

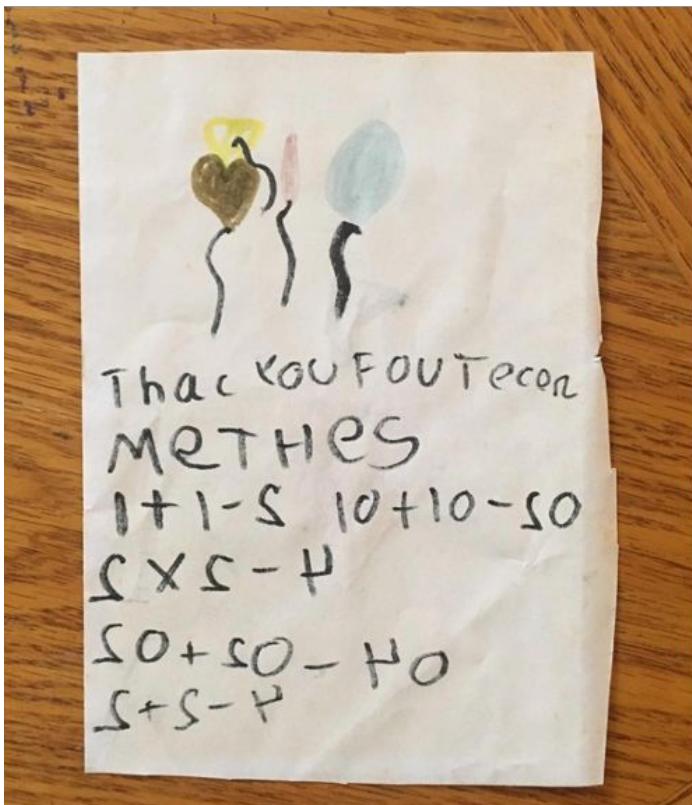
Data Science Basics in R

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
- Office hours are immediately after class

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.

Course goals

This course 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.

Course 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 Alison 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.



Everything hurts and I'm dying.

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
- Get familiar with Github (by writing poems about rainbows)

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)



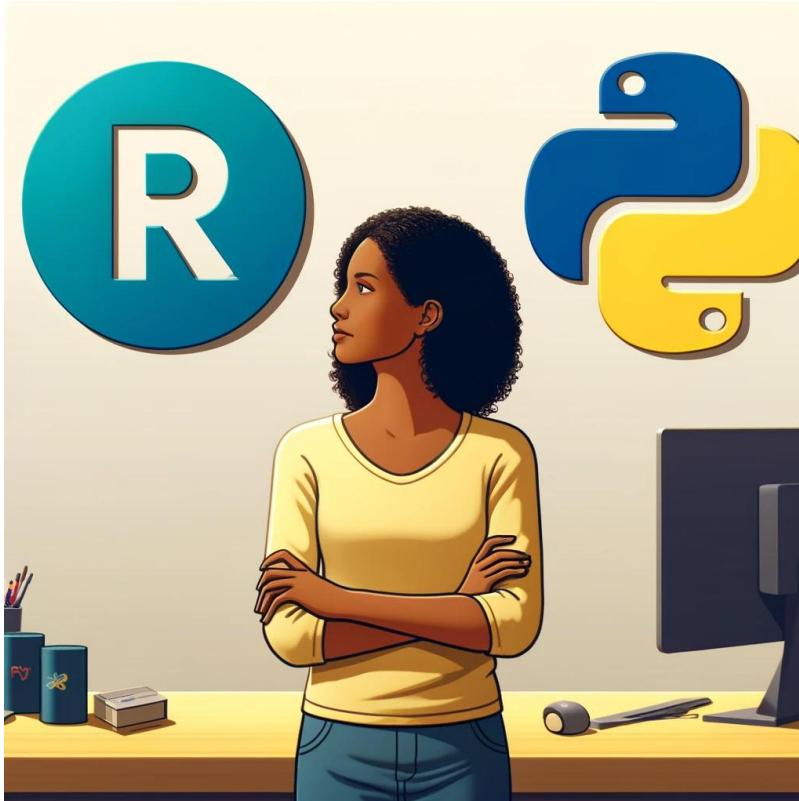
Why learn R?

- 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 in many 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 posit.co download page. At the top, there is a navigation bar with links: PRODUCTS ▾, SOLUTIONS ▾, LEARN & SUPPORT ▾, EXPLORE MORE ▾, PRICING, and a search icon. Below the navigation, there are two main sections: "1: Install R" on the left and "2: Install RStudio" on the right. The "1: Install R" section contains text about choosing an R version and a blue "DOWNLOAD AND INSTALL R" button. The "2: Install RStudio" section contains text about the RStudio desktop app and a blue "DOWNLOAD RSTUDIO DESKTOP FOR MACOS 11+" button. A red oval highlights the "DOWNLOAD RSTUDIO DESKTOP FOR MACOS 11+" button. At the bottom, there is additional text about file size, SHA-256 hash, version, and release date.

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

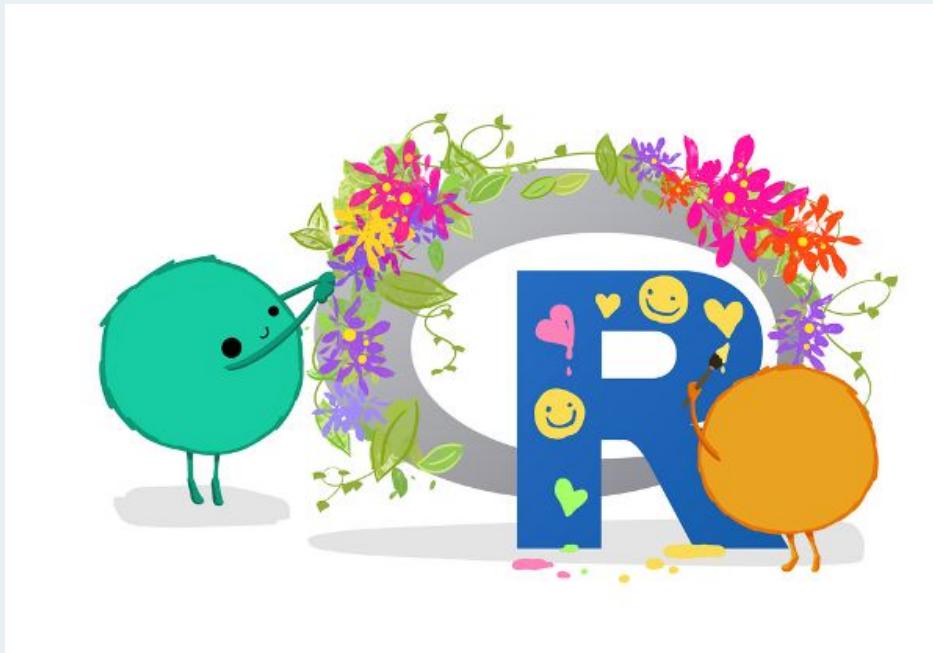
2: Install RStudio

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

Now you try!

Download R and R Studio.



How do I use RStudio?

RStudio console

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Displays the `graphic.R` script. The code reads a `countries.tsv` file, summarizes it, and prints summary info.
- Global Environment:** Shows a data frame named `countries` with 194 observations and 9 variables.
- Help Viewer:** Shows the documentation for the `duplicated` function.
- Terminal:** Shows the command line history and the output of the `countries` command.

```
graphic.R
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_je = sum(completed_je == 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 analysis of this dataset, continued below
```

```
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_je = sum(completed_je == 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_je completed_naphs published_naphs published_naphs_data machine_readable_data
1                  194              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.

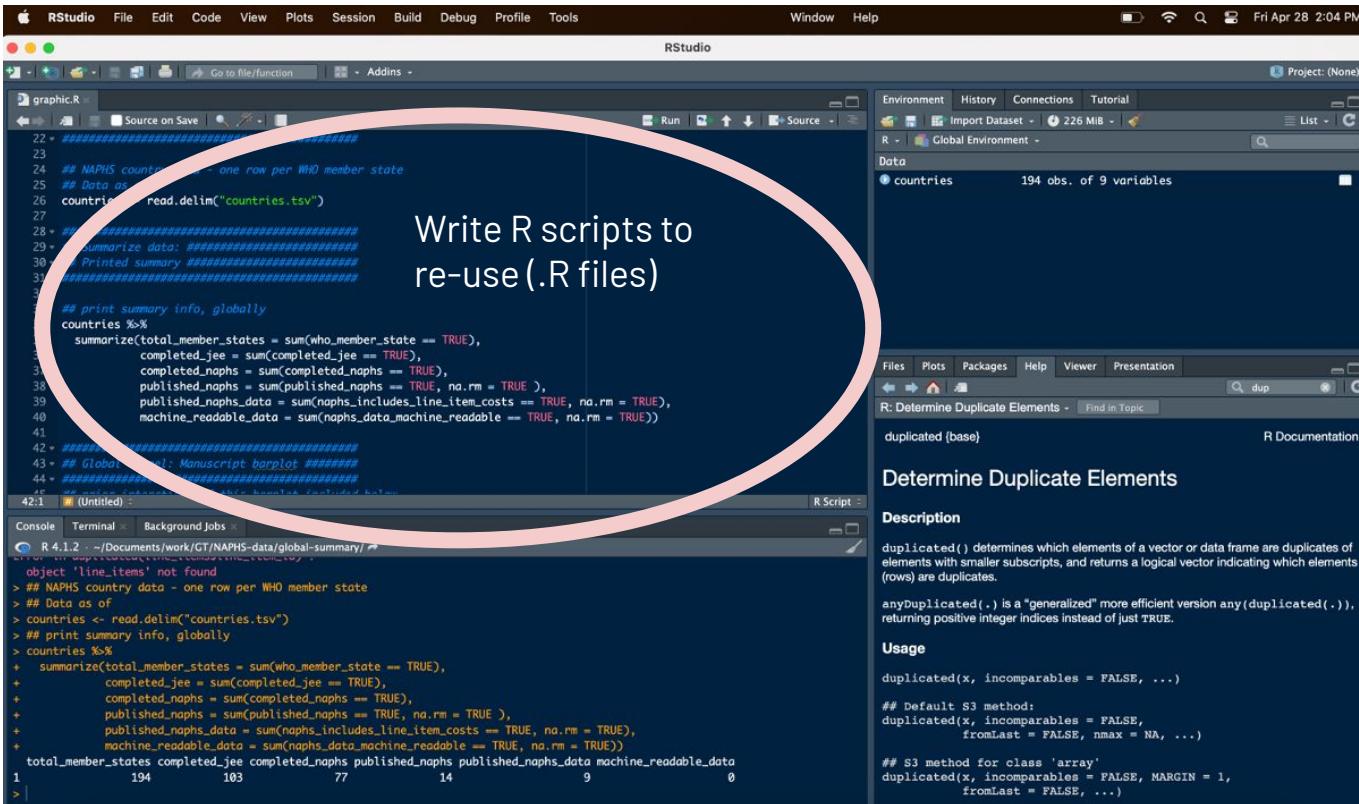
`anyDuplicated(.)` is a "generalized" more efficient version `any(duplicated(.))`, returning positive integer indices instead of just `TRUE`.

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



RStudio console

The screenshot shows the RStudio interface with several panels:

- Code Editor (left):** Displays the `graphic.R` script with R code. A pink circle highlights the **Run** button at the top of the editor.
- Global Environment (top right):** Shows the `countries` dataset with 194 observations and 9 variables.
- Help Documentation (bottom right):** Shows the `duplicated` function from the base package, with sections for Description, Usage, and Source code.
- Console (bottom left):** Shows the R command-line interface with history and output. A pink oval highlights the area where one-off code can be run.

This button runs code (points to the Run button in the code editor)

Run one-off code here, or see the results of code you ran from above (points to the Console area)

```
graphic.R
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_je = sum(completed_je == 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+
```

```
R 4.1.2 · ~/Desktop/work/GT/NAPHS-data/global-summary/ ↵
object 'naphs' 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_je = sum(completed_je == 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_je completed_naphs published_naphs published_naphs_data machine_readable_data
1                      103                  77                 14                   9                   0
```

```
duplicated {base}
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 details:

- File Menu:** RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools.
- Toolbar:** Go to file/function, Addins, Run, Source, Up, Down, Source.
- Code Editor:** A script named "graphic.R" containing R code for summarizing NAPHS country data.
- Environment Pane:** Shows the "Global Environment" tab with a data frame named "countries".

	countries	194 obs. of 9 variables
total_member_states	194	103
completed_jeo	103	77
completed_naphs	77	14
published_naphs	14	9
published_naphs_data	9	0
published_naphs_includes_line_item_costs	0	
machine_readable_data		
- Text Editor:** An "Untitled" R Script showing the same R code as the editor.
- Console:** Displays the R session history with the "line.items" object not found error.
- Help:** Shows the "duplicated" function documentation.
- Annotations:** A pink circle highlights the "Global Environment" section in the Environment pane, and the text "Stuff you have in memory" is overlaid on the circled area.

RStudio console

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Displays the `graphic.R` script. The code is a script to summarize NAPHS country data. It includes sections for reading data, summarizing, and printing summary info. It also contains a global funnel plot section.
- Global Environment:** Shows a data frame named `countries` with 194 observations and 9 variables.
- Documentation:** A circular callout highlights the documentation for the `duplicated` function. The documentation page is titled "Determine Duplicate Elements" and includes sections for "Description", "Usage", and "Documentation and figures".
- Console:** Shows the R command history and the output of the `countries` summary.

```
## 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_je = sum(completed_je == 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 #####
#####

#> countries
#> #> countries
#> countries %>%
#>   summarize(total_member_states = sum(who_member_state == TRUE),
#>             completed_je = sum(completed_je == 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_je completed_naphs published_naphs published_naphs_data machine_readable_data
#> 1          194         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`.

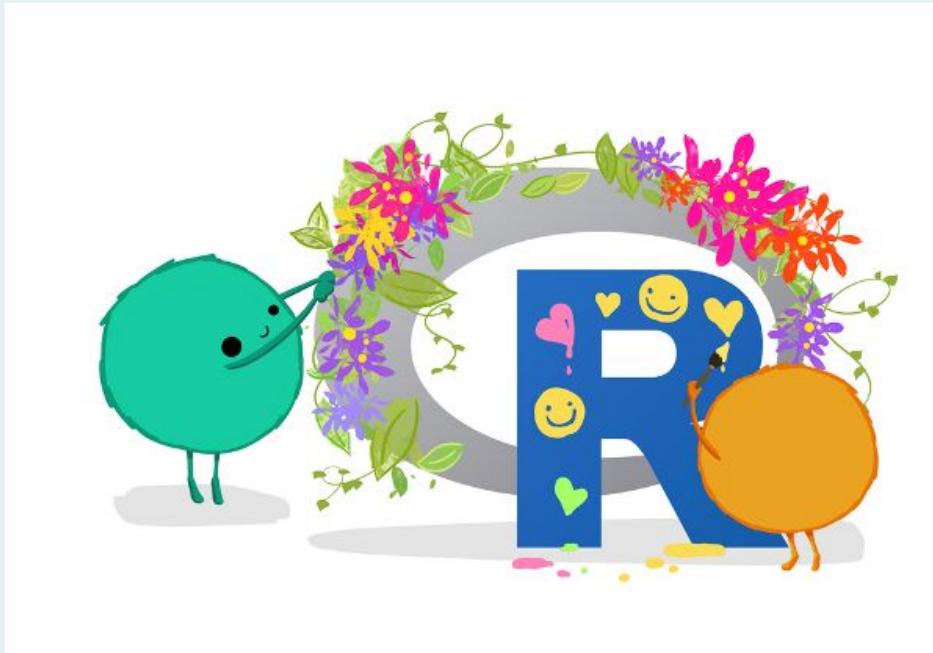
Usage

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

Documentation and figures

Now you try!

Open RStudio on your computer and click around



Downloading R

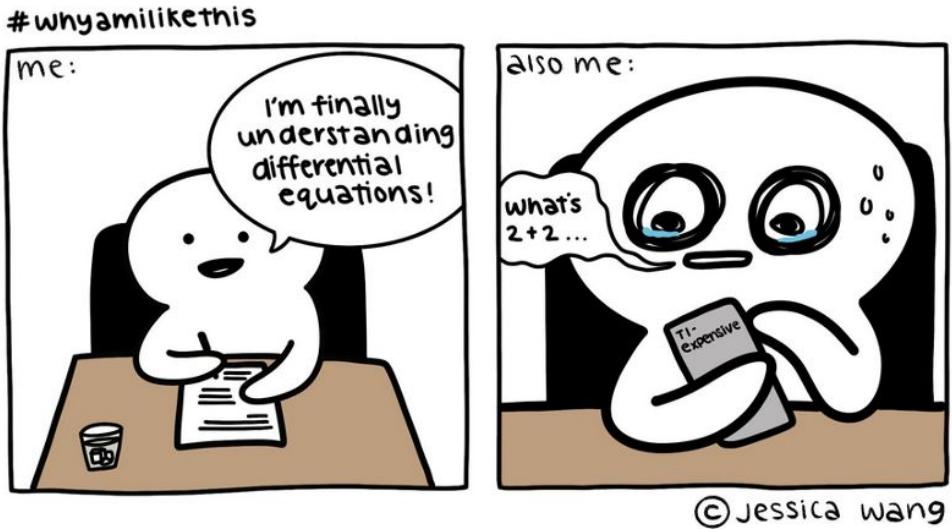
(and a note on worked examples)

- If you had issues downloading materials in today's class, 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)
- **Logarithms** ($\log(100)$)
- **Square root** ($\sqrt{9}$)
- **Absolute value** ($\text{abs}(-10)$)
- **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

The screenshot shows the RStudio interface with the following components:

- Top Bar:** Displays the title "RStudio" and a "Project: (None)" indicator.
- File Explorer:** Shows two files: "Untitled1" and "Day1.Rmd".
- Toolbar:** Includes icons for file operations like New, Open, Save, and Print, along with "Go to file/function" and "Addins" dropdowns.
- Code Editor:** An empty R script editor window with tabs for "Console", "Terminal", "Render", and "Background Jobs".
- Console:** Displays R session history:

```
R 4.1.2 · ~/Documents/learning/tidyTuesday/2024/otherdata9_gotmortality/ →  
> my_first_object <- 3  
>  
> my_second_object <- "steph"  
>  
> |
```
- Environment Tab:** Shows the "Global Environment" pane with the following objects:

Values
my_first_object 3
my_second_object "steph"
- Message Box:** A pink circle highlights the "Values" section in the Environment tab, with the text "Our new objects are shown in this corner" overlaid.
- Bottom Navigation:** Includes tabs for "Files", "Plots", "Packages", "Help", "Viewer", and "Presentation".

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.



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

```
is(-1)
```

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

```
is(TRUE)
```

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

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

```
## [1] "character"      "vector"  
## [4] "SuperClassMethod"
```

```
"data.frameRowLabels"
```

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"
```



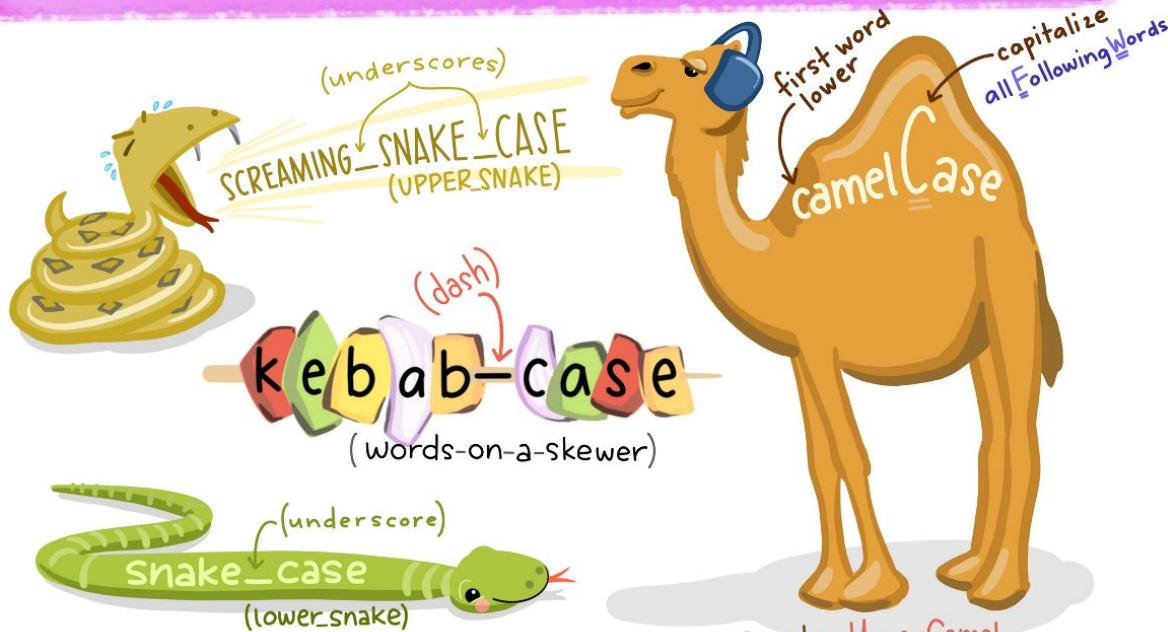
Artwork by Allison Horst
<https://allisonhorst.com/everything-else>

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
- Choose 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...



Rules for naming objects

failed programming cases



(flat case, but capitalize)
the final letter



(upper camel, but)
upside down



(screaming snake)
but with chunks
missing



@ 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

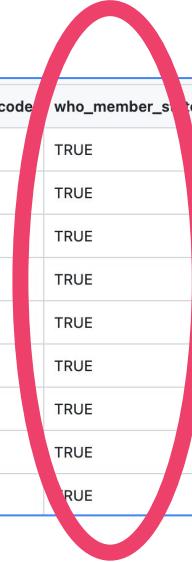
10 minute break



If at any point you wanna take a

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. In the top-left pane, there are two open files: "day1_Rslides.Rmd" and "Untitled1". The top-right pane displays the Global Environment, showing two objects: "my_first_object" with value 3.14 and "my_second_object" with value 2. The bottom-left pane is the Console, where the command "?summary" has been entered. The bottom-right pane is the Help browser, specifically the "Object Summaries" section for the "summary" function. A pink oval highlights the title "Object Summaries" and the "Description" and "Usage" sections of the help page.

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(maximum = 7,
        digits = maxgetOption("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()`

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>



data-science-basics-2024

Public



Pin



Unwatch

1



Fork

0



Star

0



About

In progress materials for the data science basics course in 2024



Readme



Activity



0 stars



1 watching



0 forks



Releases

No releases published

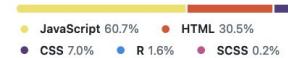
[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages



Suggested workflows

Based on your tech stack



G

Grunt

Configure

Build a NodeJS project with npm and grunt.



W

Webpack

Configure

Build a NodeJS project with npm and webpack.



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.



data-science-basics-2024

Public



Pin



Unwatch

1



Fork

0



Star

0

main

2 Branches

0

Tags

Go to file

t

Add file

Code

seaneff add mcv1 data

9e8e441 · 2 weeks ago

21 Commits

course-datasets

add mcv1 data

2 weeks ago

day1

Day 1 course materials

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.

About



In progress materials for the data science basics course in 2024

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

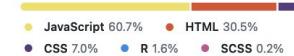
[Create a new release](#)

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>

Github, poetry, and unicorns

A brief adventure on youtube

<https://www.youtube.com/watch?v=BCQHnInPusY>



1.1: Introduction - Git and GitHub for Poets



The Coding Train
1.66M subscribers

Join

Subscribe



12K



...



Share



Thanks



Github organization

README files

- README files make it easier for your collaborators to find and understand work within your github repository
- In general, readme files should contain:
 - What the project/code does
 - Why the project/code is useful
 - Where people can get help or find more info

Read more on [github](#)

Github organization

creating a README file

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



seaneff

Repository name *



teaching example

✓ Your new repository will be created as teaching-example.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [shiny-octo-carnival](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

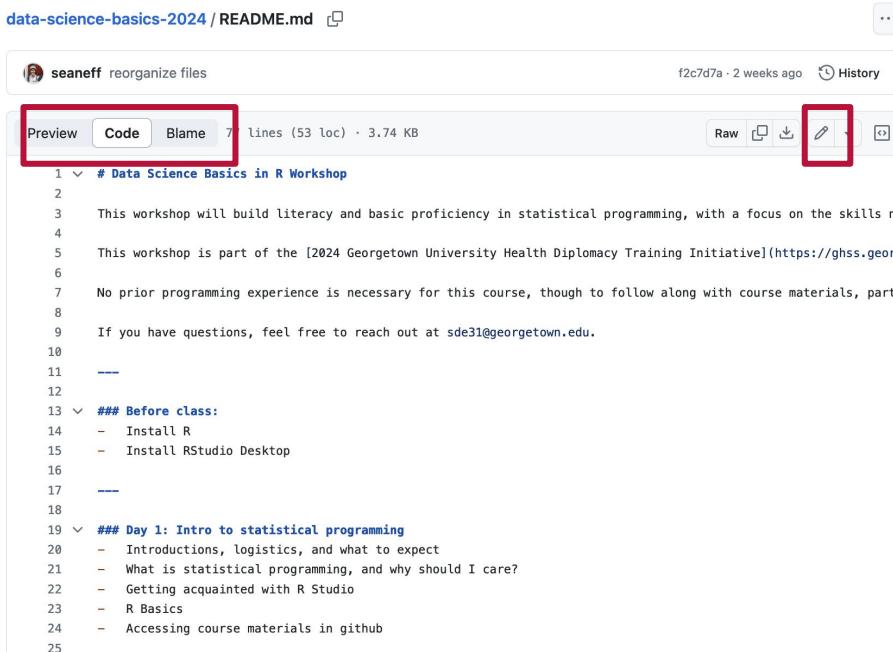
- When you create a new github repository, you can click a box to add a README file
- If you forget to do this, you can always simply add a new file to your repo and title it README.md, github will recognize this

Read more on [github](#)

Github organization

Editing a README file

data-science-basics-2024 / README.md 



seaneff reorganize files f2c7d7a · 2 weeks ago History

Preview Code Blame 71 lines (53 loc) · 3.74 KB

Raw    

```
1 # Data Science Basics in R Workshop
2
3 This workshop will build literacy and basic proficiency in statistical programming, with a focus on the skills ne
4
5 This workshop is part of the [2024 Georgetown University Health Diplomacy Training Initiative](https://ghss.georg
6
7 No prior programming experience is necessary for this course, though to follow along with course materials, parti
8
9 If you have questions, feel free to reach out at sde31@georgetown.edu.
10
11 ---
12
13 ## Before class:
14 - Install R
15 - Install RStudio Desktop
16
17 ---
18
19 ## Day 1: Intro to statistical programming
20 - Introductions, logistics, and what to expect
21 - What is statistical programming, and why should I care?
22 - Getting acquainted with R Studio
23 - R Basics
24 - Accessing course materials in github
25
```

 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.

Before class:

- Install R
- Install RStudio Desktop

Day 1: Intro to statistical programming

- Introductions, logistics, and what to expect

Course [readme file](#)
Main course [github page](#)

Github organization

Organizing files in github

- There are lots of different ways to organize files in github, and there's no one *best* solution. You should think about who is using the repo, and how, when you decide how to structure it.
- One common approach is to use different folders within one repo:
 - **data/** for datasets that open/sharable and small enough for git
 - **code/** for your analysis code, which may be split up further
 - **results/** for key outputs
 - **figures/** for figures, especially if you're publishing

Your turn!

Create a github account at <https://github.com/>

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!

Explore some useful
sources of health
diplomacy data

WHO Global Health Observatory

<https://www.who.int/data/gho>



The banner features a dark purple background with abstract white and light purple geometric patterns. In the center, the text "THE GLOBAL HEALTH OBSERVATORY" is written in a bold, white, sans-serif font. Below it, the tagline "Explore a world of health data" is displayed in a slightly smaller white font. At the bottom of the banner, there are two white rectangular buttons with rounded corners, each containing the text "Indicators >" and "Countries >" respectively, in a small blue font.

Health Topics ▾ Countries ▾ Newsroom ▾ Emergencies ▾ Data ▾ About WHO ▾

GHO Home Indicators Countries Data API ▾ Map Gallery Publications Data Search

THE GLOBAL HEALTH OBSERVATORY

Explore a world of health data

Indicators > Countries >

OECD Health Data Repository

<https://data.oecd.org/health.htm>

Screenshot of the OECD Health Data Repository website.

The page features a top navigation bar with links to OECD.org, Data, Publications, More sites, News, and Job vacancies. It also includes a search bar for "Search for OECD data" and language links for "Français".

The main content area has a header "OECD Data" with a "Health" dropdown menu. Below this, there's a sidebar titled "Topic aspects" listing "Health care use", "Health equipment", "Health resources", "Health risks", and "Health status".

The main content area displays three bar charts under the "Health care use" section:

- Doctors' consultations**: A bar chart showing data for 2021. The Y-axis ranges from 0 to 6. The chart shows values approximately 1.5, 3.0, 3.0, 5.5, and 6.5 across five categories.
- Length of hospital stay**: A bar chart showing data for 2021. The Y-axis ranges from 0 to 6. The chart shows values approximately 4.5, 5.5, 6.0, 6.5, and 6.5 across five categories.
- Health at a Glance**: A thumbnail image of the publication cover, labeled as a "PUBLICATION (2021)".

At the bottom, there's a section titled "INDICATOR GROUP" with a link to "Health care use".

OCHA Humanitarian Data Exchange

<https://data.humdata.org/>

The screenshot shows the homepage of the OCHA Humanitarian Data Exchange (HDX) at <https://data.humdata.org/>. The page has a dark header bar with the OCHA Services logo, a search bar, and navigation links for DATA, LOCATIONS, ORGANISATIONS, and DATAVIZ. A red "ADD DATA" button is also in the header. The main content area features a teal background with the title "The Humanitarian Data Exchange" and a subtitle "Find, share and use humanitarian data all in one place". Below this is a "LEARN MORE" button. To the right, there are two main sections: "FIND DATA" and "ADD DATA". The "FIND DATA" section displays statistics: 20,758 DATASETS, 254 LOCATIONS, and 1,890 SOURCES. The "ADD DATA" section includes options to "UPLOAD FILE" or "ADD METADATA". A red banner at the bottom right provides information about responsible data sharing support from the HDX team.

OCHA Services ▾

HDX Search Datasets

DATA | LOCATIONS | ORGANISATIONS | DATAVIZ ▾ ADD DATA

The Humanitarian Data Exchange

Find, share and use humanitarian data all in one place

LEARN MORE

FIND DATA

Search Datasets

20,758 DATASETS | 254 LOCATIONS | 1,890 SOURCES

ADD DATA

UPLOAD FILE ADD METADATA

Learn how the HDX team supports responsible data sharing.

Demographic and Health Surveys

<https://dhsprogram.com/>



SEARCH

LOGIN

Select Language ▾



COUNTRIES

DATA

PUBLICATIONS

METHODOLOGY

RESEARCH

TOPICS

The Demographic and Health Surveys (DHS) Program has collected, analyzed, and disseminated accurate and representative data on population, health, HIV, and nutrition through more than 400 surveys in over 90 countries.

Another and daughters in Ethiopia work with coffee beans after their house has received Indoor Residual Spraying (IRS) to reduce malaria transmission. Photo Credit: AIRS Ethiopia PMI

UN SDG Global Database

<https://unstats.un.org/sdgs/dataportal>

United Nations | Department of Economic and Social Affairs
Statistics • SDG Indicators Database

SDG Indicators Data Platform

Select indicators and country or area

SDG Global Database gives you access to data on more than 210 SDG indicators for countries across the globe

by indicator, country, region or time period

SUSTAINABLE DEVELOPMENT GOALS

UNDP Data Futures Platform

data.undp.org



Data Futures
Exchange

FOCUS AREAS

REGIONS & COUNTRIES

RESOURCES

ABOUT US

ACCESS ALL DATA



DATA FUTURES EXCHANGE

Data innovation for decision intelligence

GREENHOUSE GAS (GHG) EMISSIONS, MTCO₂E



0 50 100 250 500 1k 2.5k 5k

IPUMS

www.idhsdata.org/idhs/index.shtml

The screenshot shows the homepage of the IPUMS DHS website. At the top left is the IPUMS DHS logo. To its right is the text "DEMOGRAPHIC AND HEALTH SURVEYS". Below that is a navigation bar with links for "HOME", "SELECT DATA", "MY DATA", and "SUPPORT". A horizontal banner below the navigation bar displays six small images related to demographic and health surveys.

IPUMS DHS

ABOUT
THE DHS PROGRAM: HOME ↗
REGISTER
DONATE TO IPUMS ↗

DATA

BROWSE AND SELECT DATA
ANALYZE DATA ONLINE
DOWNLOAD OR REVISE MY DATA

SUPPLEMENTAL DATA

GEOGRAPHY & GIS
CONTEXTUAL VARIABLES

DOCUMENTATION

USER NOTES
SAMPLE DESCRIPTIONS
SAMPLE UNIVERSES
QUESTIONNAIRES
REVISION HISTORY

SUPPORT

FAQ

HEALTH-RELATED MICRODATA FOR LOW- AND MIDDLE-INCOME COUNTRIES

IPUMS-DHS facilitates analysis of Demographic and Health Surveys, administered in low- and middle-income countries since the 1980s. IPUMS-DHS contains thousands of consistently coded variables on the health and well-being of women, children, births, men, and on all members of randomly selected households, for 32 African countries and 9 Asian countries. Users can determine variable availability at a glance and create data files with just the variables and samples they need.

45 COUNTRIES – 180 SAMPLES – OVER 15,000 VARIABLES – 27 MILLION PERSON RECORDS

USE IT FOR GOOD – NEVER FOR EVIL

— CREATE AN EXTRACT — **ONLINE TOOL FOR ANALYSIS** — CREATE AN ACCOUNT —

Get Data Analyze Data Online Register

IDEA

<https://ghssidea.org/>



INTERNATIONAL DISEASE
IDEA
AND EVENTS ANALYSIS

[About](#) [Citations](#) [Contact](#)

The International Disease and Events Analysis (IDEA) platform is a suite of integrated research tools, including global health security visualization dashboards, decision support tools, and data libraries developed by the Georgetown University Center for Global Health Science and Security.

See below for a description of each tool, links to the sites, and access to a quick download of the data from each.

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
- Get familiar with Github (by writing poems about rainbows)

Plan for tomorrow

Data management and version control

- Understand the foundations of data management
- Load and clean your first dataset in R
 - explore a new dataset while learning about data structures
 - practice generating summary tables
- Learn best practices for documentation

Thank you!

See you tomorrow.

Please come with a fully charged laptop.