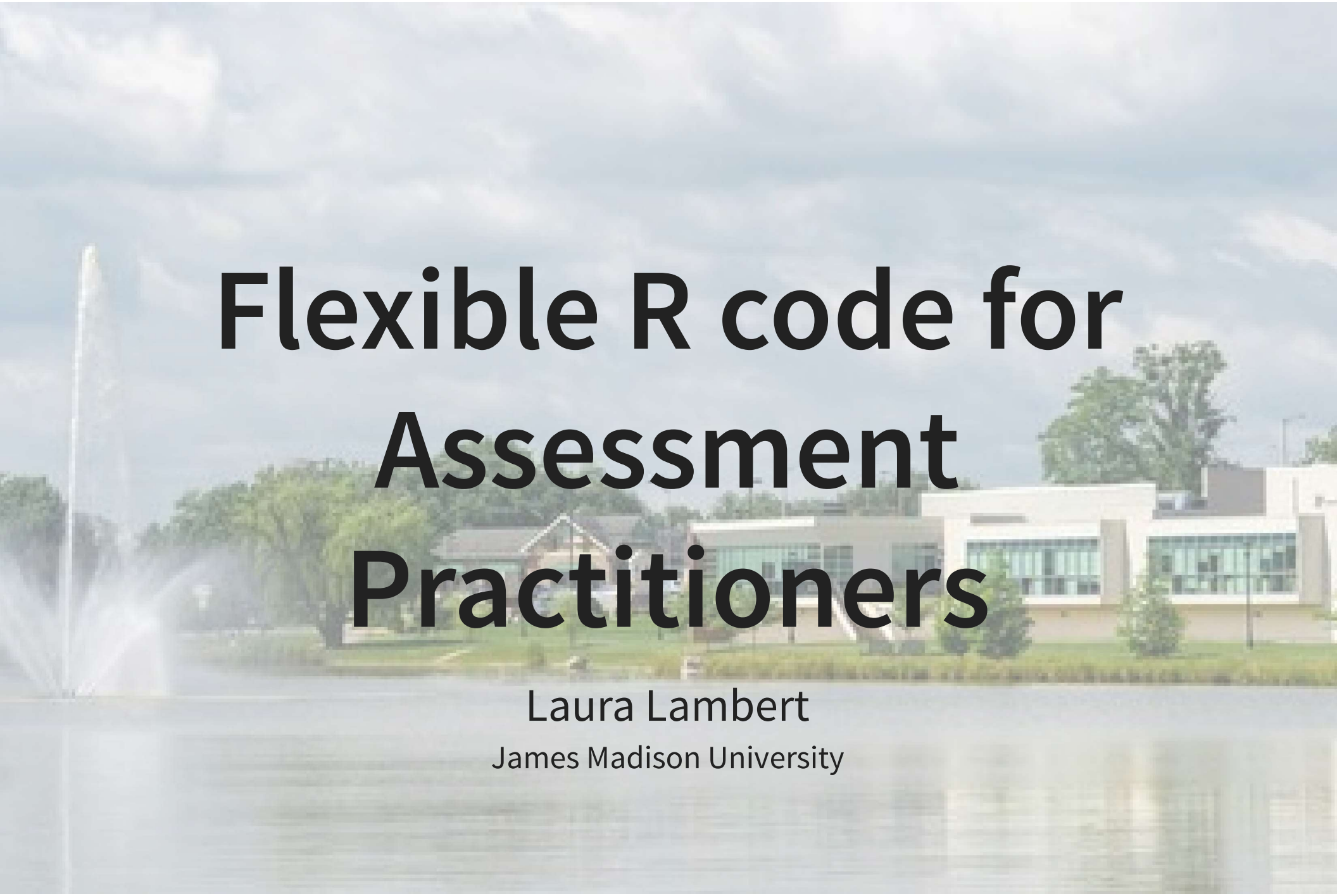


Flexible R code for Assessment Practitioners

Laura Lambert

James Madison University

Overview

- Who are you?
- What will we be doing for the next hour?
- Will I become an R expert?
- Is this going to be scary?

Overview

Who am

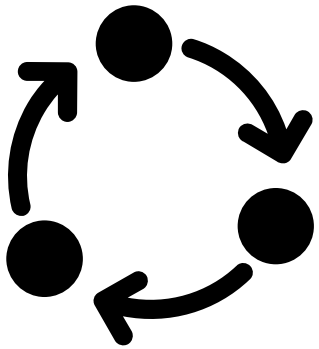
Who I am not

- An Expert Coder
- A computer science major
- Someone who has received formal training in how to ‘properly’ code

Who I am

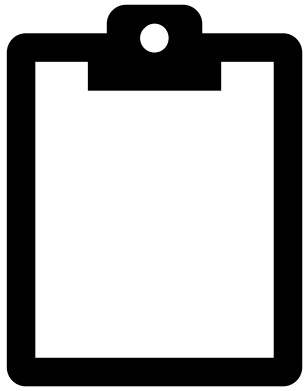
- Doctoral candidate in James Madison University's Assessment & Measurement program
- Full-time A&P faculty in Integrated Science and Technology (also at JMU)
- Assessment (and measurement) enthusiast
- Really good at crafting Google search terms
- Someone who enjoys using available tools to make my life easier!

Overview



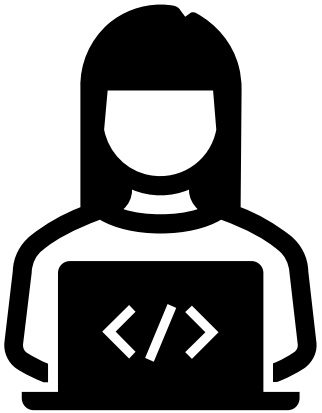
Contextualize this in the Assessment Cycle

Overview



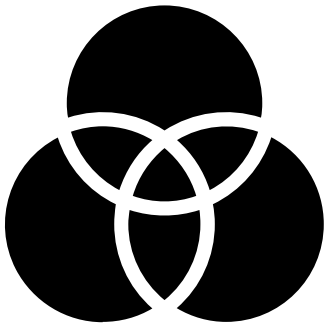
Gather some data from you

Overview



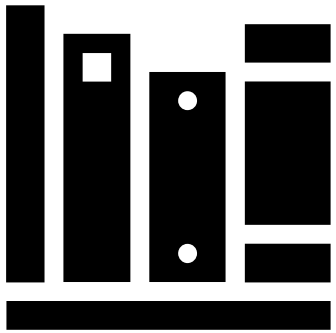
Overview of R and R Studio


Overview

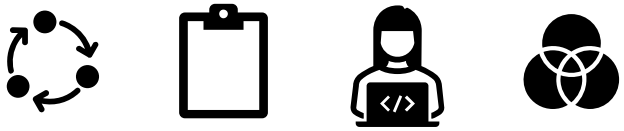


Comparison of R to other programs

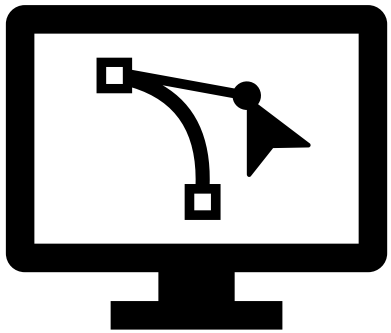
Overview



Share some resources



Overview



Live walk-through

Overview

Time for questions

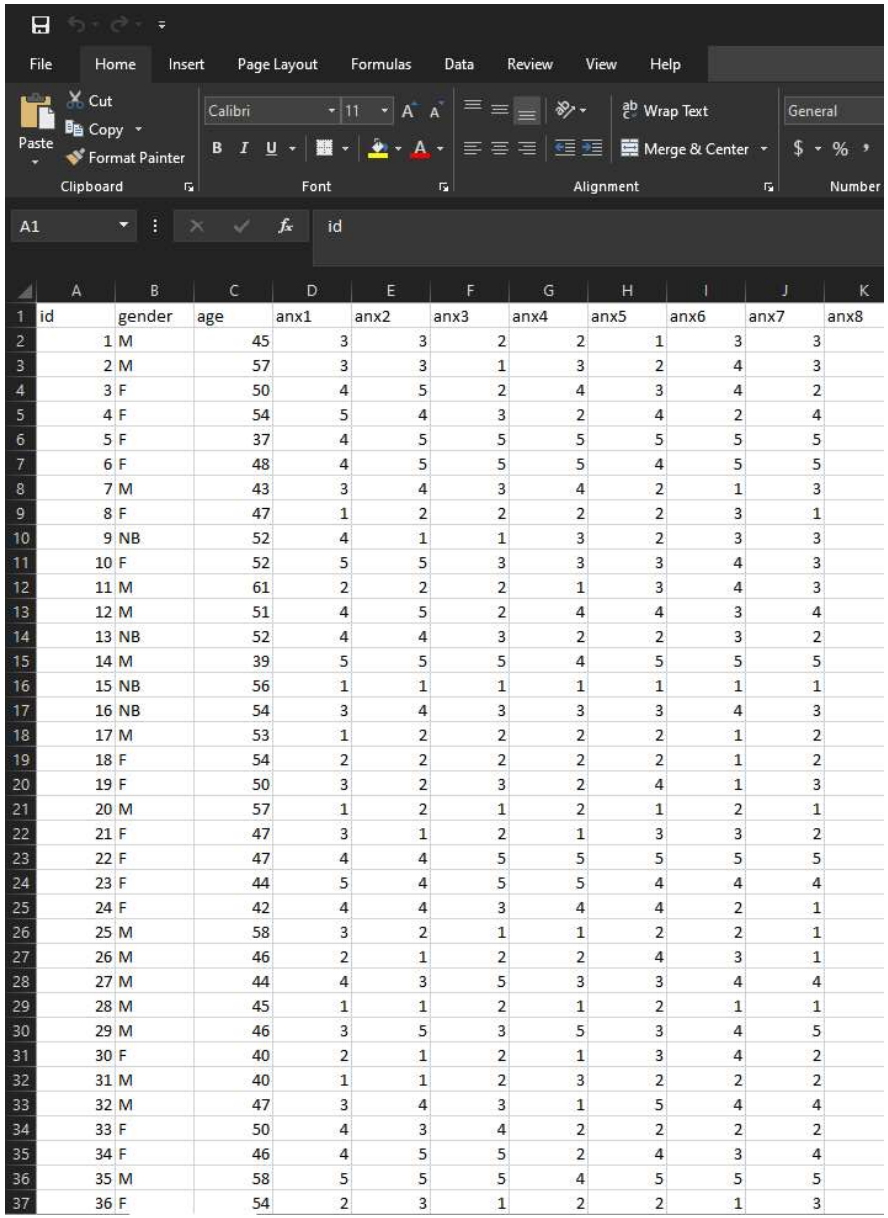
Overview

Let’s get started!

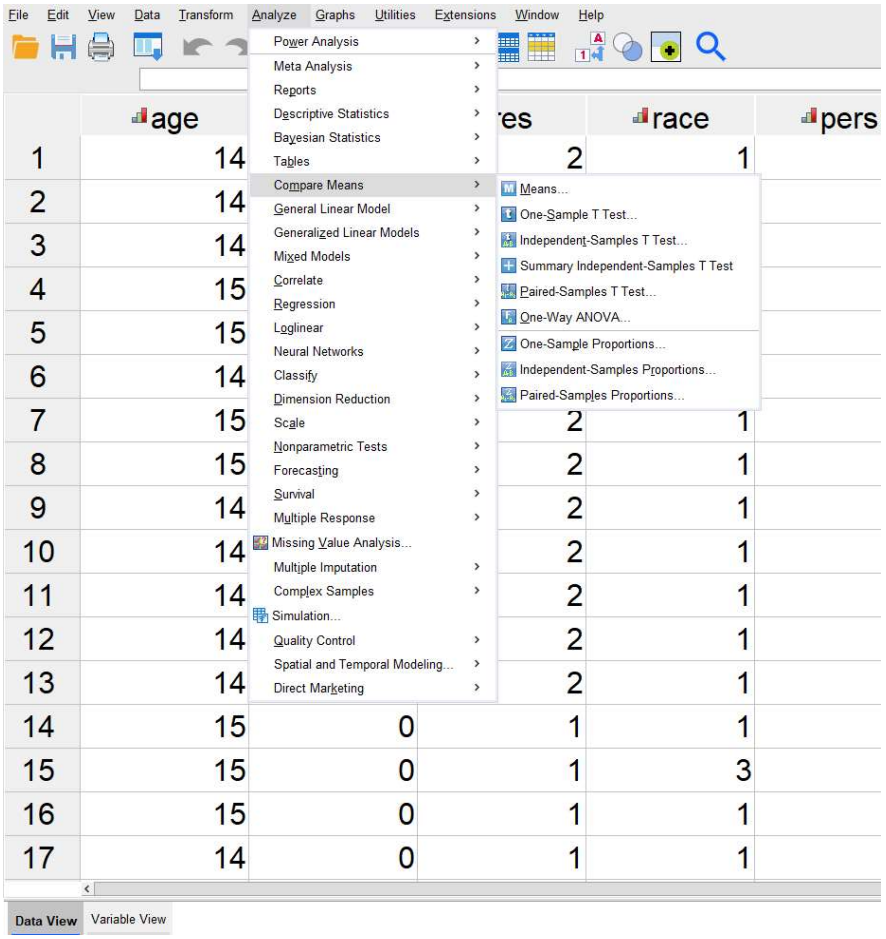
Assessment Cycle

Analyzing Data, Reporting Results, and Maintaining Information

Assessment Cycle: Analyze Data



	A	B	C	D	E	F	G	H	I	J	K
1	id	gender	age	anx1	anx2	anx3	anx4	anx5	anx6	anx7	anx8
2	1	M	45	3	3	2	2	1	3	3	
3	2	M	57	3	3	1	3	2	4	3	
4	3	F	50	4	5	2	4	3	4	2	
5	4	F	54	5	4	3	2	4	2	4	
6	5	F	37	4	5	5	5	5	5	5	
7	6	F	48	4	5	5	5	4	5	5	
8	7	M	43	3	4	3	4	2	1	3	
9	8	F	47	1	2	2	2	2	3	1	
10	9	NB	52	4	1	1	3	2	3	3	
11	10	F	52	5	5	3	3	3	4	3	
12	11	M	61	2	2	2	1	3	4	3	
13	12	M	51	4	5	2	4	4	3	4	
14	13	NB	52	4	4	3	2	2	3	2	
15	14	M	39	5	5	5	4	5	5	5	
16	15	NB	56	1	1	1	1	1	1	1	
17	16	NB	54	3	4	3	3	3	4	3	
18	17	M	53	1	2	2	2	2	1	2	
19	18	F	54	2	2	2	2	2	1	2	
20	19	F	50	3	2	3	2	4	1	3	
21	20	M	57	1	2	1	2	1	2	1	
22	21	F	47	3	1	2	1	3	3	2	
23	22	F	47	4	4	5	5	5	5	5	
24	23	F	44	5	4	5	5	4	4	4	
25	24	F	42	4	4	3	4	4	2	1	
26	25	M	58	3	2	1	1	2	2	1	
27	26	M	46	2	1	2	2	4	3	1	
28	27	M	44	4	3	5	3	3	4	4	
29	28	M	45	1	1	2	1	2	1	1	
30	29	M	46	3	5	3	5	3	4	5	
31	30	F	40	2	1	2	1	3	4	2	
32	31	M	40	1	1	2	3	2	2	2	
33	32	M	47	3	4	3	1	5	4	4	
34	33	F	50	4	3	4	2	2	2	2	
35	34	F	46	4	5	5	2	4	3	4	
36	35	M	58	5	5	5	4	5	5	5	
37	36	F	54	2	3	1	2	2	1	3	



First Survey Question

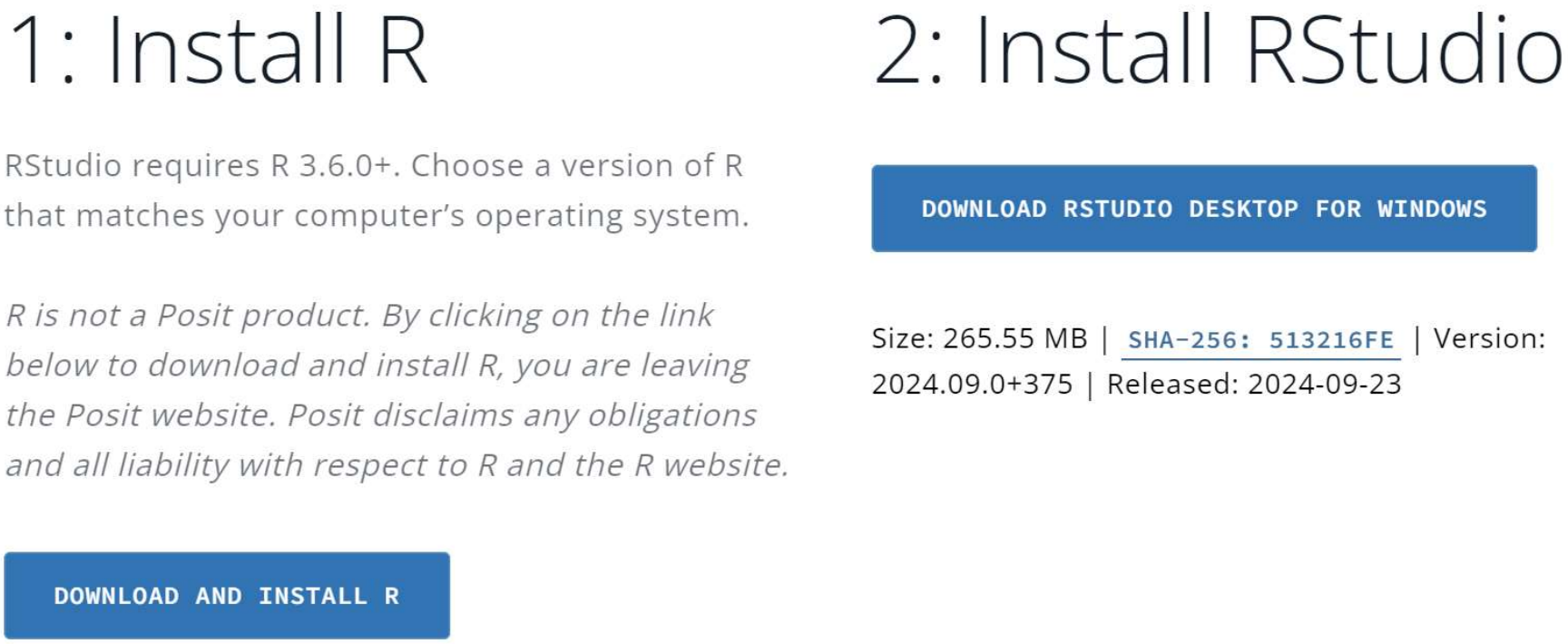
These are just for fun - we'll be illustrating some code later with this data

Introduction to R and R Studio

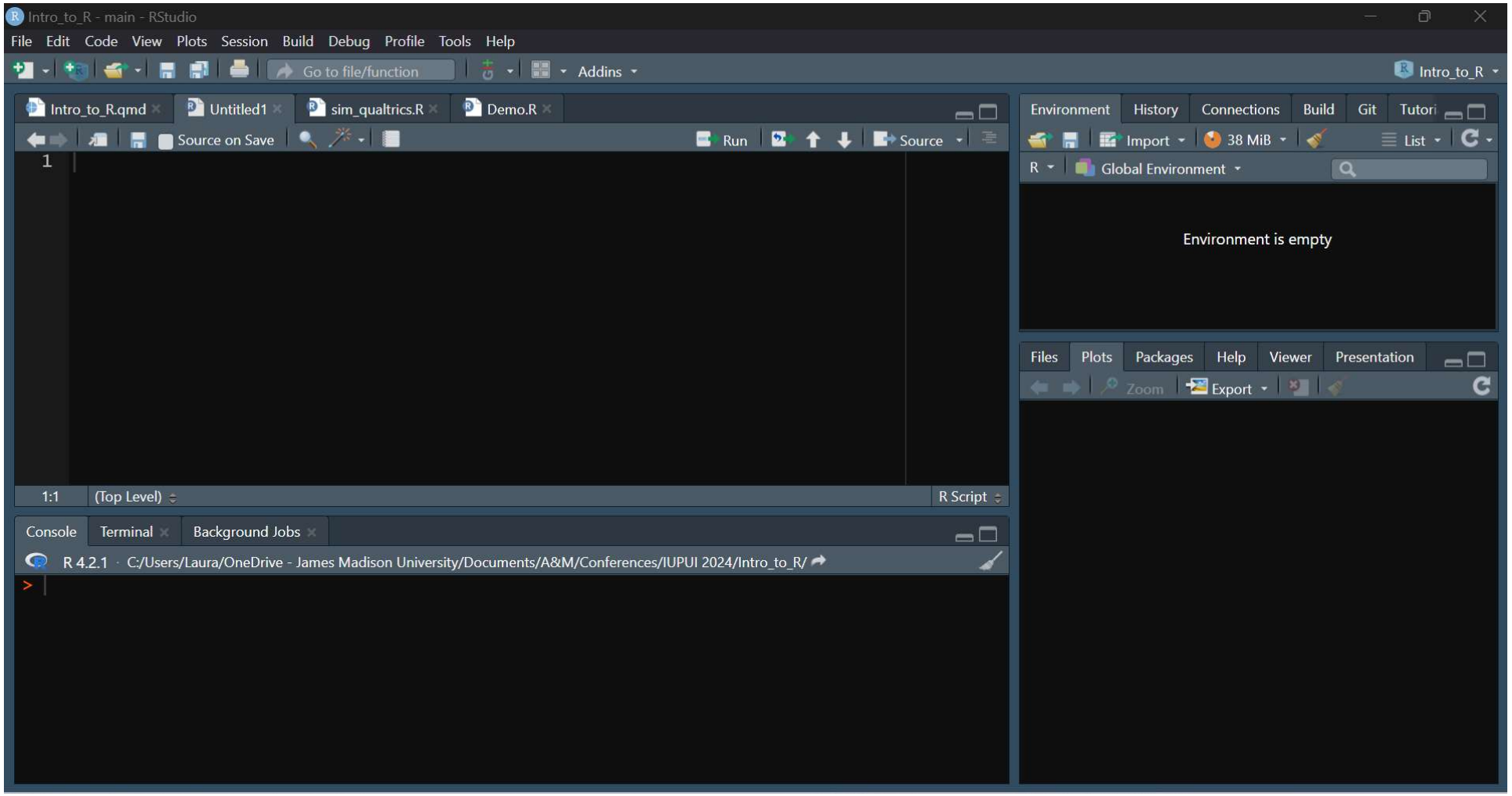
- R is a programming language
- R Studio is an IDE - a user-friendly interface
- Both R and R Studio are free and open source

Where to find them

https://posit.co/download/rstudio-desktop/



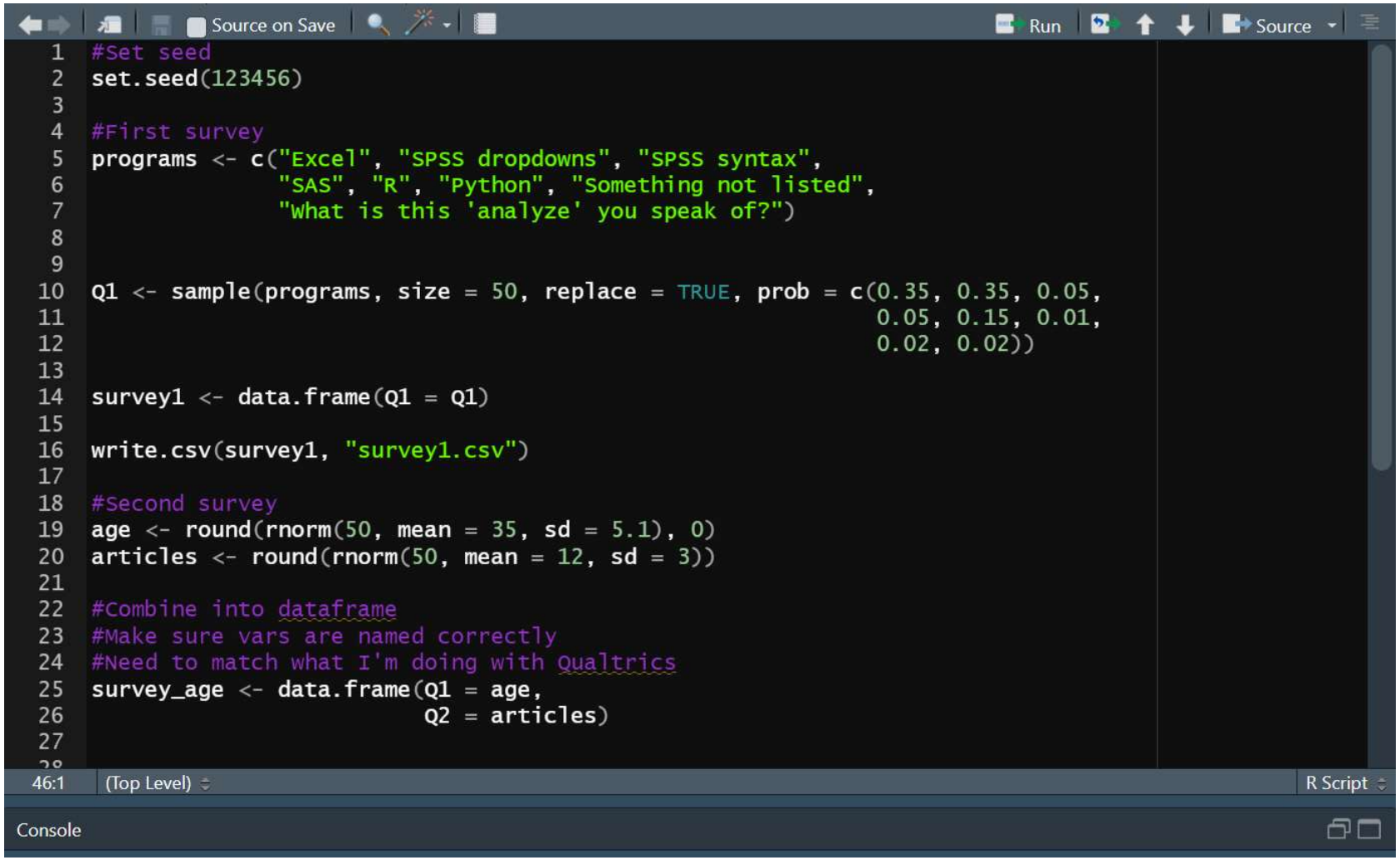
R Studio window



Script vs. Project

- A script file is a set of instructions

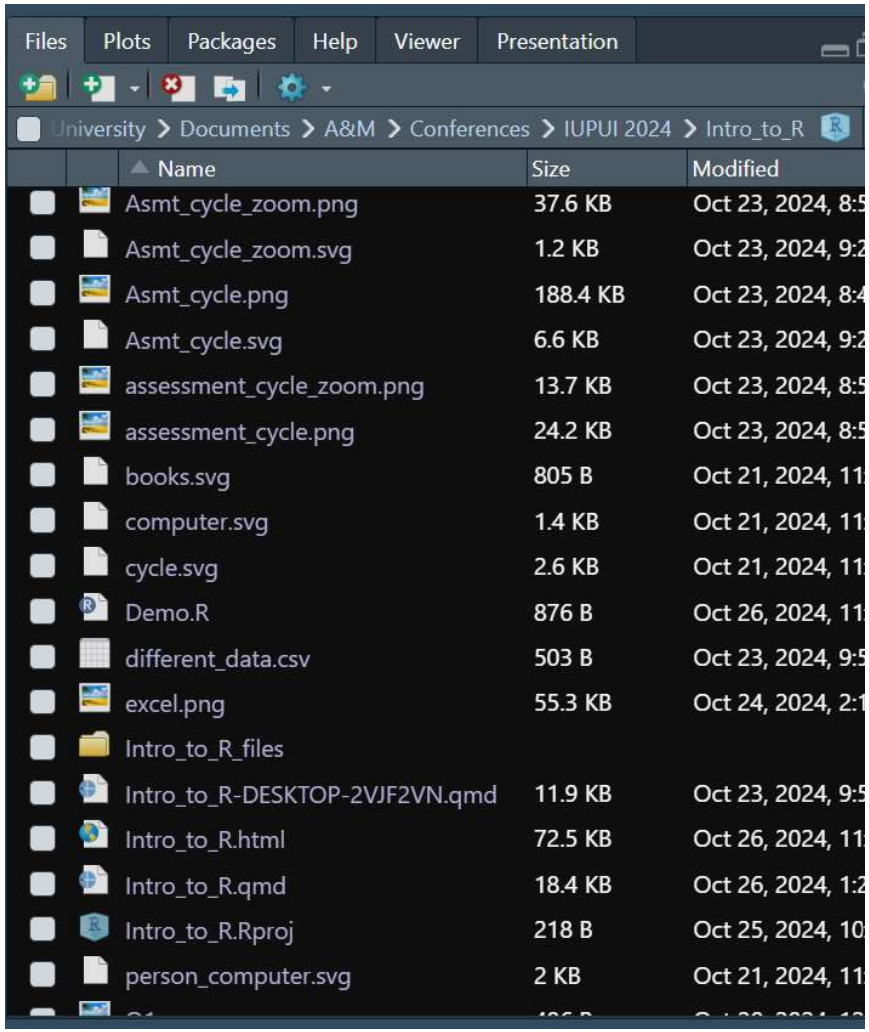
Script vs. Project



Script vs. Project

- A script file is a set of instructions
- A project is a directory (folder)

Script vs. Project



Script vs. Project

- A script file is a set of instructions
- A project is a directory (folder)
  - Script(s)
  - Data
  - Output
  - Anything, really

Script vs. Project

Main difference: calling files

Script file

```
df <- read.csv("path/to/data/here.csv")
```

Project

```
df <- read.csv("file.csv")
```

Base R and Packages

- 'base' R has quite a bit of functionality on its own
- 'packages' extend what base R can do
  - Specialized collections of functions (e.g. `psych`, `lavaan`)
  - Make data wrangling and visualization easier (e.g. `tidyverse`, `ggplot2`)
  - Advanced analysis (e.g. `lme4`, `brms`)

Installing and Loading Packages

```
#Install a package
#Only need to do this once on any given machine
#If you switch machines (e.g. office to home computer) you'll need to re-in
install.packages("ggplot2")

#Call the package
#You will need to call your packages for each R session
library(ggplot2)
```

Installing and Loading Packages

```
#Handy to have a 'set-up' area at the beginning of your code
#Load necessary packages
library(tidyverse)
library(psych)
library(car)

#Then proceed with the rest of your code....
```

Second Survey Question

Feel free to be as honest or dishonest as you'd like here…

Syntax

(Source: “Columbus, C. (1990). Home Alone. Twentieth Century

Syntax

- Like many things, an up-front investment of time can save a lot of time long-term
- MANY available resources for code
- Take and modify what you need

The Argument for Syntax

Let's imagine two scenarios where you're trying to describe your work:

Scenario 1 | Scenario 2

So, you have this Excel file and you want to make a formula in one of the cells to sum up the 'pre' and 'post' columns. Then, you need to reverse score one of the items, so you have to make that, then go ahead and delete the old column and put the new one in its place. Then, to do a t-test, you use this formula across these columns….

The Argument for Syntax

1. Documentatio
2. Reproducabil

Documentation

```
#Comments can tell you what you did
#Or, what dataset you read in

#Read in dataset A
data <- read.csv("groupA.csv")
```

Documentation

```r
#Read in data
test_data <- read.csv("some/file/path.csv")

#What does our data look like?
head(test_data)

#Take a peek at a variable
#Check if any missing data
table(test_data$sp_2024, useNA = "always")

#Two missing values
#Remove them from the data - can only have complete cases here
test_data_clean <- na.omit(test_data)
```

Documentation

But, what if you had a different criteria? What if you only removed them if some other variable was above/below a certain value?

```
#Look for missing data
table(test_data_v2$major, useNA = "always")

#Also see how these map together
table(test_data_v2$sp_2024, test_data_v2$major)

#Switching thing up this time
#Get rid of missing for all rows unless major is BIO
test_data_clean_v2 <- test_data_v2[test_data_v2$major == "BIO" |
                                     !is.na(test_data_v2$sp_2024),]
```

Reproducability

```
#Read in data
data1 <- read.csv("some_data.csv")

#Examine distributions
#Pre first
ggplot(data = data1) +
  geom_histogram(aes(pre))

#Then post
ggplot(data = data1) +
  geom_histogram(aes(post))

#Do a t-test
t.test(data1$pre, data1$post, paired = TRUE)
```

Reproducability

```r
#Read in data
data1 <- read.csv("different_data.csv")

#Examine distributions
#Pre first
ggplot(data = data1) +
  geom_histogram(aes(pre))

#Then post
ggplot(data = data1) +
  geom_histogram(aes(post))

#Do a t-test
t.test(data1$pre, data1$post, paired = TRUE)
```

Excel vs. R

- Excel can hold a decent amount of data
- Functions aid in data analysis
- Pivot tables
- Built-in data visualization

Excel vs. R

But…

How many times have you messed up an Excel sheet when trying to do an analysis and didn’t realize it until too late?

Excel vs. R

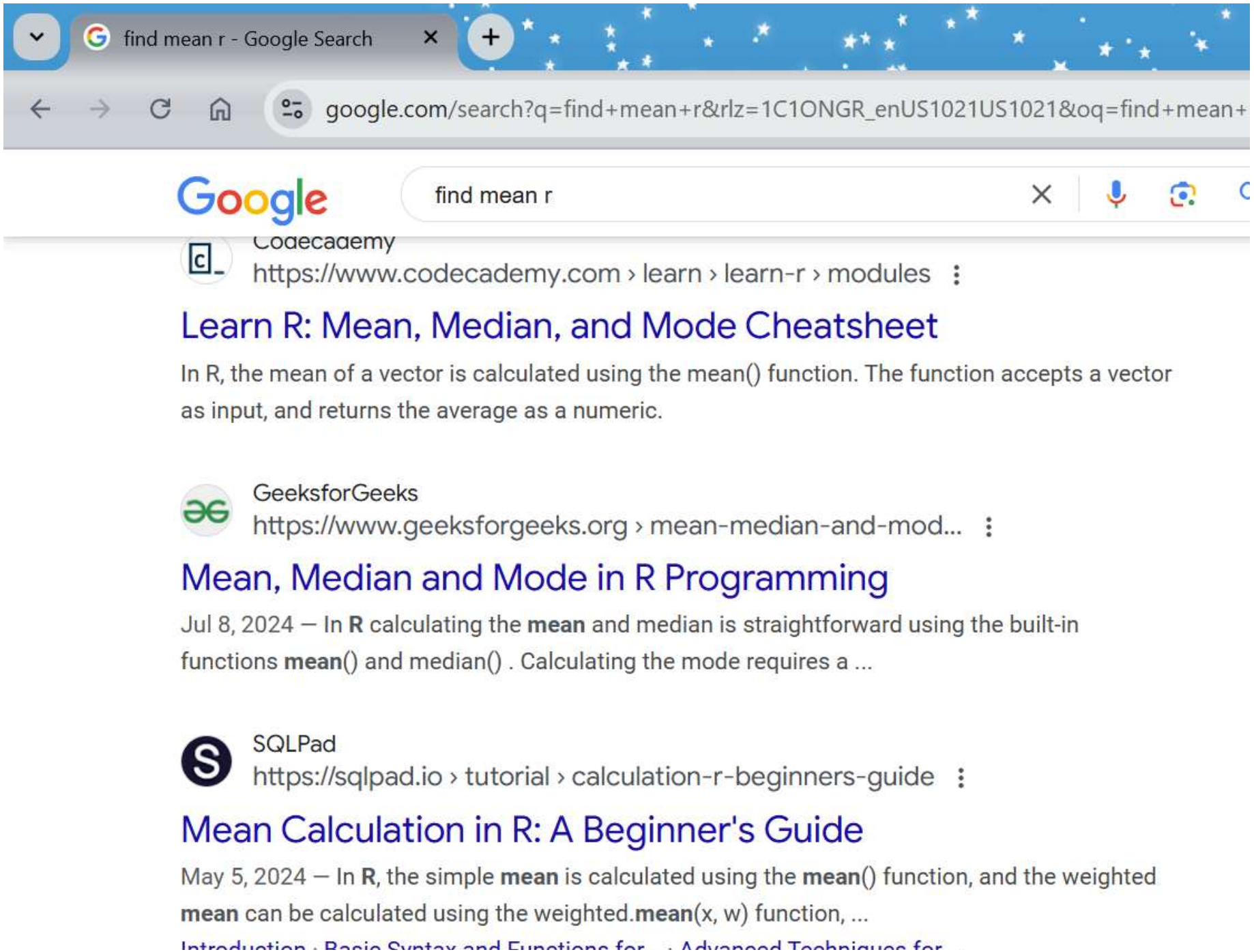
- In Excel, you perform calculations and manipulations on your original dataset
  - Unless you save a copy!
- With R, manipulations don't affect the original dataset
  - This was mind-blowing to me

Available R Resources

- A major benefit to R being open source: near infinite Google-ability
- Many other free resources if you wanted to learn more
- Can borrow code from other folks who have done it before
  - No need to reinvent the wheel

Available R Resources

Example: Google


find mean r - Google Search
google.com/search?q=find+mean+r&rlz=1C1ONGR_enUS1021US1021&oq=find+mean+
Google
find mean r
Codecademy
https://www.codecademy.com › learn › learn-r › modules
Learn R: Mean, Median, and Mode Cheatsheet
In R, the mean of a vector is calculated using the mean() function. The function accepts a vector as input, and returns the average as a numeric.
GeeksforGeeks
https://www.geeksforgeeks.org › mean-median-and-mod...
Mean, Median and Mode in R Programming
Jul 8, 2024 — In R calculating the mean and median is straightforward using the built-in functions mean() and median() . Calculating the mode requires a ...
SQLPad
https://sqlpad.io › tutorial › calculation-r-beginners-guide
Mean Calculation in R: A Beginner's Guide
May 5, 2024 — In R, the simple mean is calculated using the mean() function, and the weighted mean can be calculated using the weighted.mean(x, w) function, ...

Available R Resources

Example: YouTube

https://www.youtube.com/

Available R Resources

Example: Course materials

https://bookdown.org/laura_lambert_99/intermediate_stats/

Available R Resources

Example: Giant repository of R books

https://www.bigbookofr.com/

Available R Resources

Example: R for Data Science book

https://r4ds.hadley.nz/

Available R Resources

Example: GitHub for this presentation!

https://github.com/lauralambert99/Intro_to_R

Example Use Case

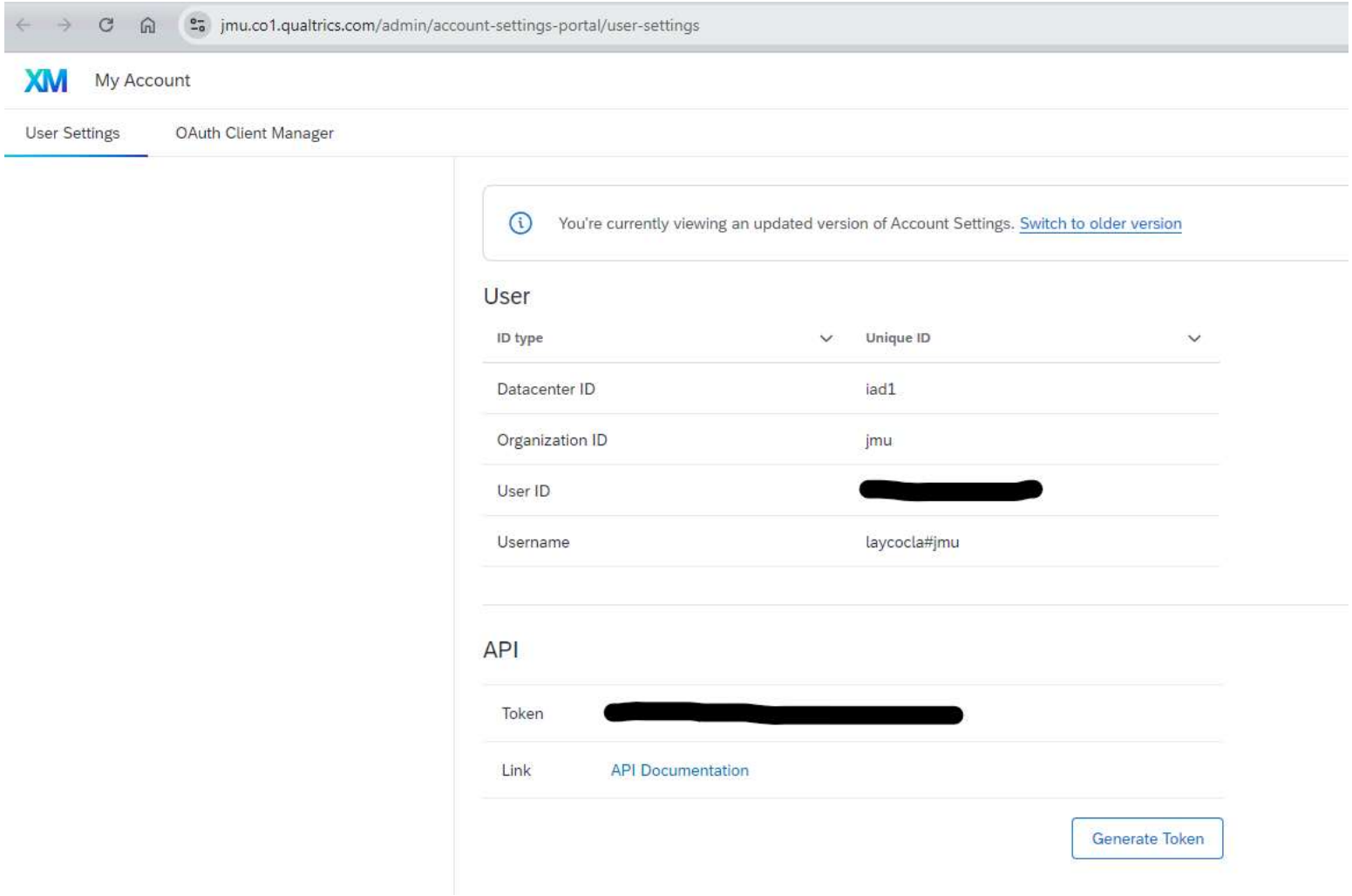
- Quick dive into some code, then we'll come back up
- Newer package: `qualtRics`
- Allows you to pull survey data straight into R for analysis

Example Use Case

```r
#Call the package
library(qualtRics)

#This is an initial set-up - only need to do once per machine!
qualtrics_api_credentials(api_key = "YOUR_KEY_HERE",
                          base_url = "URL.GOES.HERE",
                          install = TRUE)
```

Example Use Case



Example Use Case

```r
#See what surveys are associated with your account
#Here, saving to an object
surveys <- all_surveys()

#We can see what the column names are
names(surveys)

#And then see what surveys we have
surveys$name
```

Example Use Case

```r
#See what surveys are associated with your account
#Here, saving to an object
surveys <- all_surveys()

#We can see what the column names are
names(surveys)

#And then see what surveys we have
surveys$name
```

Example Use Case

```r
#See what surveys are associated with your account
#Here, saving to an object
surveys <- all_surveys()

#We can see what the column names are
names(surveys)

#And then see what surveys we have
surveys$name
```

```
[1] "id"           "name"         "ownerId"      "lastModified" "creationDate"
[6] "isActive"
```

Example Use Case

```r
#See what surveys are associated with your account
#Here, saving to an object
surveys <- all_surveys()

#We can see what the column names are
names(surveys)

#And then see what surveys we have
surveys$name
```

```
[1] "2023 ISAT Senior Survey"           "MS ISAT/MSc EMS Orientation Survey"
[3] "AI24_Q3"                           "2024 ISAT Senior Survey"
[5] "30th Anniversary Alumni Survey"    "STEM Belonging Survey"
[7] "AI24_Q2"                           "AI24_Q1"
[9] "STEM Focus Group Availability"
```

Example Use Case

```r
#Get survey data
survey_data <- fetch_survey(surveyID = surveys$id[8])
```

```
  |
  |                                                                      |
0%
  |
  |======================================================================|
100%
```

Example Use Case

```r
#Save question names to an object
questions <- survey_questions(surveyID = surveys$id[8])

#See what the questions are
print(questions)
```

```
# A tibble: 1 × 4
  qid   qname question
force_resp
  <chr> <chr> <chr>                                                    <lgl>
1 QID1  Q1    What software do you currently use to analyze your ass… FALSE
```

Example Use Case

```r
#Now, we can do things!
table(survey_data$Q1)
```

```
                     Excel                SPSS dropdowns
                        16                             2
               SPSS syntax                           SAS
                         4                             1
                         R                        Python
                         8                             0
Something not listed What is this 'analyze' you speak of?
                         4                             6
```

Parting Thoughts Before Demo

- You don’t need to be an expert in R to use it
- Use it however works for **you** - there isn’t a “right” way
- Projects make things easier
- Don’t reinvent the wheel!
- You’ll learn real fast how well you spell

Last Survey Question

Last bit of data!

You Promised Flexibility!

creative_name <- read.csv(“some/data/file.csv”)

You Promised Flexibility!

creative_name <- read.csv(“some/data/file.csv”)

You Promised Flexibility!

creative_name <- read.csv(“some/data/file.csv”)

You Promised Flexibility!

creative_name <- read.csv(“some/data/file.csv”)

You Promised Flexibility!

creative_name <- read.csv(“some/data/file.csv”)

You Promised Flexibility!

creative_name$some_variable

You Promised Flexibility!

creative_name$some_variable

You Promised Flexibility!

creative_name$some_variable

Live Demo time

Now, let's run some code, using the data you have so kindly provided

If you want to run it on your computer, access some fake data here: https://bit.ly/3NINVha

What questions can I answer for you now?

My Contact Info

Laura Lambert laycocla@jmu.edu