

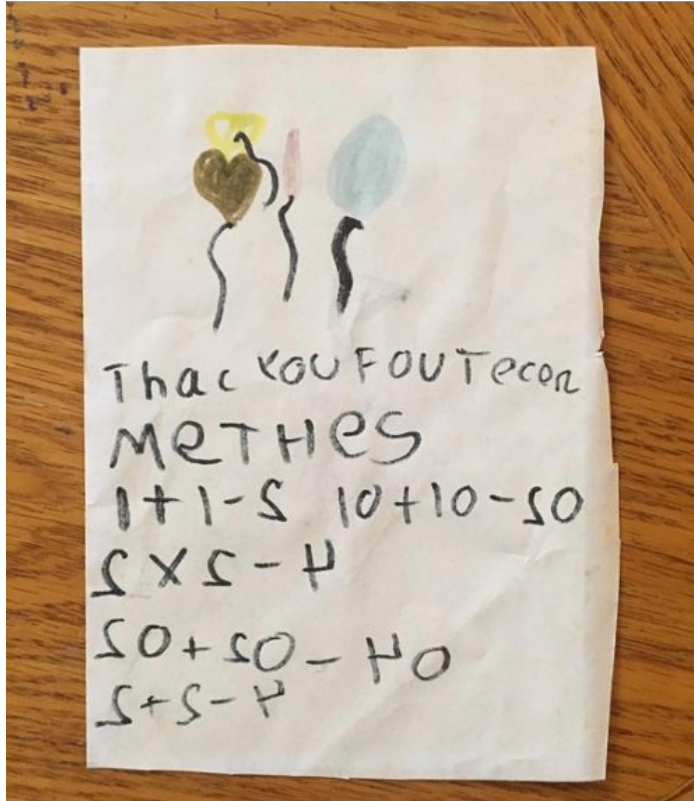
Data Science Basics in R

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



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.

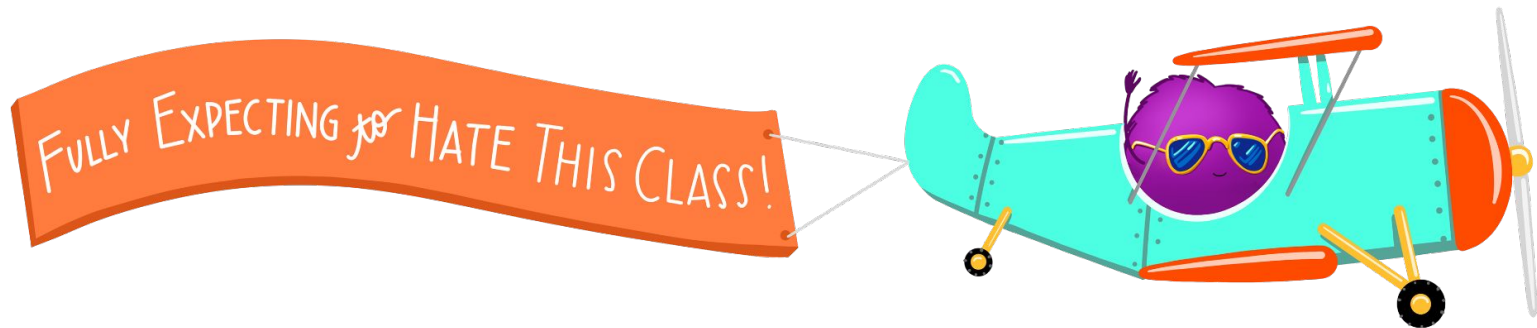


Learning by doing

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.



@allison_horst

Artwork by Allison Horst

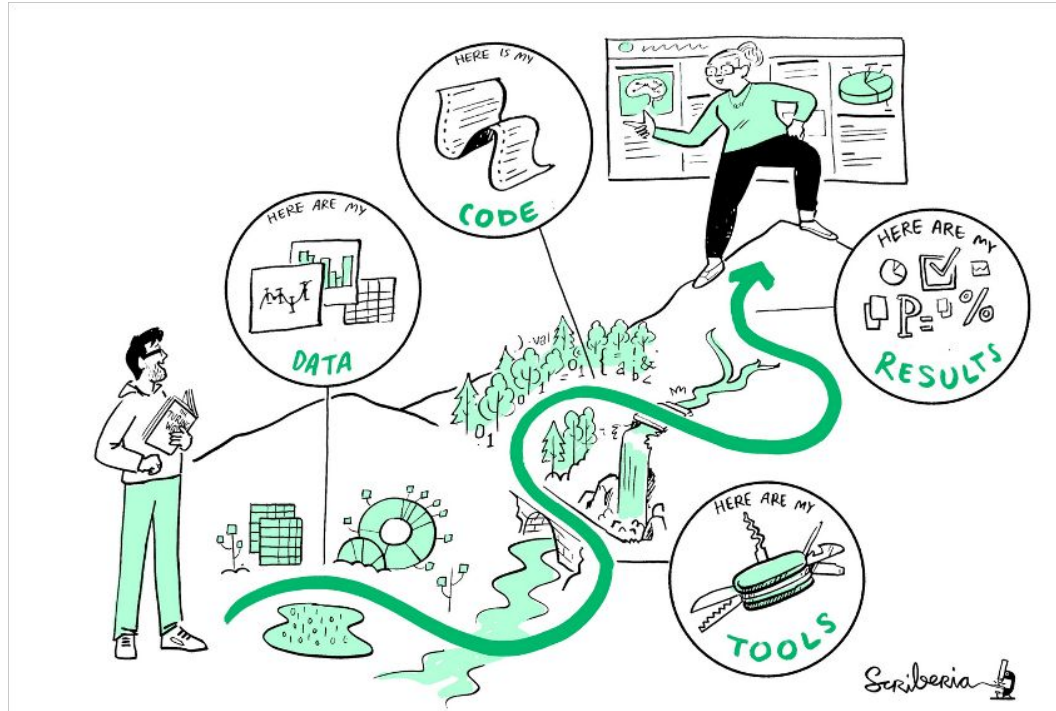
Goals for today

- 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

**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)



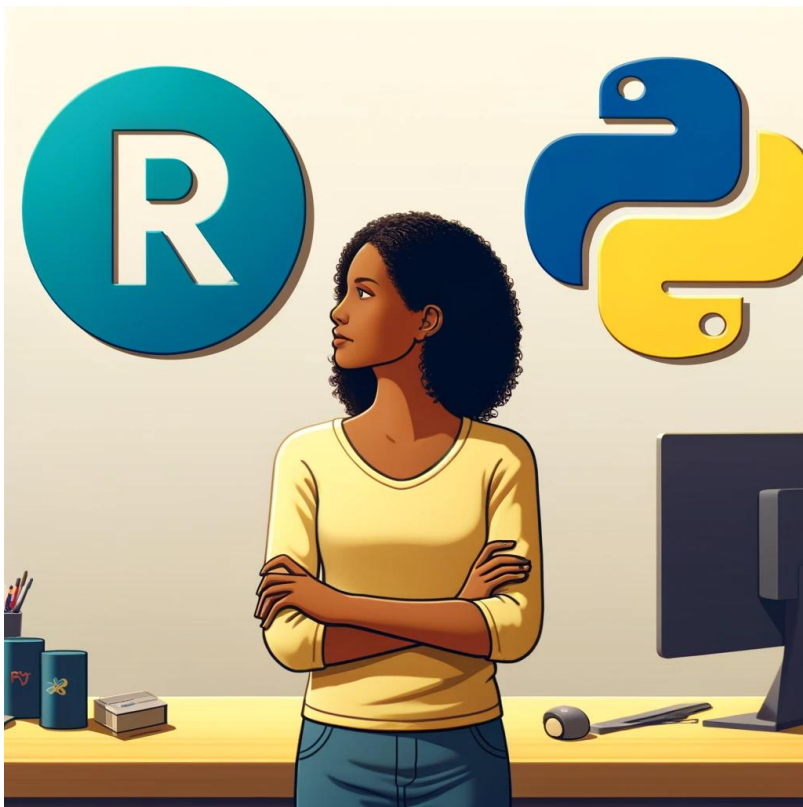
Why learn R?

- 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)
- Shareable, reproducible code and research
- Lots of academics/companies/agencies use it
- It's fun (honestly)

Alternatives to R

Tool	Primary focus	Open source?	Great for
R	Data analysis	Yes	Statistical analysis, data visualization
Python	General-purpose programming language	Yes	Huge datasets, deploying to production, bioinformatics
Julia	General-purpose programming language	Yes	Huge datasets, time-consuming calculations
SAS	Data analysis	No	Highly regulated work (e.g., clinical trials)
SPSS	Data analysis	No	Easy to use (point and click)
Stata	Data analysis	No	Easy to use (point and click)
Matlab	Data analysis (especially engineering)	No	Mathematical programming
Excel	Data management	No	Quick pivot tables, looking at raw data

R vs. Python



- Python is the primary open-source alternative to R, and some people *love* to argue about which is better
- They are both great tools. Most data analysts/data scientists I've worked with use both, for different use cases.

Downloading R and RStudio

Download R:

<https://cran.r-project.org/>

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is packaged on new Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

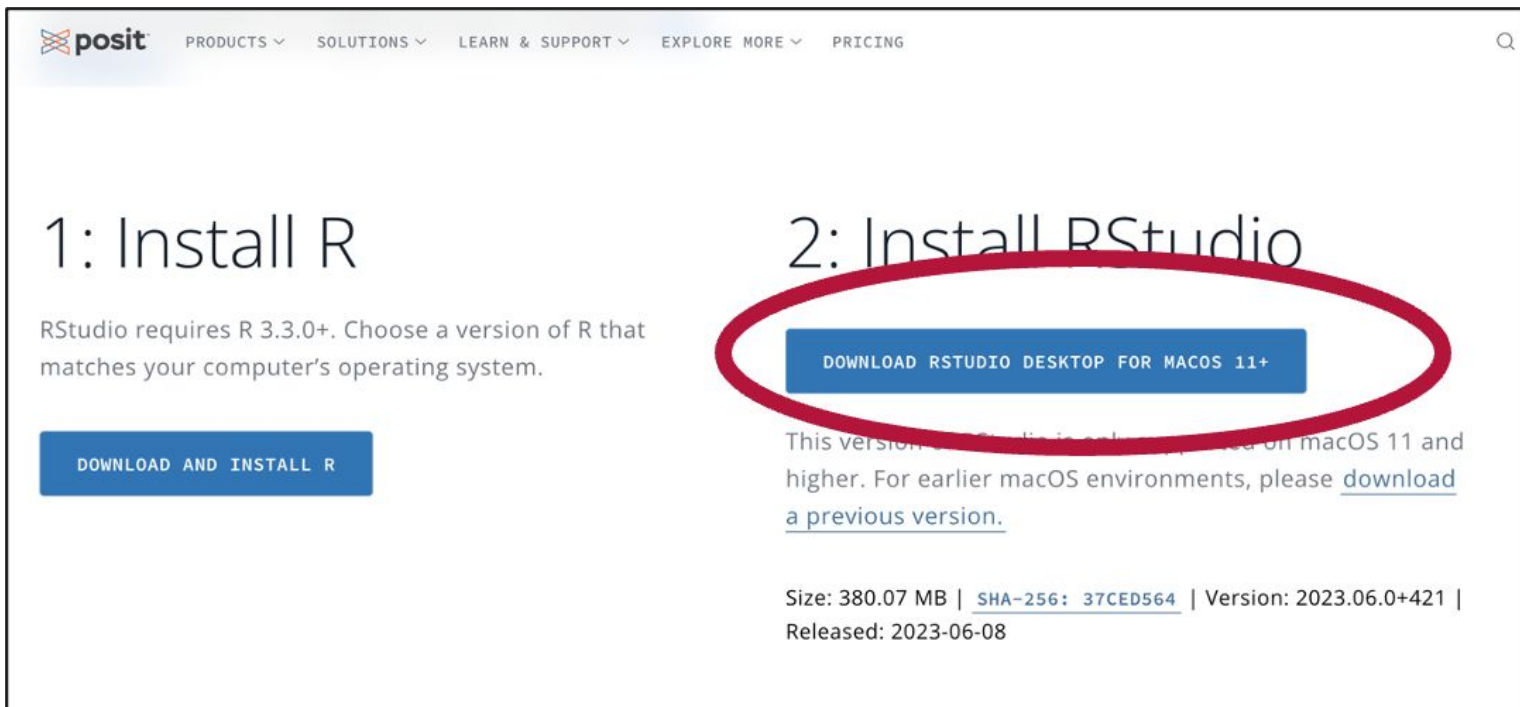
- The latest release (2023-04-21, Already Tomorrow) [R-4.3.0.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Download RStudio:

<https://posit.co/download/rstudio-desktop/#download>



The screenshot shows the RStudio download page. The navigation bar at the top includes the Posit logo and links for PRODUCTS, SOLUTIONS, LEARN & SUPPORT, EXPLORE MORE, and PRICING. The page is divided into two main sections: '1: Install R' and '2: Install RStudio'. The '2: Install RStudio' section is highlighted with a red circle. Below the heading '2: Install RStudio' is a blue button labeled 'DOWNLOAD RSTUDIO DESKTOP FOR MACOS 11+'. Below this button, text states: 'This version of RStudio is only supported on macOS 11 and higher. For earlier macOS environments, please [download a previous version.](#)' At the bottom of the section, technical details are provided: 'Size: 380.07 MB | [SHA-256: 37CED564](#) | Version: 2023.06.0+421 | Released: 2023-06-08'.

posit[™] PRODUCTS ▾ SOLUTIONS ▾ LEARN & SUPPORT ▾ EXPLORE MORE ▾ PRICING

1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

DOWNLOAD AND INSTALL R

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR MACOS 11+

This version of RStudio is only supported on macOS 11 and higher. For earlier macOS environments, please [download a previous version.](#)

Size: 380.07 MB | [SHA-256: 37CED564](#) | Version: 2023.06.0+421 | Released: 2023-06-08

How do I use RStudio?

RStudio console

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains an R script named `graphic.R` with the following code:

```
22 #####
23
24 ## NAPHS country data - one row per WHO member state
25 ## Data as of
26 countries <- read.delim("countries.tsv")
27
28 #####
29 ## Summarize data: #####
30 ## Printed summary #####
31 #####
32
33 ## print summary info, globally
34 countries %>%
35   summarize(total_member_states = sum(who_member_state == TRUE),
36             completed_jees = sum(completed_jees == TRUE),
37             completed_naphs = sum(completed_naphs == TRUE),
38             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE),
39             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
40             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
41
42 #####
43 ## Global funnel: Manuscript barplot #####
44 #####
45 ## end of script - end of script - end of script
```
- Console:** Shows the execution of the script, resulting in a summary table:

```
R 4.1.2 ~|/Documents/work/CT/NAPHS-data/global-summary/
> ## 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_jees = sum(completed_jees == 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_jees completed_naphs published_naphs published_naphs_data machine_readable_data
1                194             103              77              14                9                0
>
```
- Environment:** Shows the `Global Environment` with a data frame `countries` containing 194 observations and 9 variables.
- Viewer:** Displays the documentation for the `duplicated()` function, titled "Determine Duplicate Elements".

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, ...)
```

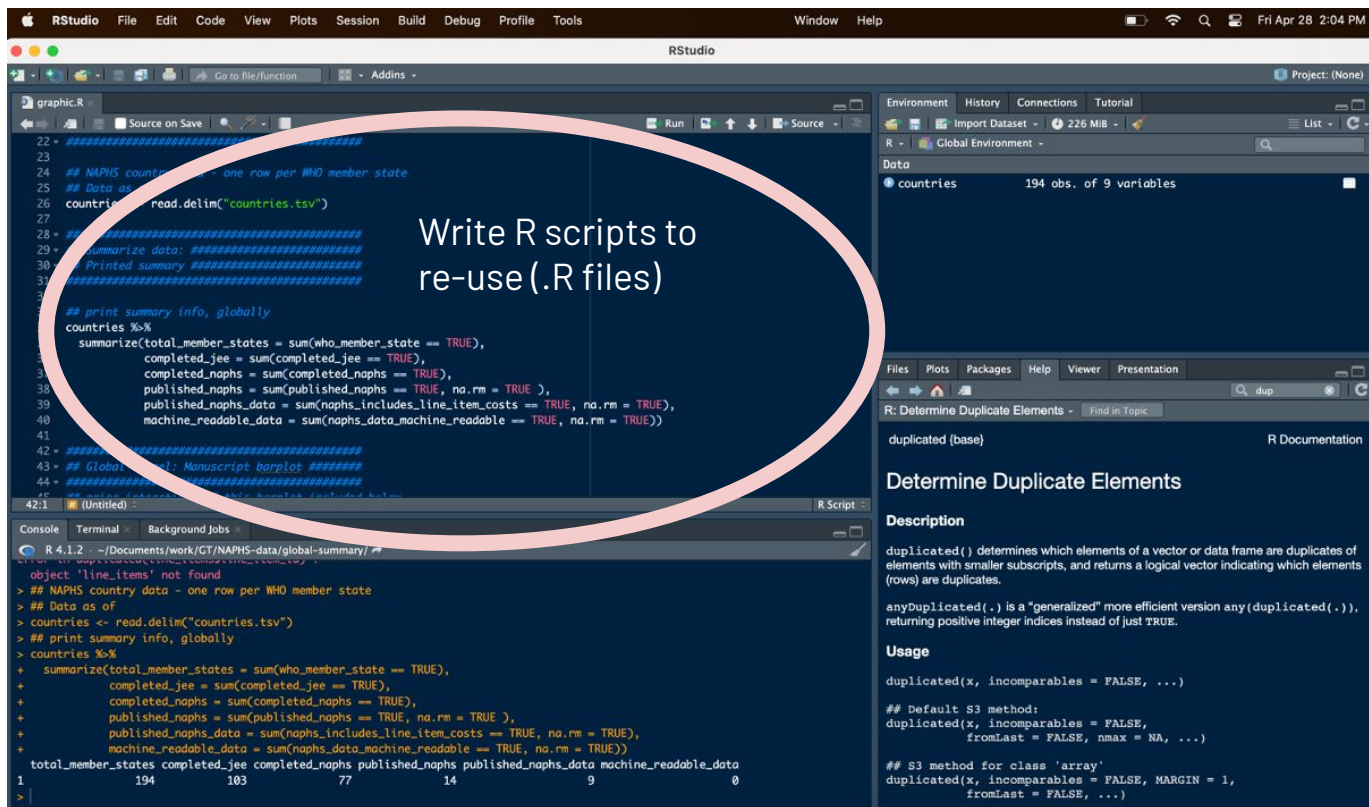
Default S3 method:

```
duplicated(x, incomparables = FALSE,
          fromLast = FALSE, nmax = NA, ...)
```

S3 method for class 'array'

```
duplicated(x, incomparables = FALSE, MARGIN = 1,
          fromLast = FALSE, ...)
```


RStudio console



The screenshot shows the RStudio interface. A pink oval highlights the script editor and the console. The script editor contains R code for reading and summarizing data. The console shows the output of the code, including a summary of the data and a table of counts.

Script Editor (graphic.R):

```
22 #####
23
24 ## NAPHS country data - one row per WHO member state
25 ## Data as
26 countries <- read.delim("countries.tsv")
27
28 #####
29
30 ## print summary info, globally
31 countries %>%
32   summarize(total_member_states = sum(who_member_state == TRUE),
33             completed_jee = sum(completed_jee == TRUE),
34             completed_naphs = sum(completed_naphs == TRUE),
35             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE),
36             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
37             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
38
39 #####
40
41 ## Global goal: Manuscript barplot #####
42
43 #####
44
45 ## end of script - RStudio console output below
46
47 42:1 (Untitled) R Script
```

Console:

```
R 4.1.2 ~/Documents/work/CT/NAPHS-data/global-summary/
> object 'line_items' not found
> ## NAPHS country data - one row per WHO member state
> ## Data as
> 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
>
```

Environment:

R - Global Environment - 226 MiB

Data

countries 194 obs. of 9 variables

Files: R: Determine Duplicate Elements - Find in Topic

Determine Duplicate Elements

Description

`uplicated()` 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

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

Default S3 method:

```
uplicated(x, incomparables = FALSE,
          fromLast = FALSE, nmax = NA, ...)
```

S3 method for class 'array'

```
uplicated(x, incomparables = FALSE, MARGIN = 1,
          fromLast = FALSE, ...)
```

RStudio console

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for loading and summarizing data. The **Run** button (a green play icon) in the toolbar is circled in pink. Text next to it says: "This button runs code".
- Environment:** Shows the **Global Environment** with a variable **countries** of type **Data**, containing 194 observations of 9 variables.
- Console:** Shows the output of the code executed in the source editor. It is circled in pink. Text next to it says: "Run one-off code here, or see the results of code you ran from above".
- Documentation:** The bottom right pane shows the documentation for the **duplicated()** function, including a description and usage examples.

```
## NAPHIS 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 #####
#####
```

Run one-off code here, or see the results of code you ran from above

```
R 4.1.2 ~D:\work\CT\NAPHIS-data\global-summary>
> object 'countries' not found
> ## NAPHIS 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
1 103 77 14 9 0
```

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, ...)
```

RStudio console

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains an R script named `graphic.R` with the following code:

```
22 #####
23
24 ## NAPHS country data - one row per WHO member state
25 ## Data as of
26 countries <- read.delim("countries.tsv")
27
28 #####
29 ## Summarize data: #####
30 ## Printed summary #####
31 #####
32
33 ## print summary info, globally
34 countries %>%
35   summarize(total_member_states = sum(who_member_state == TRUE),
36             completed_jeec = sum(completed_jeec == TRUE),
37             completed_naphs = sum(completed_naphs == TRUE),
38             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE),
39             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),
40             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))
41
42 #####
43 ## Global funnel: Manuscript barplot #####
44 #####
45 ## end of script - end of script - end of script
```
- Console:** Shows the execution of the script, with the following output:

```
R 4.1.2 ~|/Documents/work/CT/NAPHS-data/global-summary/
> ## 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_jeec = sum(completed_jeec == 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_jeec completed_naphs published_naphs published_naphs_data machine_readable_data
1          194          103           77           14           9           0
>
```
- Environment Pane:** A pink oval highlights this pane, which contains the text "Stuff you have in memory". It shows the `countries` data frame with 194 observations and 9 variables.
- Help Pane:** Displays the documentation for the `duplicated()` function, titled "Determine Duplicate Elements".

RStudio console

The screenshot displays the RStudio environment with the following components:

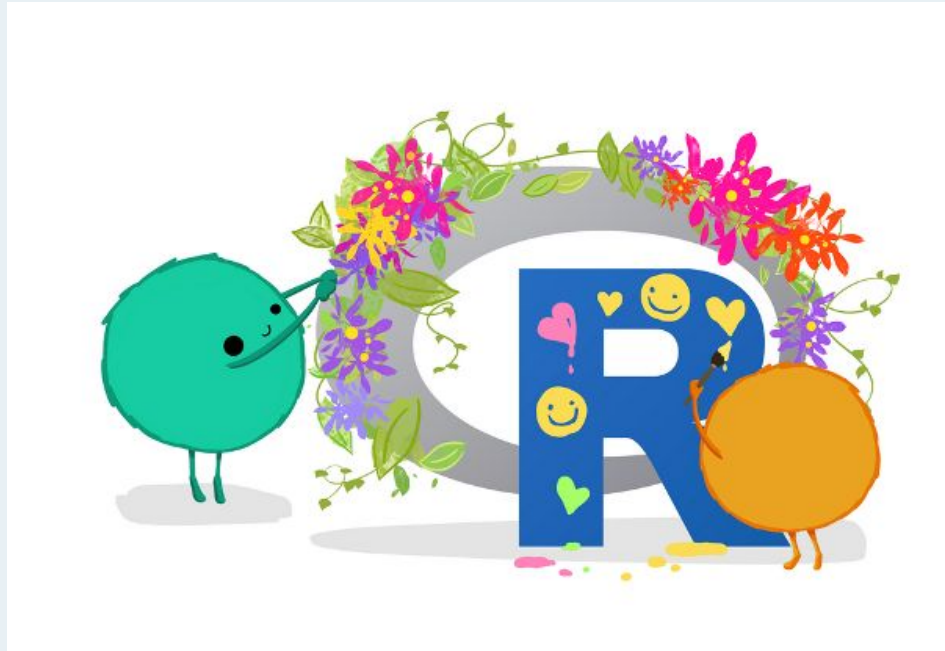
- Source Editor:** Contains an R script named `graphic.R` with the following code:

```
22 #####  
23  
24 ## NAPHS country data - one row per WHO member state  
25 ## Data as of  
26 countries <- read.delim("countries.tsv")  
27  
28 #####  
29 ## Summarize data: #####  
30 ## Printed summary #####  
31 #####  
32  
33 ## print summary info, globally  
34 countries %>%  
35   summarize(total_member_states = sum(who_member_state == TRUE),  
36             completed_jees = sum(completed_jees == TRUE),  
37             completed_naphs = sum(completed_naphs == TRUE),  
38             published_naphs = sum(published_naphs == TRUE, na.rm = TRUE),  
39             published_naphs_data = sum(naphs_includes_line_item_costs == TRUE, na.rm = TRUE),  
40             machine_readable_data = sum(naphs_data_machine_readable == TRUE, na.rm = TRUE))  
41  
42 #####  
43 ## Global funnel: Manuscript barplot #####  
44 #####  
45 ## end of script - end of script - end of script
```
- Console:** Shows the execution of the script, with the following output:

```
R 4.1.2 ~|/Documents/work/CT/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_jees = sum(completed_jees == 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_jees completed_naphs published_naphs published_naphs_data machine_readable_data  
1          194          103           77           14           9           0  
>
```
- Environment:** Shows the `Global Environment` with a data frame `countries` containing 194 observations and 9 variables.
- Documentation Pane:** Displays the documentation for the `duplicate()` function, titled "Determine Duplicate Elements". The text includes a description of the function and its usage. A pink circle highlights the "Documentation and figures" section.

Now you try!

Open RStudio on your computer and click around



15 minute break

(and a note on worked examples)

- If you haven't already, please try to download R and Rstudio before tomorrow's class. I'll be around by 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.

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

```
1+1
```

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

```
10-8
```

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

```
(6-3)*4
```

```
## [1] 12
```

Using R as a calculator

```
abs(-18)
```

```
## [1] 18
```

```
log(1)
```

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

```
log(1)
```

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

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

```
my_first_object <- 3
```

```
my_first_object
```

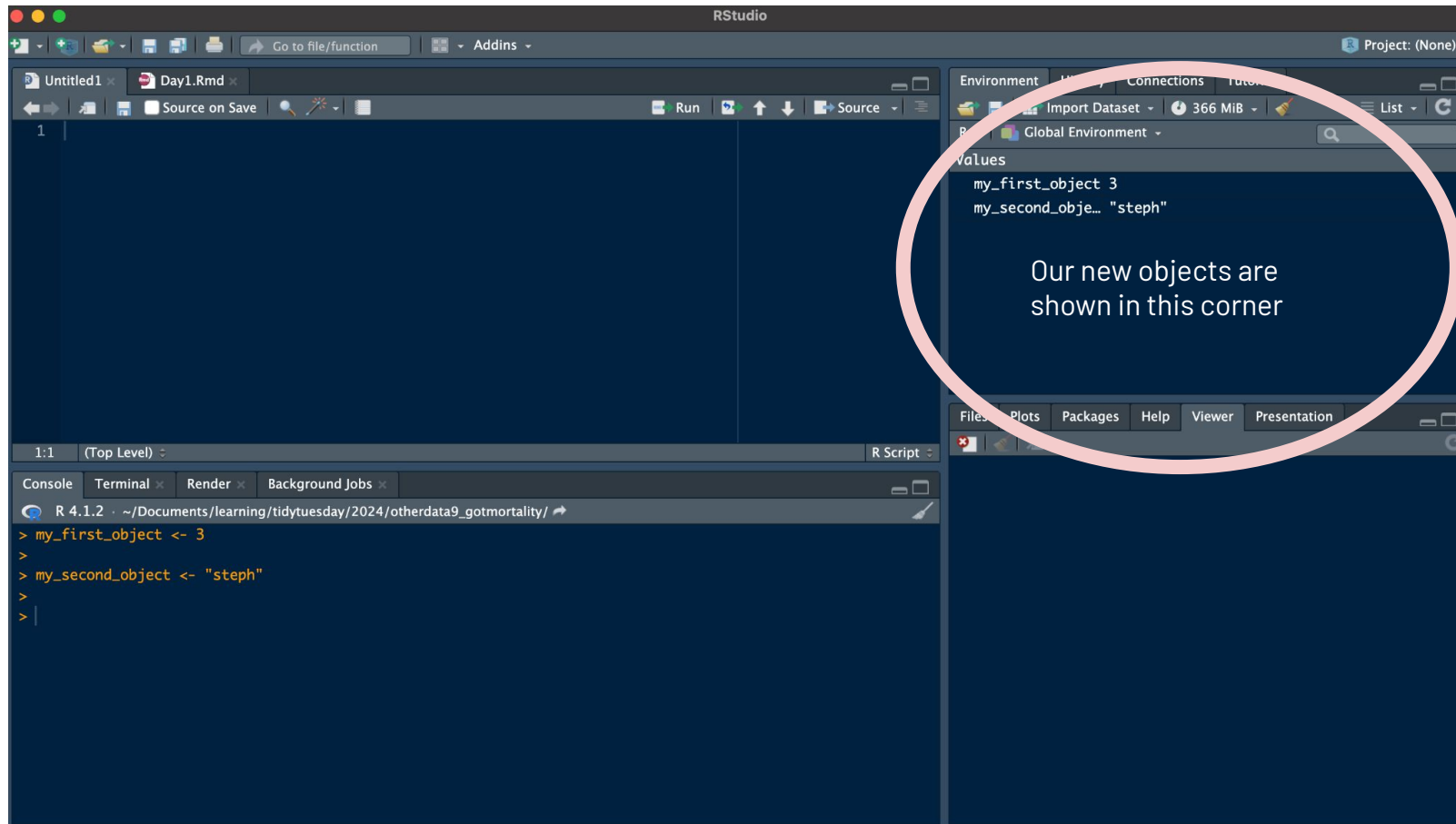
```
## [1] 3
```

```
my_second_object <- "steph"
```

```
my_second_object
```

```
## [1] "steph"
```

Objects



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

```
my_first_object
```

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

```
my_first_object*2
```

```
## [1] 6
```

```
my_first_object * my_first_object
```

```
## [1] 9
```

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.



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

```
is(-1)
```

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

```
is(TRUE)
```

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

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

```
## [1] "character"
```

```
"vector"
```

```
"data.frameRowLabels"
```

```
## [4] "SuperClassMethod"
```

Numeric data

```
my_favorite_number <- 3  
my_favorite_number
```

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

```
my_house_number <- 1416  
my_house_number
```

```
## [1] 1416
```

```
example_result <- 3*4  
example_result
```

```
## [1] 12
```

Character data

```
policy <- "International Health Regulations (2005)"  
policy
```

```
## [1] "International Health Regulations (2005)"
```

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

```
## [1] "UNAIDS"
```

Logical data

```
logical_example <- TRUE  
logical_example
```

```
## [1] TRUE
```

```
second_logical_example <- FALSE  
second_logical_example
```

```
## [1] FALSE
```

Check your understanding!

`is(5)`

`is(FALSE)`

`is("Georgetown")`

Check your understanding!

```
is(5)
```

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

```
is(FALSE)
```

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

```
is("Georgetown")
```

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



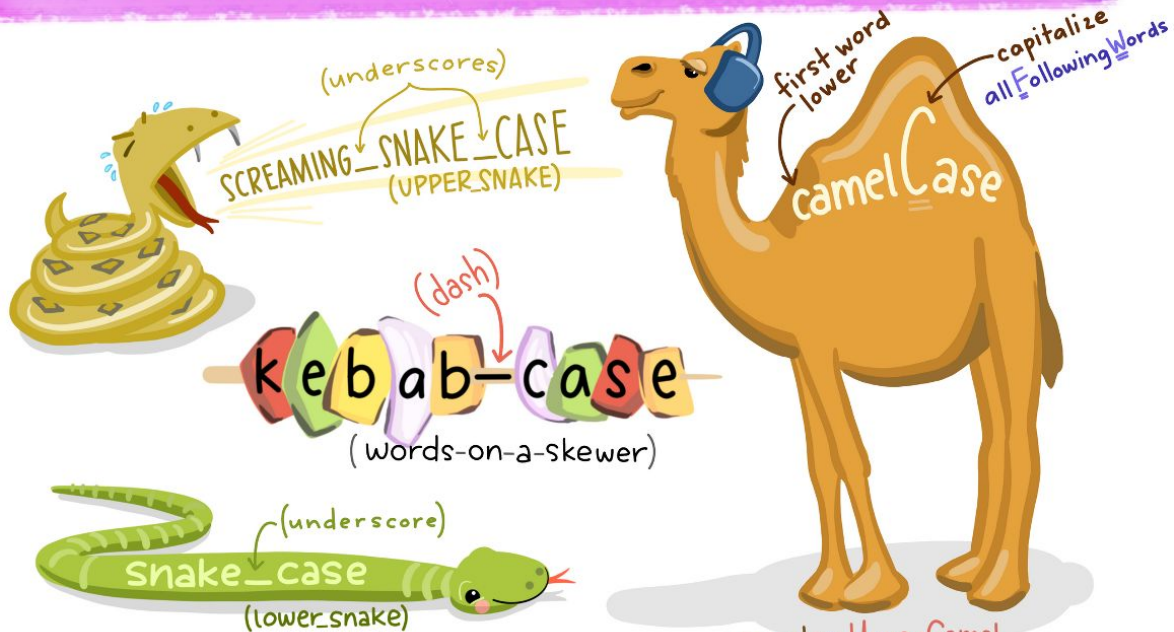

Artwork by Allison Horst
<https://allisonhorst.com/everything-else>

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

in that case...



See also: UpperCamel
aka PascalCase

Rules for naming objects

failed programming cases



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

name	iso_3166	stanag_code	internet_code	who_member_state
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	TRUE

Each column is a vector

Vectors

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

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

```
c(1, 2, 6, 87)
```

```
## [1] 1 2 6 87
```

Vectors

The `c()` stands for “concatenate”

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

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

```
c(1, 2, 6, 87)
```

```
## [1] 1 2 6 87
```


Vectors

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

Vectors can contain strings

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

```
c(1, 2, 6, 87)
```

Or numbers

```
## [1] 1 2 6 87
```

Vectors

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

Vectors can contain strings

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

```
c(1, 2, 6, 87)
```

Or numbers

```
## [1] 1 2 6 87
```

```
c("CDC", "FDA", 897)
```

... but not both
What happened here?

```
## [1] "CDC" "FDA" "897"
```

Now you try!

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

Vectorized calculations

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

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

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

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

```
c(1, 2, 3, 4) + c(5, 6, 7, 8)
```

```
## [1] 6 8 10 12
```

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

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

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

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

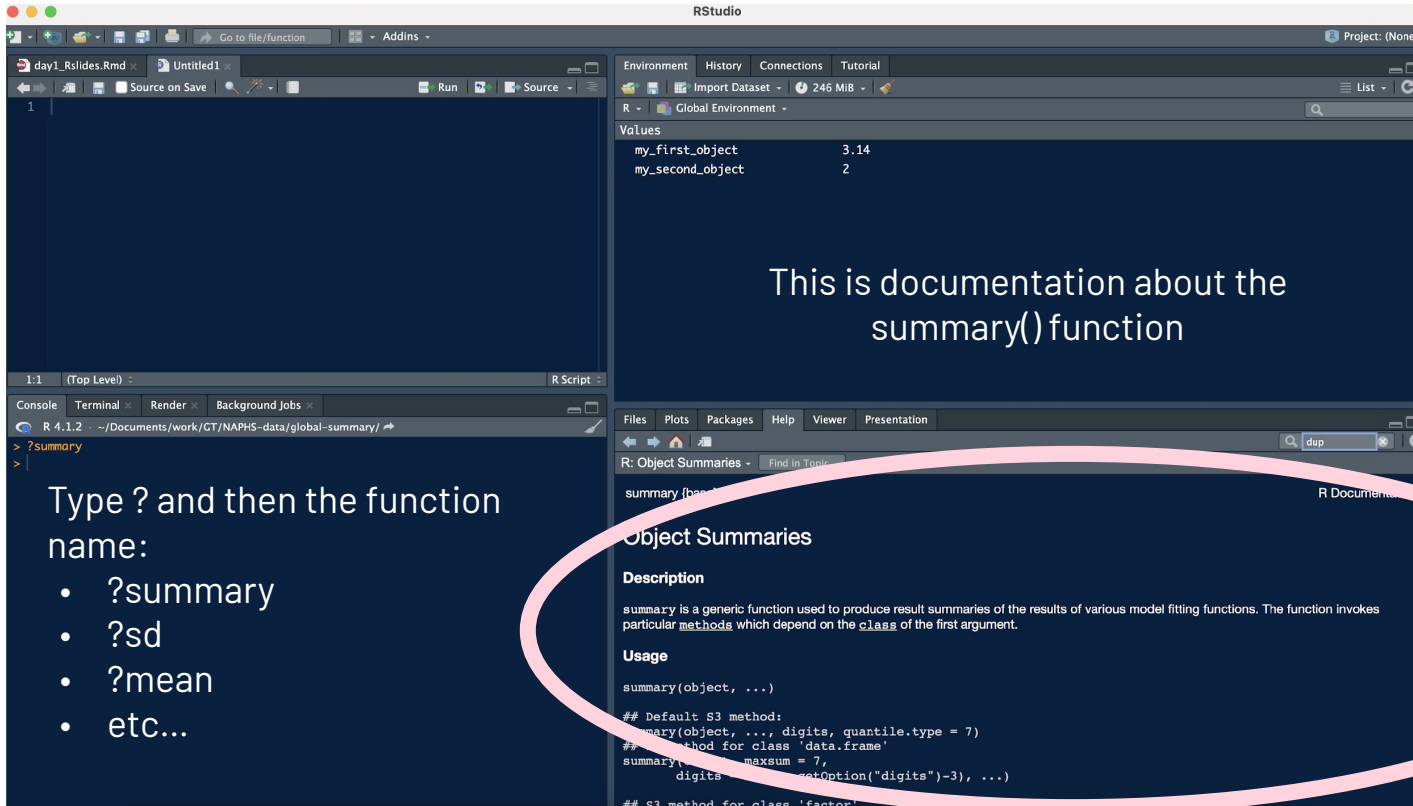
```
## [1] 1.581139
```

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

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

Learn more about functions

?function or help(function)



The screenshot shows the RStudio interface. The top-left pane contains a script editor with a single line of code: `?summary`. The bottom-left pane shows the console output: `> ?summary`. The right-hand side of the interface displays the help documentation for the `summary()` function. A pink oval highlights the 'Object Summaries' section, which includes the 'Description' and 'Usage' sections. The 'Description' section states: '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.' The 'Usage' section shows the function signature: `summary(object, ...)` and provides examples for the default S3 method and the method for class 'data.frame'.

This is documentation about the `summary()` function

Type `?` and then the function name:

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

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:

```
summary(object, ..., digits, quantile.type = 7)
```

Method for class 'data.frame'

```
summary(object, maxsum = 7,
  digits = getOption("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>

main

2 Branches 0 Tags

Go to file

t

Add file

<> Code

About



seaneff add mcv1 data

9e8e441 · 2 weeks ago

21 Commits

course-datasets	add mcv1 data	2 weeks ago
day1	add new dataset on measles vaccine coverage	2 weeks ago
day2	add information on organizing github repos	2 weeks ago
day3	add new dataset on measles vaccine coverage	2 weeks ago
extras	add mcv1 data	2 weeks ago
.gitignore	figuring out how to host slides	6 months ago
README.md	reorganize files	last month

README



Data Science Basics in R Workshop

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 workspaces. 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.

This workshop is part of the [2024 Georgetown University Health Diplomacy Training Initiative](#) led by the [Center for Health Science and Security](#).

No prior programming experience is necessary for this course, though to follow along with course materials, participants will require access to a fully charged laptop or computer. There are no required course materials or textbooks, though we will work together through select materials from the online [Intro to R](#) book developed by Alex Douglas, Deon Roos, Francesca Mancini, Ana Couto and David Lusseau.

If you have questions, feel free to reach out at sde31@georgetown.edu.

In progress materials for the data science basics course in 2024

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

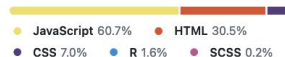
[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages



Suggested workflows

Based on your tech stack

**Grunt**

Configure

Build a NodeJS project with npm and grunt.

**Webpack**

Configure

Build a NodeJS project with npm and webpack.

main

2 Branches 0 Tags

Go to file

t

Add file

<> Code

About



seaneff add mcv1 data

9e8e441 · 2 weeks ago

21 Commits

course-datasets	add mcv1 data	2 weeks ago
day1	add new dataset on measles vaccine coverage	2 weeks ago
day2	add information on organizing github repos	2 weeks ago
day3	add new dataset on measles vaccine coverage	2 weeks ago
extras	add mcv1 data	2 weeks ago
.gitignore	figuring out how to host slides	6 months ago
README.md	reorganize files	last month

README



Data Science Basics in R Workshop

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 workspaces. 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.

This workshop is part of the [2024 Georgetown University Health Diplomacy Training Initiative](#) led by the [Center for Health Science and Security](#).

No prior programming experience is necessary for this course, though to follow along with course materials, participants will require access to a fully charged laptop or computer. There are no required course materials or textbooks, though we will work together through select materials from the online [Intro to R](#) book developed by Alex Douglas, Deon Roos, Francesca Mancini, Ana Couto and David Lusseau.

If you have questions, feel free to reach out at sde31@georgetown.edu.

In progress materials for the data science basics course in 2024

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

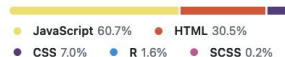
[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages



Suggested workflows

Based on your tech stack

**Grunt**

Configure

Build a NodeJS project with npm and grunt.

**Webpack**

Configure

Build a NodeJS project with npm and webpack.

Now you try!

Open the course github and click around for a few minutes

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

Recap for today

What we talked about

- 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

OPTIONAL homework

choose your own dataset

- Tomorrow, we will begin using worked examples with **real data**
- The materials I have prepared focus on measles and vaccine policy
- If you have other datasets you know you want to explore, let me know!
 - We'll still focus primarily on the measles dataset as a class
 - There is also room to explore other areas! Last year students explored sanctions, extinction-risk, and public safety datasets
- Please email me before tomorrow's class if there is a dataset you are interested in (limit 1 per person), and I will do my best to work it in!
sde31@georgetown.edu

OPTIONAL homework

choose your own dataset: options to explore

- TidyTuesday: <https://github.com/rfordatascience/tidytuesday>
- Security Studies: <https://guides.library.georgetown.edu/Security/data>
- Spotify: www.kaggle.com/datasets/iamsumat/spotify-top-2000s-mega-dataset

The world is your oyster. Datasets don't need to be related to your major, they can also be other things you find interesting or fun!

Plan for tomorrow

Data management and version control

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
 - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorns)

Thank you!

See you tomorrow.

Please come with a fully charged laptop.