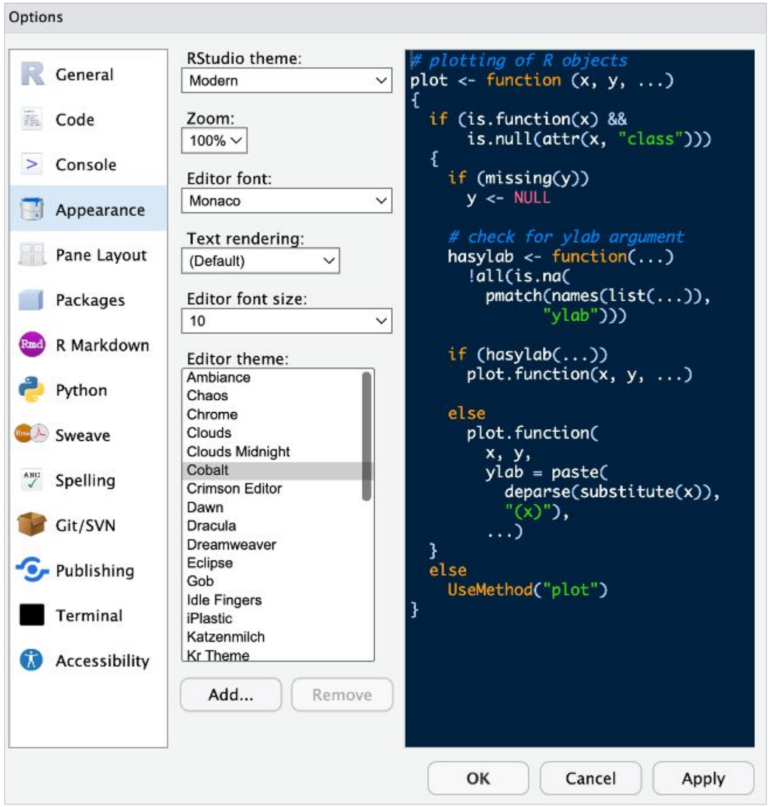
```
 plicated(x, incomparables = FAL

##  fault S3 method:
dup  ated(x, incomparables = FALSE,
           fromLast = FALSE, nmax = NA, ...)

## S3 meth  for class 'array'
duplicated(x,   mparables = FALSE, MARGIN = 1,
           fromLa    ALSE, ...)
```

Changing RStudio’s apperance

- Navigate from the top bar
- -> Tools
- -> Global Options
- -> Appearance
- (then click “apply”)



RStudio theme: Modern

Zoom: 100%

Editor font: Monaco

Text rendering: (Default)

Editor font size: 10

Editor theme: Cobalt

```
# plotting of R objects
plot <- function (x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
          "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)"),
        ...)
  }
  else
    UseMethod("plot")
}
```

15 minute break
(and a note on worked examples)

- If you haven’t already, try to download R and Rstudio before tomorrow’s class. I’ll be around by Zoom or email if you have any questions, and can help troubleshoot.
- Today, we’ll do some worked examples sharing my laptop. If you already have R and Rstudio installed on your laptop, feel free to follow along there.

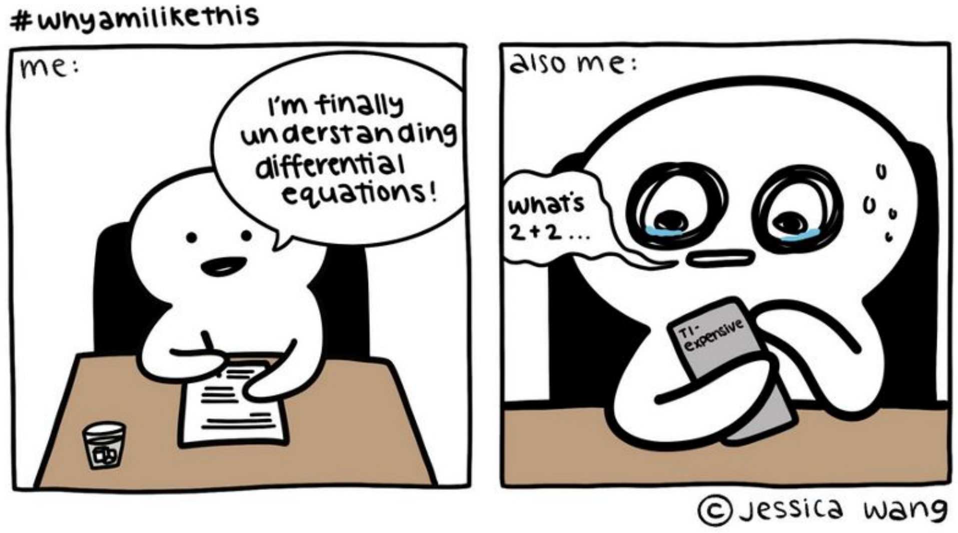
Goals for today

- Understand what statistical programming is
- Get acquainted with Rstudio
- Write your very first R code (at least, of this workshop)
  - vectors
  - functions
  - accessing documentation
- Explore github to access course materials

R Basics

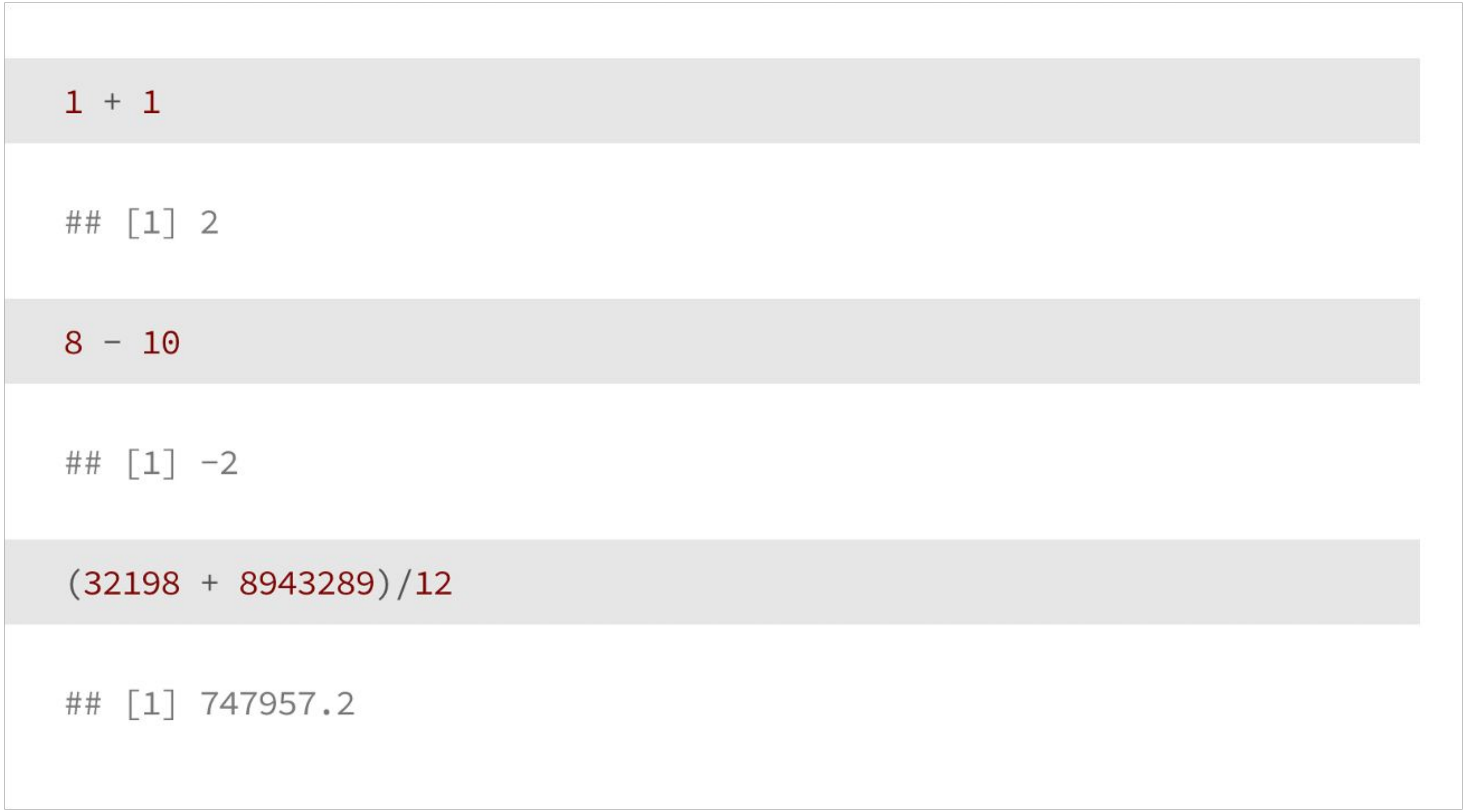
R as a calculator

- R can do everything a basic calculator can do
- Using R as a calculator is a great first step



Comic by Jessica Wang. Accessed online: https://i.redd.it/dmayt2tc3e551.jpg

Using R as a calculator



```r
1 + 1
```

```
## [1] 2
```

```r
8 - 10
```

```
## [1] -2
```

```r
(32198 + 8943289)/12
```

```
## [1] 747957.2
```

Using R as a calculator

```r
log(1)
```

```
## [1] 0
```

```r
sqrt(64)
```

```
## [1] 8
```

```r
abs(-14)
```

```
## [1] 14
```

Using R as a calculator
Symbols and syntax

- **Addition** (1+1)
- **Subtraction** (2-1)
- **Multiplication** (3*4)
- **Division** (7.2/9)
- **Exponents** (2^7)
- **Square root** (sqrt(9))
- **Order of operations** (7/(3*2))

Now you try!

- Use R to do some basic math
    - Add two numbers together
    - Multiply three or more numbers
    - Take the square root of a number

Objects

- An object is something you save to R's working memory
- It can be almost anything
  - A string (e.g., your name)
  - A number (e.g., 3.14)
  - A dataset (e.g., that file you have in Excel)
- We assign objects using a little arrow with the syntax (<-)
- When doing data analysis, the most common object you'll probably save is a dataframe, like an Excel or .csv file that you can access from within R (more on this later)

Objects

```r
my_first_object <- 3.14

my_first_object
```

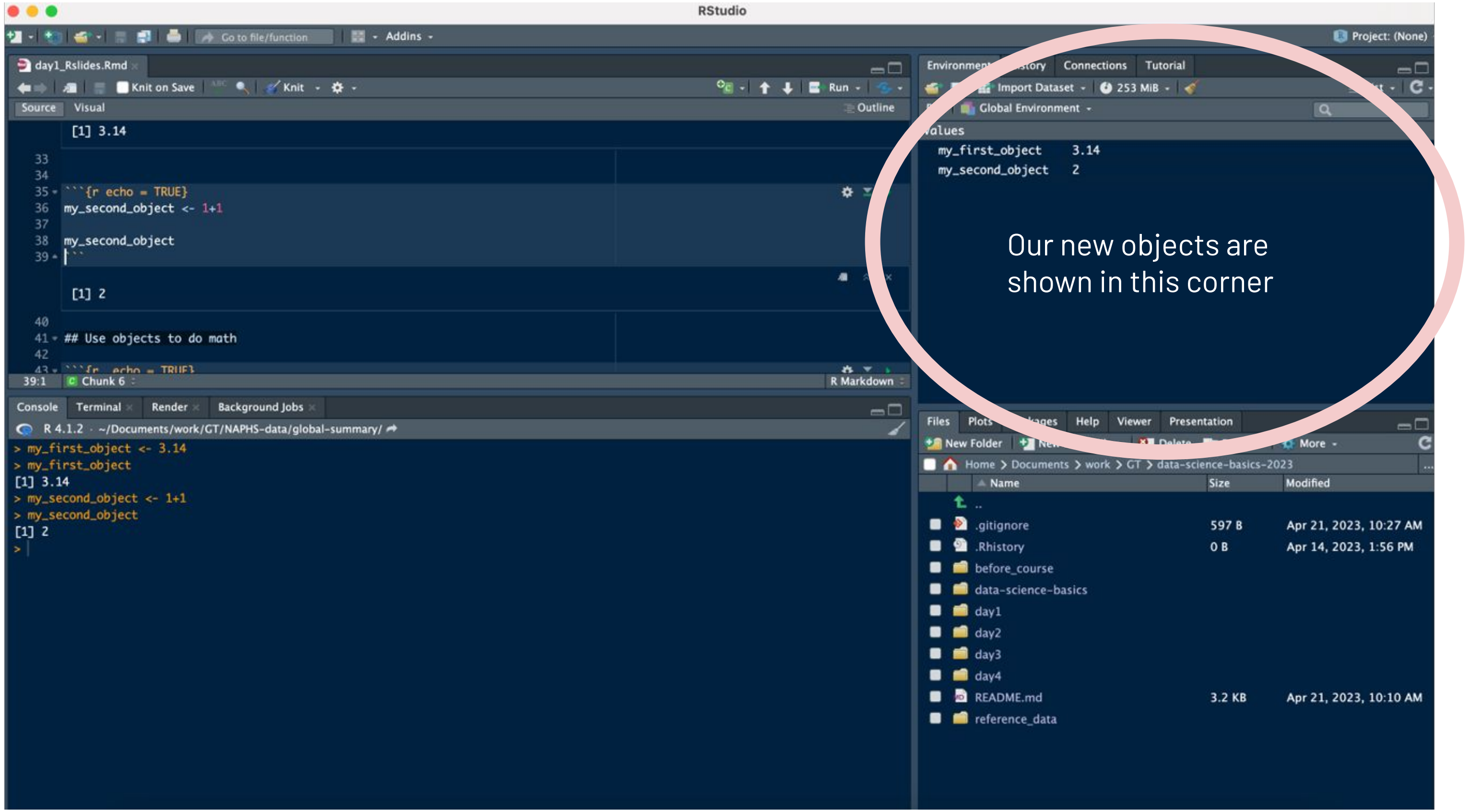
```
## [1] 3.14
```

```r
my_second_object <- 1+1

my_second_object
```

```
## [1] 2
```

Objects



Editor:

~~~
[1] 3.14

```{r echo = TRUE}
my_second_object <- 1+1

my_second_object
~~~

[1] 2

Use objects to do math

```{r echo = TRUE}
```

Console:

```
> my_first_object <- 3.14
> my_first_object
[1] 3.14
> my_second_object <- 1+1
> my_second_object
[1] 2
>
```

Environment:

```
Values
  my_first_object    3.14
  my_second_object   2
```

Files: Home > Documents > work > GT > data-science-basics-2023

Name	Size	Modified
.gitignore	597 B	Apr 21, 2023, 10:27 AM
.Rhistory	0 B	Apr 14, 2023, 1:56 PM
before_course		
data-science-basics		
day1		
day2		
day3		
day4		
README.md	3.2 KB	Apr 21, 2023, 10:10 AM
reference_data		

Our new objects are shown in this corner

Using (numeric) objects to do math

- Just like we did when we used R as a calculator, you can also use numeric objects to do math
- When you do this, the objects themselves don’t change unless you explicitly re-assign them to new variables

Using (numeric) objects to do math

```r
my_first_object * my_second_object
```

```
## [1] 6.28
```

```r
my_first_object/my_second_object
```

```
## [1] 1.57
```

```r
my_first_object^my_second_object
```

```
## [1] 9.8596
```

Now you try!

- Pick your favorite number, and save it as an object
- Pick another number, and save it as another object
- Do one basic calculation (e.g., addition) with your objects

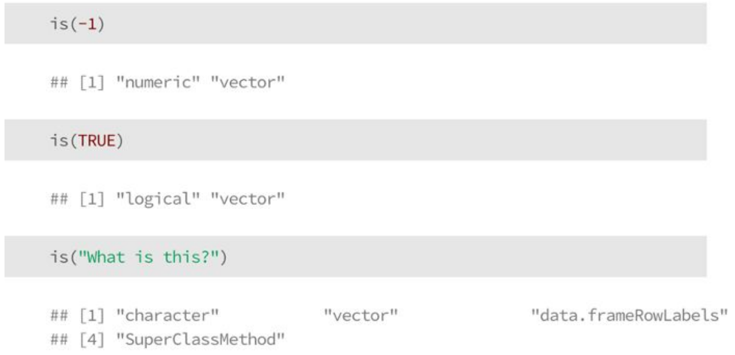
- You may run into issues. That's okay! We'll talk them through.

HEY, MISS LENHART! I FORGOT EVERYTHING ABOUT ALGEBRA THE MOMENT I GRADUATED, AND IN 20 YEARS NO ONE HAS NEEDED ME TO SOLVE *ANYTHING* FOR X!
I *TOLD* YOU I'D NEVER USE IT!
IN YOUR *FACE!*
IT'S WEIRD HOW PROUD PEOPLE ARE OF NOT LEARNING MATH WHEN THE SAME ARGUMENTS APPLY TO LEARNING TO PLAY MUSIC, COOK, OR SPEAK A FOREIGN LANGUAGE.

Source: XKCD

Data types in R

- **numeric:** a number (e.g., -1, 0, 893243.343)
- **logical:** TRUE or FALSE (no quotations)
- **character:** letters and words (tricky: or a number stored as letter!)
- The function is() helps us figure out what type of data we have



```r
is(-1)
```

```
## [1] "numeric" "vector"
```

```r
is(TRUE)
```

```
## [1] "logical" "vector"
```

```r
is("What is this?")
```

```
## [1] "character"           "vector"              "data.frameRowLabels"
## [4] "SuperClassMethod"
```

Numeric data

```r
my_first_object <- 3.14

my_first_object
```

```
## [1] 3.14
```

```r
my_second_object <- 1+1

my_second_object
```

```
## [1] 2
```

Character data

```r
policy <- "International Health Regulations (IHR)"
```

```r
organization <- "UNAIDS"
```

Logical data

```
logical_example <- TRUE
```

```
second_logical_example <- FALSE
```

Check your understanding!

```r
is(-1)
```

```r
is(TRUE)
```

```r
is("What is this?")
```

Check your understanding!

```r
is(-1)
```

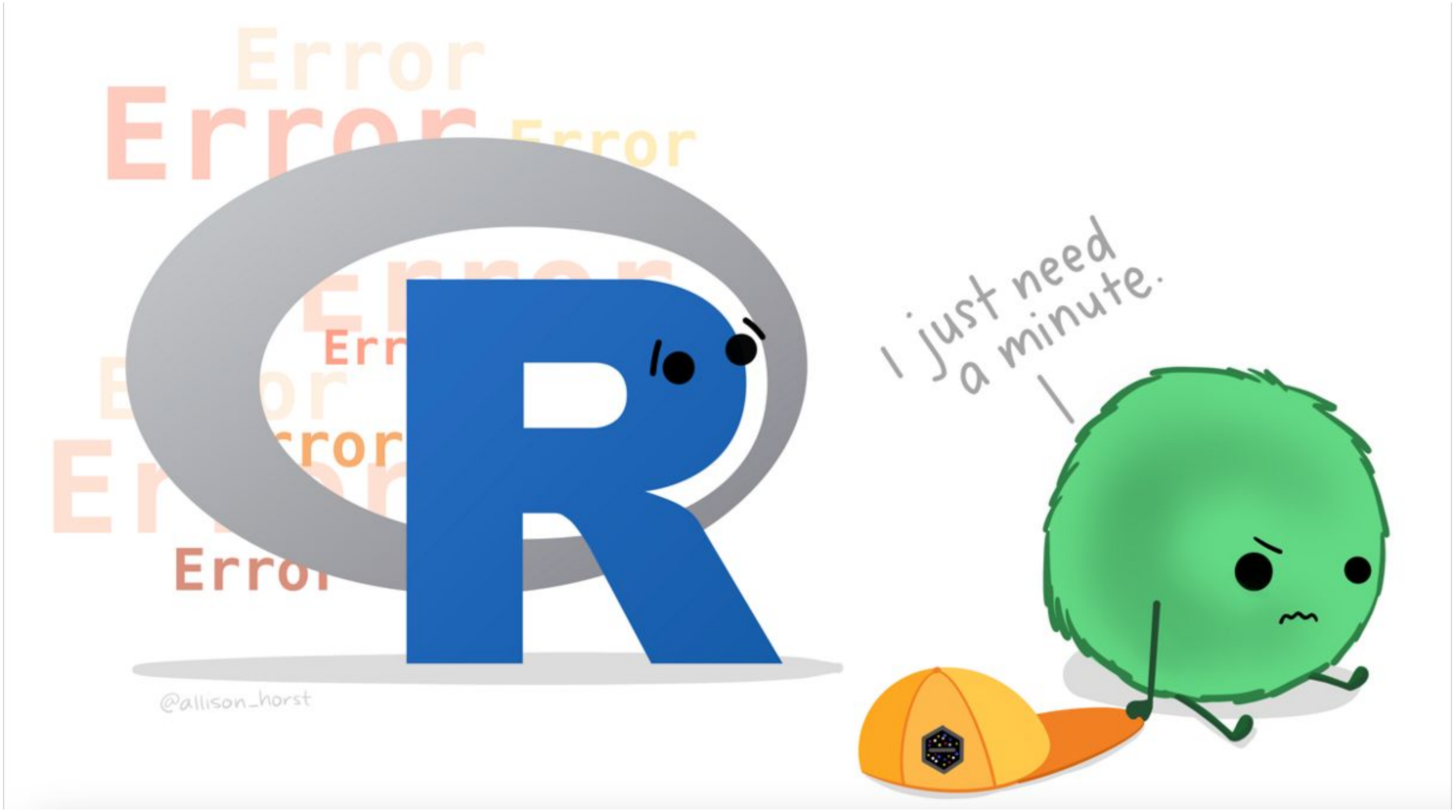
```
## [1] "numeric" "vector"
```

```r
is(TRUE)
```

```
## [1] "logical" "vector"
```

```r
is("What is this?")
```

```
## [1] "character"              "vector"                 "data.frameRowLabels"
## [4] "SuperClassMethod"
```

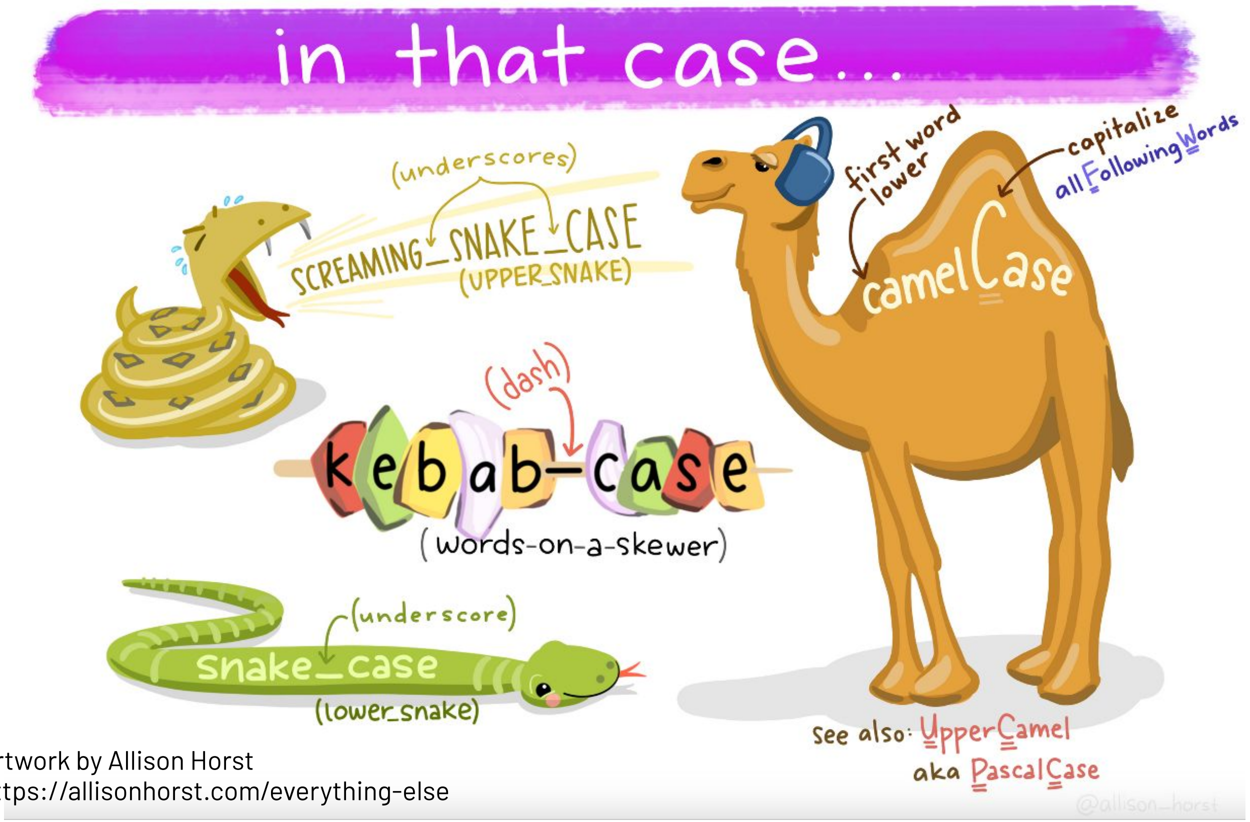


Artwork by Allison Horst
https://allisonhorst.com/everything-else

Rules for naming objects

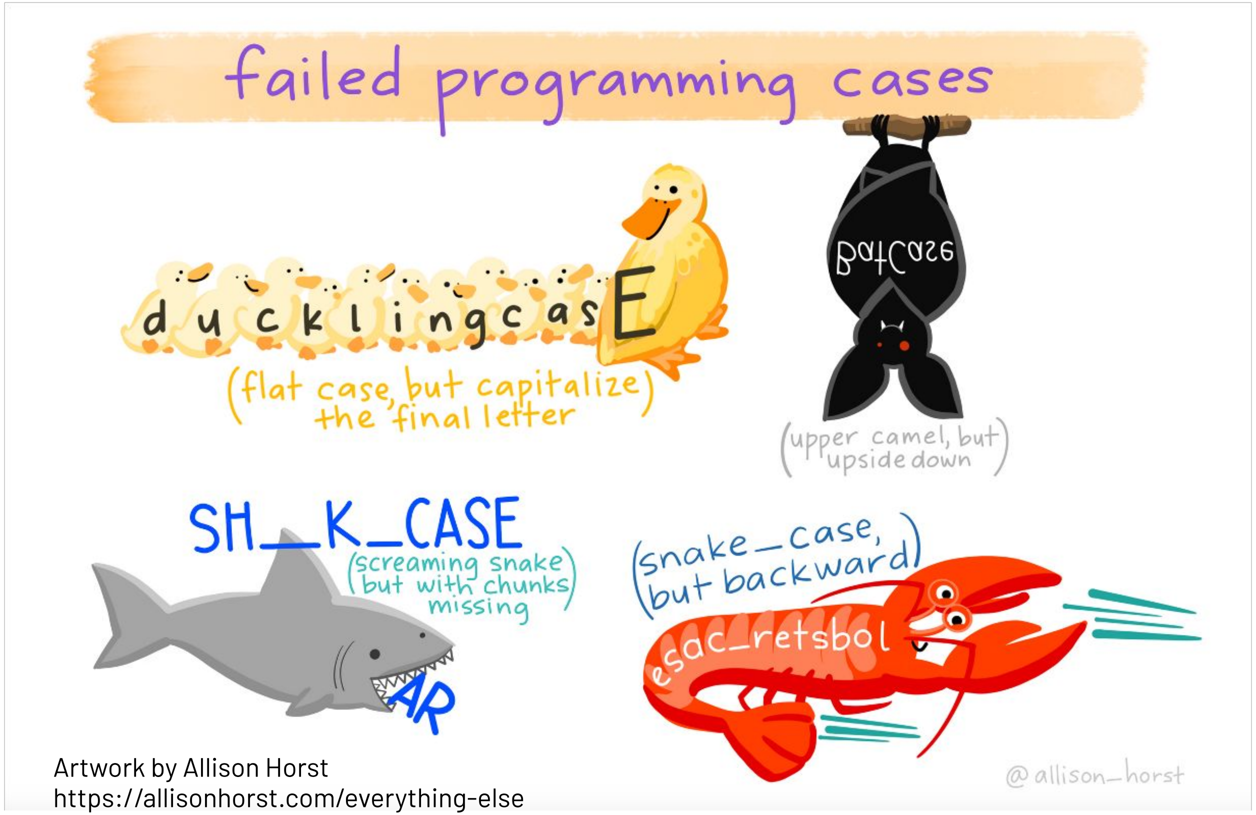
- General naming requirement: a variable name can’t start with a number or a dot (.)
- R is case sensitive (‘A’ is different than ‘a’)
- General rules of thumb: aim for consistency
    - snake_case
    - camelCase
    - whatever.this.is
- Chose a name you’ll understand when you open your code the next day, or when someone else reviews it

Rules for naming objects



Artwork by Allison Horst
https://allisonhorst.com/everything-else

Rules for naming objects


failed programming cases
ducklingcasE
(flat case, but capitalize the final letter)
BatCase
(upper camel, but upside down)
SH_K_CASE
(screaming snake but with chunks missing)
AR
(snake_case, but backward)
esac_ekans... esac_retsbol
@allison_horst

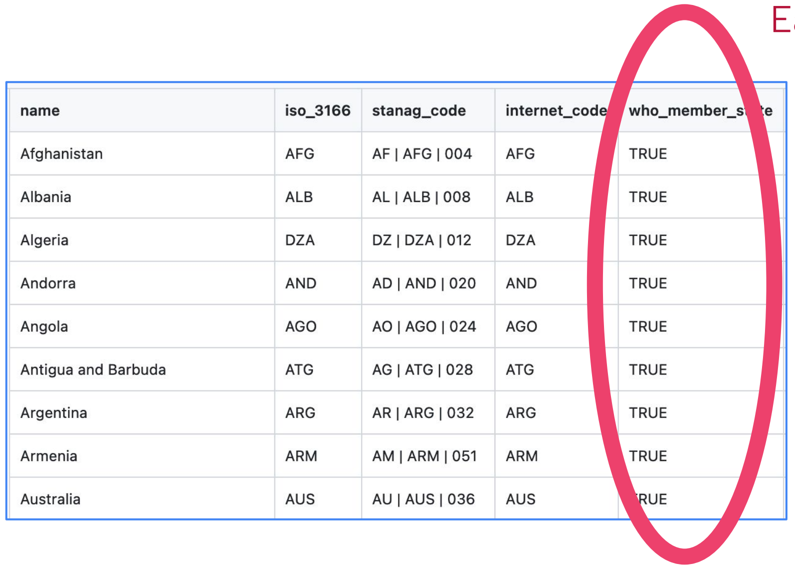
Artwork by Allison Horst
https://allisonhorst.com/everything-else

Now you try!

- Create three new objects, with any allowable names you want. Try to use a consistent naming style.
    - Numeric (we already did this, but practice is good)
    - Character
    - Logical

Vectors

- Vectors are grouped data elements in a specific order
- For example, data in a specific column in Excel



name	iso_3166	stanag_code	internet_code	who_member_s te
Afghanistan	AFG	AF | AFG | 004	AFG	TRUE
Albania	ALB	AL | ALB | 008	ALB	TRUE
Algeria	DZA	DZ | DZA | 012	DZA	TRUE
Andorra	AND	AD | AND | 020	AND	TRUE
Angola	AGO	AO | AGO | 024	AGO	TRUE
Antigua and Barbuda	ATG	AG | ATG | 028	ATG	TRUE
Argentina	ARG	AR | ARG | 032	ARG	TRUE
Armenia	ARM	AM | ARM | 051	ARM	TRUE
Australia	AUS	AU | AUS | 036	AUS	RUE

Each column is a vector

- When you've thought previously about data analysis, you probably think about vectors, even if you didn't use that name

Vectors

```r
c("HIV", "malaria", "TB")
```

```
## [1] "HIV"     "malaria" "TB"
```

```r
c(1419, 4832, 10342)
```

```
## [1]  1419  4832 10342
```

Vectors

The c() stands for "concatenate"

```
c("HIV", "malaria", "TB")
```

```
## [1] "HIV"     "malaria" "TB"
```

```
c(1419, 4832, 10342)
```

```
## [1]  1419  4832 10342
```

Vectors

```r
c("HIV", "malaria", "TB")
```

Vectors can contain strings

```
## [1] "HIV"     "malaria" "TB"
```

```r
c(1419, 4832, 10342)
```

Or numbers

```
## [1]  1419  4832 10342
```

Vectors

```
c("HIV", "malaria", "TB")
```

Vectors can contain strings

```
## [1] "HIV"     "malaria" "TB"
```

```
c(1419, 4832, 10342)
```

Or numbers

```
## [1]  1419  4832 10342
```

```
c("HIV", "malaria", 10342)
```

... but not both

What happened here?

```
## [1] "HIV"     "malaria" "10342"
```

Now you try!

- Make two vectors in R and assign them to objects.
    - Numeric
    - String

Vectorized calculations

```r
c(1,2,3,4,5) + 1
```

```
## [1] 2 3 4 5 6
```

```r
c(1,2,3,4,5) * 2
```

```
## [1]  2  4  6  8 10
```

```r
c(1,2,3,4,5) + c(8,0,0,0,0)
```

```
## [1] 9 2 3 4 5
```

Vectorized calculations

```
c(1,2,3,4,5) + c(8,0)
```

```
## Warning in c(1, 2, 3, 4, 5) + c(8, 0): longer object length is not a mult
## of shorter object length
```

```
## [1]  9  2 11  4 13
```

Functions

- Functions are instructions to perform a task
  - They are *algorithms*, or consistent set of rules
- R has built-in functions for many basic things
- Functions generally look like this: *function(object)*
- We can also “add on” extra functions by loading new libraries (we’ll get to this later), or we can write our own functions to do whatever we want

Functions

- Most functions in R are vectorized
  - This means they act on all items in a vector
- Why does this matter?
  - If you misunderstand it, your math will be wrong
  - It’s useful for basic calculations and analysis:
    - divide all numbers by 100 to calculate a %
    - multiply per-capita rates by total population

Functions

```
mean(c(1,2,3,4,5))
```

```
## [1] 3
```

```
sd(c(1,2,3,4,5))
```

```
## [1] 1.581139
```

```
summary(c(1,2,3,4,5))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1       2       3       3       4       5 
```

Functions

```r
mean(c(1,2,3,4,5))
```

```
## [1] 3
```

```r
sd(c(1,2,3,4,5))
```

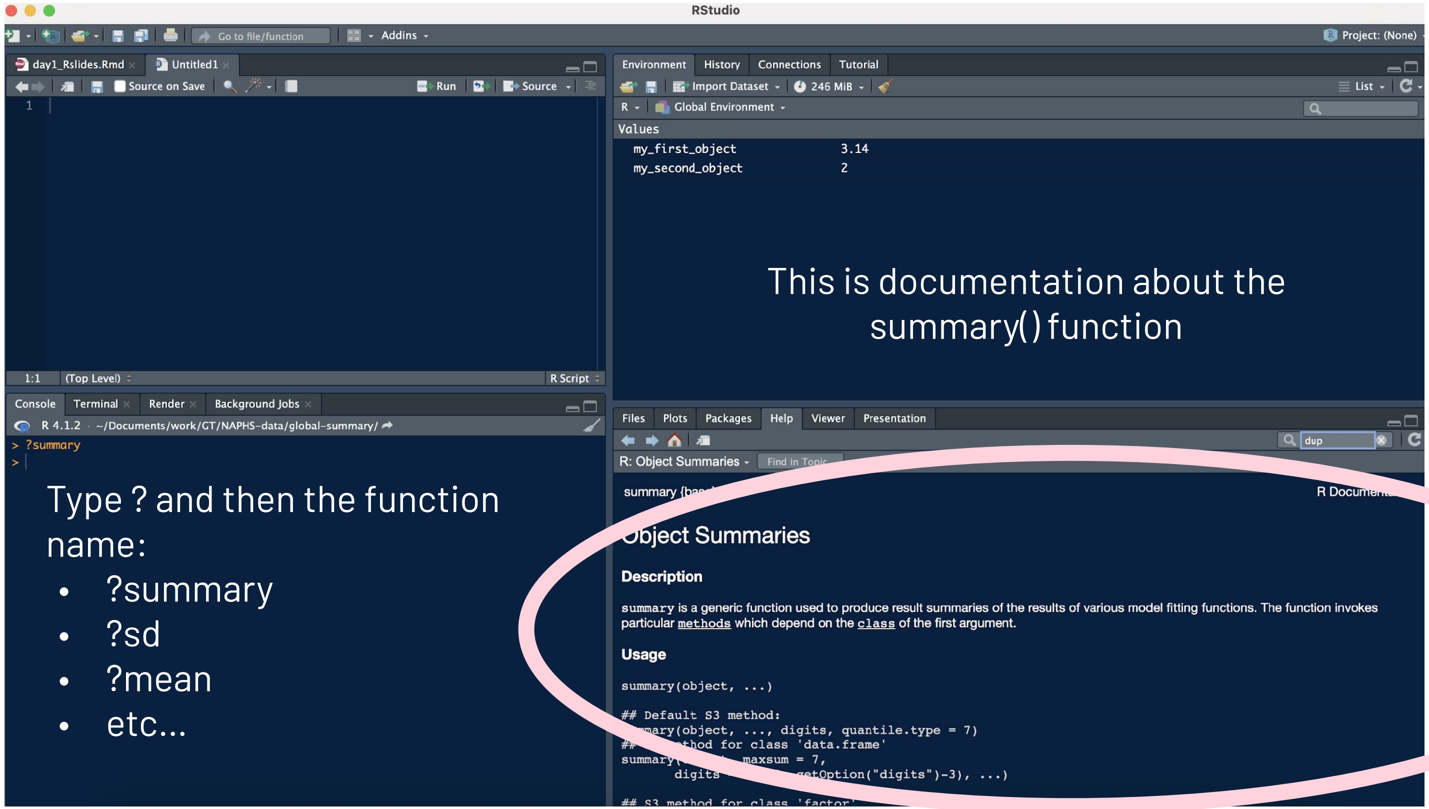
```
## [1] 1.581139
```

```r
summary(c(1,2,3,4,5))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1       2       3       3       4       5
```

Learn more about functions

?function or help(function)



```
Values
  my_first_object         3.14
  my_second_object        2
```

```
R 4.1.2 · ~/Documents/work/GT/NAPHS-data/global-summary/
> ?summary
>
```

This is documentation about the summary() function

Type ? and then the function name:

- ?summary
- ?sd
- ?mean
- etc...

summary {bas R Documen

Object Summaries

Description

summary is a generic function used to produce result summaries of the results of various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

Usage

```
summary(object, ...)

## Default S3 method:
  mary(object, ..., digits, quantile.type = 7)
##      thod for class 'data.frame'
summary(       maxsum = 7,
       digits          etOption("digits")-3), ...)

## S3 method for class 'factor'
```

Now you try!

- Take the average of three or more numbers
- Use “?” to learn more about the function sd()

Goals for today

- Understand what statistical programming is
- Get acquainted with Rstudio
- Write your very first R code (at least, of this workshop)
  - vectors
  - functions
  - accessing documentation
- Explore github to access course materials

What is github?

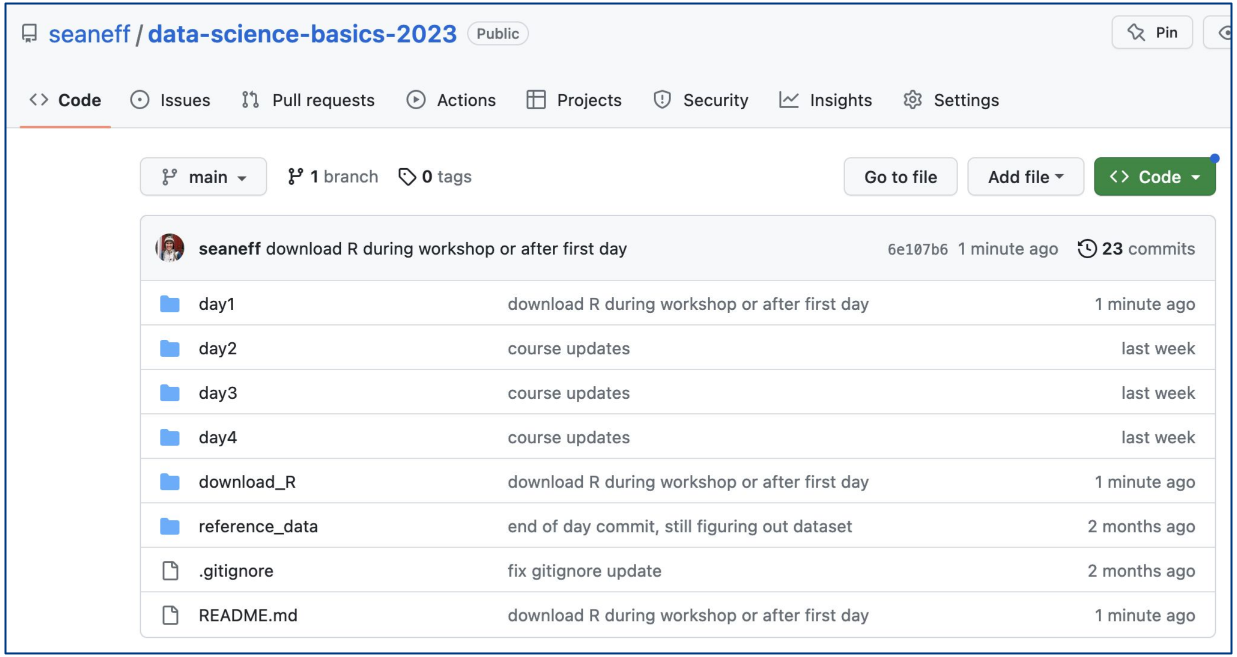
What is github?

- Have you ever saved a bunch of versions of a paper on your computer with different file names at different dates or times of day?
- Backups are useful to save progress, understand what we’ve done before, and look into problems/bugs
- Github is a tool do help do this with code

What is github?

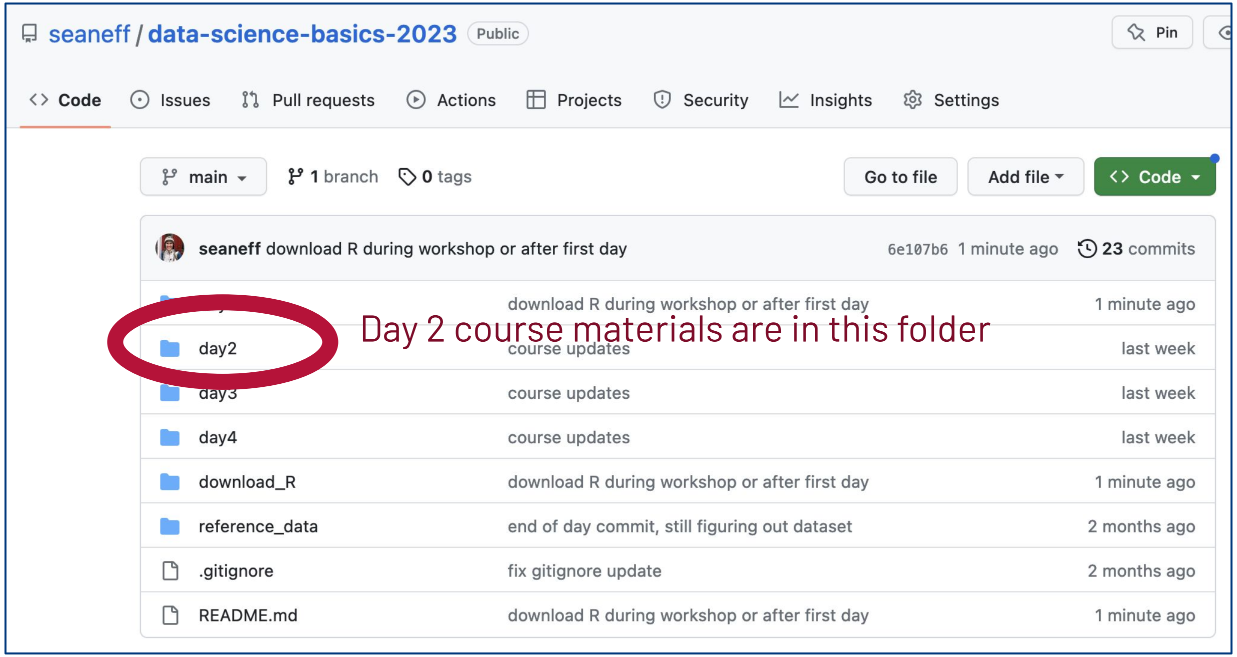
We'll talk more about github later in this workshop. For now, I'd like you to be able to use it to access course materials any time you'd like to.

https://github.com/seaneff/data-science-basics-2024



seaneff download R during workshop or after first day 6e107b6 1 minute ago 23 commits

day1	download R during workshop or after first day	1 minute ago
day2	course updates	last week
day3	course updates	last week
day4	course updates	last week
download_R	download R during workshop or after first day	1 minute ago
reference_data	end of day commit, still figuring out dataset	2 months ago
.gitignore	fix gitignore update	2 months ago
README.md	download R during workshop or after first day	1 minute ago



seaneff / data-science-basics-2023 Public

1 branch 0 tags

seaneff download R during workshop or after first day 6e107b6 1 minute ago 23 commits

	download R during workshop or after first day	1 minute ago
day2	course updates	last week
day3	course updates	last week
day4	course updates	last week
download_R	download R during workshop or after first day	1 minute ago
reference_data	end of day commit, still figuring out dataset	2 months ago
.gitignore	fix gitignore update	2 months ago
README.md	download R during workshop or after first day	1 minute ago

Day 2 course materials are in this folder

Now you try!

Open the course github and click around for a few minutes

https://github.com/seaneff/data-science-basics-2024

Thank you!

See you tomorrow.

Please come with a fully charged laptop.