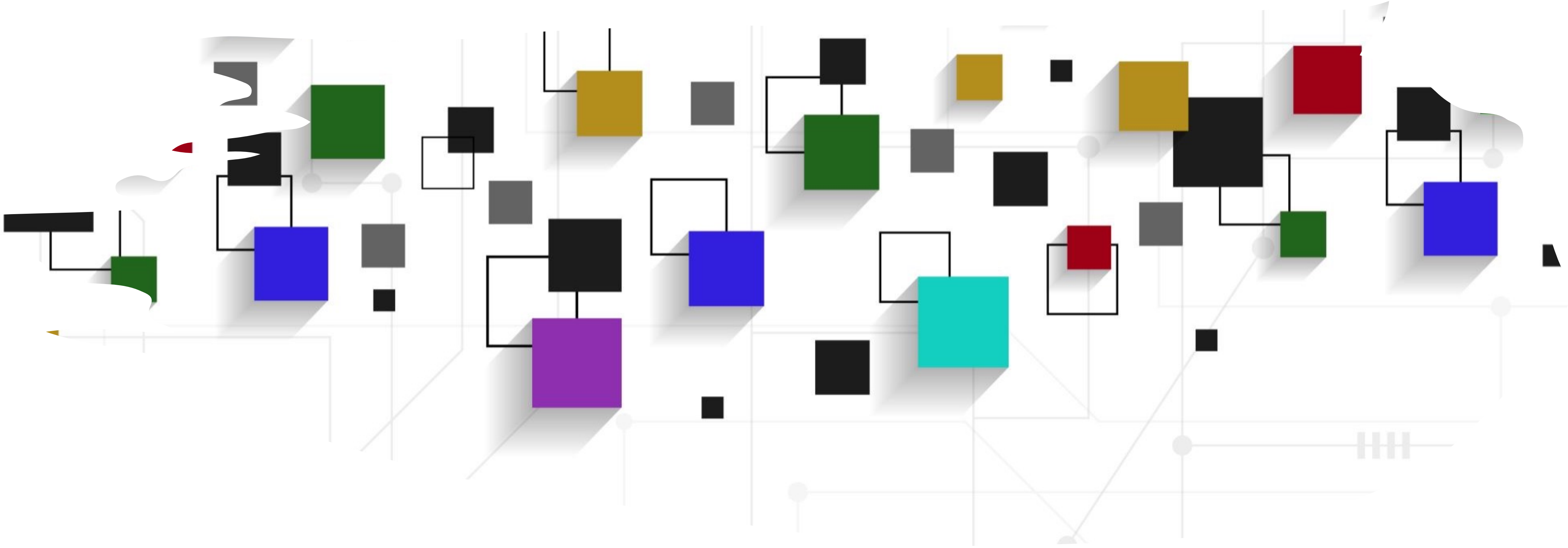


CogLab: Visualize Data

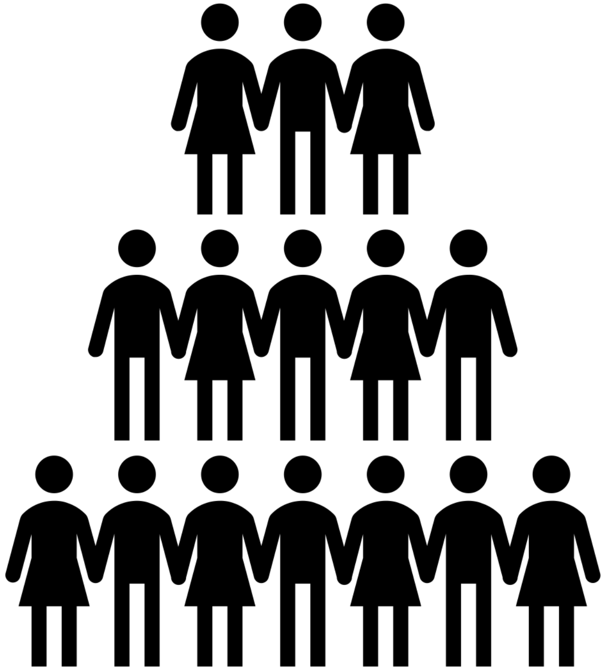
WEEK 8 / R101

logistics: formative assignment #1

- feedback sent to everyone
- resubmission due Oct 22
- common errors
  - creating separate plugin trials for practice and experiment
  - using repetitions
  - recording accuracy + feedback
- two choices:
  - keep original score: worth 10%
  - revise: worth 8%

logistics: project milestone #5

- full experiment code is due Oct 22
- incorporate feedback from design draft (milestone #4)
- include a demographics survey
  - after experiment, before thank you
  - decide which questions can go on the same screen vs. different screens
  - think about how to record the data



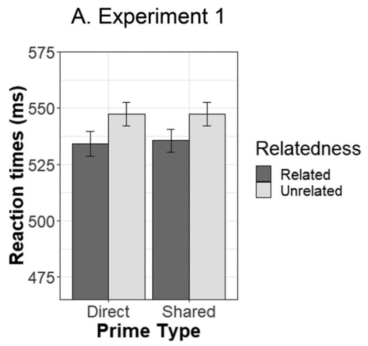
today’s agenda

- R 101
- programming basics
- visualizing data

recap: Oct 12, 2023

- what we covered:
  - reviewing design
  - intuitions about data analysis
- your to-do's were:
  - *prep*: download R for mac (first .pkg link)
  - *prep*: download Rstudio
  - *prep*: Programming Basics primer from posit
  - *apply*: formative assignment #1

analysis preview



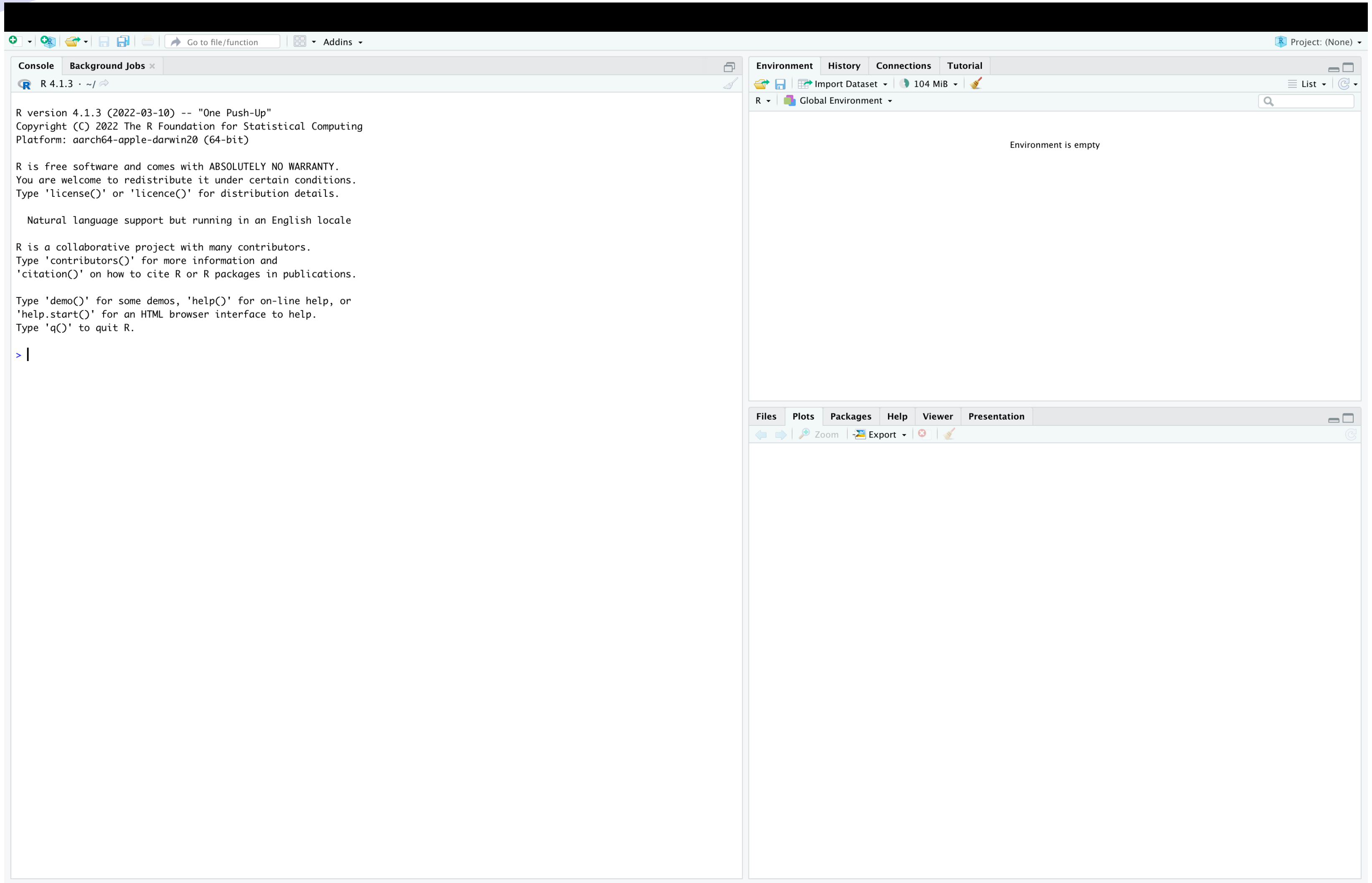
phase	measure	type	exclusion criteria
attention	accuracy	descriptive	< 0.75
association	proportion of correct/congruent responses + direct/shared responding	descriptive	
priming	$RT_{related}$ vs. $RT_{unrelated}$ for direct and shared pairs	inferential (mixed effects model / ANOVA)	RT < 200 ms and RT > 1500 ms correct responses related/unrelated and direct/shared trials

R 101

- R is a statistical programming language
- Rstudio is a graphical user interface for R that makes R a little less scary!
- we will use R within Rstudio to analyze data

installation check-in

Rstudio: basics

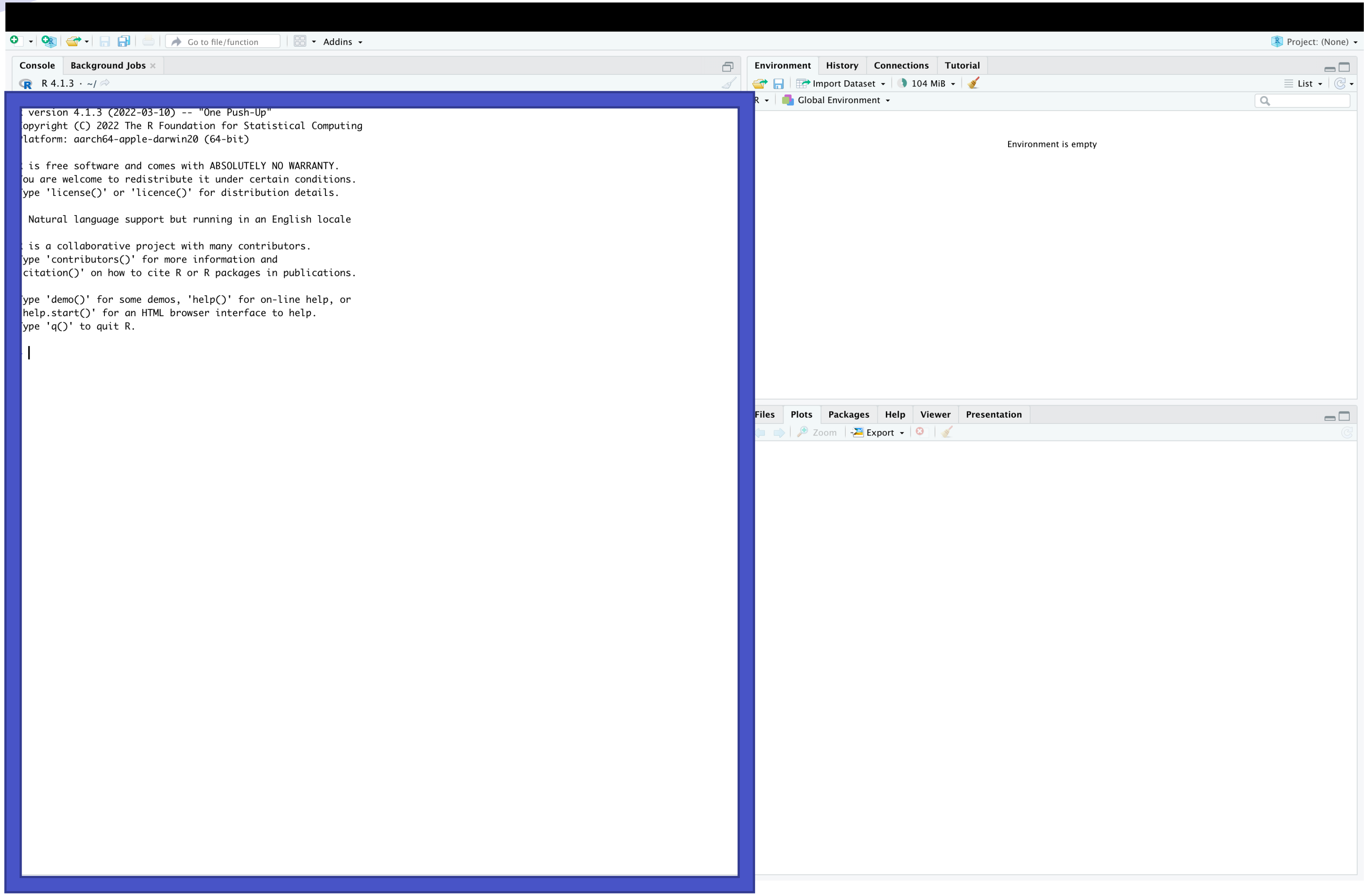


```
R version 4.1.3 (2022-03-10) -- "One Push-Up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Environment is empty

Rstudio: basics

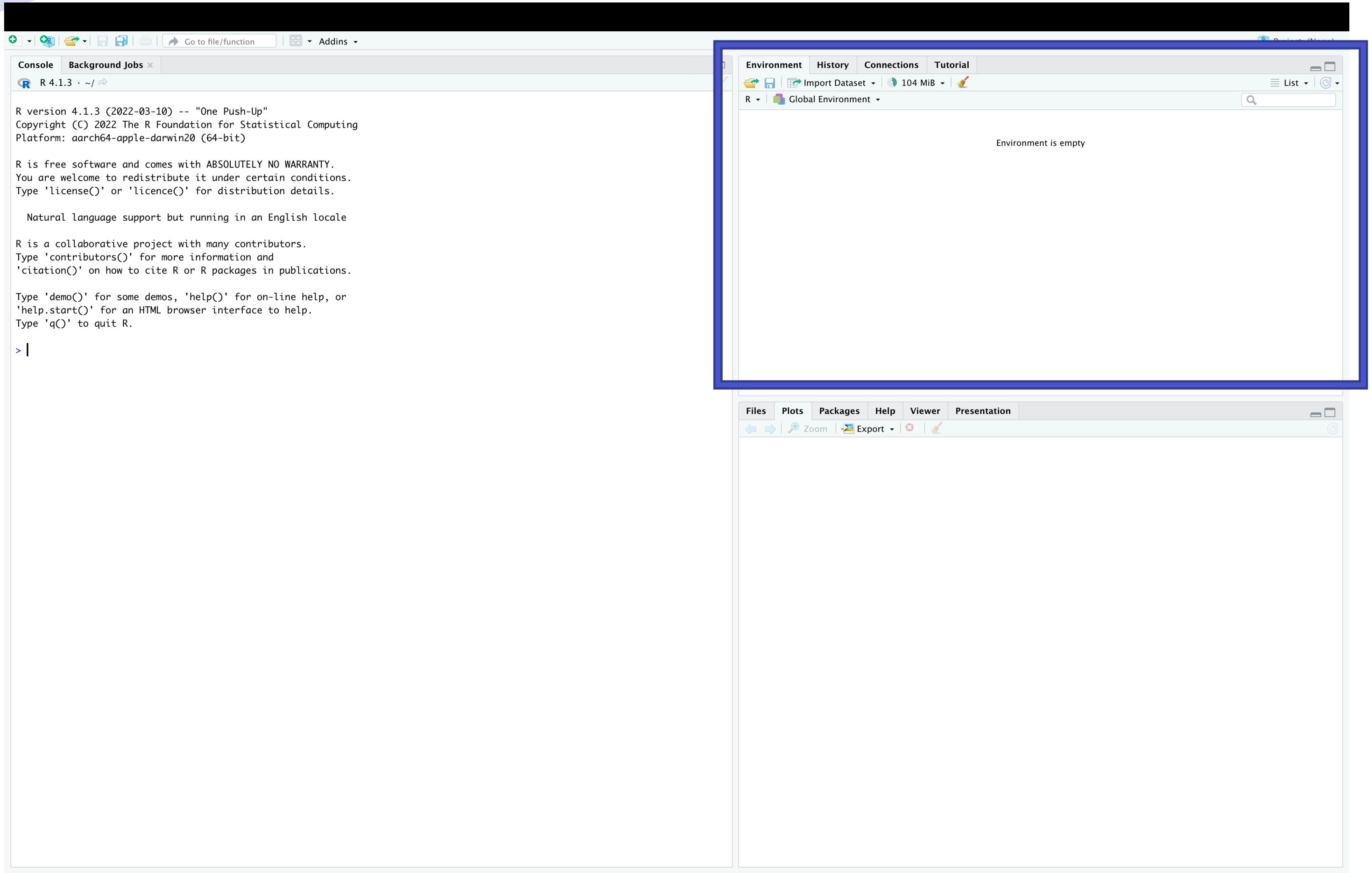
This is called the “console”. This is where any code you write is run by R



```
 version 4.1.3 (2022-03-10) -- "One Push-Up"
opyright (C) 2022 The R Foundation for Statistical Computing
latform: aarch64-apple-darwin20 (64-bit)

 is free software and comes with ABSOLUTELY NO WARRANTY.
ou are welcome to redistribute it under certain conditions.
ype 'license()' or 'licence()' for distribution details.

 Natural language support but running in an English locale

 is a collaborative project with many contributors.
ype 'contributors()' for more information and
citation()' on how to cite R or R packages in publications.

ype 'demo()' for some demos, 'help()' for on-line help, or
help.start()' for an HTML browser interface to help.
ype 'q()' to quit R.

 |
```

Environment is empty

Rstudio: basics

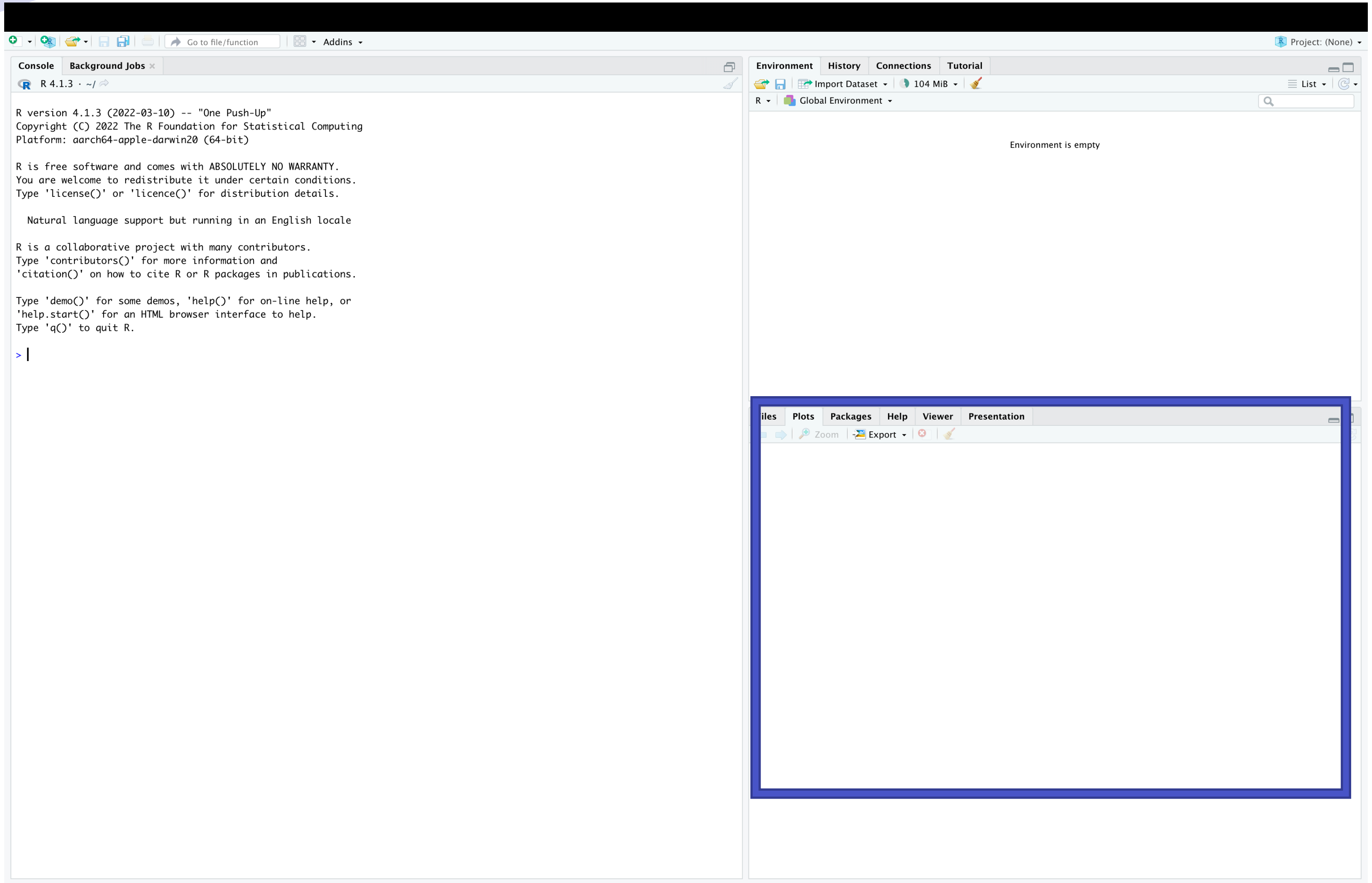


[redacted]

```
R version 4.1.3 (2022-03-10) -- "One Push-Up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

This is called the “environment”, this is where any data you import will show up. This is also where any variables you define or create will show up

Rstudio: basics

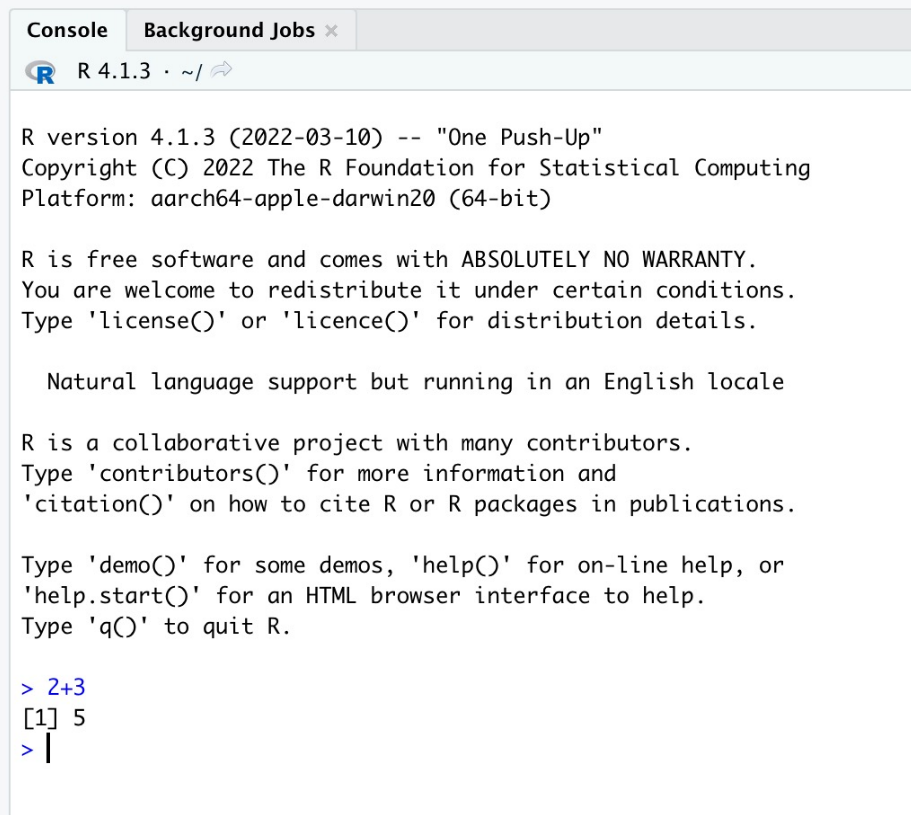


```
R version 4.1.3 (2022-03-10) -- "One Push-Up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

This is called the files and plots pane. Currently it is on “plots”, so any plots you make in the R console in R will show up here as a preview

testing the R console

- you can type any code in the console and it will try to understand and run it
- try typing 2 + 3 where you see the blinking cursor and press return
- voila! R just added those up for you!



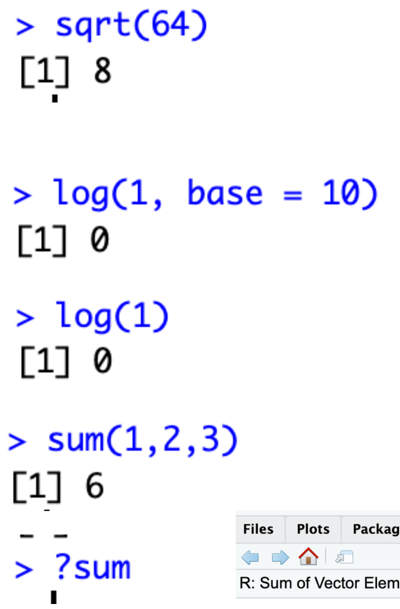
```
R version 4.1.3 (2022-03-10) -- "One Push-Up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 2+3
[1] 5
> |
```

programming basics

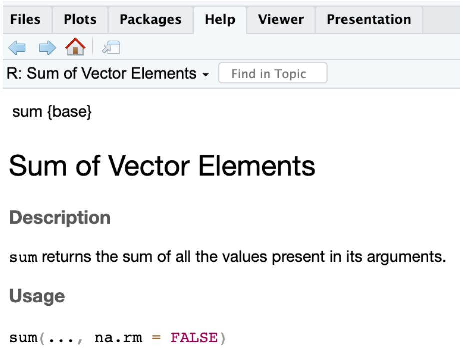
- functions
- arguments
- objects
- vectors
- types
- lists
- packages

exercises: functions

- functions do things in R
- compute
  - the square root of 64
  - log of 1 with base 10
  - log of 1 with default base
  - sum of 1, 2, and 3
- how do we get more info about a function (e.g., sum)?



```
> sqrt(64)
[1] 8

> log(1, base = 10)
[1] 0

> log(1)
[1] 0

> sum(1,2,3)
[1] 6

> ?sum
```



sum {base}

Sum of Vector Elements

Description

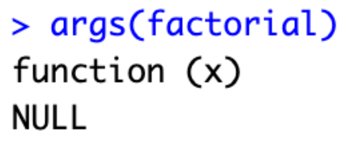
`sum` returns the sum of all the values present in its arguments.

Usage

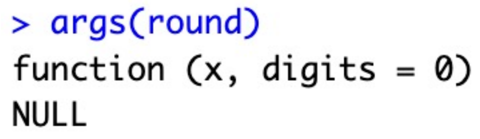
```
sum(..., na.rm = FALSE)
```

exercises: arguments

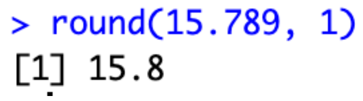
- arguments are inputs to functions
- find out the arguments for:
  - factorial
  - round
- round 15.789 to 1 digit
- compute 10 random values from a normal distribution with mean 15 and sd 5



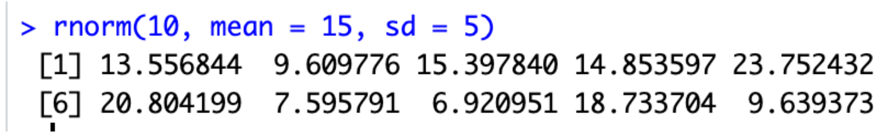
```
> args(factorial)
function (x)
NULL
```



```
> args(round)
function (x, digits = 0)
NULL
```



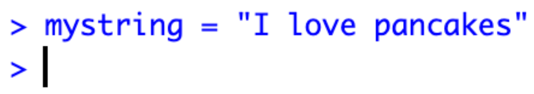
```
> round(15.789, 1)
[1] 15.8
```



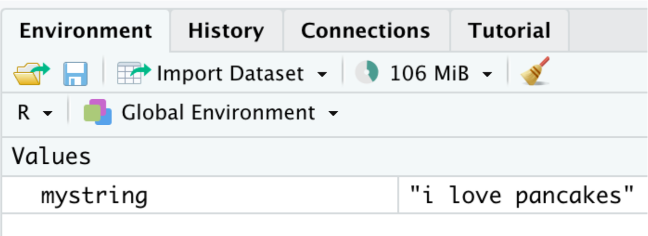
```
> rnorm(10, mean = 15, sd = 5)
 [1] 13.556844  9.609776 15.397840 14.853597 23.752432
 [6] 20.804199  7.595791  6.920951 18.733704  9.639373
```

exercises: objects

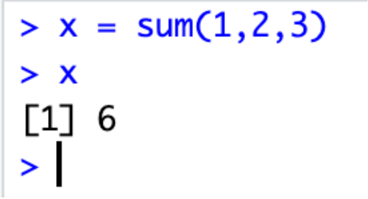
- objects store information
- store:
  - “I love pancakes” in an object called mystring
  - sum of 1,2, and 3 in x
  - 50 random values from a normal distribution with mean 20 and sd 1 in an object called r
- plot the histogram of r
- what is the mean of r?
- what is the standard deviation of r?



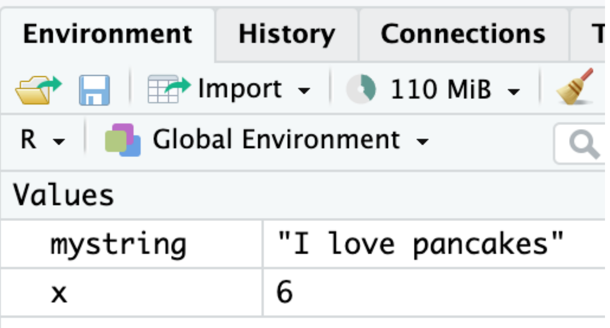
```
> mystring = "I love pancakes"
> |
```



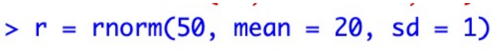
```
Values
  mystring                 "i love pancakes"
```



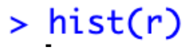
```
> x = sum(1,2,3)
> x
[1] 6
> |
```



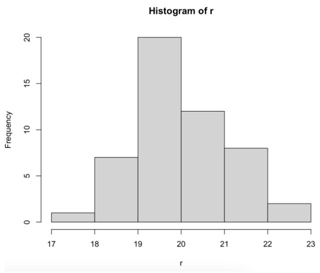
```
Values
  mystring           "I love pancakes"
  x                  6
```

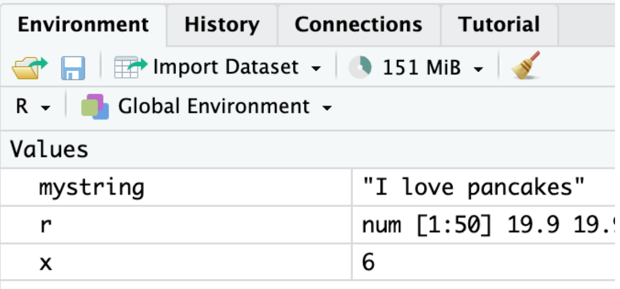


```
> r = rnorm(50, mean = 20, sd = 1)
```



```
> hist(r)
```

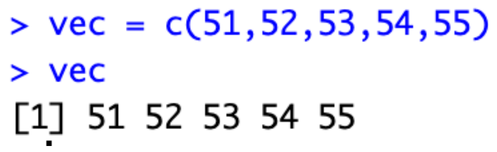




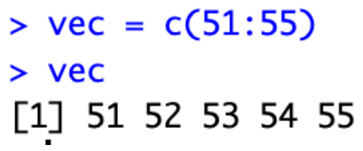
```
Values
  mystring             "I love pancakes"
  r                    num [1:50] 19.9 19.
  x                    6
```

exercises: vectors

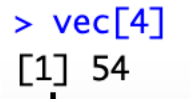
- vectors are 1-d arrays of values
- create a vector of integers from 51 to 55
- R counts from 1 (unlike JavaScript!)
- retrieve the fourth value from this vector



```
> vec = c(51,52,53,54,55)
> vec
[1] 51 52 53 54 55
```



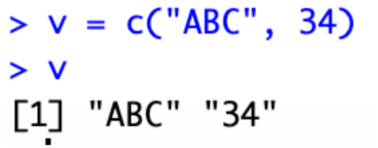
```
> vec = c(51:55)
> vec
[1] 51 52 53 54 55
```



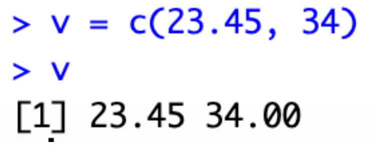
```
> vec[4]
[1] 54
```

exercises: data types

- R has 6 basic data types: integer, double, character (string), logical, complex, raw
- vectors can only contain values of the same data type
- try storing a character string and a number together in a vector
- R will automatically convert them to the same datatype (character is default)



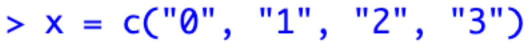
```
> v = c("ABC", 34)
> v
[1] "ABC" "34"
```



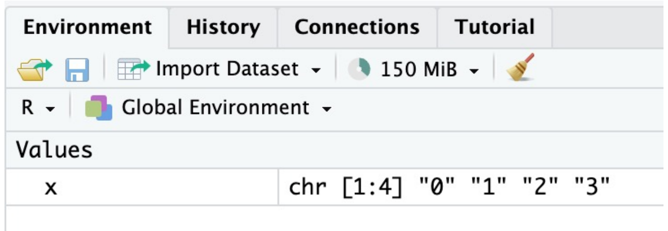
```
> v = c(23.45, 34)
> v
[1] 23.45 34.00
```

factors and levels

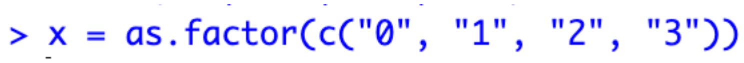
- when your variables have specific "conditions", they are referred to as factors in R and these factors are said to have levels
- R may not immediately recognize a column as a factor
  - typically, just treated as a or "numeric" or "character" vector
- but we might want to designate our independent variables as factors
  - for plotting
  - for analyses



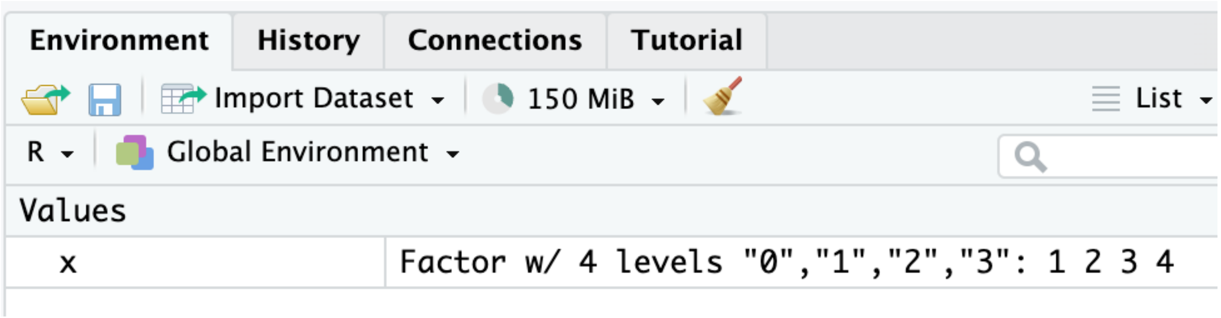
```
> x = c("0", "1", "2", "3")
```



```
Values
  x                chr [1:4] "0" "1" "2" "3"
```



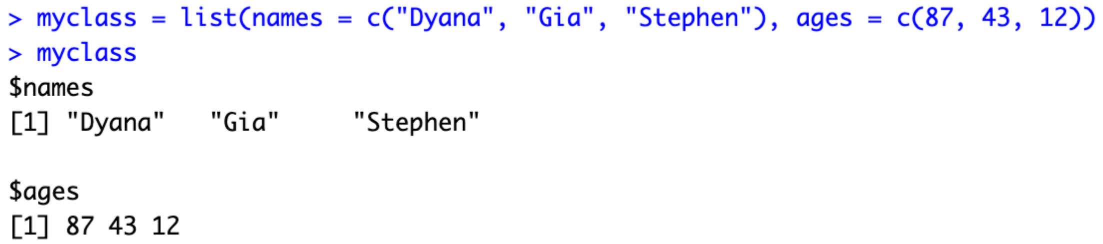
```
> x = as.factor(c("0", "1", "2", "3"))
```



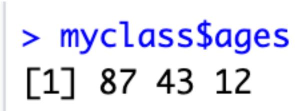
```
Values
  x                Factor w/ 4 levels "0","1","2","3": 1 2 3 4
```

exercises: lists

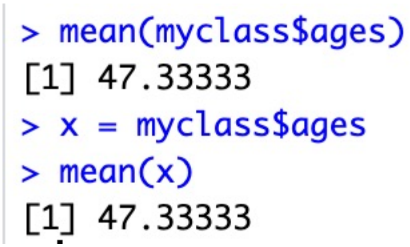
- lists store multiple data types
- create a list called myclass of names and ages
- retrieve the ages from myclass
- compute the mean of the ages



```
> myclass = list(names = c("Dyana", "Gia", "Stephen"), ages = c(87, 43, 12))
> myclass
$names
[1] "Dyana"   "Gia"     "Stephen"

$ages
[1] 87 43 12
```



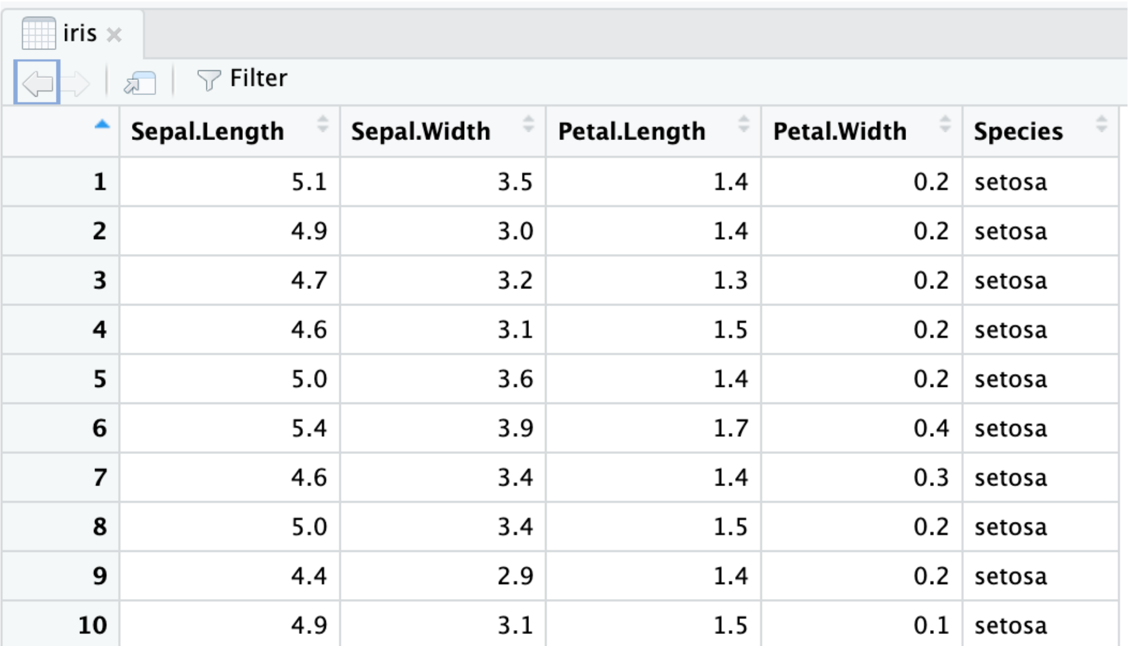
```
> myclass$ages
[1] 87 43 12
```



```
> mean(myclass$ages)
[1] 47.33333
> x = myclass$ages
> mean(x)
[1] 47.33333
```

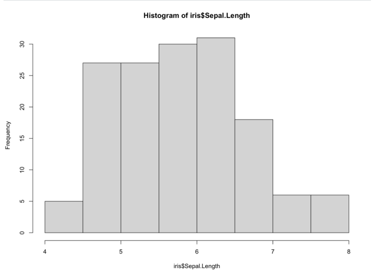
exercises: dataframes

- tabular data is stored as a dataframe in R
- dataframes are simply a list of vectors
- View(iris)
- create a histogram of sepal lengths from iris



	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

```
> View(iris)
> hist(iris$Sepal.Length)
```

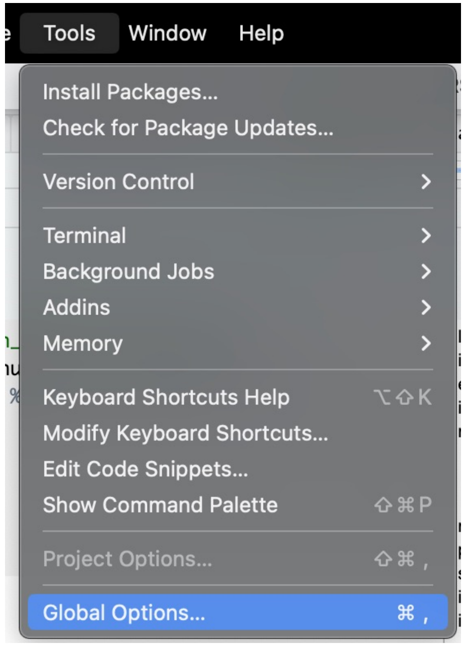


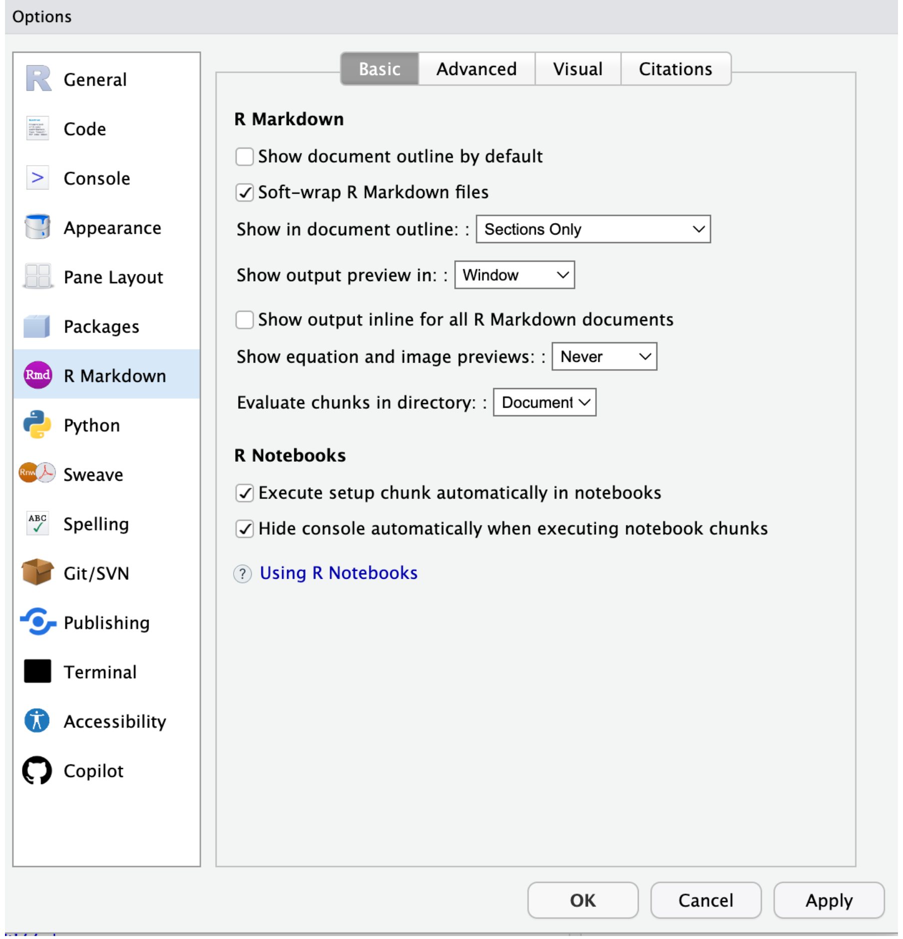
storing code

- anything you type in the console will go away when you close R
- R projects and notebooks are a good way to keep track of your code and maintain a reproducible workflow

setting global options

- go to tools > global options
- uncheck “show output inline”
- apply





R Markdown

□ Show document outline by default

☑ Soft-wrap R Markdown files

Show in document outline: : Sections Only

Show output preview in: : Window

□ Show output inline for all R Markdown documents

Show equation and image previews: : Never

Evaluate chunks in directory: : Document

R Notebooks

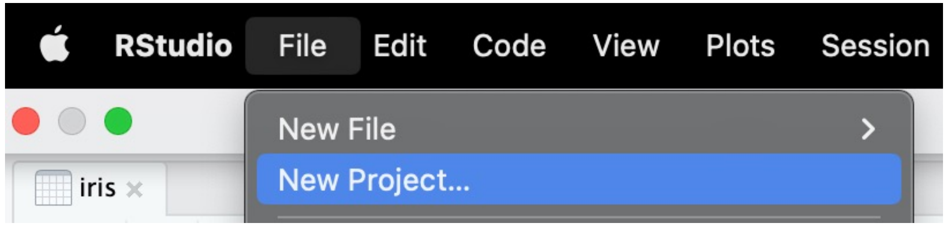
☑ Execute setup chunk automatically in notebooks

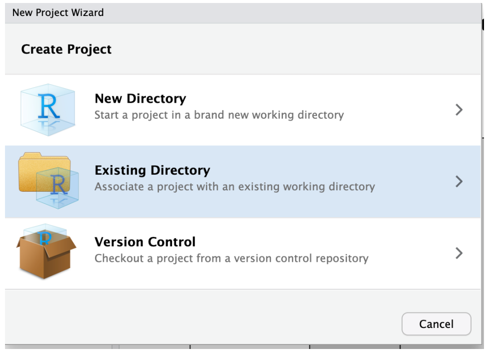
☑ Hide console automatically when executing notebook chunks

Using R Notebooks

creating a project in R

- File > New Project > existing directory
  - select the directory of first-jspsych-experiment
- File > New file > R Markdown
  - opens a notebook in Markdown format
- you can add “chunks” of code
  - between ```{r} and ```
- save (Command +S) this file as “first-R-notebook” with the .Rmd extension
- we will write all our analysis code in this file!




New Project Wizard

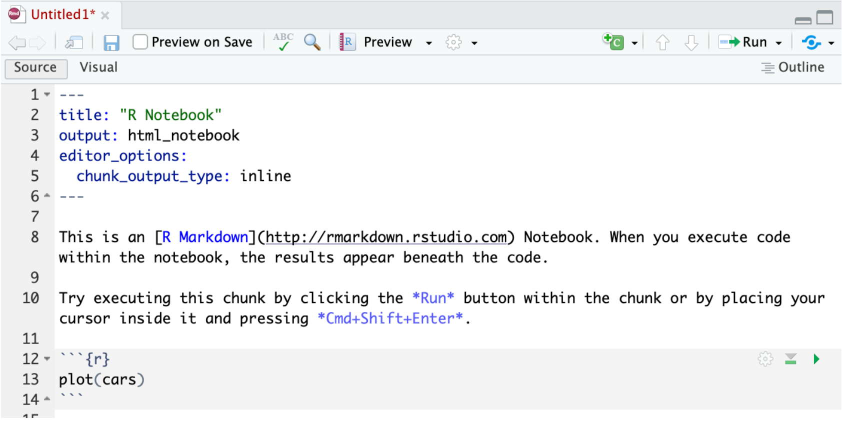
Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

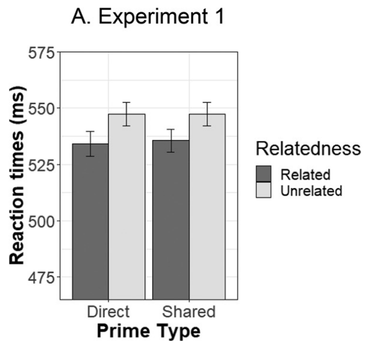
Version Control
Checkout a project from a version control repository

Cancel



~~~
---
title: "R Notebook"
output: html_notebook
editor_options:
  chunk_output_type: inline
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code
within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your
cursor inside it and pressing *Cmd+Shift+Enter*.

```{r}
plot(cars)
```
~~~

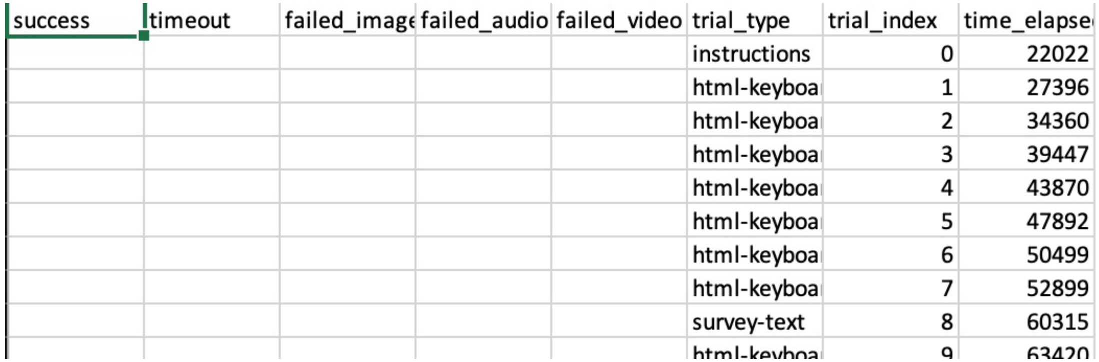
analysis preview



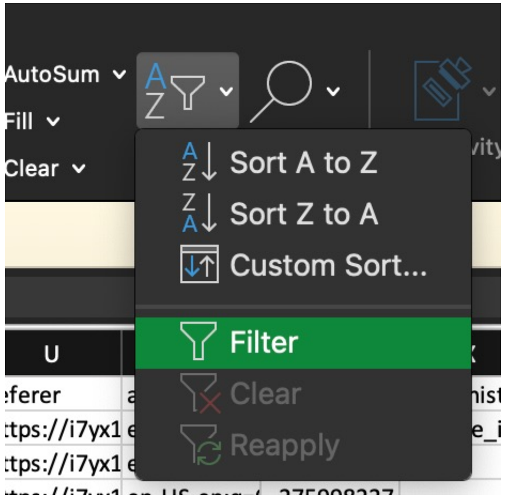
phase	measure	type	exclusion criteria
attention	accuracy	descriptive	< 0.75
association	proportion of correct/congruent responses + direct/shared responding	descriptive	
priming	$RT_{related}$ vs. $RT_{unrelated}$ for direct and shared pairs	inferential (mixed effects model / ANOVA)	RT < 200 ms and RT > 1500 ms correct responses related/unrelated and direct/shared trials

exploring our data

- download and save class_data.csv in first-jspsych-experiment folder
- open class_data in Excel
- apply a filter to the data

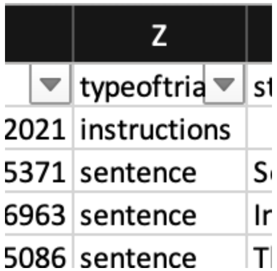


success	timeout	failed_image	failed_audio	failed_video	trial_type	trial_index	time_elapse
					instructions	0	22022
					html-keyboa	1	27396
					html-keyboa	2	34360
					html-keyboa	3	39447
					html-keyboa	4	43870
					html-keyboa	5	47892
					html-keyboa	6	50499
					html-keyboa	7	52899
					survey-text	8	60315
					html-keyboa	9	63420

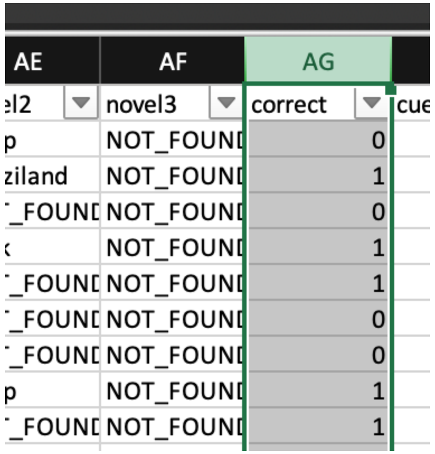


filtering rows in excel

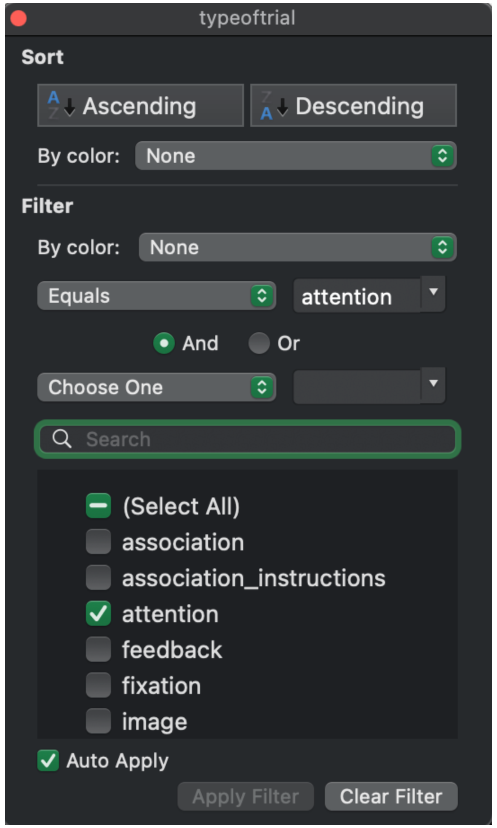
- locate the column that identifies each type of trial in our experiment
- filter to only attention trials
- locate the column that contains the accuracy



	Z	
	typeoftria	st
2021	instructions	
5371	sentence	S
6963	sentence	Ir
5086	sentence	T



AE	AF	AG	
el2	novel3	correct	cue
p	NOT_FOUNI	0	
ziland	NOT_FOUNI	1	
_FOUNI	NOT_FOUNI	0	
k	NOT_FOUNI	1	
_FOUNI	NOT_FOUNI	1	
_FOUNI	NOT_FOUNI	0	
_FOUNI	NOT_FOUNI	0	
p	NOT_FOUNI	1	
_FOUNI	NOT_FOUNI	1	



typeoftrial

Sort

Ascending Descending

By color: None

Filter

By color: None

Equals attention

● And ○ Or

Choose One

Search

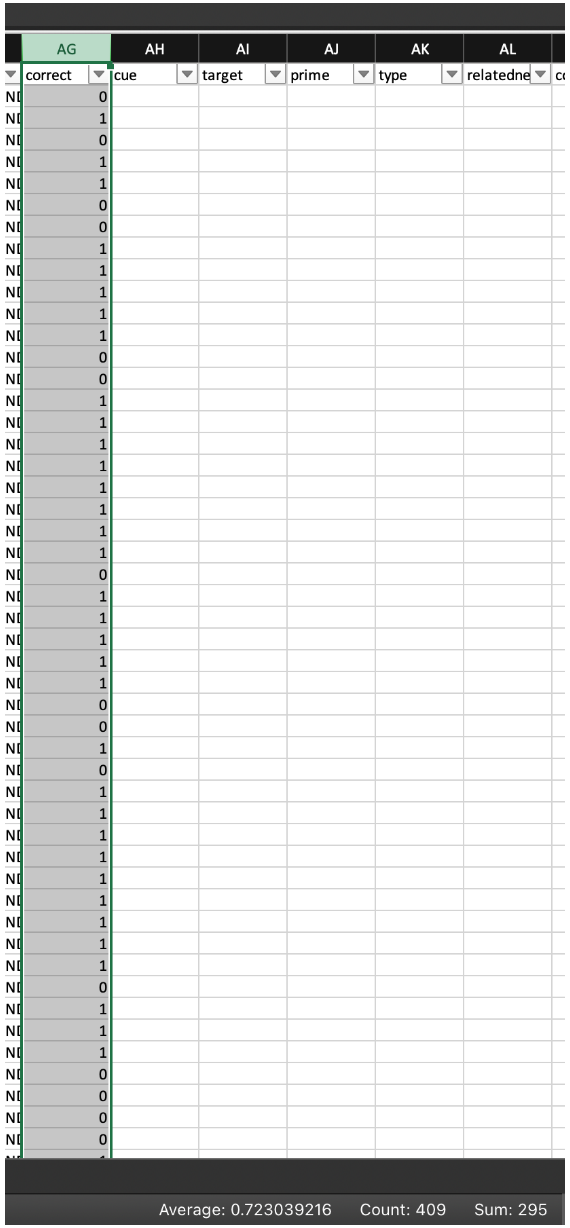
- ▬ (Select All)
- □ association
- □ association_instructions
- ☑ attention
- □ feedback
- □ fixation
- □ image

☑ Auto Apply

Apply Filter Clear Filter

exploring averages

- selecting the whole column will provide some summary statistics
- what do you learn?
- what were the exclusion criteria for attention?



	AG	AH	AI	AJ	AK	AL	
	correct	cue	target	prime	type	relatedne	c
N	0						
N	1						
N	0						
N	1						
N	1						
N	0						
N	0						
N	1						
N	1						
N	1						
N	1						
N	1						
N	0						
N	0						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	0						
N	1						
N	1						
N	1						
N	1						
N	1						
N	0						
N	0						
N	1						
N	0						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	1						
N	0						
N	1						
N	1						
N	1						
N	0						
N	0						
N	0						
N	0						

Average: 0.723039216 Count: 409 Sum: 295

next class

- **before** class
  - *prep*: Visualization Basics primer from posit
  - *prep*: work on project milestone #5
  - *prep*: work on formative milestone #1 resubmission
- **during** class
  - plotting data