Exploratory Data Analysis with R

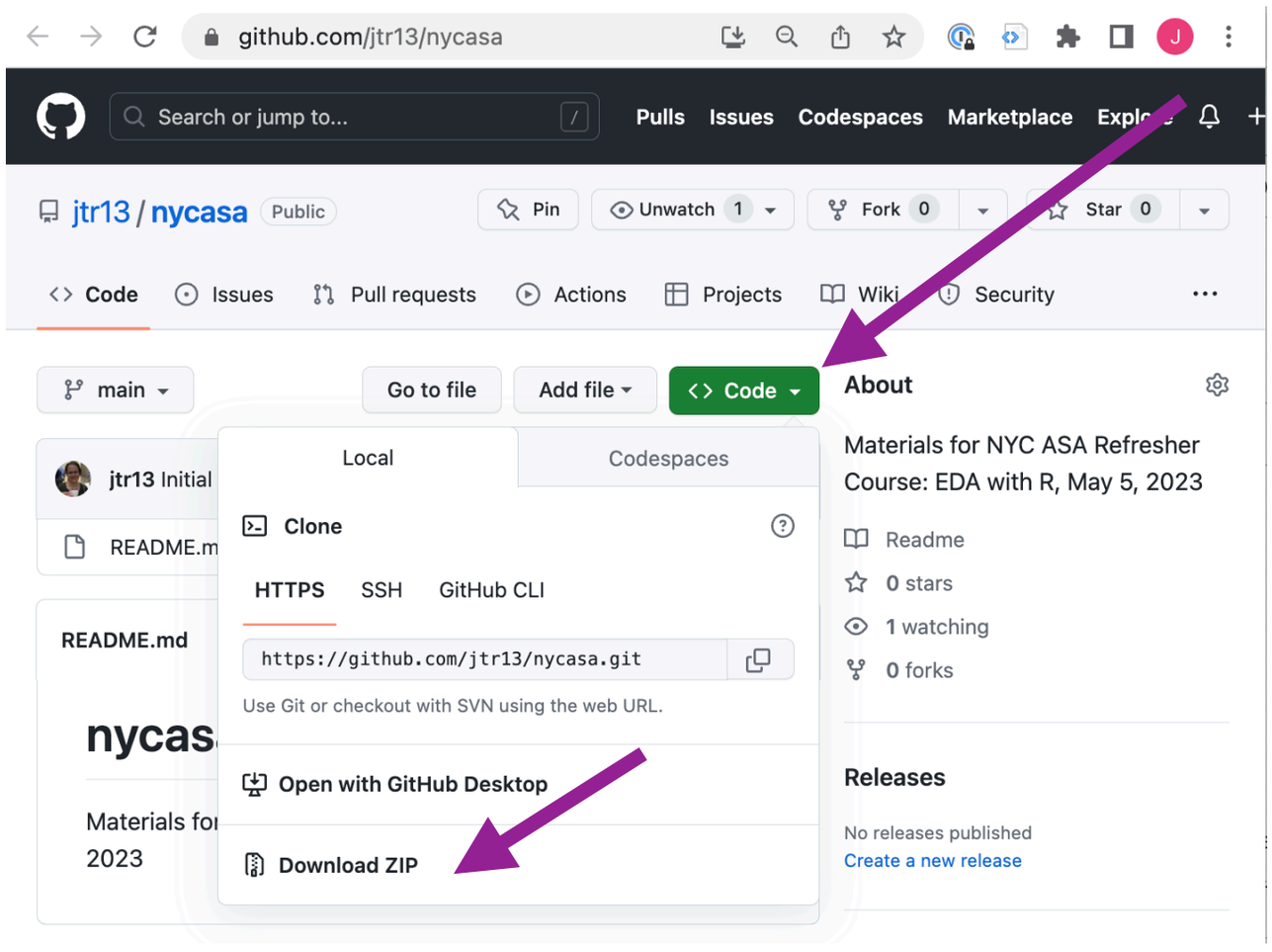
ASA NYC Chapter
Friday, May 5, 2023, 3pm-5pm

Joyce Robbins
Lecturer
Department of Statistics
Columbia University
jtr13@columbia.edu

Slides and code files

---

www.github.com/jtr13/nycasa



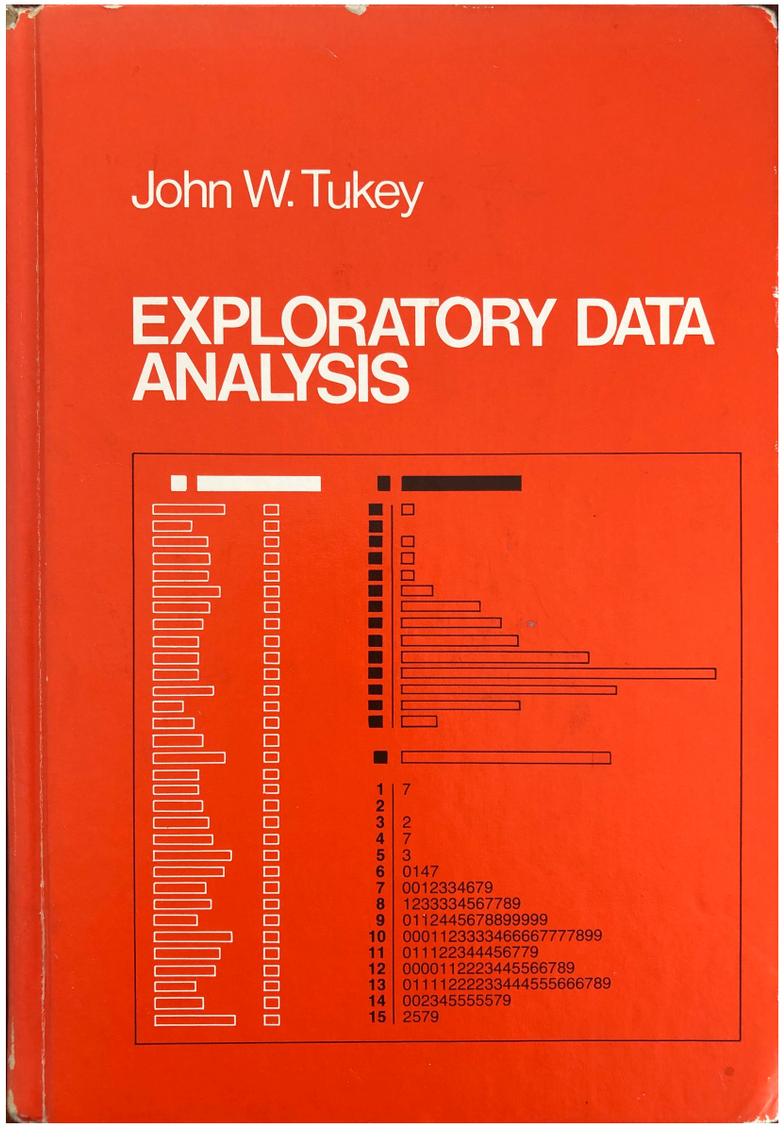
These slides: EDAwR-Slides1.pdf

Agenda

- What is EDA?
- Update / Setup R
- Finding data
- Exploring data

Exploratory Data Analysis

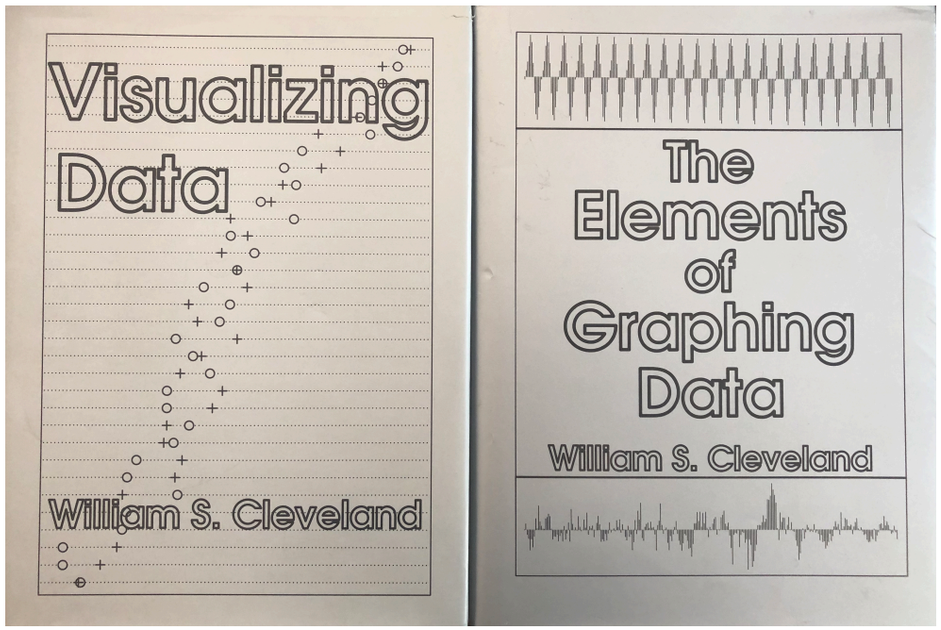
Exploratory Data Analysis



"Exploratory data analysis is detective work."

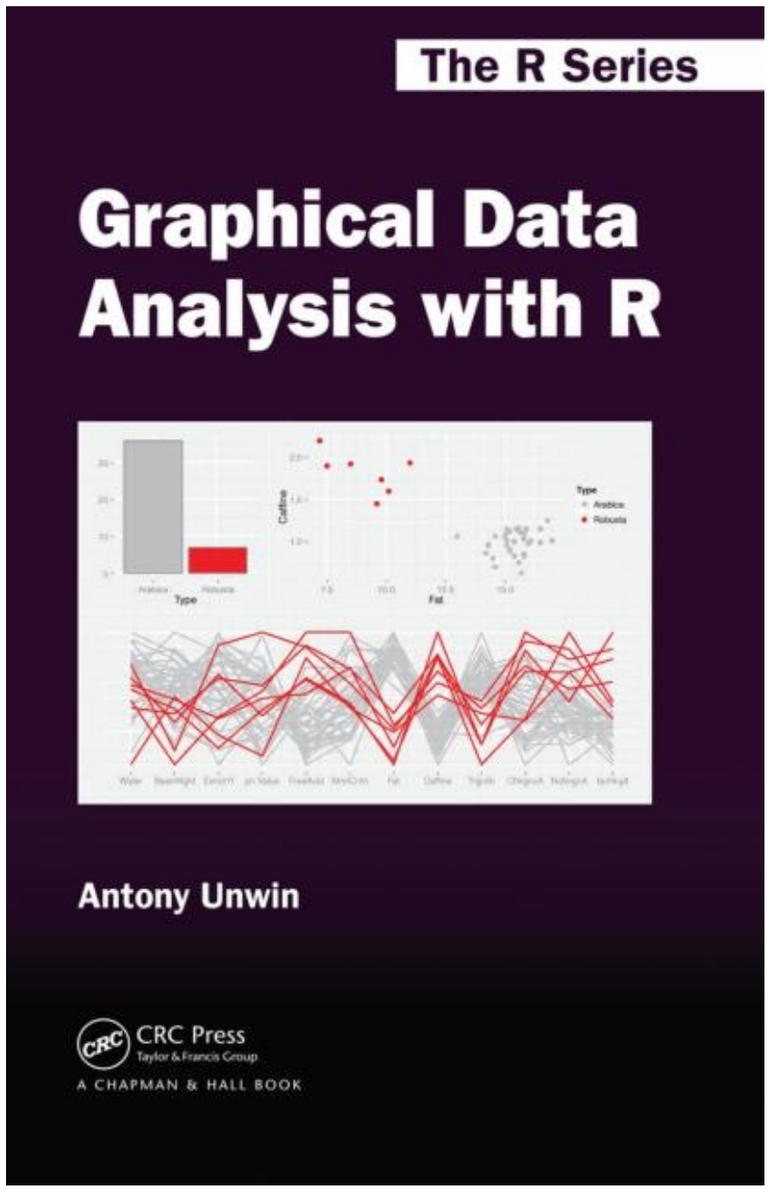
"Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone--as the first step."

William Cleveland 1980s, 1990s



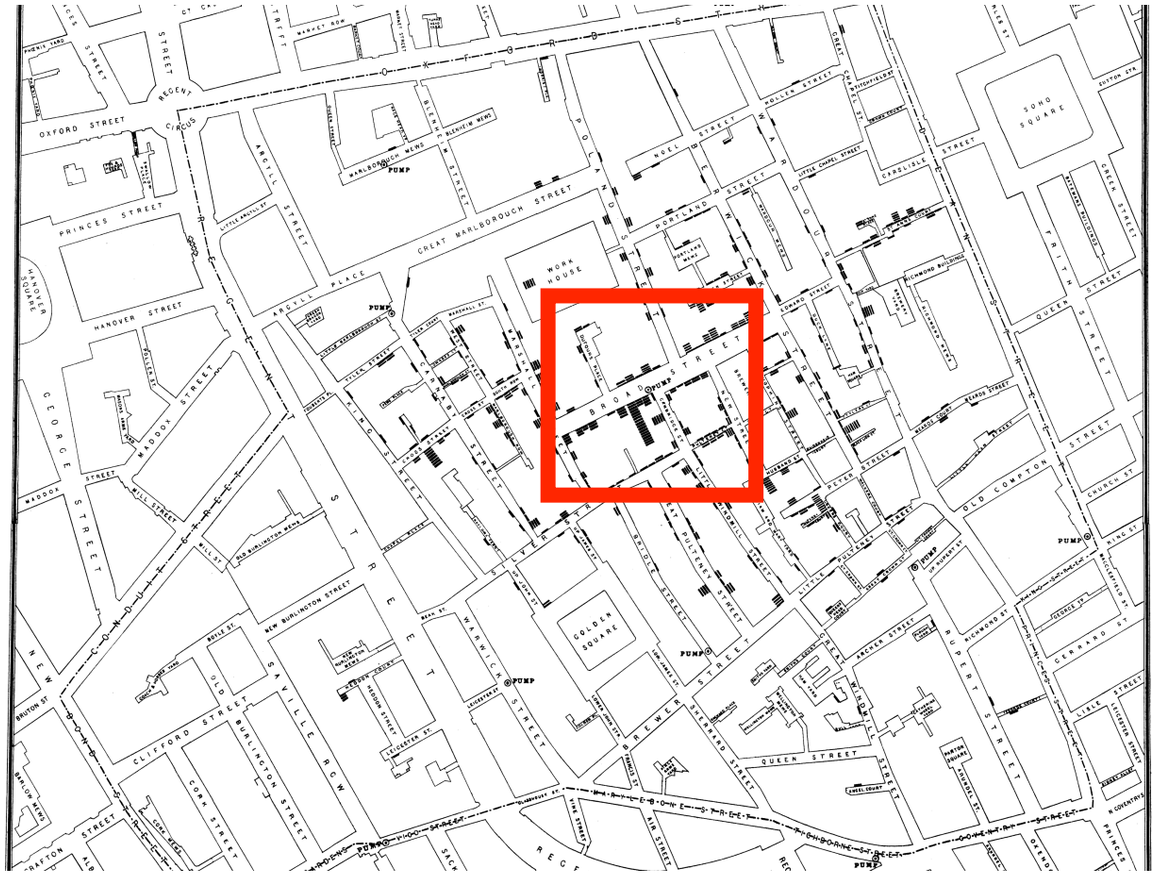
"Visualization is critical to data analysis. It provides a front line of attack, revealing intricate structure in data that cannot be absorbed in any other way. We discover unimagined effects, and we challenge imagined ones."

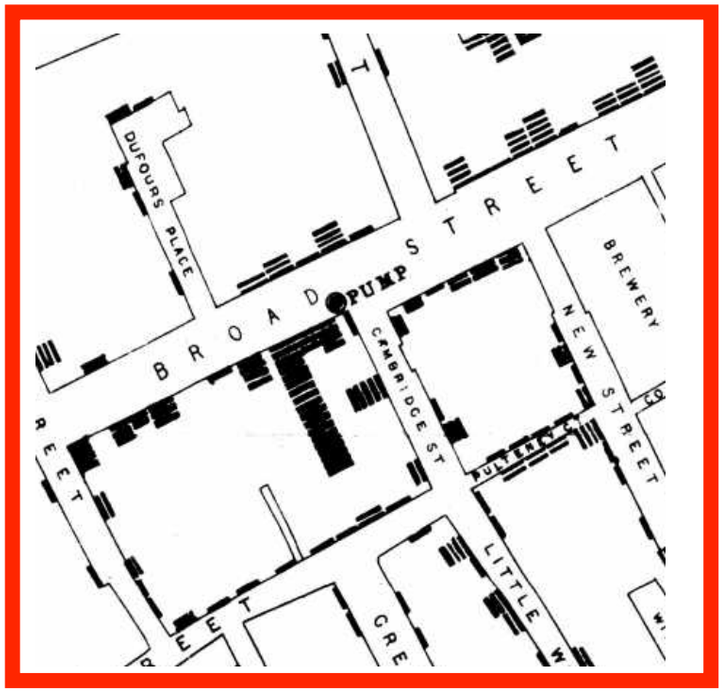
Graphical Data Analysis with R



- Strong EDA focus
- Code examples
- Textbook for my EDA course
- Multivariate approaches

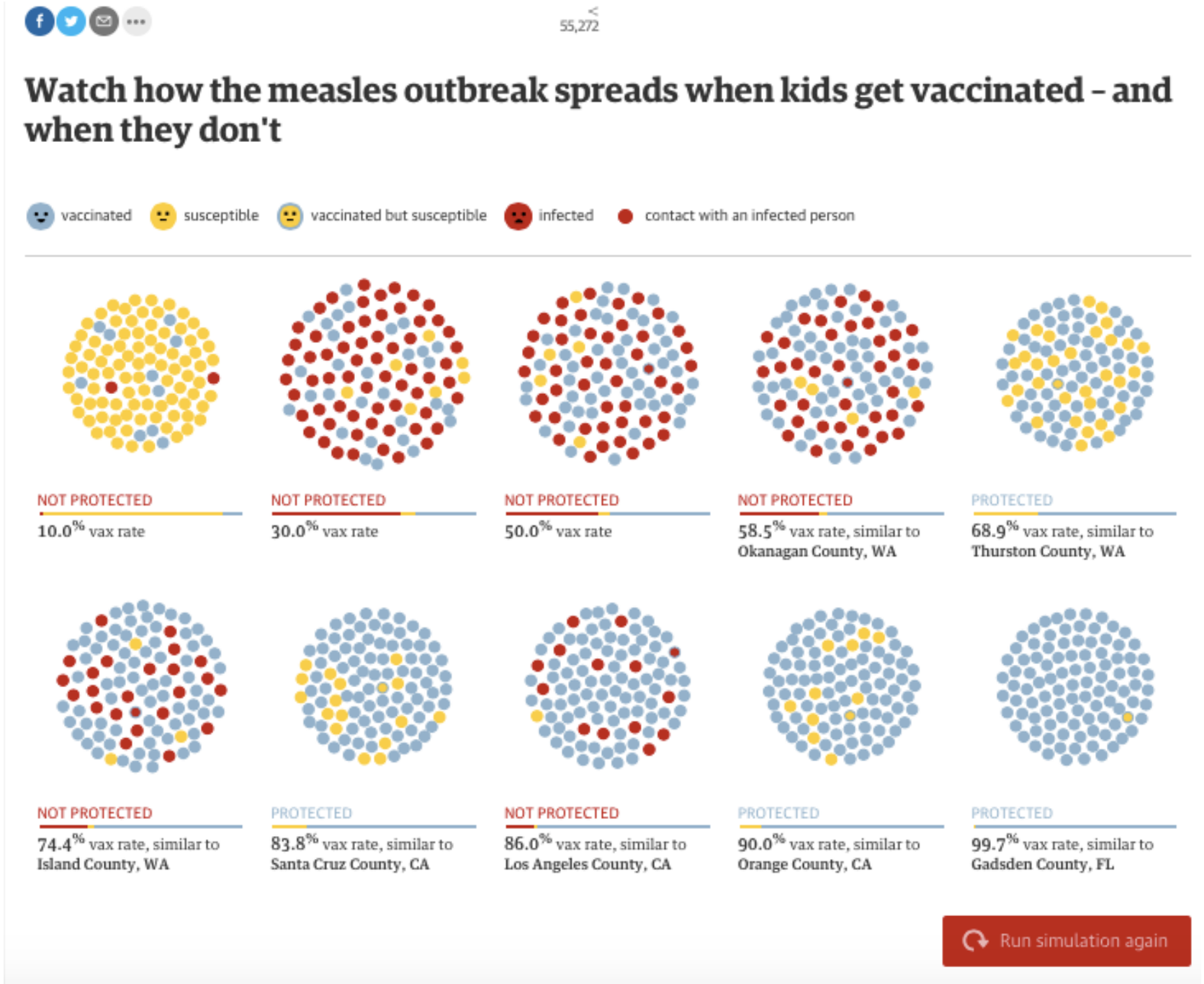
John Snow, Cholera Map 1854





Broad St. pump

Data Visualization



EDA vs. Data Visualization

- Data Visualization: how can I help the reader understand how vaccination affects the spread of disease?
- EDA: How does vaccination affect the spread of disease?

Setup

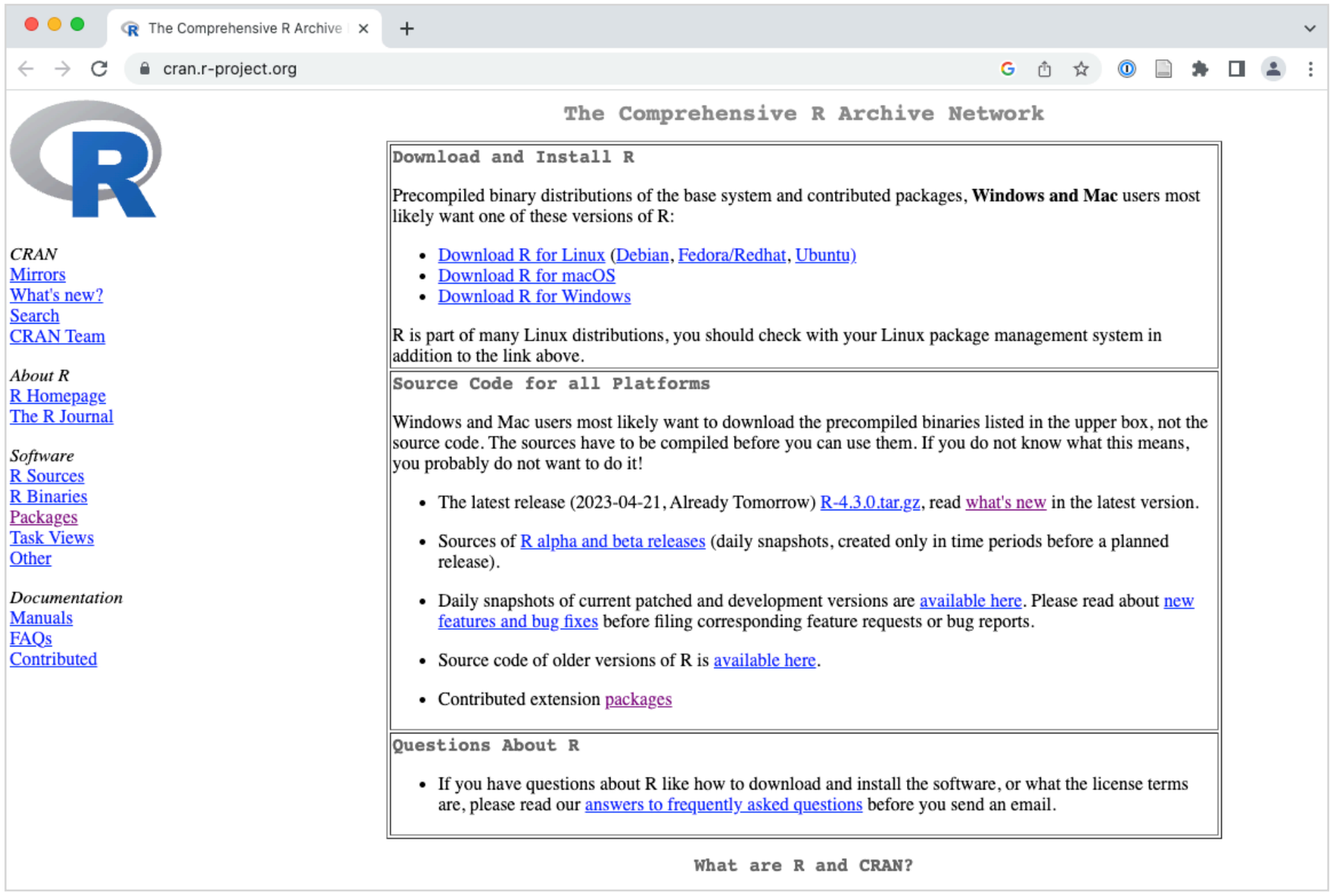
Update R (or install)

```
> R.version
```

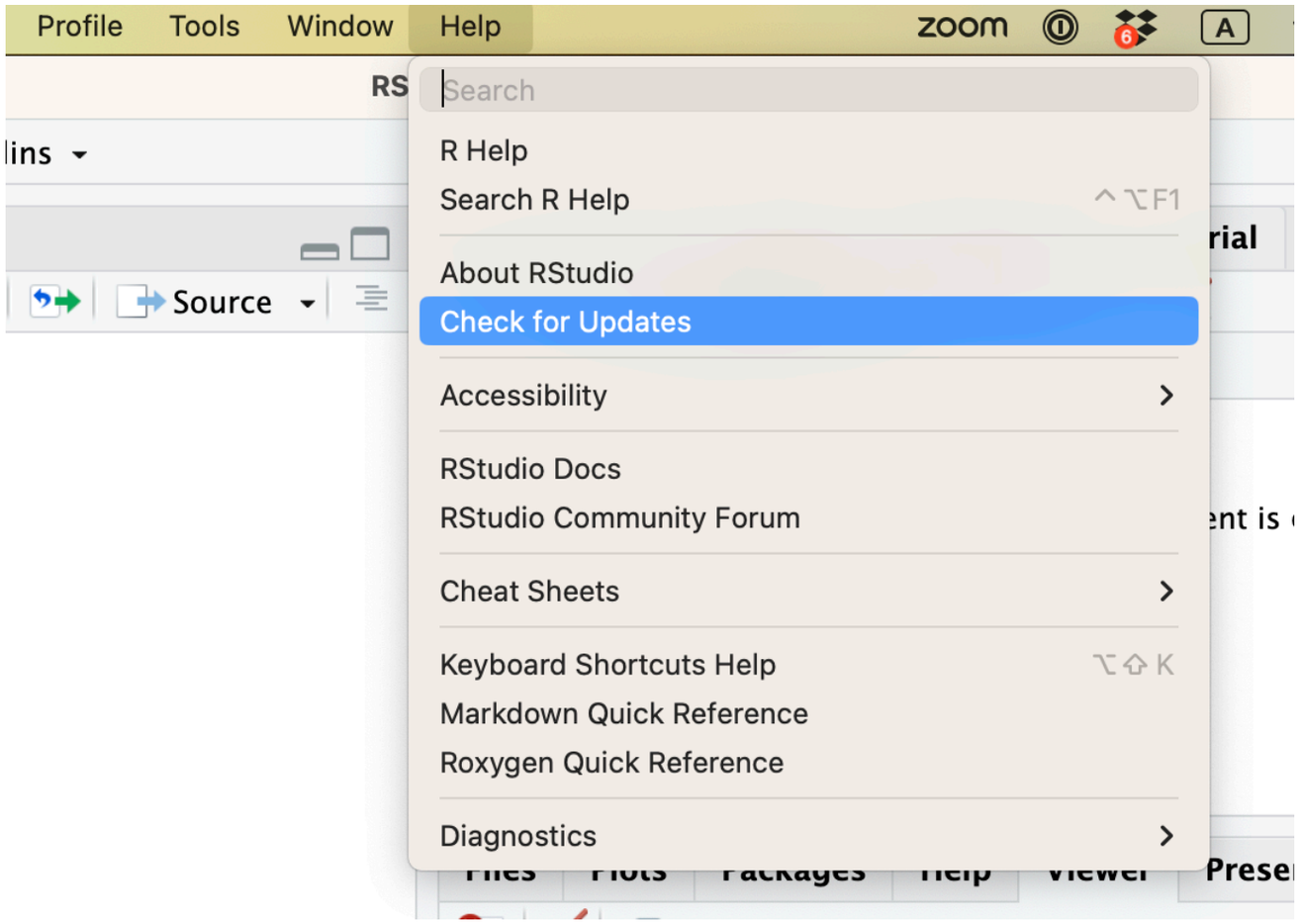
```
R.version
#>
#> platform       aarch64-apple-darwin20
#> arch           aarch64
#> os             darwin20
#> system         aarch64, darwin20
#> status
#> major          4
#> minor          3.0
#> year           2023
#> month          04
#> day            21
#> svn rev        84292
#> language       R
#> version.string R version 4.3.0 (2023-04-21)
#> nickname       Already Tomorrow
```

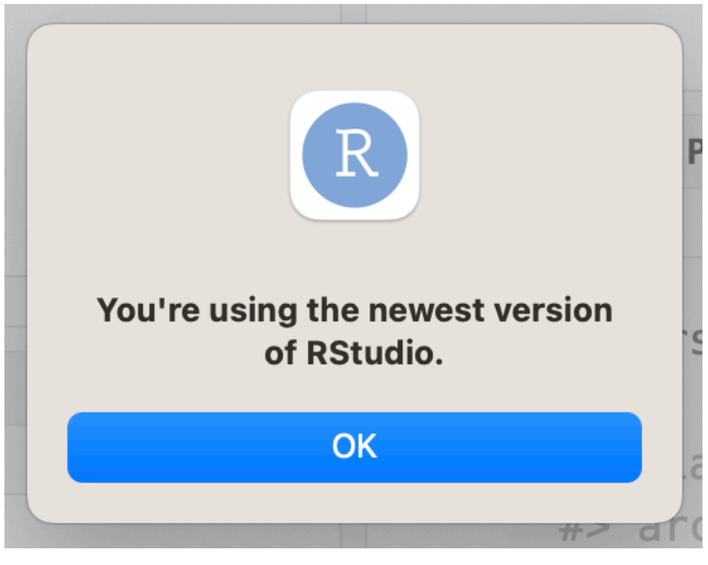
Update R (or install)

cran.r-project.org



Update RStudio

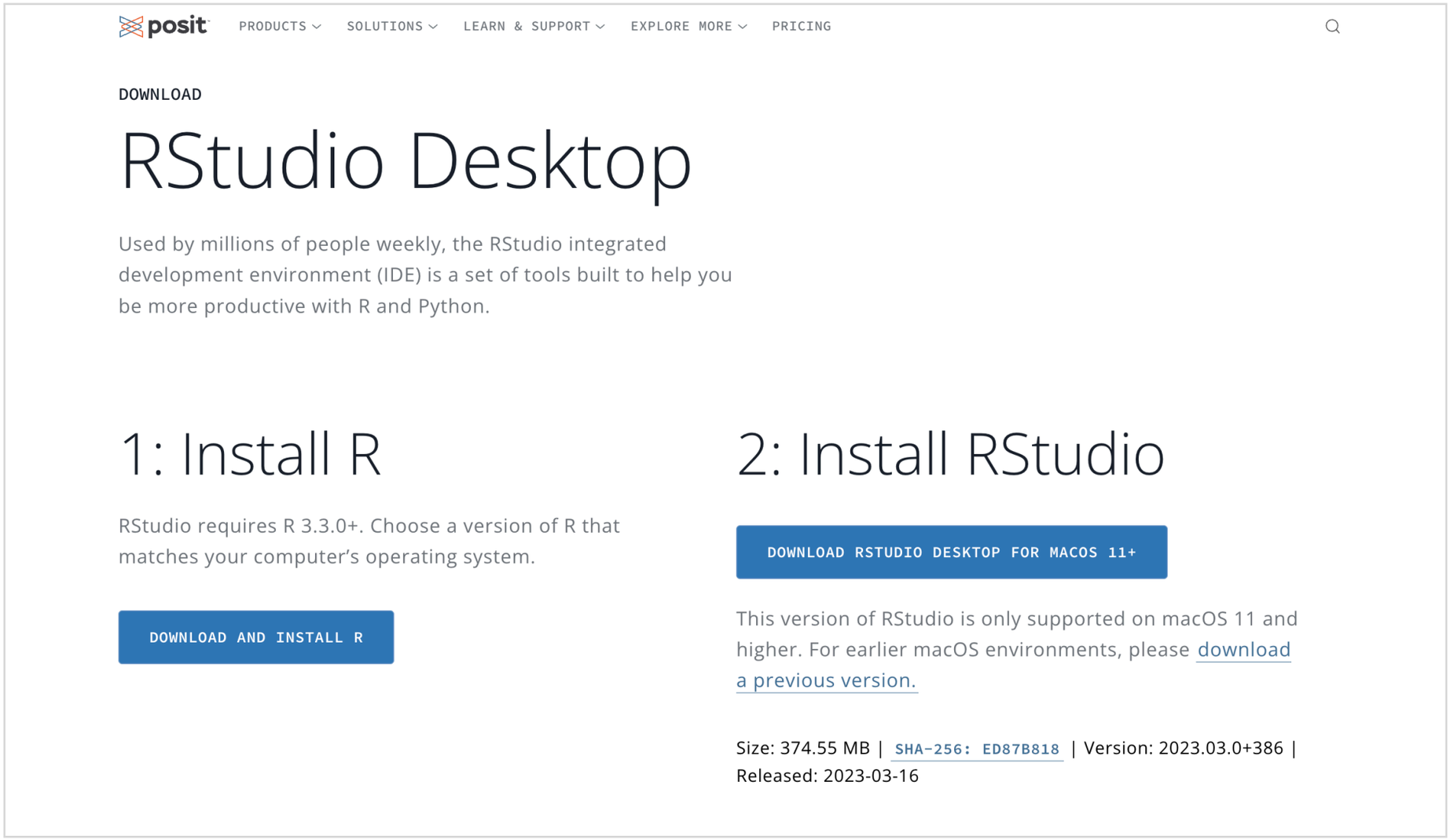




Or install RStudio

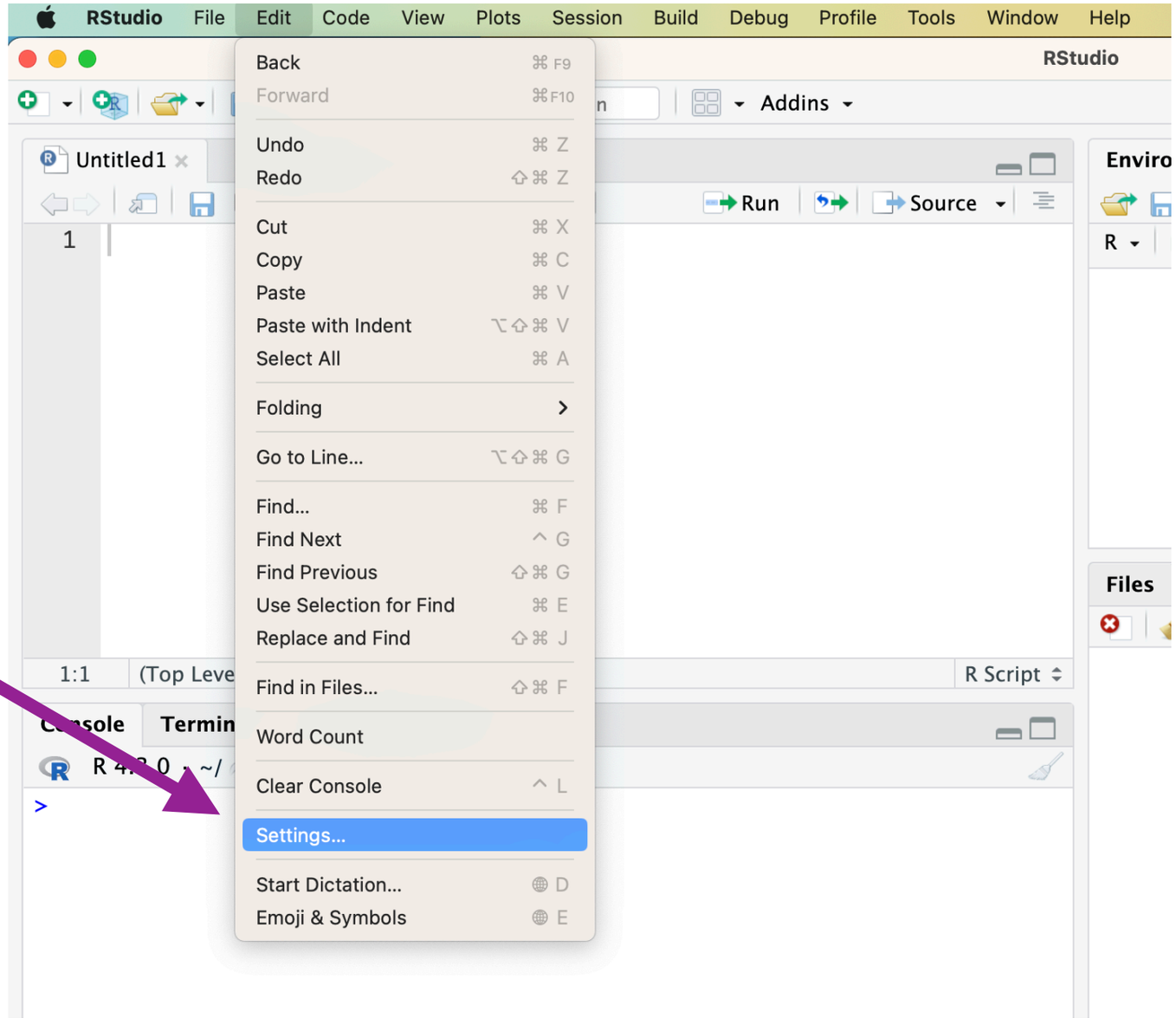
---

https://posit.co/download/rstudio-desktop/ (free)



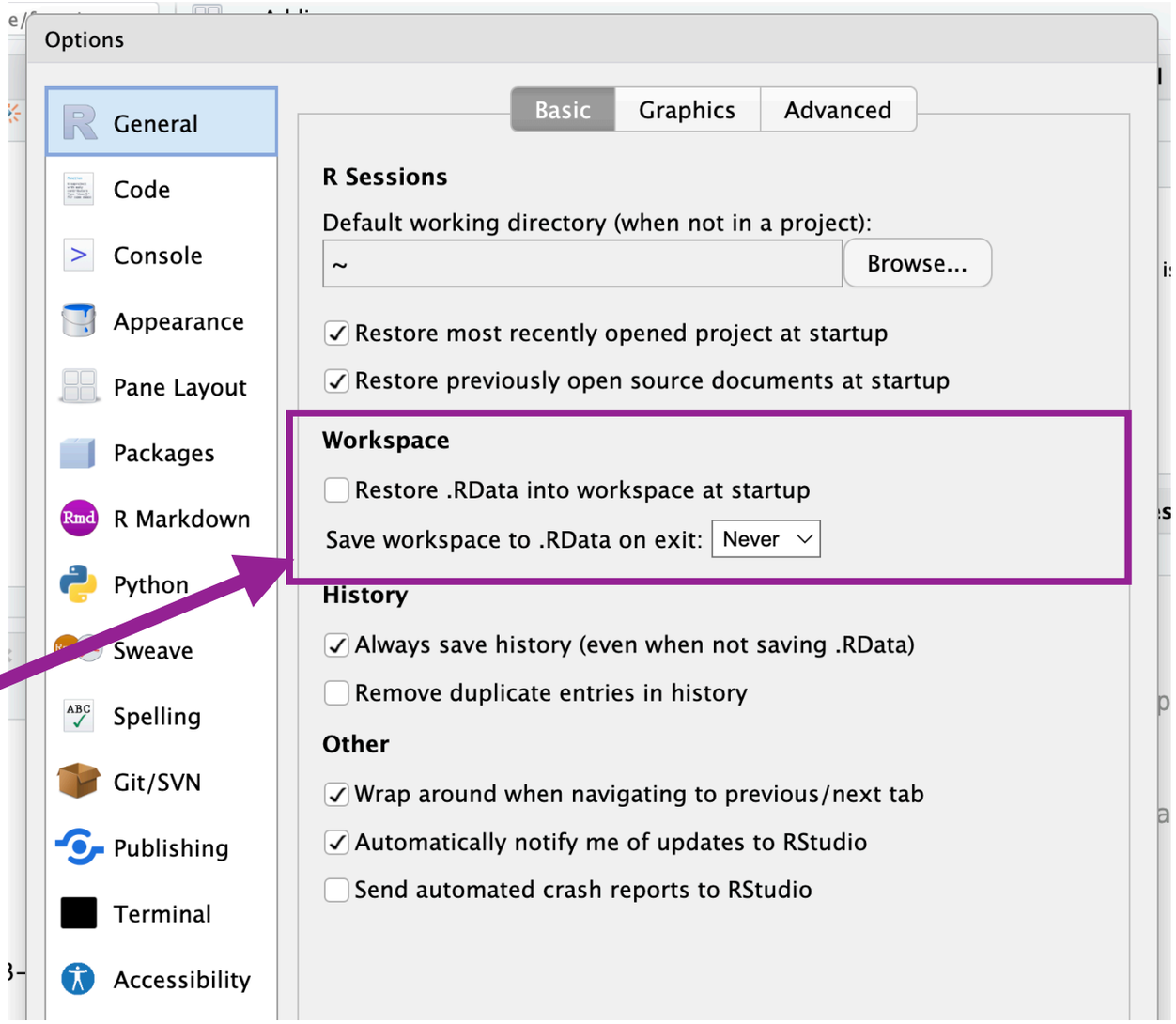
Setup RStudio

Choose "Edit", "Settings..."



Setup RStudio

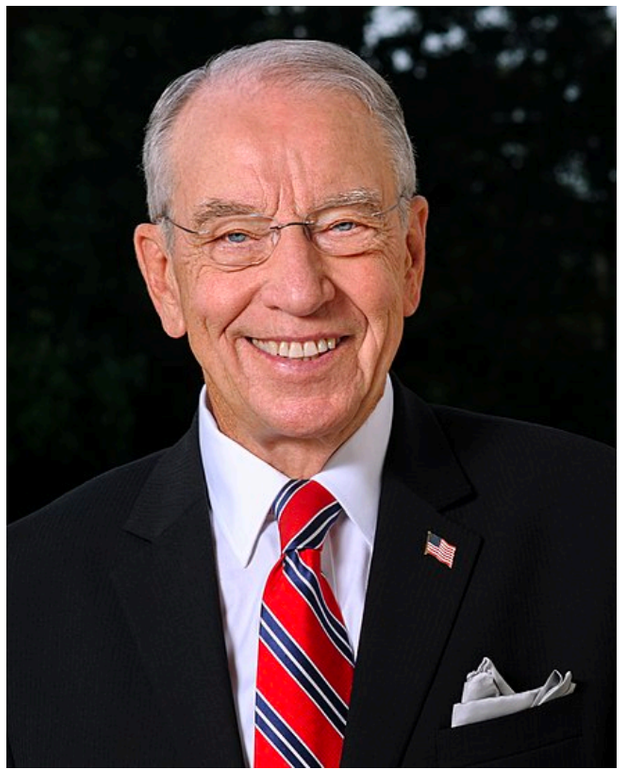
In the "General" "Workspace" section, choose these settings.



Work in projects

- Your files will be organized and in the right place.
- You will never have to use `getwd()` or `setwd()` ever again!
- It will be easy to switch between projects.
- It will be easy to sync with GitHub.
- It is simple -- a project is just a folder (project=folder=repo)

Senator Chuck Grassley's advice



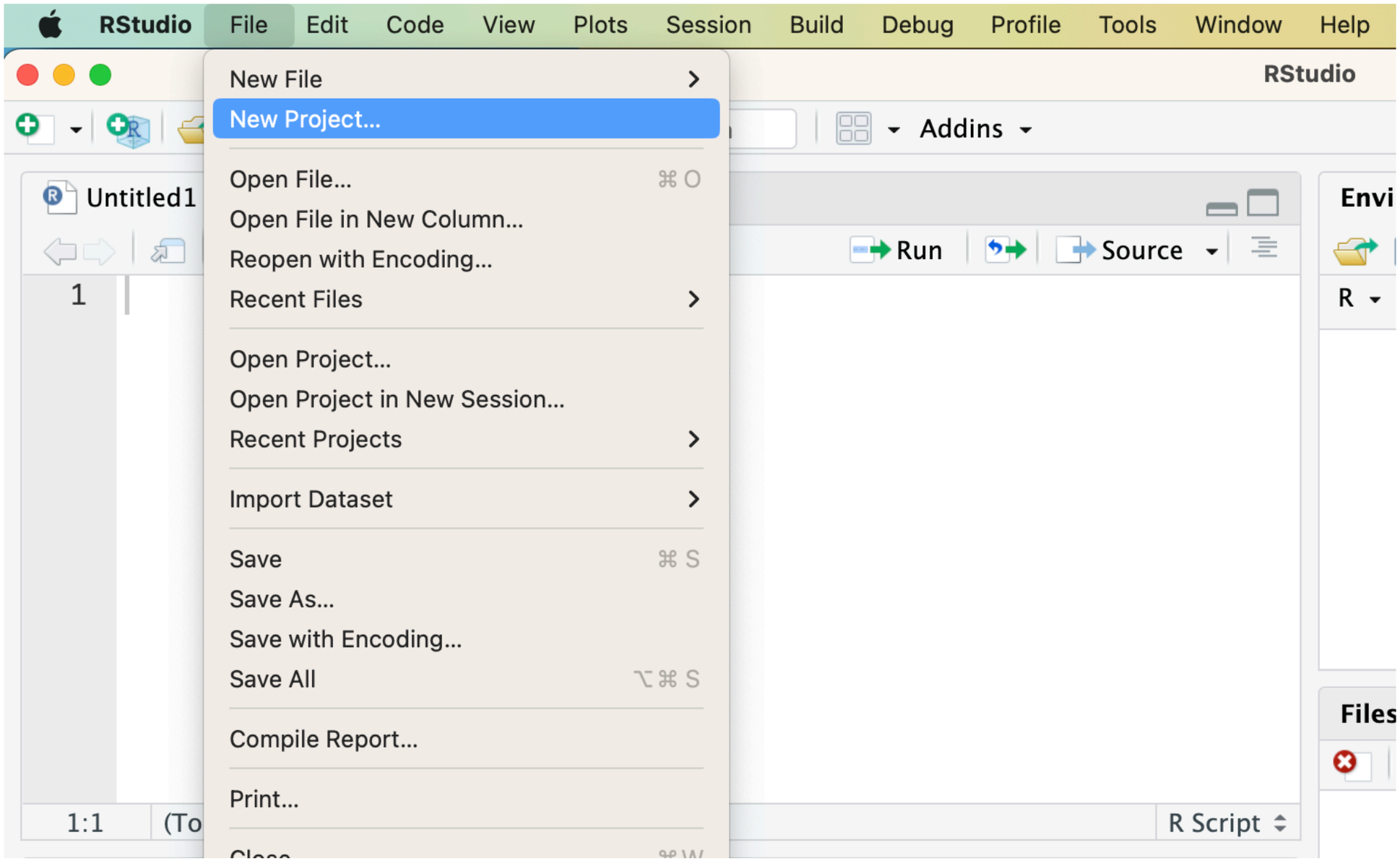
Wikipedia

- 2nd oldest senator (DOB: 9-17-33)

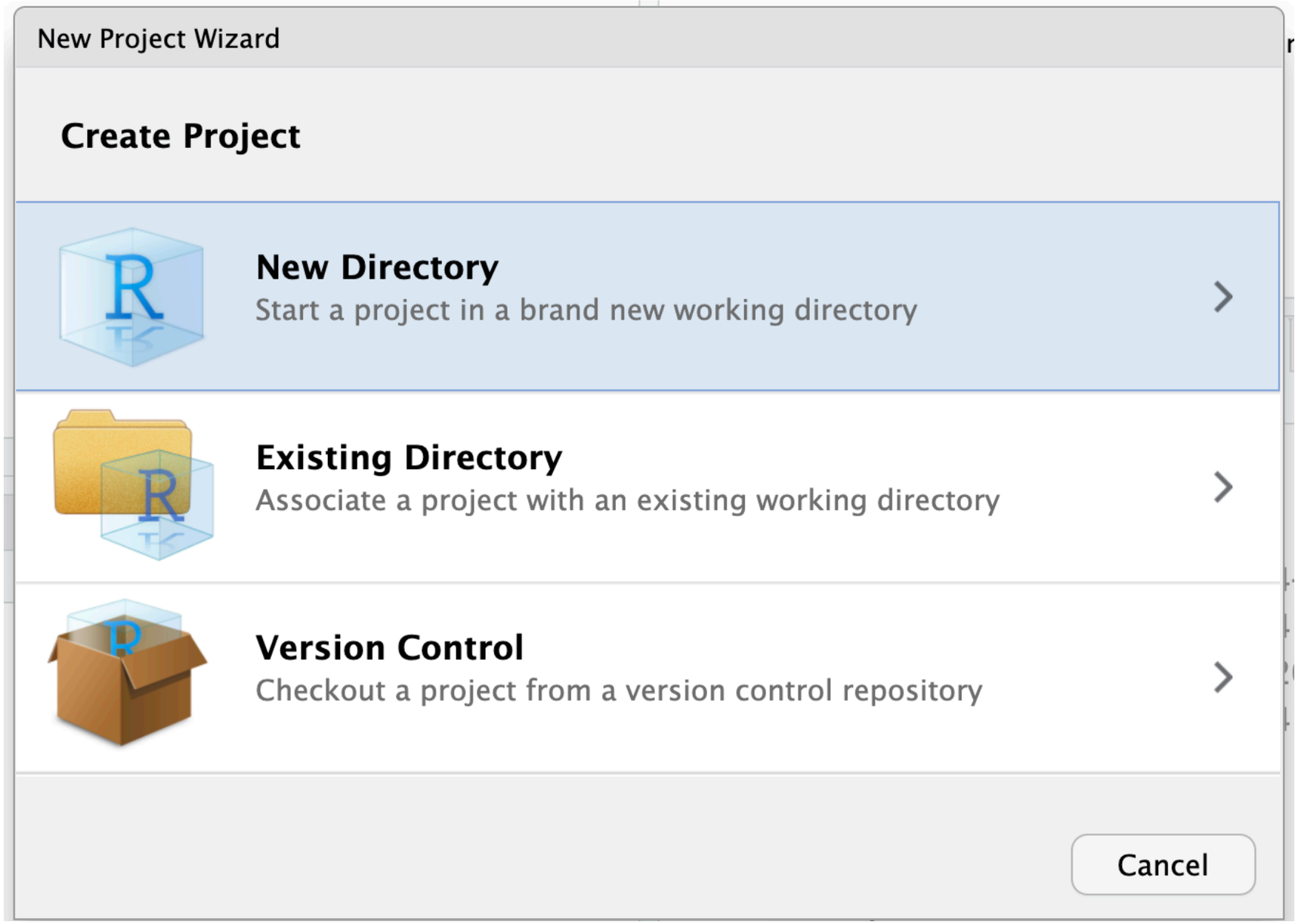
- *Master the Latest Tools*
  "I like to be on the forefront of technology that I can use to better serve Iowans. Whether by fax, satellite, Instagram, or Twitter, I'd use whatever platform works. In the digital age, tweets are an instant, unfiltered way to get my work and my message [out]."

https://www.washingtonian.com/2019/03/18/4-twitter-tips-from-u-s-senator-chuck-grassley/

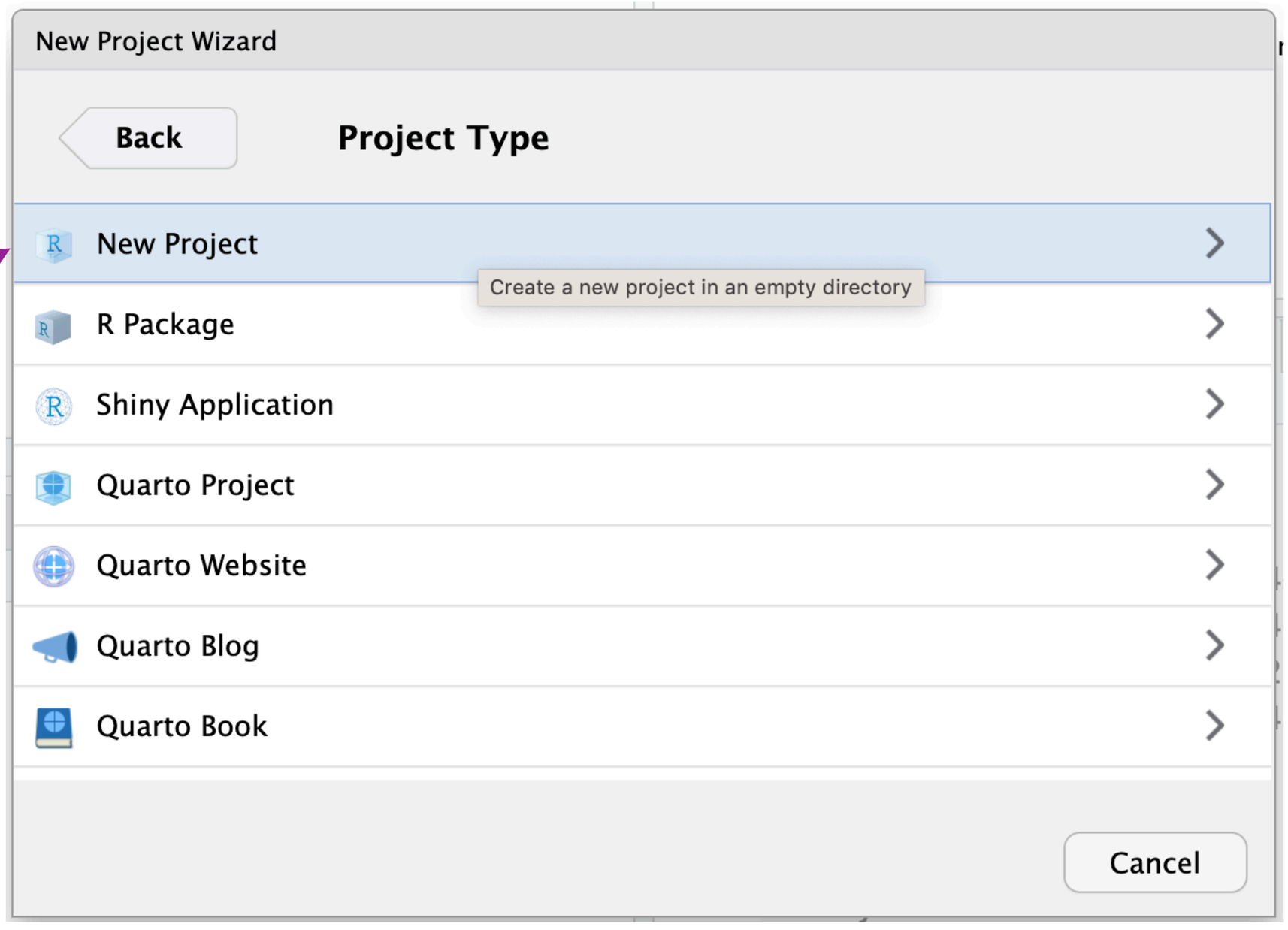
Create a project



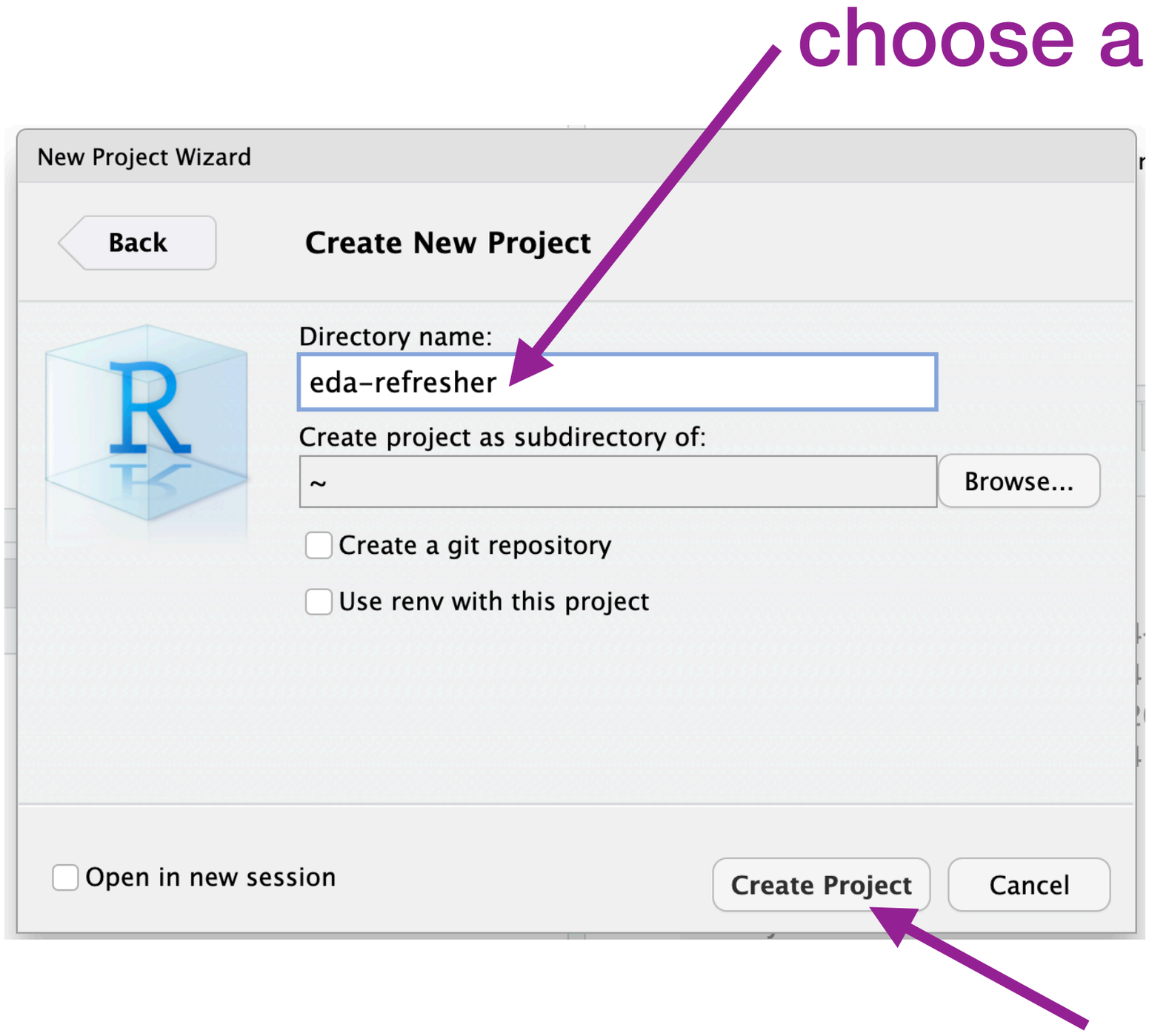
Create a project



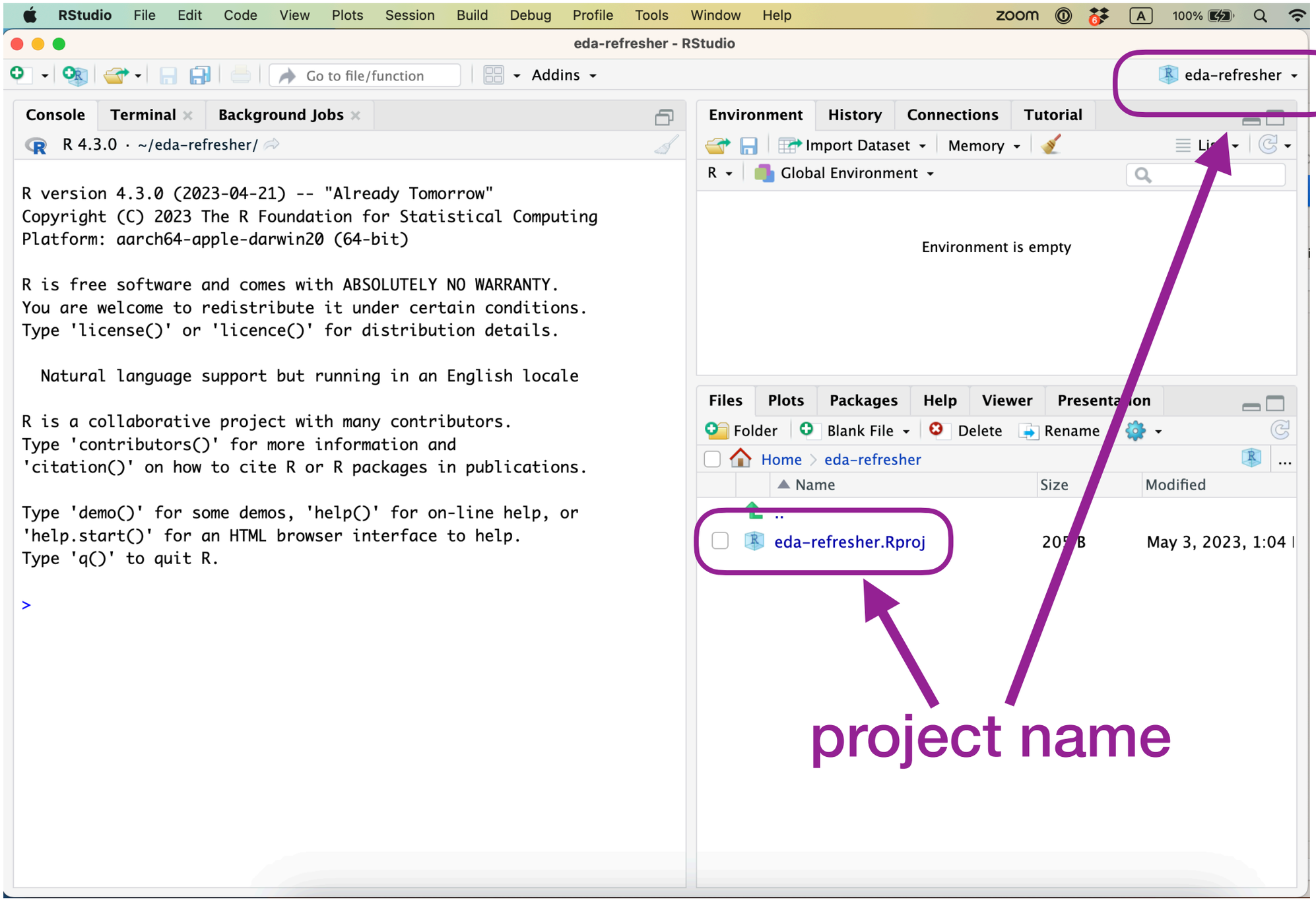
Create a project



Create a project



Create a project



Use Quarto or RMarkdown

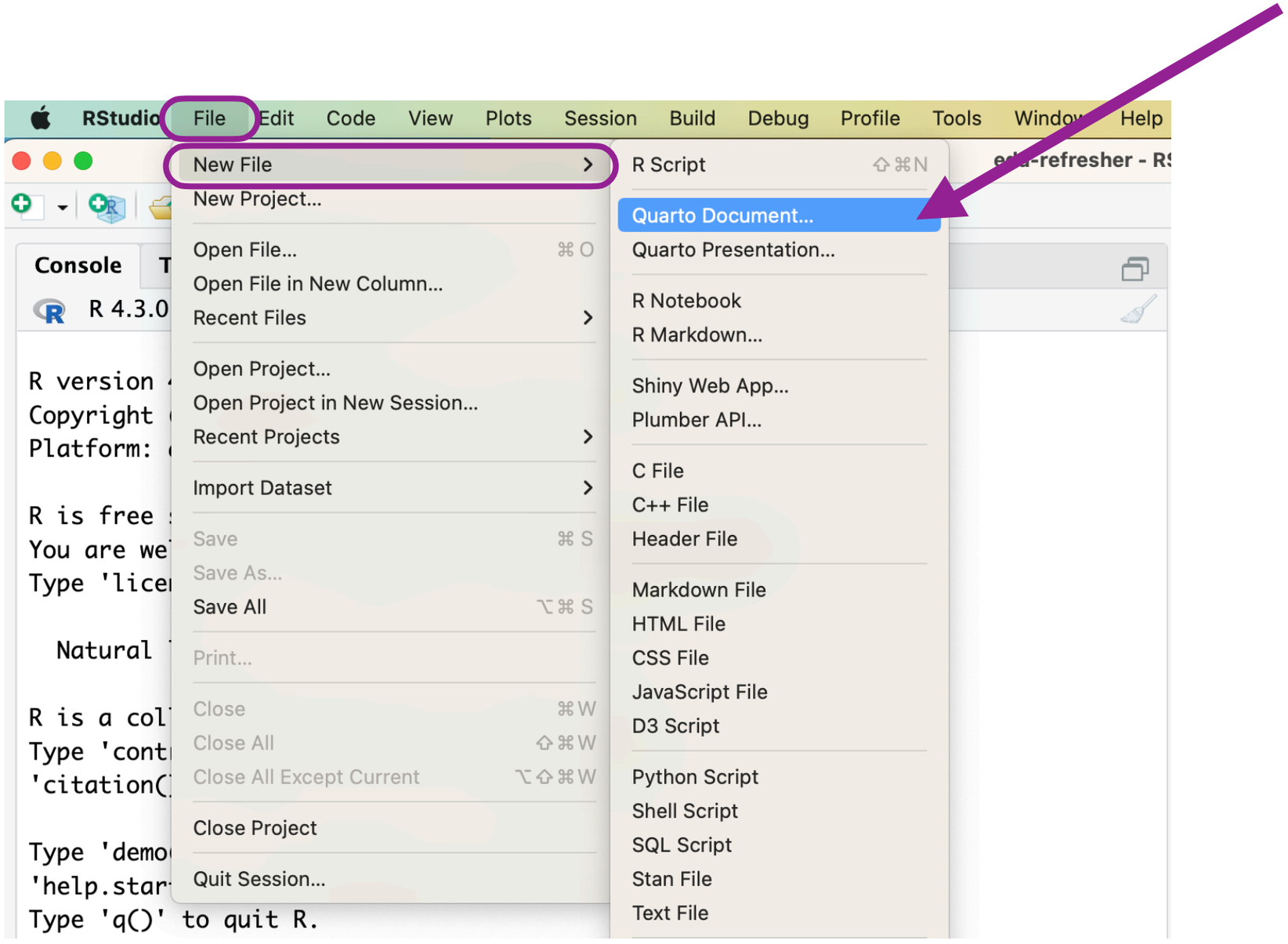
- Your workflow will be reproducible -- change the data file and you will have a revised output with one click

- No more copying and pasting text, code, output, and plots.

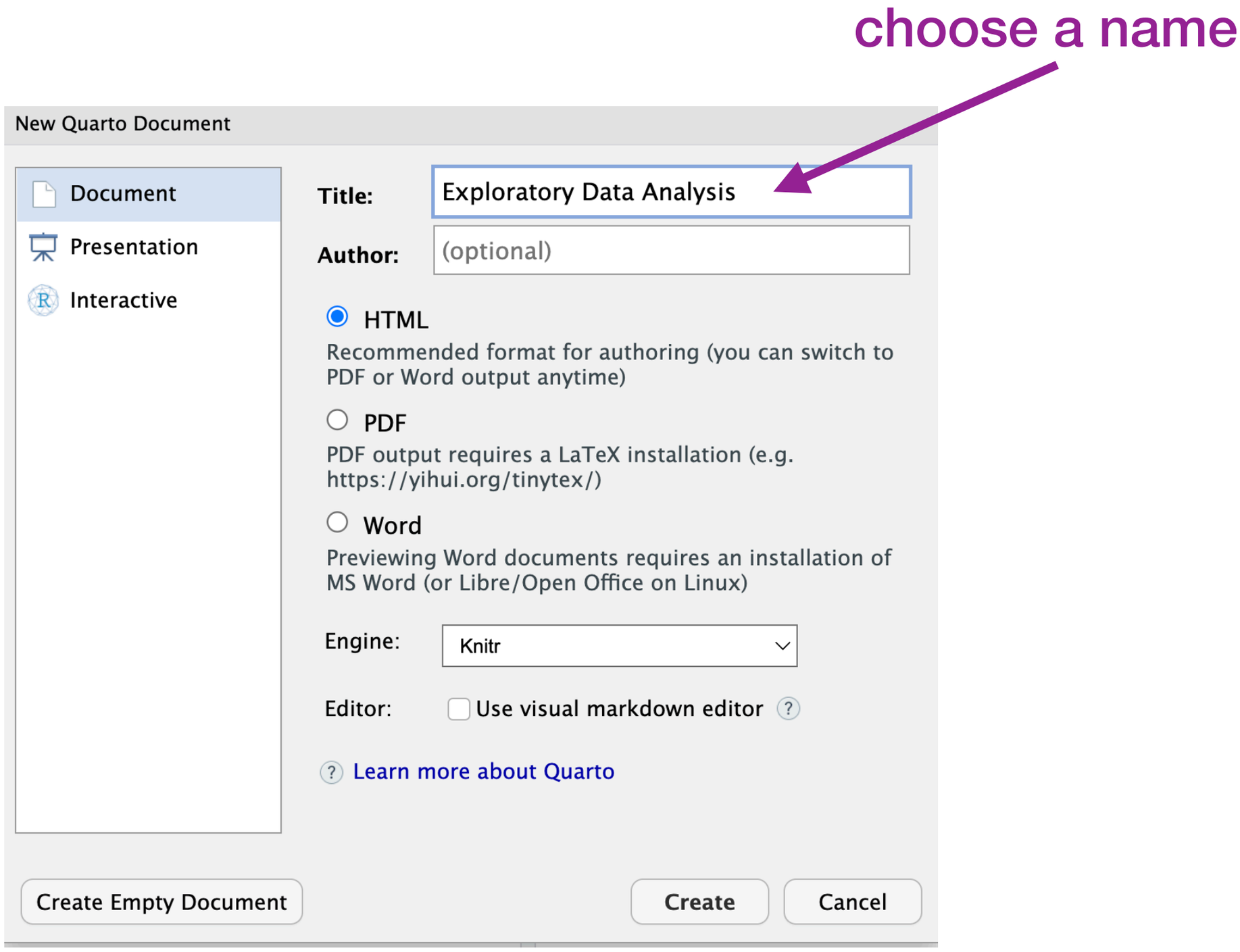
- You will have fewer mistakes and mismatches between stages of analysis.

- It will be easy to share your work.

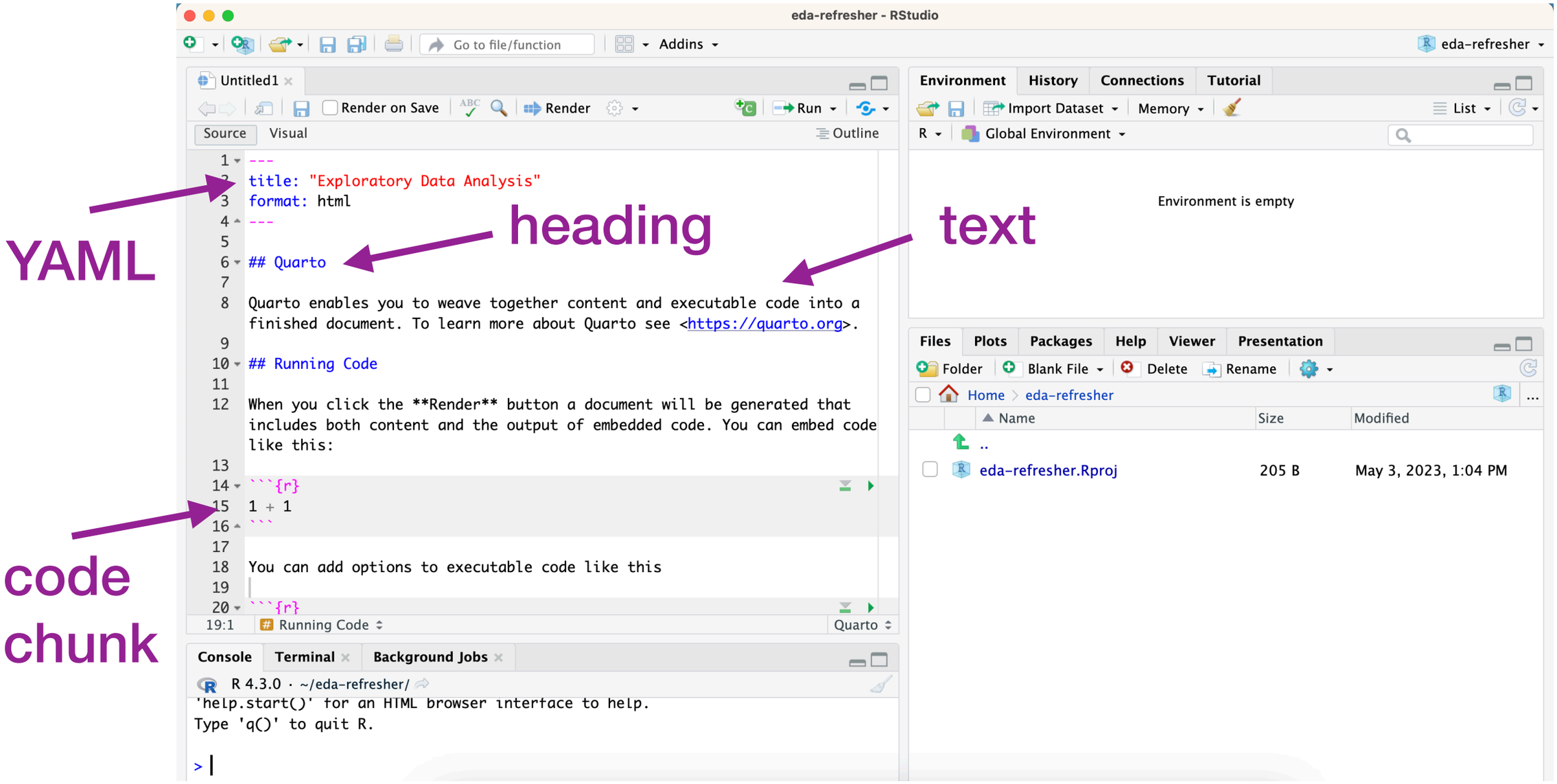
Create a Quarto document



Create a Quarto document



Parts of a Quarto document



Base R vs. tidyverse

- Not either / or
- Statistical functions and packages are the same
- Tidyverse fills in gaps to make R more versatile
- I find tidyverse more intuitive and convenient
- At least know what it's about and make an informed decision

Tidyverse

tidyverse.org

Tidyverse

Packages Blog Learn Help Contribute

R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

What is the philosophy?

- *(partial, unofficial list)*
- consistency
- human readability
  - `grepl() --> str_detect()`
  - `gsub() --> str_replace_all()`
- fail faster
- focus on data transformation

Data transformation

```
# base R
crime.by.state <- read.csv("CrimeStatebyState.csv")
crime.ny.2005 <- crime.by.state[crime.by.state$Year==2005 &
                                  crime.by.state$State=="New York",
                                c("Type.of.Crime", "Count")]
crime.ny.2005 <- crime.ny.2005[order(crime.ny.2005$Count,
                                     decreasing=TRUE), ]
crime.ny.2005$Proportion <- crime.ny.2005$Count /
                            sum(crime.ny.2005$Count)
summary1 <- aggregate(Count ~ Type.of.Crime,
                      data=crime.ny.2005, FUN=sum)
summary2 <- aggregate(Count ~ Type.of.Crime,
                      data=crime.ny.2005, FUN=length)
final <- merge(summary1, summary2, by="Type.of.Crime")
```

https://www.r-bloggers.com/2014/02/how-dplyr-replaced-my-most-common-r-idioms/

Data transformation

dplyr package

```
# dplyr
crime.by.state <- read.csv("CrimeStatebyState.csv")
final <- crime.by.state |>
filter(State=="New York", Year==2005) |>
arrange(desc(Count)) |>
select(Type.of.Crime, Count) |>
mutate(Proportion=Count/sum(Count)) |>
group_by(Type.of.Crime) |>
summarise(num.types = n(), counts = sum(Count))
```

base R pipe (R 4.1)

https://www.r-bloggers.com/2014/02/how-dplyr-replaced-my-most-common-r-idioms/

ggplot2

- underlying logic (gg = grammar of graphics)
- faceting
- automatic legends
- scales
- themes

Finding / importing data

Finding data

- Look for reputable sources
- Don't necessarily go with the first find
- Consider pros and cons of different options
  - R API package
  - Read from URL (or download file(s))
  - Web interface, then download
  - Web scraping

R API package

computerworld.com/article/3109890/these-r-packages-import-sports-weather-stock-data-and-more.html

R packages to import public data

PACKAGE	CATEGORY	DESCRIPTION	SAMPLE CODE	MORE INFO
blscrapeR	Economics, Government	For specific information about U.S. salaries and employment info, the Bureau of Labor Statistics offers a wealth of data available via this new package. blsAPI package is another option. CRAN.	bls_api(c("LEU0254530800", "LEU0254530600"), startyear = 2000, endyear = 2015)	Package vignettes
quantmod	Finance, Government	This package is designed for financial modelling but also has functions to easily pull data from Google Finance, Yahoo Finance and the St. Louis Federal Reserve (FRED). CRAN.	getSymbols("DEXJPUS",src="FRED")	Intro on getting data
Bureau of Economic Analysis	Economics, Government	Maintained by Andera Batch at BEA, this taps into the bureau's API to download data sets. CRAN.	beaSpecs <- list('UserID' = beaKey , 'Method' = 'GetData', 'datasetname' = 'NIPA', 'TableName' = 'T20305', 'Frequency' = 'Q', 'Year' = 'X', 'ResultFormat' = 'json'); beaPayload <- beaGet(beaSpecs);	See the GitHub repo, including info about recent project changes
edgarWebR	Finance, Government	This package is designed to let you search and download data from the U.S. Securities and Exchange Commission, including corporate and mutual-fund financial filings. CRAN.	getSymbols("DEXJPUS",src="FRED")	See the package vignette

https://www.computerworld.com/article/3109890/these-r-packages-import-sports-weather-stock-data-and-more.html

R API package: quantmod

```
library(quantmod)
getSymbols("AAPL")
#> [1] "AAPL"
tail(AAPL)
#>            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
#> 2023-04-25    165.19    166.31   163.73     163.77    48714100        163.77
#> 2023-04-26    163.06    165.28   162.80     163.76    45498800        163.76
#> 2023-04-27    165.19    168.56   165.19     168.41    64902300        168.41
#> 2023-04-28    168.49    169.85   167.88     169.68    55209200        169.68
#> 2023-05-01    169.28    170.45   168.64     169.59    52472900        169.59
#> 2023-05-02    170.09    170.35   167.54     168.54    48329100        168.54
```

```
plot(AAPL$AAPL.Close)
```

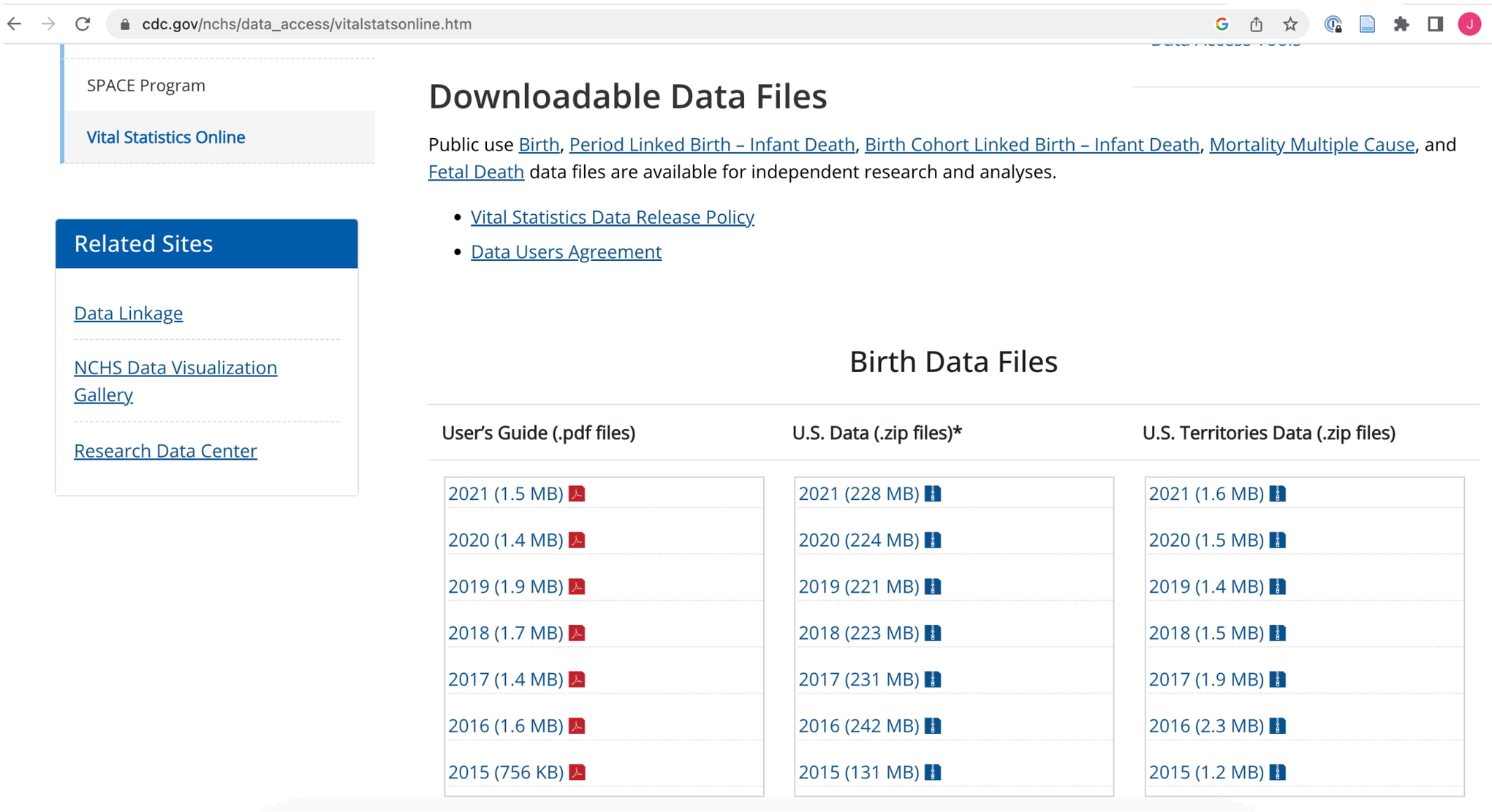
R API Package

- Particularly useful if data has frequent updates
- For some analyses you may want to set an end date so results don't change every time you run the code
- Add "R package" to internet searches for data to find these packages
- Many require API keys, but not hard to obtain
- Check for data transformations, etc.

Read from URL or download file(s)

- Very common, convenient way to get data
- Often data is available in multiple formats
- Content is not always the same
- Consider reading directly rather than downloading first, at least for small files

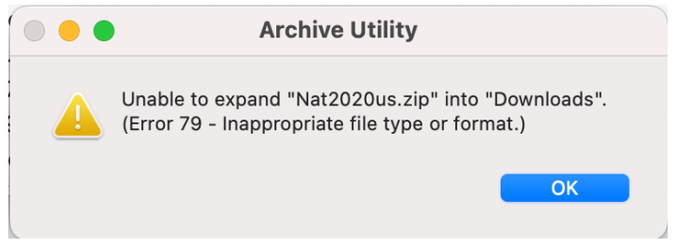
CDC Natality Database



https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm

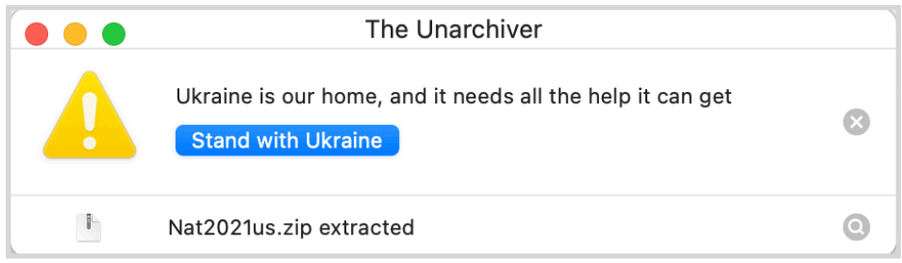
CDC Natality Database ZIP files

- 2021 ZIP file (228MB) downloaded in less than a minute



- Couldn't unzip with default utility

- Worked with another app, The Unarchiver



- Unzipped .txt file is almost 5GB, but could read the file in less than a minute

https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm

CDC Natality Database .txt files

BUT .txt file is fixed width, need to specify column widths:

Position	Length	Field	Description	Values	Definition
1-8	6	FILLER	**Filler**	Blank	
9-12	4	DOB_YY	**Birth Year**	2021	Year of birth
13-14	2	DOB_MM	**Birth Month**	01	January
				02	February
				03	March
				04	April
				05	May
				06	June
				07	July
				08	August
				09	September
				10	October
				11	November
				12	December

https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/DVS/natality/UserGuide2021.pdf

CDC Natality Database .txt files

to:

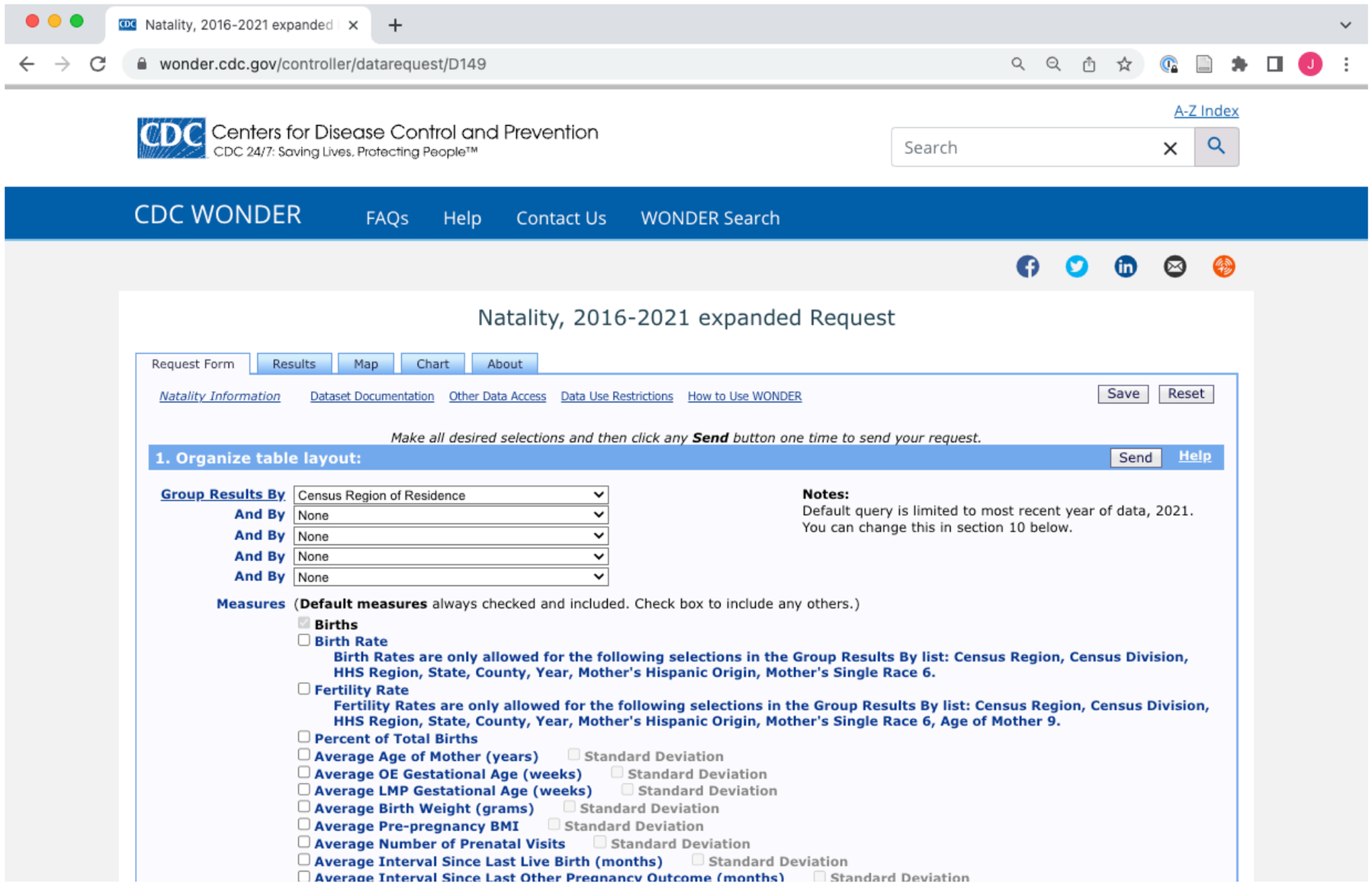
Position	Length	Field	Description	Values	Definition
				U	Unknown or not stated
569	1	BFED	**Infant Breastfed at Discharge**	Y N U	Yes No Unknown or not stated
570	1	F_BFED	**Reporting Flag for Breastfed at Discharge**	0 1	Non-Reporting Reporting
571-1330	760	FILLER_X	**Filler**	Blank	

https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/DVS/natality/UserGuide2021.pdf

CDC Natality Database .txt files

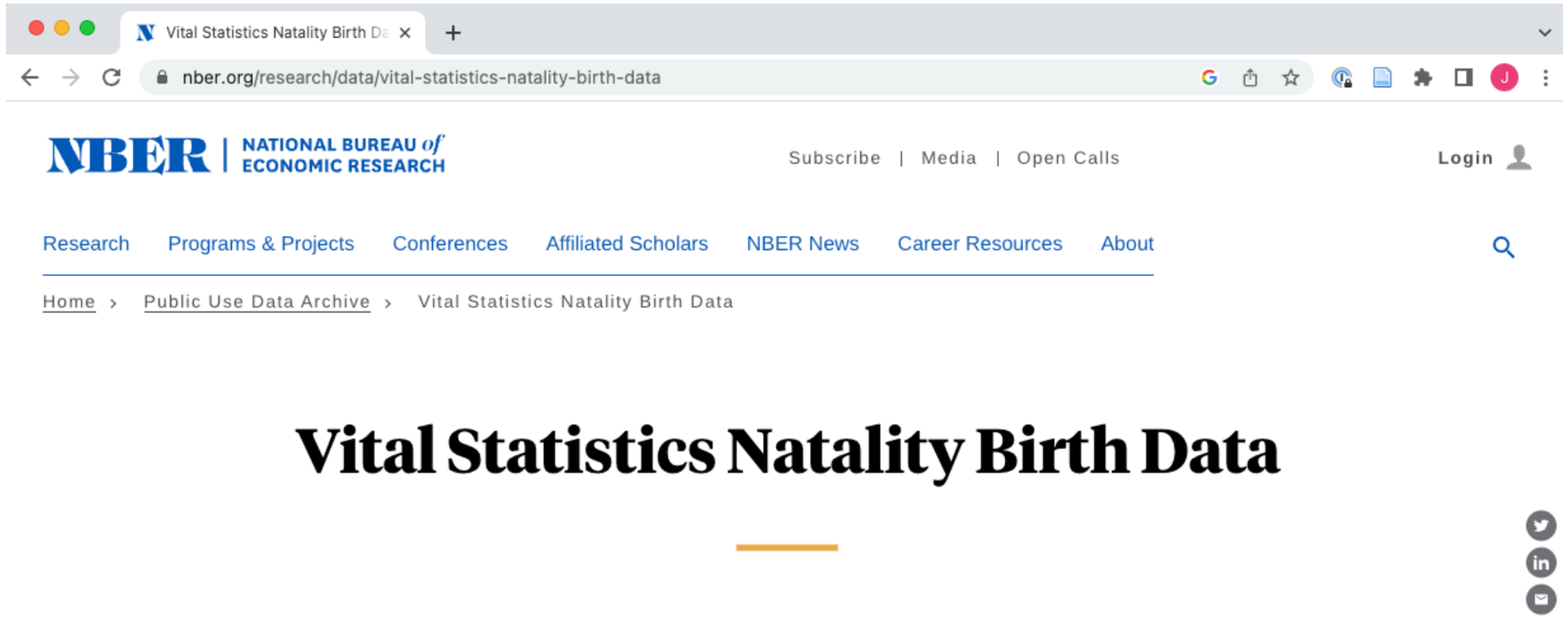
```
library(readr)
read_fwf("~/Downloads/Nat2021us.txt",
                 col_positions = fwf_widths(c(8, 4, 2, 4, 4, 1)),
                 n_max = 8, show_col_types = FALSE)
#> # A tibble: 8 × 6
#>   X1        X2 X3    X4    X5        X6
#>   <lgl> <dbl> <chr> <lgl> <chr> <dbl>
#> 1 NA     2021 01    NA    0636      7
#> 2 NA     2021 01    NA    0259      7
#> 3 NA     2021 01    NA    0223      1
#> 4 NA     2021 01    NA    0241      1
#> 5 NA     2021 01    NA    0503      1
#> 6 NA     2021 01    NA    2341      7
#> 7 NA     2021 01    NA    1800      7
#> 8 NA     2021 01    NA    0652      1
```

CDC Wonder web interface



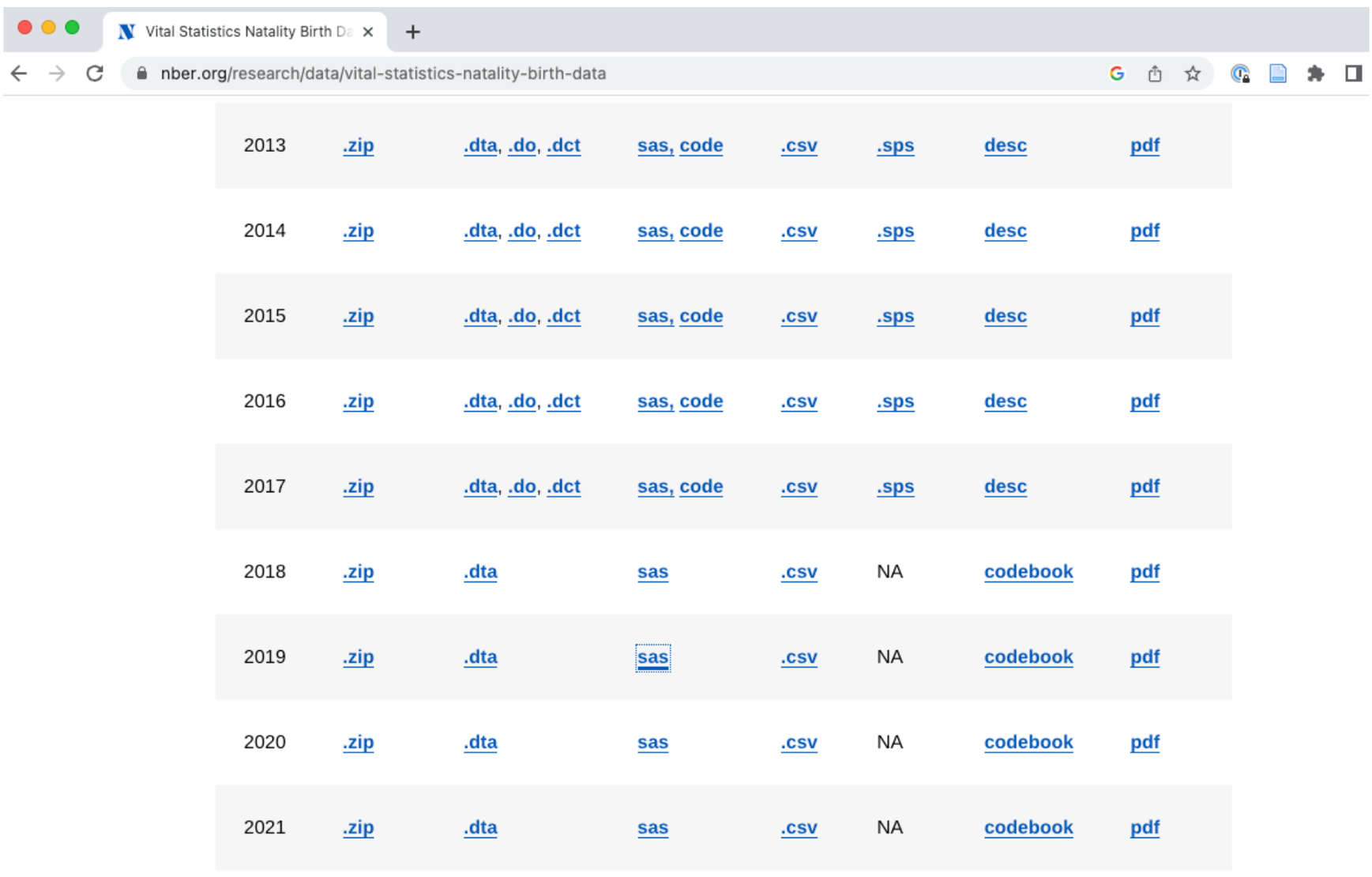
https://wonder.cdc.gov/controller/datarequest/D149

NBER Natality Birth Data



https://www.nber.org/research/data/vital-statistics-natality-birth-data

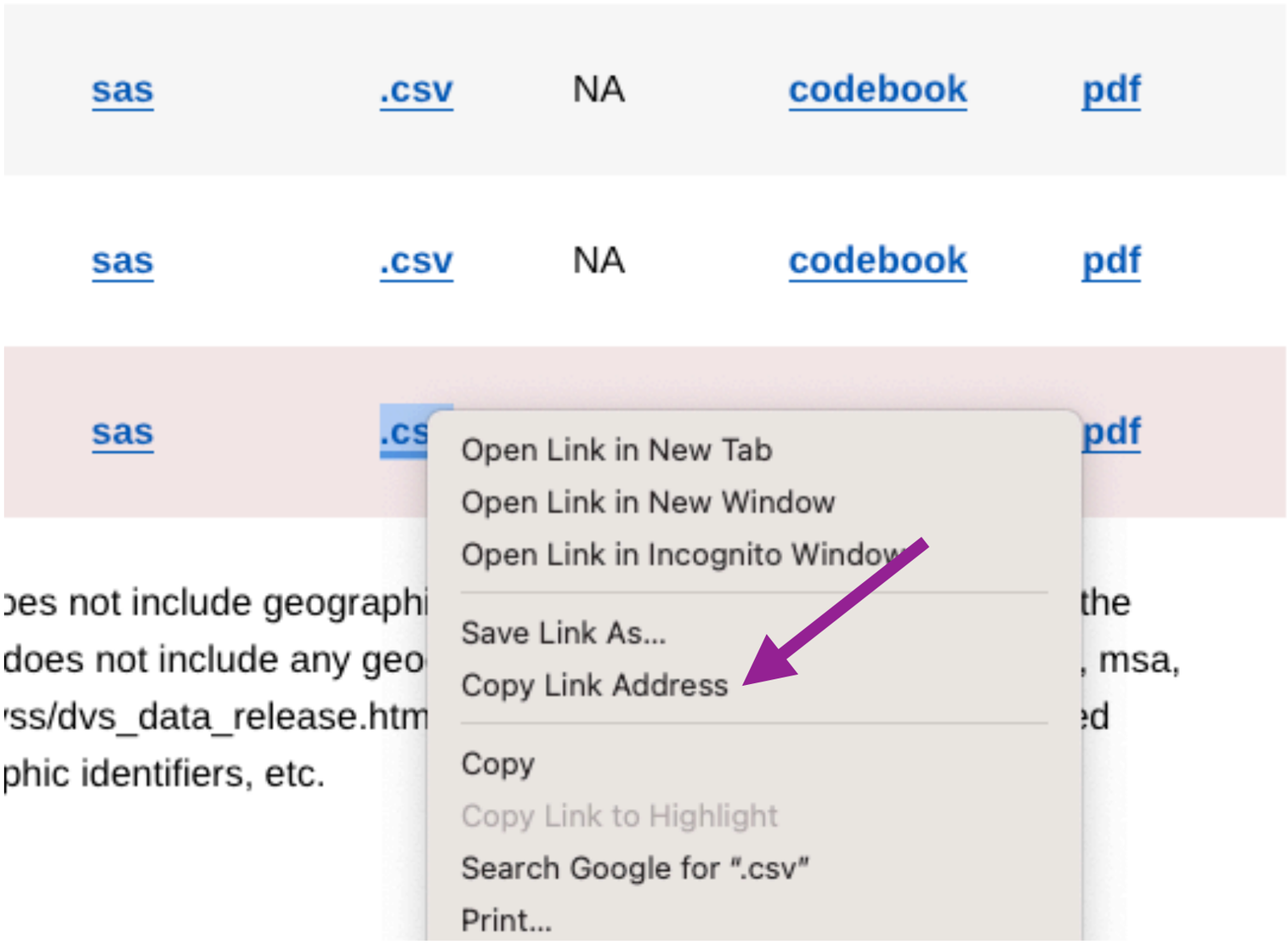
NBER Natality Birth Data



2013	.zip	.dta, .do, .dct	sas, code	.csv	.sps	desc	pdf
2014	.zip	.dta, .do, .dct	sas, code	.csv	.sps	desc	pdf
2015	.zip	.dta, .do, .dct	sas, code	.csv	.sps	desc	pdf
2016	.zip	.dta, .do, .dct	sas, code	.csv	.sps	desc	pdf
2017	.zip	.dta, .do, .dct	sas, code	.csv	.sps	desc	pdf
2018	.zip	.dta	sas	.csv	NA	codebook	pdf
2019	.zip	.dta	sas	.csv	NA	codebook	pdf
2020	.zip	.dta	sas	.csv	NA	codebook	pdf
2021	.zip	.dta	sas	.csv	NA	codebook	pdf

https://www.nber.org/research/data/vital-statistics-natality-birth-data

NBER Natality Birth Data



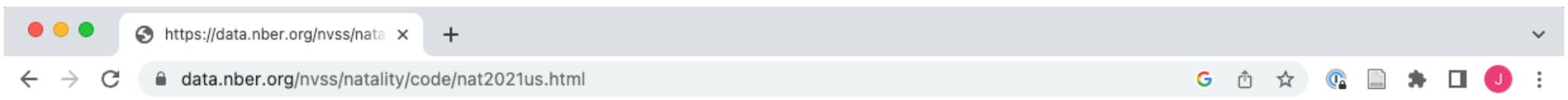
https://www.nber.org/research/data/vital-statistics-natality-birth-data

NBER Natality Birth Data

```
df <- read.csv("https://data.nber.org/nvss/natality/csv/
nat2021us.csv", nrows = 10)
df[,1:7]
#>    dob_yy dob_mm dob_tt dob_wk bfacil f_facility bfacil3
#> 1    2021      1    636      7       1          1       1
#> 2    2021      1    259      7       1          1       1
#> 3    2021      1    223      1       1          1       1
#> 4    2021      1    241      1       1          1       1
#> 5    2021      1    503      1       1          1       1
#> 6    2021      1   2341      7       1          1       1
#> 7    2021      1   1800      7       1          1       1
#> 8    2021      1    652      1       1          1       1
#> 9    2021      1    227      6       1          1       1
#> 10   2021      1   2151      6       1          1       1
```

https://www.nber.org/research/data/vital-statistics-natality-birth-data

Web scraping



Number of observations: 3,669,928

dob_yy		Num	Birth Year 2015 Year of birth
Frequency	Percent	Value	Label
3669928	100%	2021	

dob_mm		Num	Birth Month 01 January NOTE: Smallest 5 and largest 5 values are displayed.
Frequency	Percent	Value	Label
277533	8%	1	
266725	7%	2	
303139	8%	3	
293630	8%	4	
301343	8%	5	
640635	17%	6-7	NOTE: Range of values omitted from display
330740	9%	8	
326280	9%	9	
315909	9%	10	
302309	8%	11	
311685	8%	12	

https://data.nber.org/nvss/natality/code/nat2021us.html

Web scraping with `rvest`

```
library(rvest)
df <- read_html("https://data.nber.org/nvss/natality/code/
nat2021us.html") |>
  html_table()
df[[5]]
#> # A tibble: 9 × 4
#>   X1        X2        X3    X4
#>   <chr>     <chr>     <chr> <chr>
#> 1 dob_wk    ""        Num   "Birth Day of Week 1 Sunday"
#> 2 Frequency "Percent" Value "Label"
#> 3 355442    "10%"     1     ""
#> 4 548910    "15%"     2     ""
#> 5 593276    "16%"     3     ""
#> 6 593664    "16%"     4     ""
#> 7 591182    "16%"     5     ""
#> 8 579740    "16%"     6     ""
#> 9 407714    "11%"     7     ""
```

https://data.nber.org/nvss/natality/code/nat2021us.html

The winner is...

- Download .csv file from:

https://data.nber.org/nvss/natality/csv/nat2021us.csv

- Read once and save small versions:
  - One with 4 of 225 variables (all rows)
  - One with a sample of 1000 rows