Introduction to Data Analysis in R using the Tidyverse

SACNAS National Conference
San Juan, Puerto Rico
Oct 27, 2022

WIFI: **SACNAS2022**

Password: **NDISTEM@22**

Go to RStudio Cloud and make an account:
https://rstudio.cloud/content/4686217

Outline

1. Intro to R and RStudio
2. Data Management Systems
3. Data Visualization
4. Tidy Verse Functions and Examples

WIFI: **SACNAS2022**

Password: **NDISTEM@22**

Go to RStudio Cloud and make an account: https://rstudio.cloud/content/4686217

Introductions

Tina Del Carpio
cad17@g.ucla.edu
They/Them
Univ of CA, Los Angeles (UCLA)
Biology PhD Student

Angelica Riojas, PhD
riojasam@uthscsa.edu
She/Her
University of Texas Health, San Antonio
Postdoctoral Fellow
Radiation Imaging Institute

Jazlyn Mooney, PhD
jazlynmo@usc.edu
She/Her
University of Southern California
Gabilan Assistant Professor
Dept. of Quantitative and
Computational Biology

Jesse Garcia
jessegarcia562@ucla.edu
He/Him
Univ of CA, Los Angeles (UCLA)
Bioinformatics PhD Student

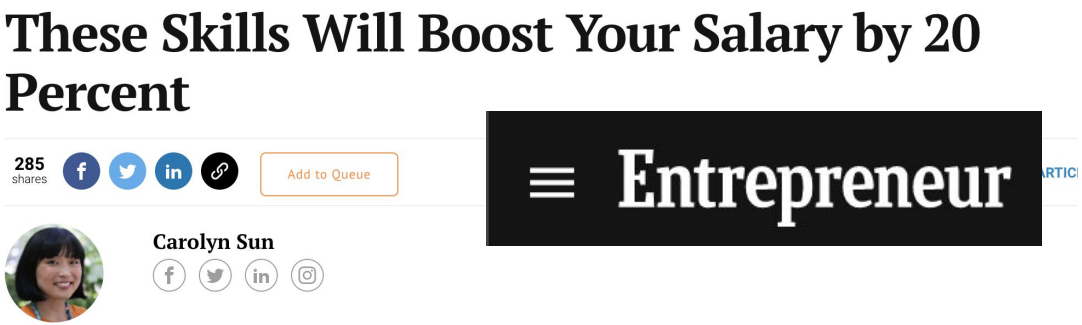
Introductions

Turn to your neighbor(s) and share

- Name
- Pronouns (optional)
- Institution
- Positon (junior, grad student, staff, etc)
- Coding experience
- One thing you hope to get out of this session

Why learn to code?

- Useful across STEM fields
- Fastest growing jobs according to Forbes.com includes
  - Software developer - median pay $110K
  - Data scientist - median pay $98K
- Coding give highest boost in salary up to ~20% (payscale.com)



Slides Modified from Chris Robles

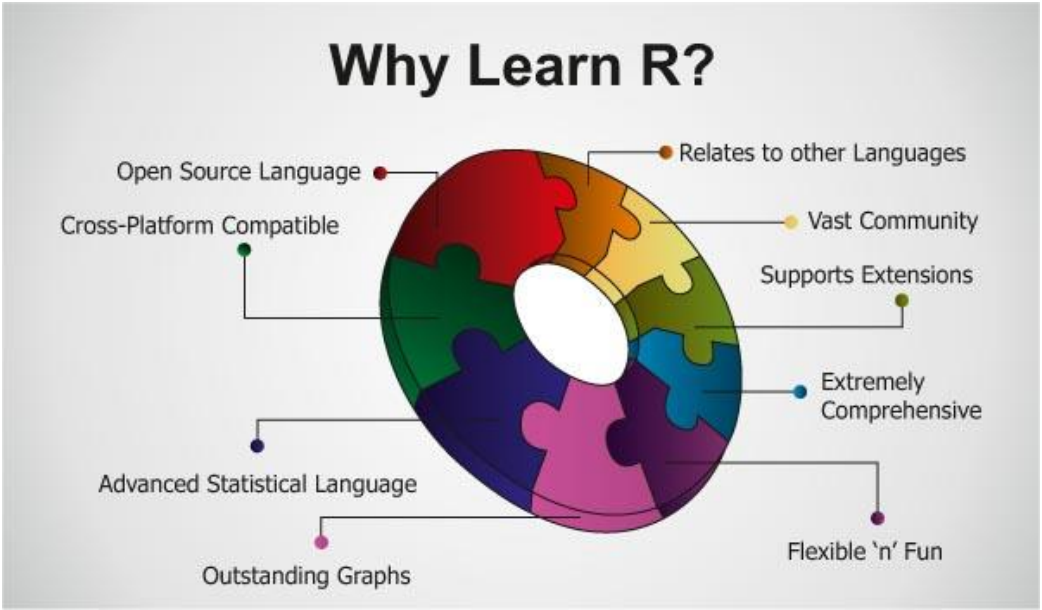
What is R and RStudio?

- R is a **programming language** for statistical computing and graphics
  - Widely used amongst scientists, statisticians and data scientists
  - Accessed through the command line
  - Ranked #2 best programming language to learn for data science (https://www.technotification.com)
- Rstudio is an **Integrated development environment** (IDE)
  - Contains a debugger, automation tools, and code editor
  - Has a GUI (graphical user interface) making it more user friendly

Slides Modified from Chris Robles

Why learn R vs excel?

- Pros
  - Can handle very large datasets
  - Faster calculations
  - Easily reproduced
  - More complex and advanced data visualization
  - FREE!
- Cons
  - Steeper learning curve but definitely surmountable!



Slides Modified from Chris Robles

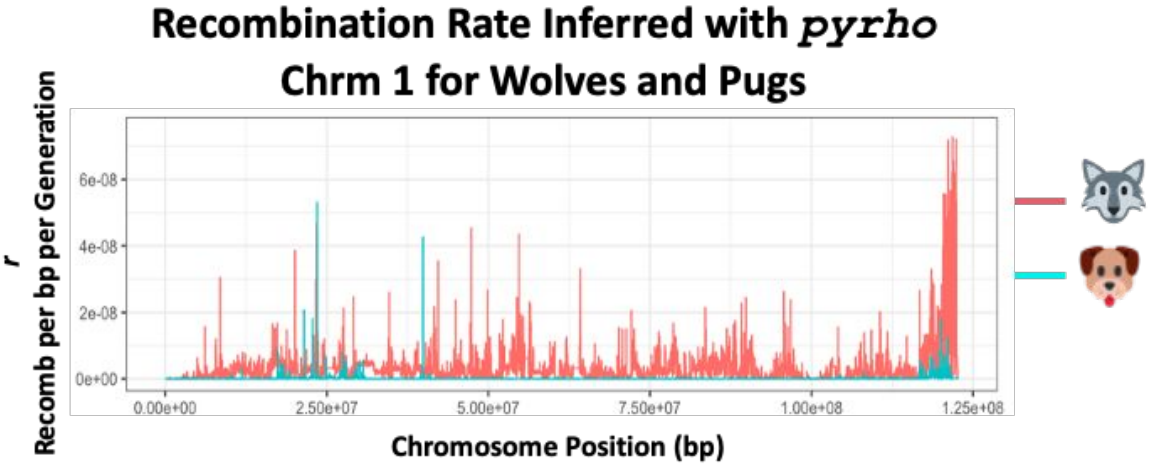
Examples of how it can be used

- Import your data into R
- Perform statistical analysis on an experiment
- Plot data from an experiment

Slides Modified from Chris Robles

Examples of how it can be used

- Import your data into R
- Perform statistical analysis on an experiment
- Plot data from an experiment



Slides Modified from Chris Robles

Where to Download?

- R: http://cran.r-project.org/bin/windows/base
- RStudio: http://www.rstudio.com/products/rstudio/download/
- Rstudio Cloud: https://rstudio.cloud/
  - Browser based **(We will use this today!)**

Go to RStudio Cloud and make an account:
https://rstudio.cloud/content/4686217

Slides Modified from Chris Robles

Your Workspace / SACNAS 2022 Jesse Garcia

TEMPORARY COPY RAM Save a Permanent Copy Tina Del Carpio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins R 4.2.1

Console Terminal Background Jobs

R 4.2.1 · /cloud/project/

```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

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 History Connections Tutorial

Import Dataset 124 MiB List

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

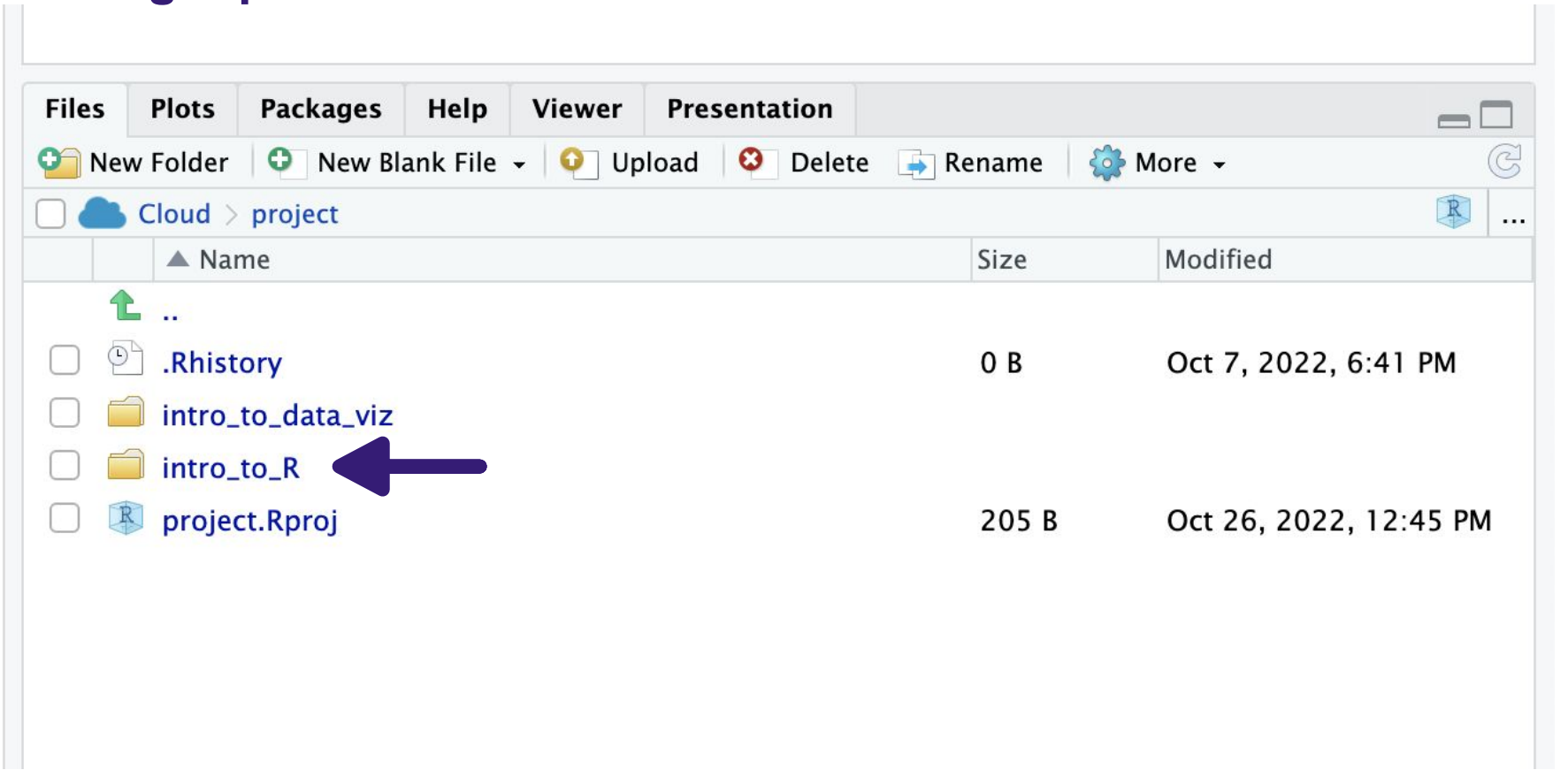
New Folder New Blank File Upload Delete Rename More

Cloud > project

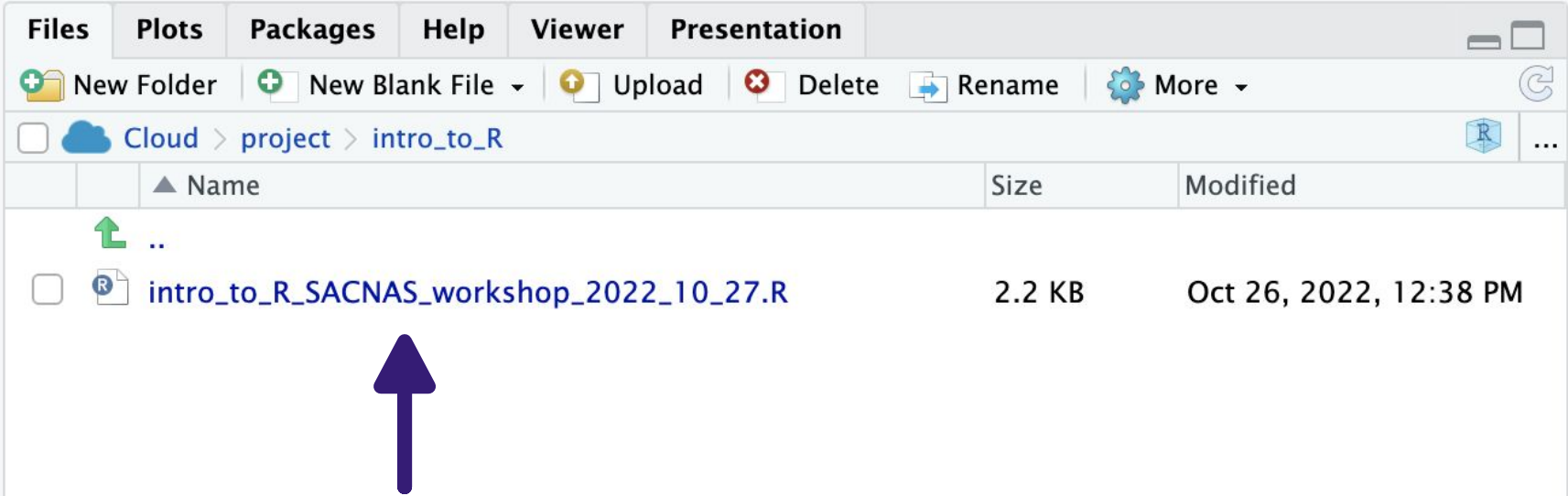
Name	Size	Modified
..		
.Rhistory	0 B	Oct 7, 2022, 6:41 PM
collegeboard_data_most_recent_cohorts.csv	548.7 KB	Oct 12, 2022, 1:12 AM
collegeboard_data_most_recent_cohorts.tsv	543.3 KB	Oct 12, 2022, 1:12 AM
Data_visualization.Rmd	12.3 KB	Oct 12, 2022, 3:32 PM
project.Rproj	205 B	Oct 26, 2022, 1:03 AM

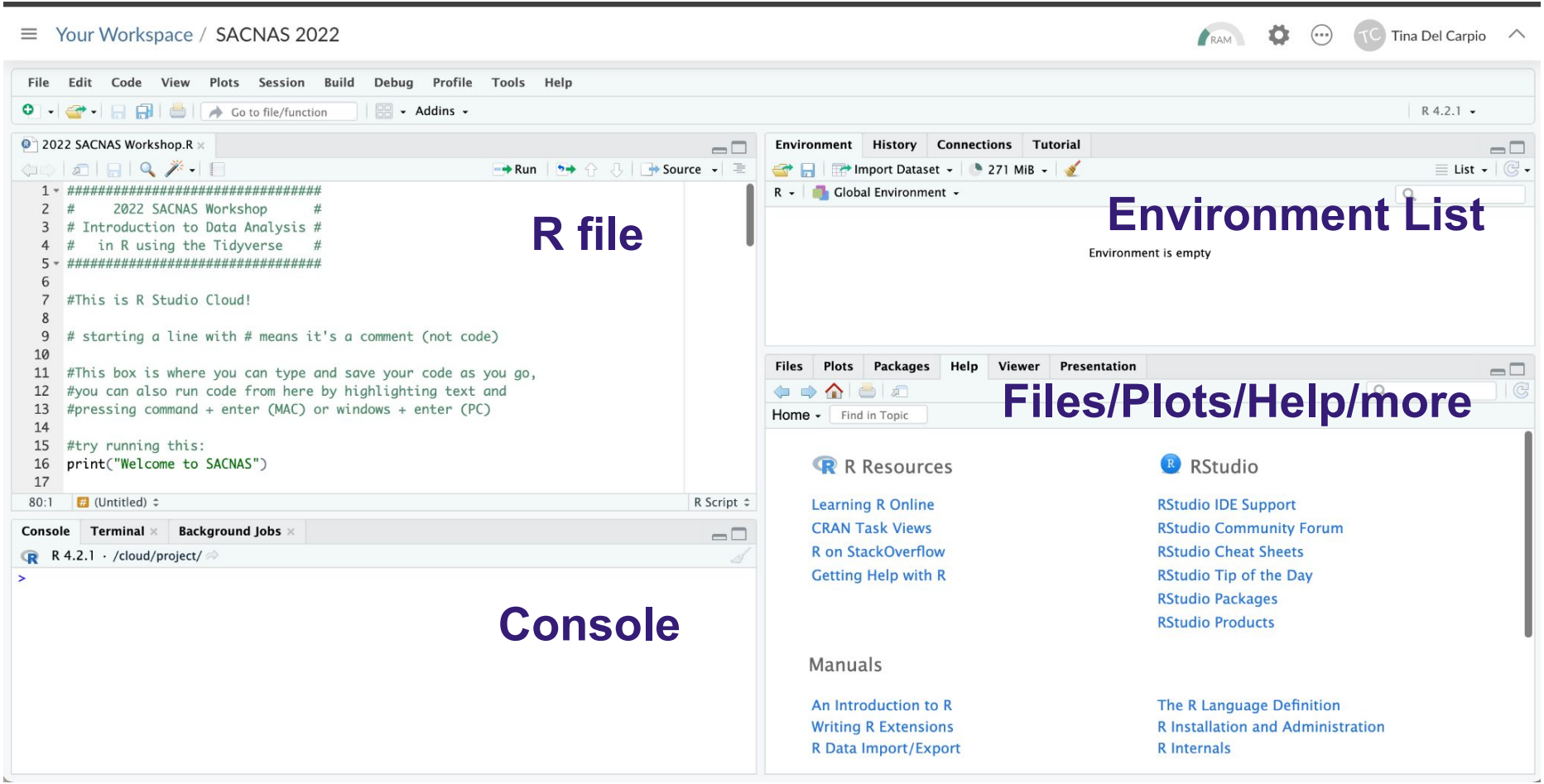
Link to RStudio Cloud Files for Today:
https://rstudio.cloud/content/4686217

Bottom right panel



Bottom right panel




Your Workspace / SACNAS 2022
Tina Del Carpio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 4.2.1
2022 SACNAS Workshop.R
Run
Source
1 ################################
2 #   2022 SACNAS Workshop   #
3 # Introduction to Data Analysis #
4 #   in R using the Tidyverse   #
5 ################################
6
7 #This is R Studio Cloud!
8
9 # starting a line with # means it's a comment (not code)
10
11 #This box is where you can type and save your code as you go,
12 #you can also run code from here by highlighting text and
13 #pressing command + enter (MAC) or windows + enter (PC)
14
15 #try running this:
16 print("Welcome to SACNAS")
17
80:1
(Untitled)
R Script
R file
Console
Terminal
Background Jobs
R 4.2.1 · /cloud/project/
>
Console
Environment
History
Connections
Tutorial
Import Dataset
271 MiB
List
R
Global Environment
Environment is empty
Environment List
Files
Plots
Packages
Help
Viewer
Presentation
Home
Find in Topic
Files/Plots/Help/more
R Resources
Learning R Online
CRAN Task Views
R on StackOverflow
Getting Help with R
RStudio
RStudio IDE Support
RStudio Community Forum
RStudio Cheat Sheets
RStudio Tip of the Day
RStudio Packages
RStudio Products
Manuals
An Introduction to R
Writing R Extensions
R Data Import/Export
The R Language Definition
R Installation and Administration
R Internals

```
10
11  #This box is where you can type and save your code as you go,
12  #you can also run code from here by highlighting text and
13  #pressing command + enter (MAC) or windows + enter (PC)
14
15  #try running this:
16  print("Welcome to SACNAS")
17
18  #you should now see Welcome to SACNAS in the box below
19  #that's the console where your code will run and your output will appear
20
21
22  ########################
```

15:1 (Untitled) R Script

Console | Terminal | Background Jobs

R 4.2.1 · /cloud/project/

```
> #try running this:
> print("Welcome to SACNAS")
[1] "Welcome to SACNAS"
>
```

← **Should create this output in the console**

```
######################
#now let's make our first object in R

#you may be asking what the heck is an object?
#find it in this list of R jargon
#https://link.springer.com/content/pdf/bbm:978-1-4419-1318-0/1.pdf

#our object will list the names of the attendes sitting next to us
#don't be shy about reminding your neighbors of your name and how to spell it

#to make this list we need to use the c() function
#learn what c() does by running
?c
```

Combine Values into a Vector or List

Description

This is a generic function which combines its arguments.

The default method combines its arguments to form a vector. All arguments are coerced to a common type which is the type of the returned value, and all attributes except names are removed.

Usage

```
## S3 Generic function
c(...)

## Default S3 method:
c(..., recursive = FALSE, use.names = TRUE)
```

Arguments

`...`	objects to be concatenated. All `NULL` entries are dropped before method dispatch unless at the very beginning of the argument list.

```
35
36  #now we came make the object that lists our neighbors names
37  names_list <- c("Angelica", "Jesse", "Jazlyn")
38
39
40  #see what happens when you run the object
41  #also notice what RStudio does when you type at least the first three letters
42  #of your objects' name
43  #this is an advantage of RStudio!
44
45  #also look at the box on the top right - can you find your object there?
46
47
```

- **Functions are always followed by ( )**
- **Inside the parentheses are your “arguments”**
- **Arguments are separated by commas**
- **The names are in “” because they are character strings not objects**

```
################
#now let's store this information in a matrix
#you can go back to the list of jargon to wee what is a matrix

#to do this, we'll need to use the matrix function
#pull up the help section for matrix like you did for the c function
```

Matrices

Description

`matrix` creates a matrix from the given set of values.

`as.matrix` attempts to turn its argument into a matrix.

`is.matrix` tests if its argument is a (strict) matrix.

Usage

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,
       dimnames = NULL)

as.matrix(x, ...)
## S3 method for class 'data.frame'
as.matrix(x, rownames.force = NA, ...)

is.matrix(x)
```

This is the default order and values of the arguments

Arguments

`data`	an optional data vector (including a list or `expression` vector). Non-atomic classed `R` objects are coerced by `as.vector` and all attributes discarded.
`nrow`	the desired number of rows.
`ncol`	the desired number of columns.
`byrow`	logical. If `FALSE` (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.
`dimnames`	A `dimnames` attribute for the matrix: `NULL` or a `list` of length 2 giving the row and column names respectively. An empty list is treated as `NULL`, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.
`x`	an `R` object.
`...`	additional arguments to be passed to or from methods.
`rownames.force`	logical indicating if the resulting matrix should have character (rather than `NULL`) `rownames`. The default, `NA`, uses `NULL` rownames if the data frame has 'automatic' row.names or for a zero-row data frame.

```r

#the matrix we're going to make will contain the names list from above
#and now where they are sitting relative to you

#make a object that lists everyone's seat position in the same order as the names list

position <- c("left", "right", "far right")

```

```
56
57  #the matrix we're going to make will contain the names list from above
58  #and now where they are sitting relative to you
59
60  #make a object that lists everyone's seat position in the same order as the names list
61
62  position <- c("left", "right", "far right")
63
64  #okay now we have the pieces to make our first matrix
65  names_matrix <- matrix(c(names_list, position), nrow = 3, ncol = 2)
66
```

You can nest a function within a function!

```
67
68  #call your matrix to see what it looks like
69
```

```
> names_matrix
     [,1]       [,2]
[1,] "Angelica" "left"
[2,] "Jesse"    "right"
[3,] "Jazlyn"   "far right"
```

```
71
72  #in this short time you've already learned how to
73    #start RStudio cloud
74    #look up R jargon
75    #run code from a file
76    #find help with any function in R
77    #use functions in R
78    #make objects including a matrix
79
80
```

Electronic Notebooks

Best practices for lab notebooks

Safekeeping	• Lab notebook belongs to the organization (NIH) • **Always** stays at the work place (make photocopies) • Record observations into the notebook as you are doing the work
Organization and Readability	• Bound notebook with numbered pages • Black ink (no pencil) – minimizes chances of erasures; easily scanned and photocopied • No erasing! One line cross-outs; initial the change, add note why you made that change
Quality of Record Keeping	• Provides sufficient details so work can be reproduced by others. • Allows experimental findings to be fully understood. • One day you or someone else will write your results up as a paper. Make it easy for them to find things.

“If it’s not written down, it didn’t happen.”

Your notebook must answer the questions below:

1) What was done?

2) How was it done?

3) When was the work performed?

4) Who performed the work?

This applies to *both* hardcopy and electronic notebooks

Electronic Notebooks

Problem:

It’s not realistic to put all data and results from a big data experiment into a bound notebook…

but a bound notebook is still required.

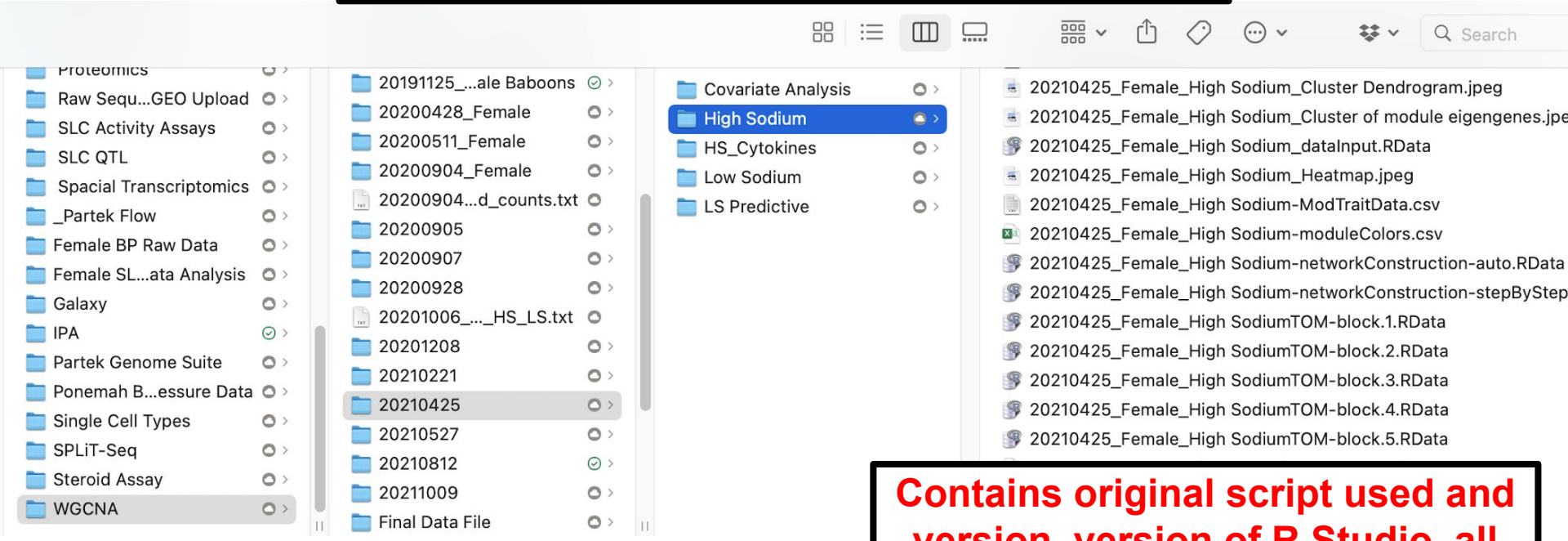
The solution:

An electronic notebook that is referenced and maintained within the bound notebook.

What does an electronic notebook look like?

- A project folder/directory that contains:
  1. summary project file; Excel is great for this!
  2. all data files
  3. raw data
  4. output files from analyses
- Files containing raw data must be backup and preserved to that it's always possible to go back to the raw data and analyze.

File naming scheme:
Date_Experiment_File Type



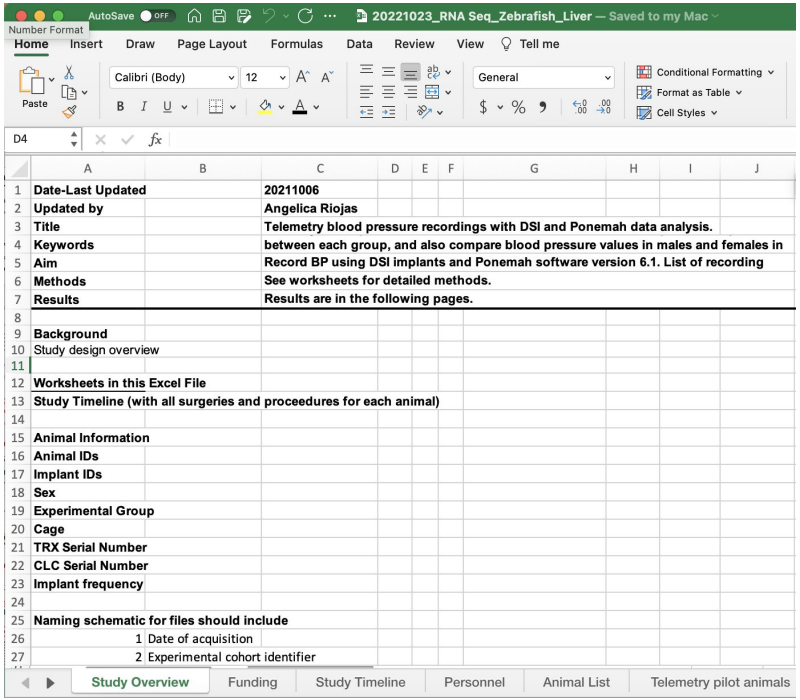
Each project file represents a new experiment run in R Studio on a given date

Contains original script used and version, version of R Studio, all input data files and all newly generated files

Summary project file in Excel

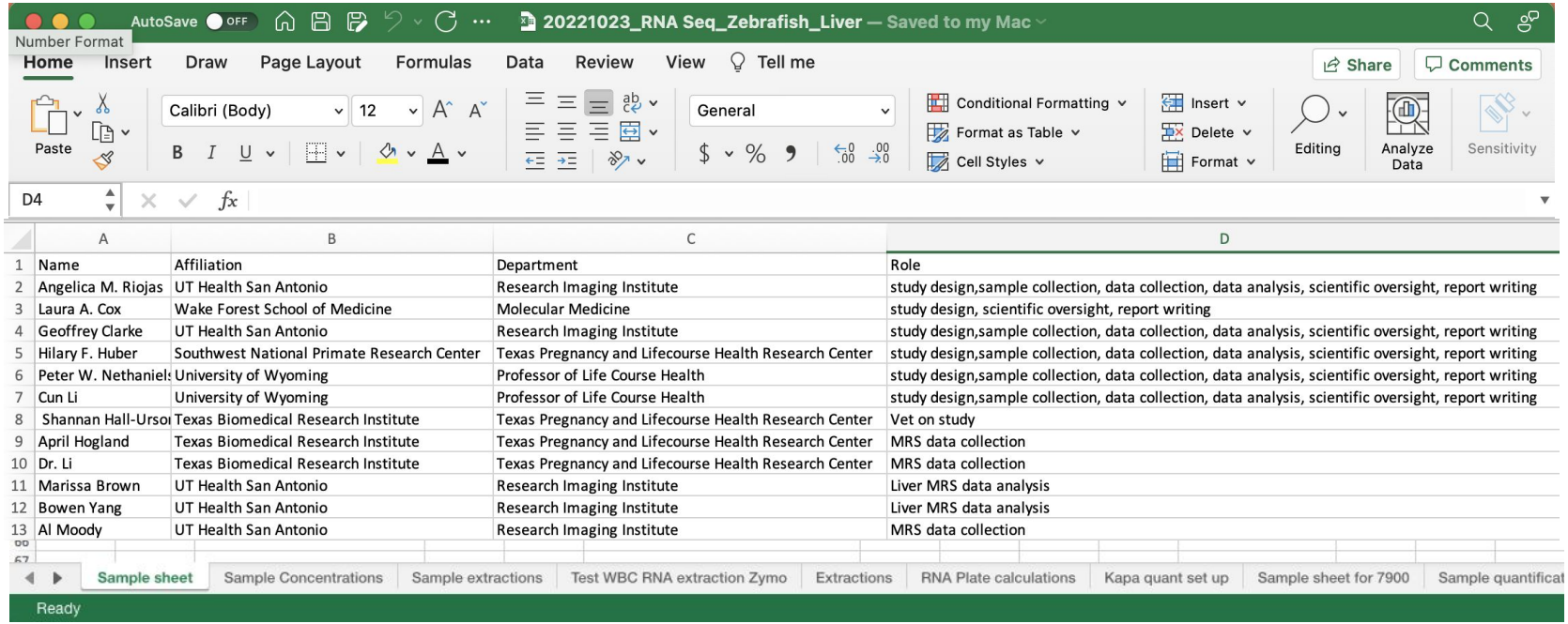
File naming scheme:
Date_Experiment Type_Species_Tissue

- Date-Last Updated
- Updated by
- Title
- Keywords
- Aim
- Methods
- Results



	A	B	C
1	**Date-Last Updated**		**20211006**
2	**Updated by**		**Angelica Riojas**
3	**Title**		**Telemetry blood pressure recordings with DSI and Ponemah data analysis.**
4	**Keywords**		**between each group, and also compare blood pressure values in males and females in**
5	**Aim**		**Record BP using DSI implants and Ponemah software version 6.1. List of recording**
6	**Methods**		**See worksheets for detailed methods.**
7	**Results**		**Results are in the following pages.**
8			
9	**Background**		
10	Study design overview		
11			
12	**Worksheets in this Excel File**		
13	**Study Timeline (with all surgeries and proceedures for each animal)**		
14			
15	**Animal Information**		
16	**Animal IDs**		
17	**Implant IDs**		
18	**Sex**		
19	**Experimental Group**		
20	**Cage**		
21	**TRX Serial Number**		
22	**CLC Serial Number**		
23	**Implant frequency**		
24			
25	**Naming schematic for files should include**		
26	1	Date of acquisition	
27	2	Experimental cohort identifier	

Summary project file in Excel

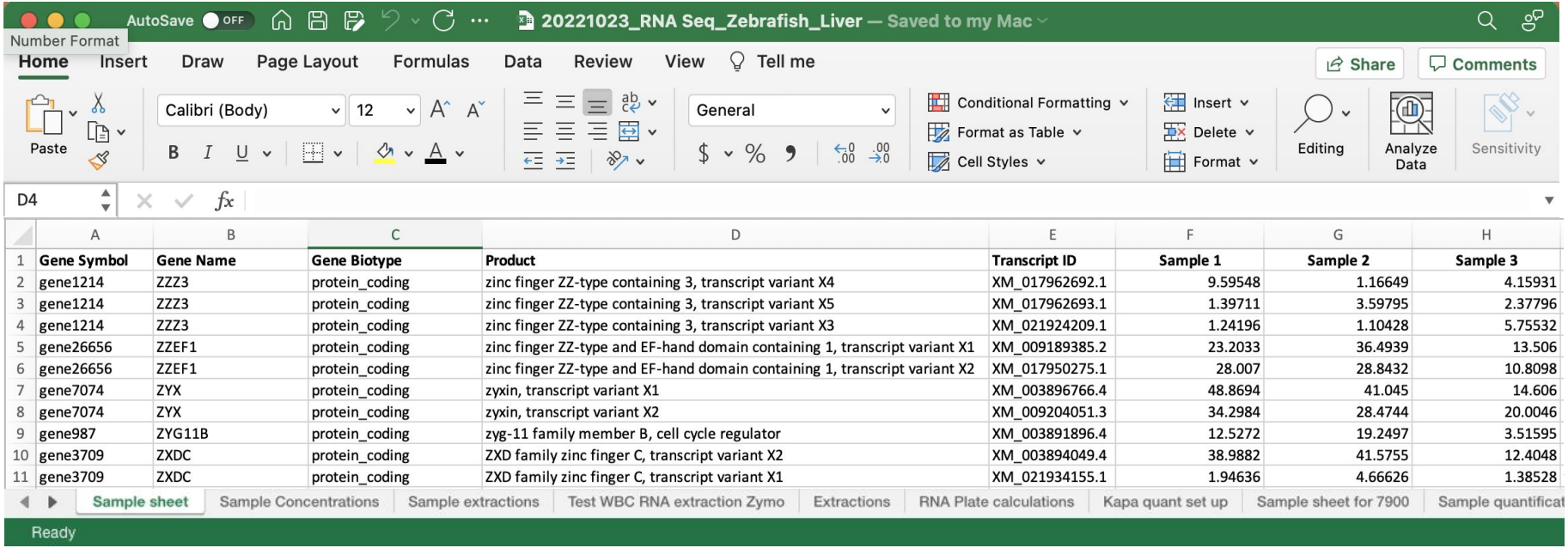


	A	B	C	D
1	Name	Affiliation	Department	Role
2	Angelica M. Riojas	UT Health San Antonio	Research Imaging Institute	study design,sample collection, data collection, data analysis, scientific oversight, report writing
3	Laura A. Cox	Wake Forest School of Medicine	Molecular Medicine	study design, scientific oversight, report writing
4	Geoffrey Clarke	UT Health San Antonio	Research Imaging Institute	study design,sample collection, data collection, data analysis, scientific oversight, report writing
5	Hilary F. Huber	Southwest National Primate Research Center	Texas Pregnancy and Lifecourse Health Research Center	study design,sample collection, data collection, data analysis, scientific oversight, report writing
6	Peter W. Nethaniel	University of Wyoming	Professor of Life Course Health	study design,sample collection, data collection, data analysis, scientific oversight, report writing
7	Cun Li	University of Wyoming	Professor of Life Course Health	study design,sample collection, data collection, data analysis, scientific oversight, report writing
8	Shannan Hall-Urso	Texas Biomedical Research Institute	Texas Pregnancy and Lifecourse Health Research Center	Vet on study
9	April Hogland	Texas Biomedical Research Institute	Texas Pregnancy and Lifecourse Health Research Center	MRS data collection
10	Dr. Li	Texas Biomedical Research Institute	Texas Pregnancy and Lifecourse Health Research Center	MRS data collection
11	Marissa Brown	UT Health San Antonio	Research Imaging Institute	Liver MRS data analysis
12	Bowen Yang	UT Health San Antonio	Research Imaging Institute	Liver MRS data analysis
13	Al Moody	UT Health San Antonio	Research Imaging Institute	MRS data collection

Define authorship early and keep track of who has been involved in a study. People move, graduate ect.

Summary project file in Excel

Useful for raw data, methods, and results pages



	A	B	C	D	E	F	G	H
1	**Gene Symbol**	**Gene Name**	**Gene Biotype**	**Product**	**Transcript ID**	**Sample 1**	**Sample 2**	**Sample 3**
2	gene1214	ZZZ3	protein_coding	zinc finger ZZ-type containing 3, transcript variant X4	XM_017962692.1	9.59548	1.16649	4.15931
3	gene1214	ZZZ3	protein_coding	zinc finger ZZ-type containing 3, transcript variant X5	XM_017962693.1	1.39711	3.59795	2.37796
4	gene1214	ZZZ3	protein_coding	zinc finger ZZ-type containing 3, transcript variant X3	XM_021924209.1	1.24196	1.10428	5.75532
5	gene26656	ZZEF1	protein_coding	zinc finger ZZ-type and EF-hand domain containing 1, transcript variant X1	XM_009189385.2	23.2033	36.4939	13.506
6	gene26656	ZZEF1	protein_coding	zinc finger ZZ-type and EF-hand domain containing 1, transcript variant X2	XM_017950275.1	28.007	28.8432	10.8098
7	gene7074	ZYX	protein_coding	zyxin, transcript variant X1	XM_003896766.4	48.8694	41.045	14.606
8	gene7074	ZYX	protein_coding	zyxin, transcript variant X2	XM_009204051.3	34.2984	28.4744	20.0046
9	gene987	ZYG11B	protein_coding	zyg-11 family member B, cell cycle regulator	XM_003891896.4	12.5272	19.2497	3.51595
10	gene3709	ZXDC	protein_coding	ZXD family zinc finger C, transcript variant X2	XM_003894049.4	38.9882	41.5755	12.4048
11	gene3709	ZXDC	protein_coding	ZXD family zinc finger C, transcript variant X1	XM_021934155.1	1.94636	4.66626	1.38528

Each page represents a new experiment, and be sequentially ordered.

Running a code/script is an experiment

1. What parameters were used & how do they impact your results?
    a. Quality filters
    b. Stringency filters
    c. Positive controls
    d. Negative controls
2. Default parameters
    a. Defined by others…
    b. What are they and how do they impact your results?

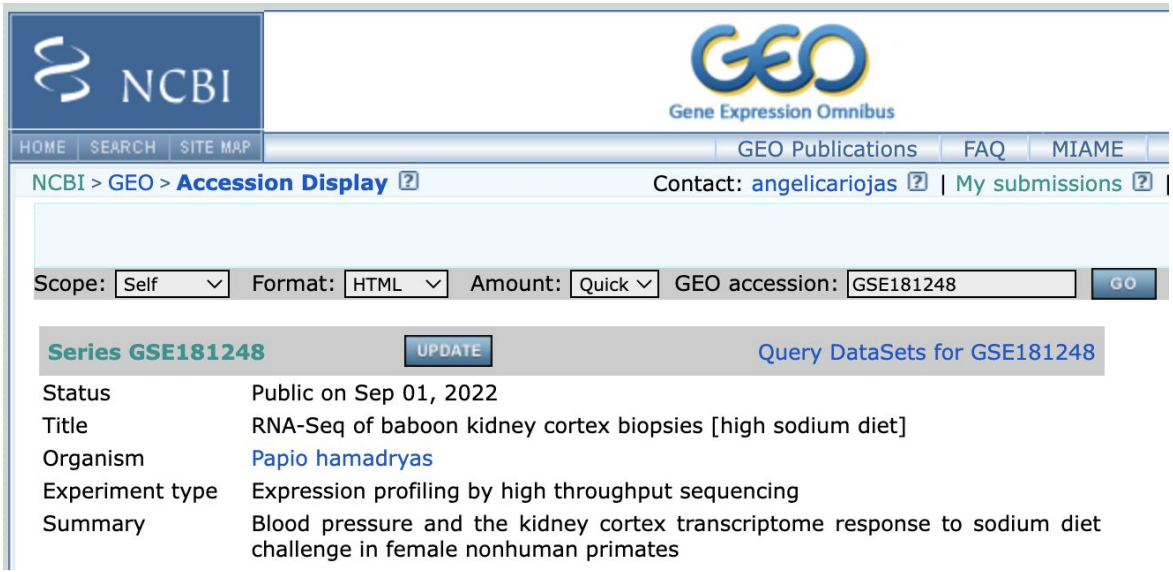
Version Control

- It is common for big datasets to be analyzed by multiple scientists.
- For any software or script used for data analysis, the version of the software used must be recorded.
- Include dates for experiments, analysis, or data downloads.

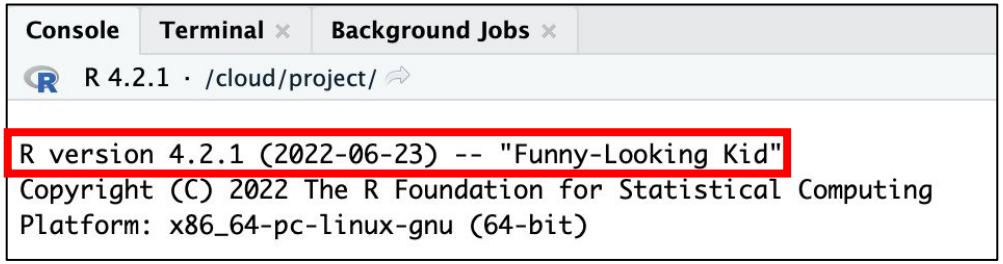
Publishing large datasets

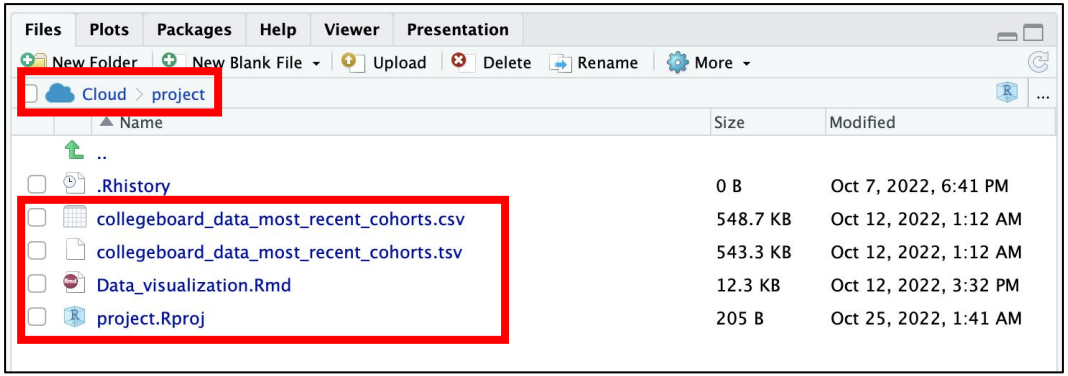
1. Sequence data & gene array data – requires depositing raw and processed data in an NIH database for public access.

2. Not all data types have a standardized format



RStudio Cloud



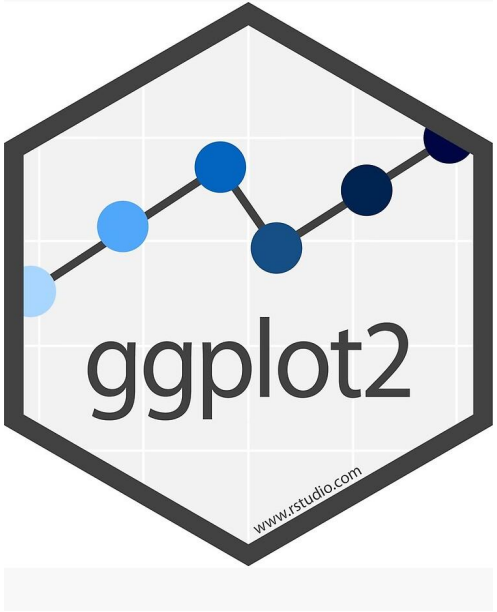


Ex. file naming scheme:
Date_collegeboard_file description

Visualizing your data with ggplot2

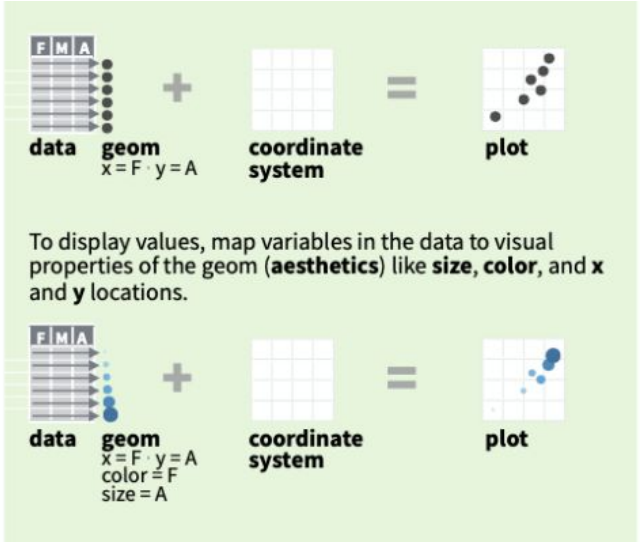
ggplot2 and the Grammar of Graphics

Originally developed by Leland Wilkinson, the Grammar of Graphics was adapted by Hadley Wickham for the R package ggplot2



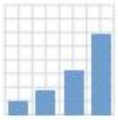
Every graph can be made with three things

- A **data** set
- A **coordinate system**
- A "**Geom**" (Visual marks that represent data)

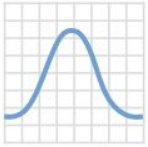


(Rstudio Cheat Sheets)

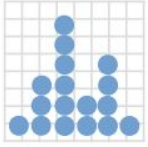
geom_* Examples



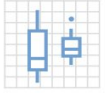
d + geom_bar()
x, alpha, color, fill, linetype, size, weight



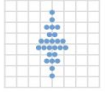
c + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, group, linetype, size, weight



c + geom_dotplot()
x, y, alpha, color, fill

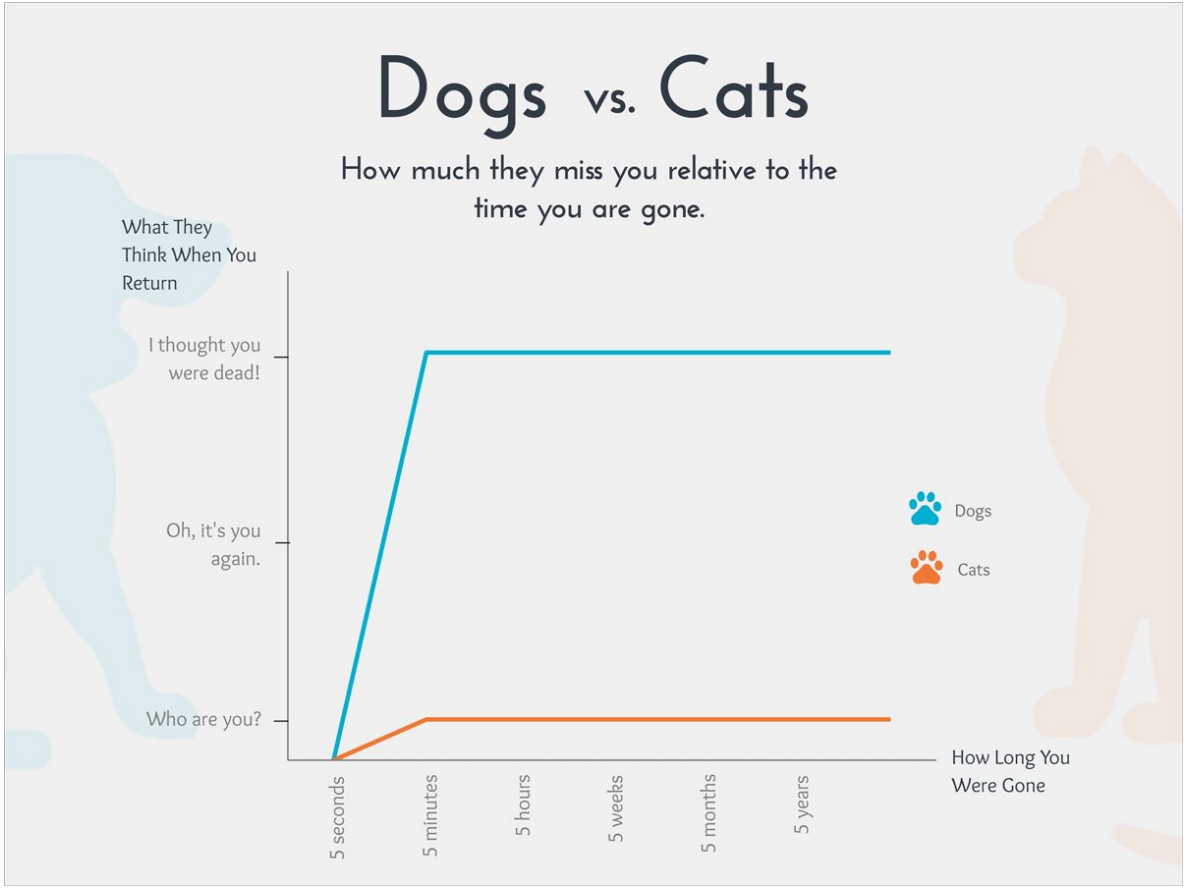


f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight



f + geom_dotplot(binaxis = "y", stackdir = "center"**)**, x, y, alpha, color, fill, group

f + geom_violin(scale = "area"**)**, x, y, alpha, color, fill, group, linetype, size, weight



https://visme.co/blog/funny-graphs/

Read the data in

```
# college_board_data<-read_tsv(file = "collegeboard_data_most_recent_cohorts.tsv")
college_board_data <- read_csv(file = "collegeboard_data_most_recent_cohorts.csv")
```

Examining the data

```
college_board_data
```

```
## # A tibble: 6,681 x 7
##    institution_name funding longitude latitude percent_of_stud… state
##    <chr>            <chr>       <dbl>    <dbl>           <dbl> <chr>
##  1 Alabama A & M U… Public      -86.6     34.8            70.9 AL
##  2 University of A… Public      -86.8     33.5            34.0 AL
##  3 Amridge Univers… Privat…     -86.2     32.4            74.5 AL
##  4 University of A… Public      -86.6     34.7            24.0 AL
##  5 Alabama State U… Public      -86.3     32.4            73.7 AL
##  6 The University … Public      -87.5     33.2            17.2 AL
##  7 Central Alabama… Public      -85.9     32.9            38.2 AL
##  8 Athens State Un… Public      -87.0     34.8            43.3 AL
##  9 Auburn Universi… Public      -86.2     32.4            46.5 AL
## 10 Auburn Universi… Public      -85.5     32.6            13.4 AL
## # … with 6,671 more rows, and 1 more variable: admission_rate <dbl>
```

Checking column names

```
colnames(college_board_data)
```

```
## [1] "institution_name"
## [2] "funding"
## [3] "longitude"
## [4] "latitude"
## [5] "percent_of_students_with_pell_grants"
## [6] "state"
## [7] "admission_rate"
```

What’s the question?

What's the question?

Do private, public and for profit schools have the same percentage of Pell Grant recipients?

"Federal Pell Grants usually are awarded only to undergraduate students who display exceptional financial need" (Student.aid.ed.gov)

What's the question?

Do **private, public and for profit** schools have the same percentage of Pell Grant recipients?

"Federal Pell Grants usually are awarded only to undergraduate students who display exceptional financial need" (Student.aid.ed.gov)

What's the question?

Do private, public and for profit schools have the same **percentage of Pell Grant recipients**?

"Federal Pell Grants usually are awarded only to undergraduate students who display exceptional financial need" (Student.aid.ed.gov)

What’s your coordinate system?

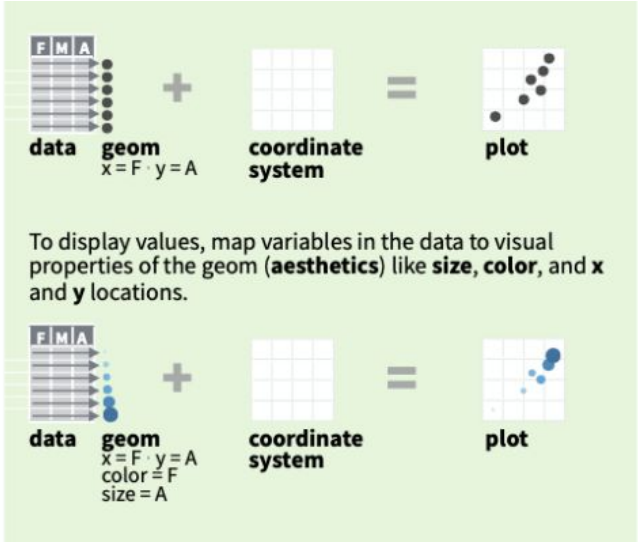
- X axis?
  - School funding type
    - This is the “funding” variable
- Y axis?
  - Percentage of undergraduates who receive Pell Grant aid
    - This is the “percent_of_students_with_pell_grants” variable

Loading data into ggplot()

```
ggplot(data = college_board_data)
```

Every graph can be made with three things

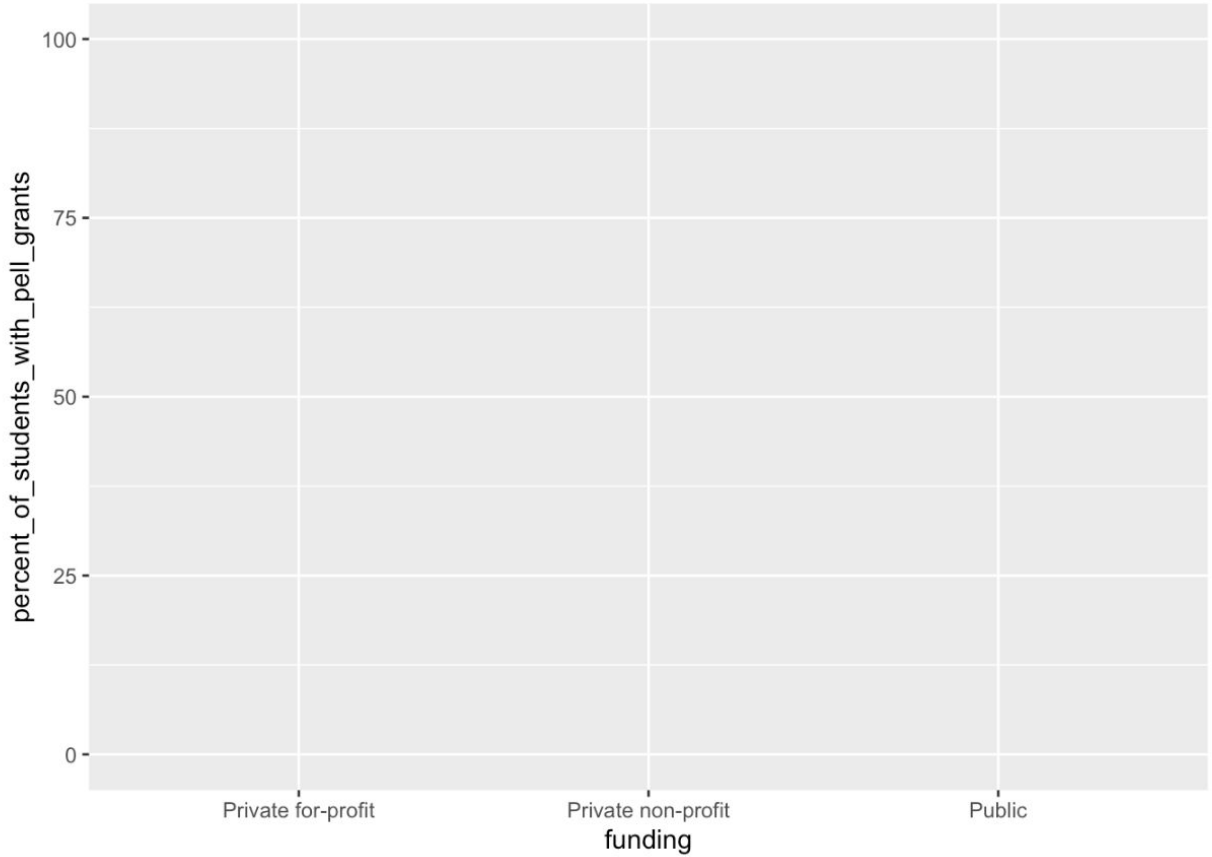
- A **data** set ✅
- A **coordinate system**
- A "**Geom**" (Visual marks that represent data)



(Rstudio Cheat Sheets)

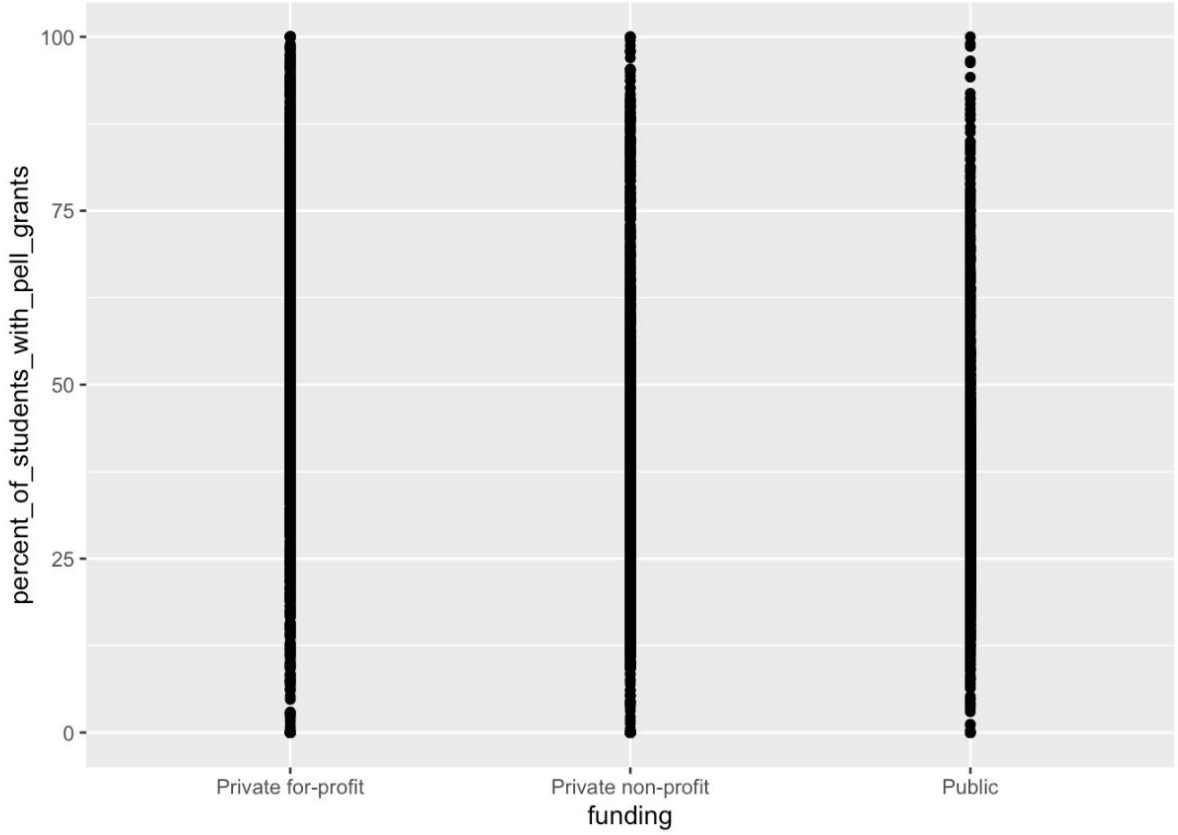
Setting up coordinate system

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants))
```



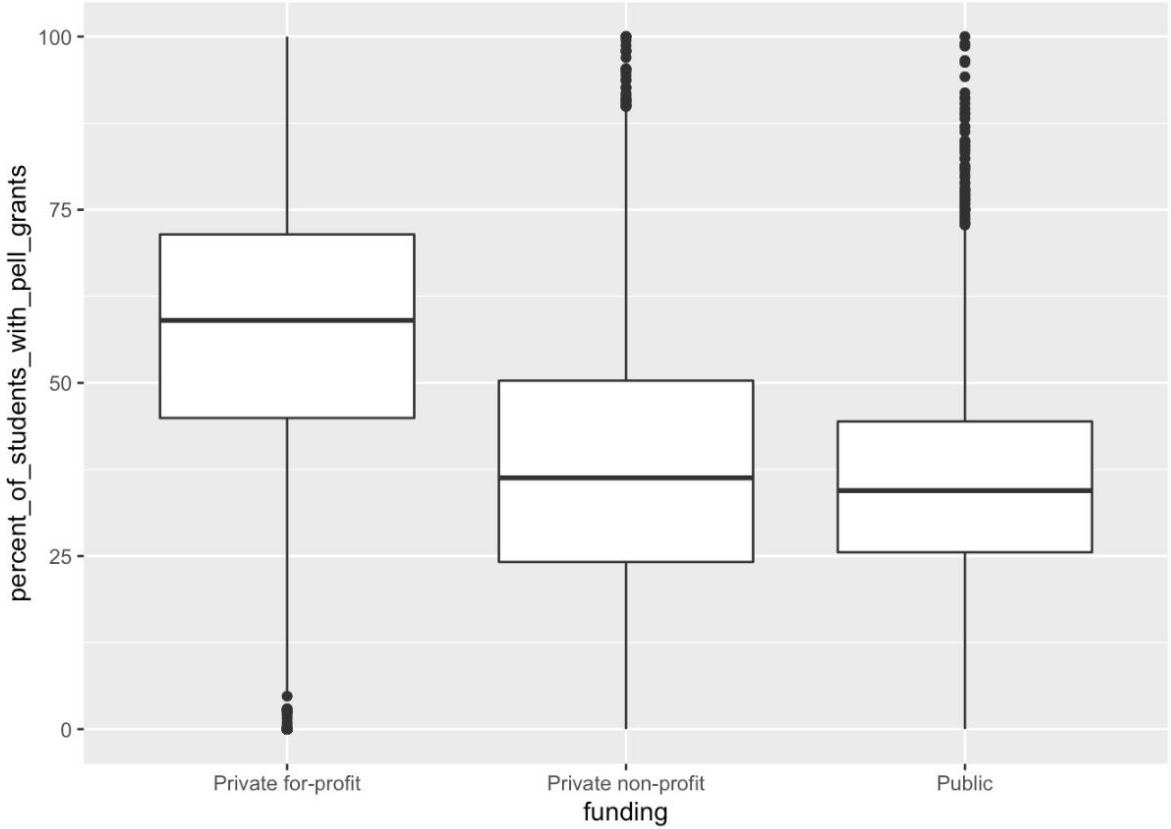
Making a scatter plot

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants)) +
  geom_point()
```



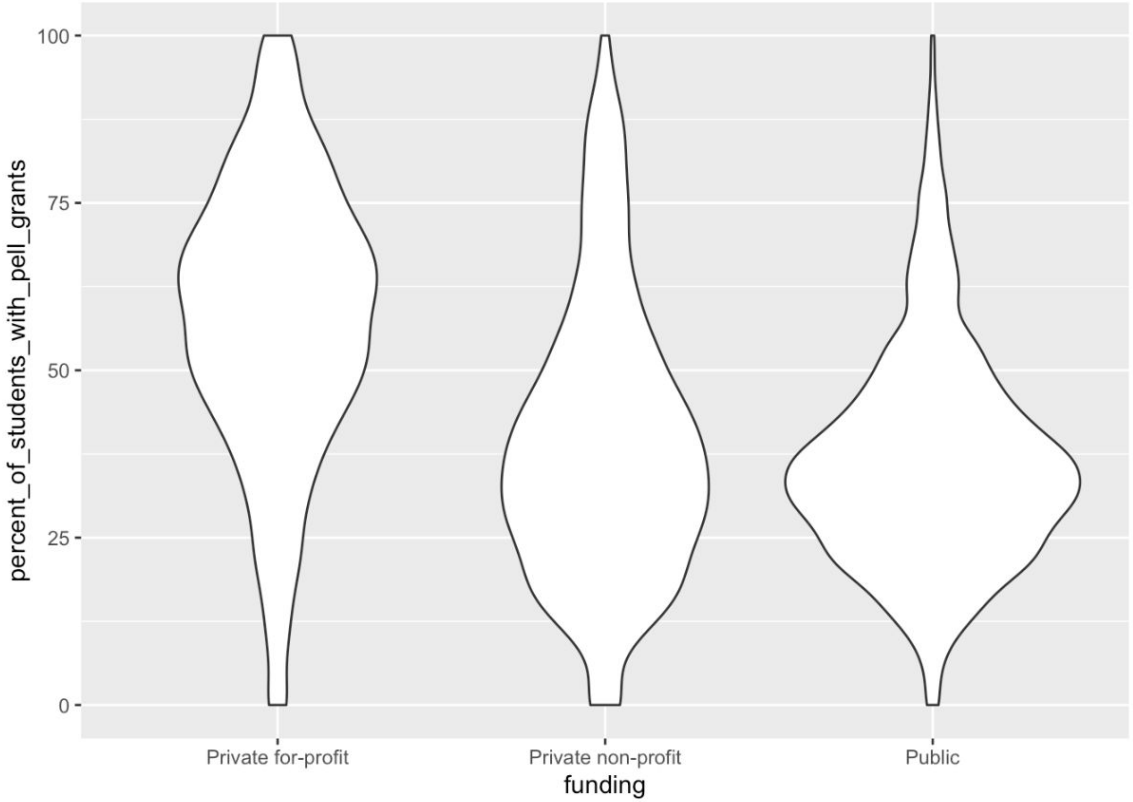
Making a boxplot

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants)) +
  geom_boxplot()
```



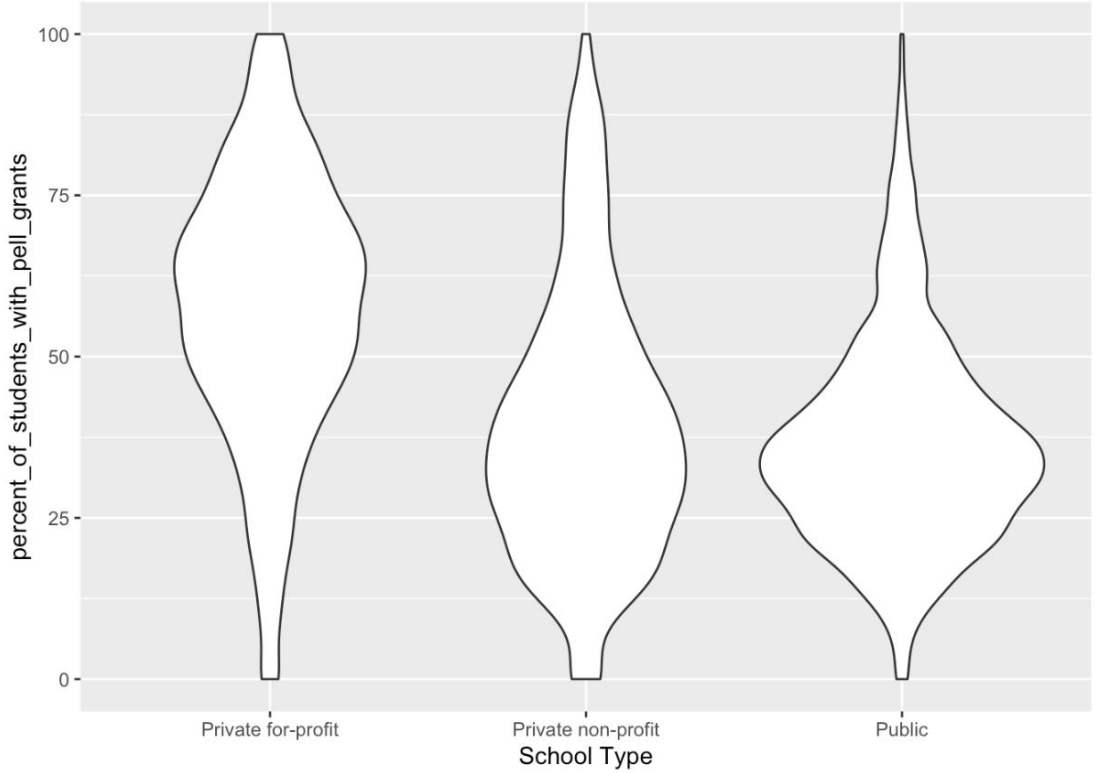
Making violin plot

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants)) +
  geom_violin()
```



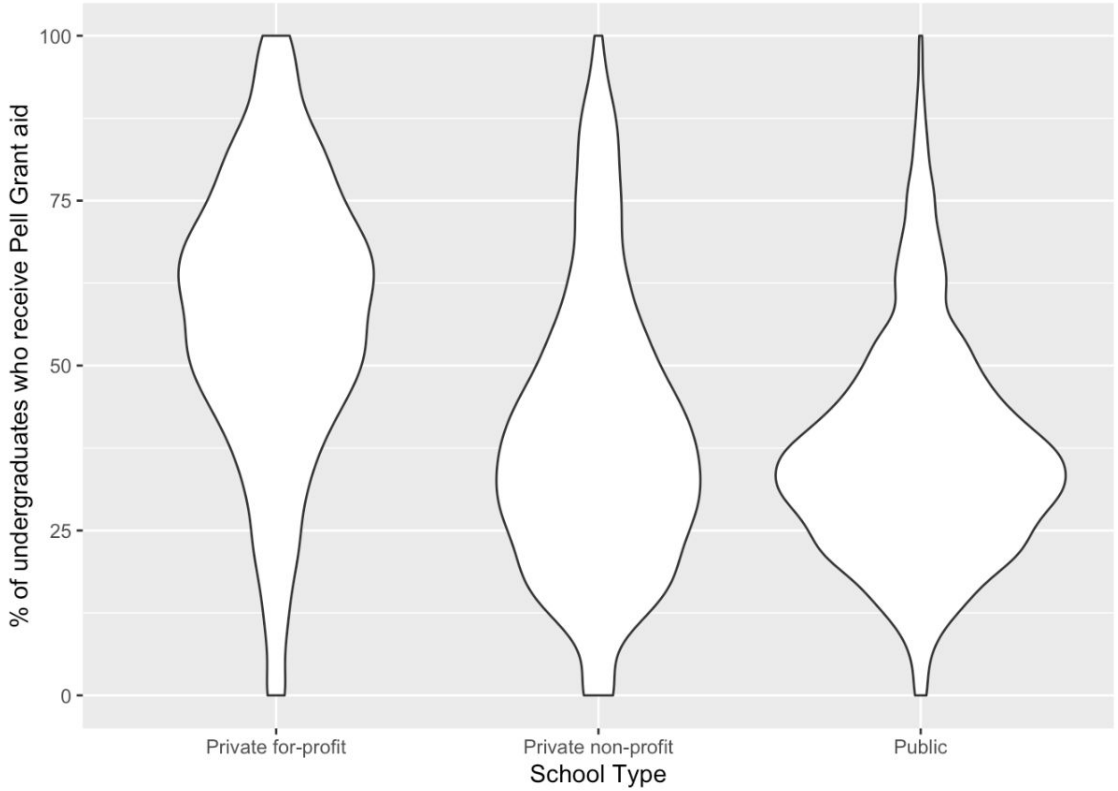
Labeling X axis

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants)) +
  geom_violin() +
  labs(x="School Type")
```



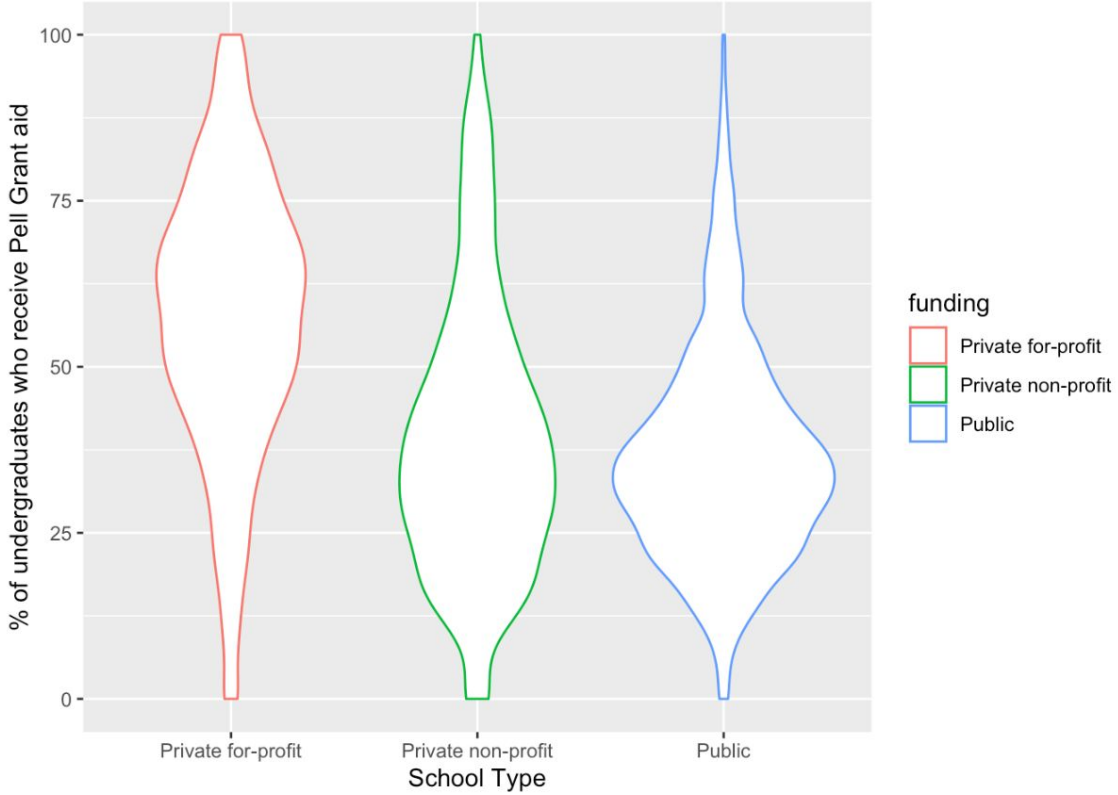
Labeling Y axis

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid")
```



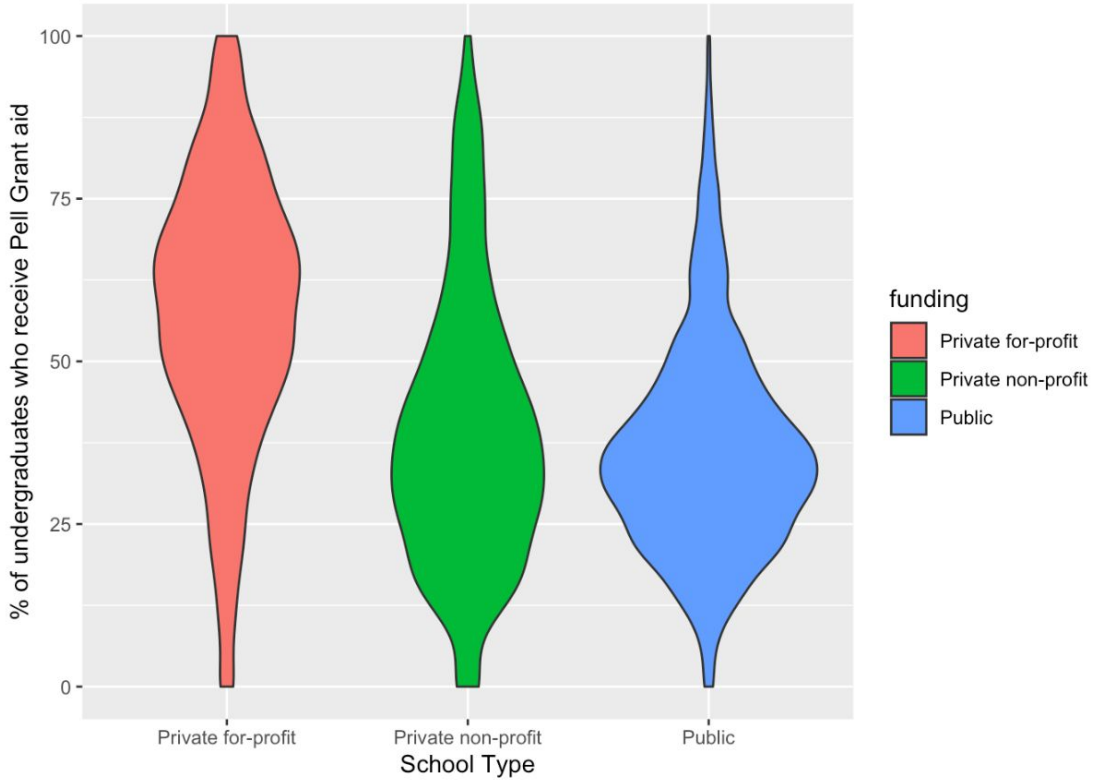
Using the color aesthetic

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, color=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid")
```



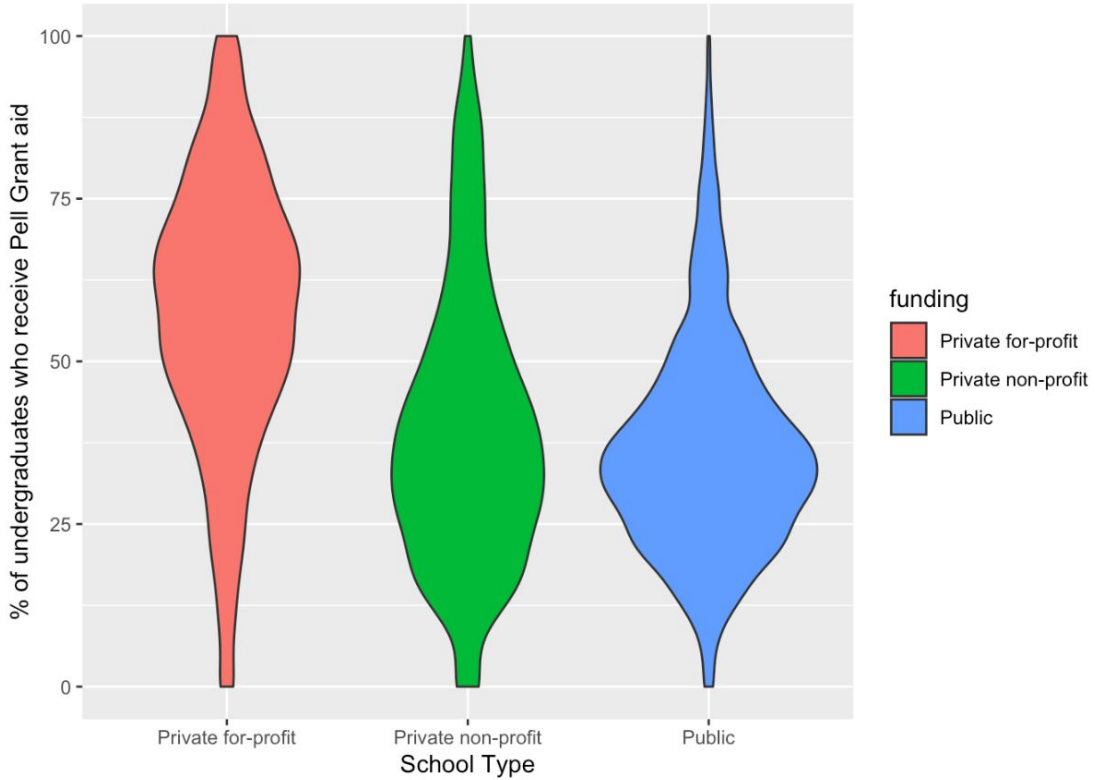
Using the fill aesthetic

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid")
```



Using the fill aesthetic

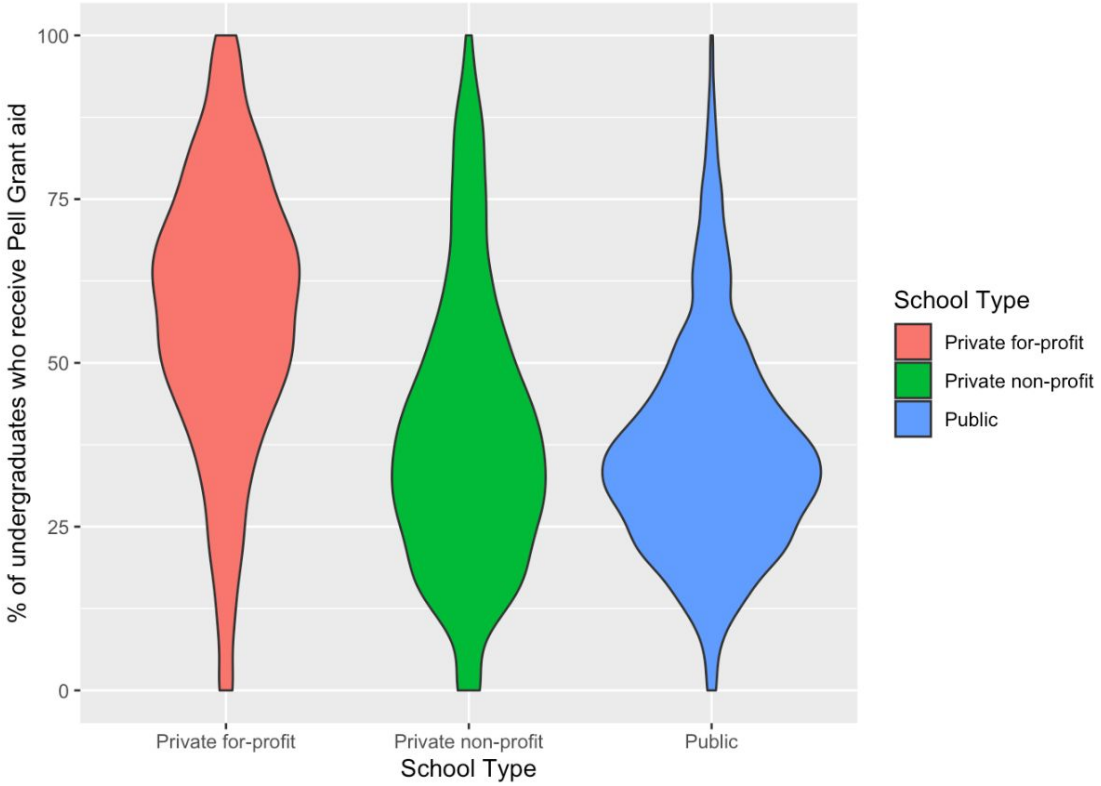
```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid")
```



How could these color choices be more

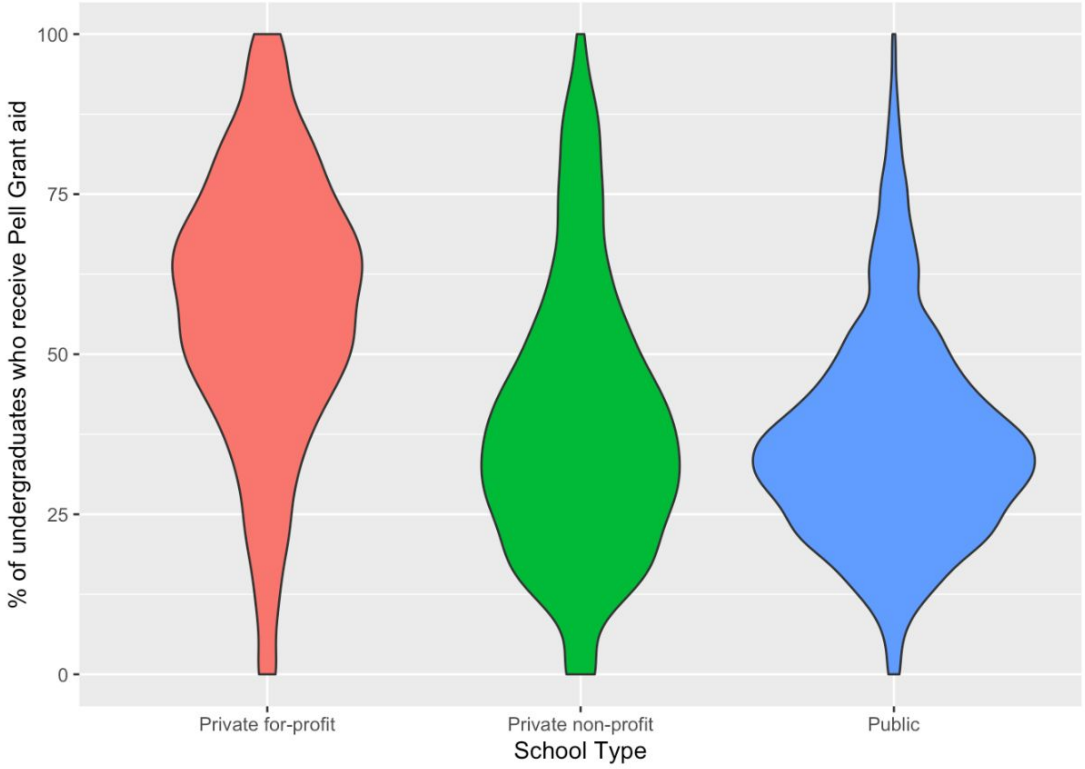
Changing legend label using labs()

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type")
```



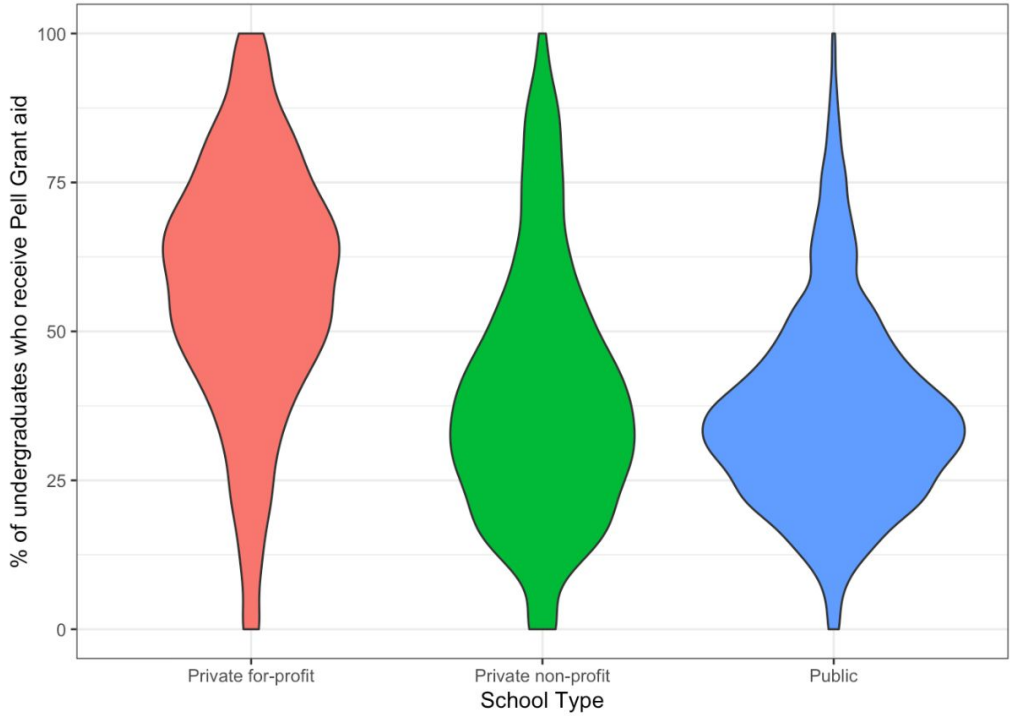
Removing the plot's legend with "guides()"

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
  guides(fill=FALSE)
```



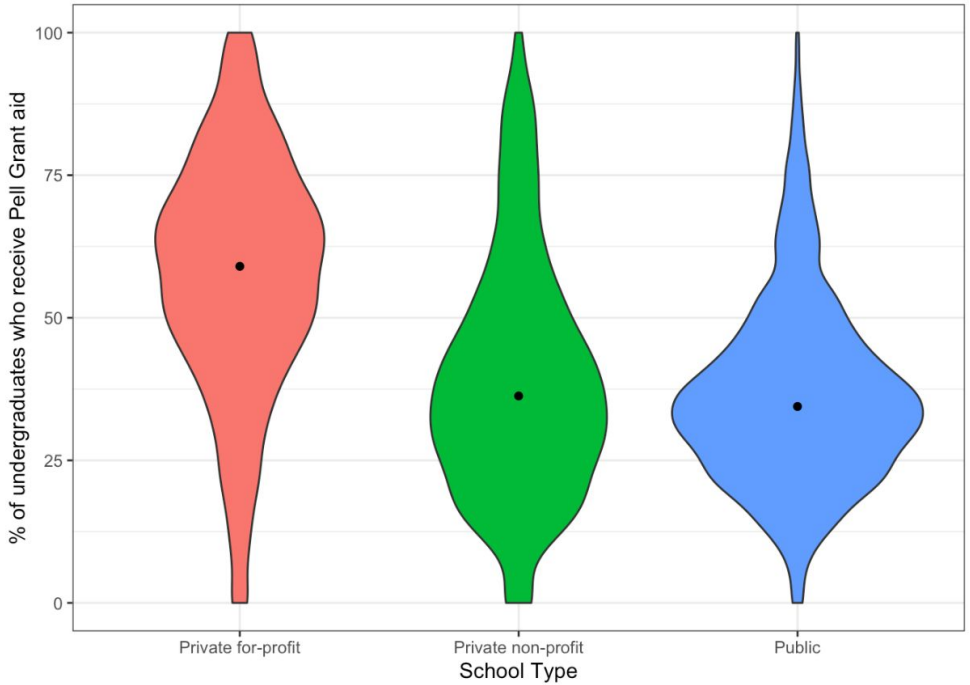
Removing grey background

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
  guides(fill=FALSE) +
  theme_bw()
```



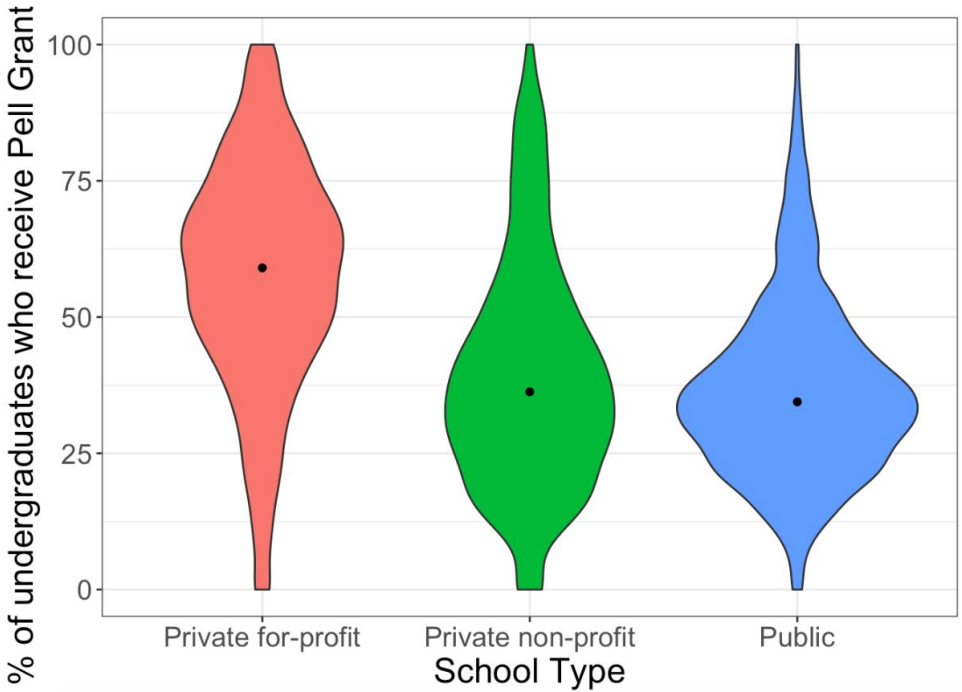
Adding median of distribution as points

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
  guides(fill=FALSE) +
  theme_bw() +
  stat_summary(fun.y="median", geom="point")
```



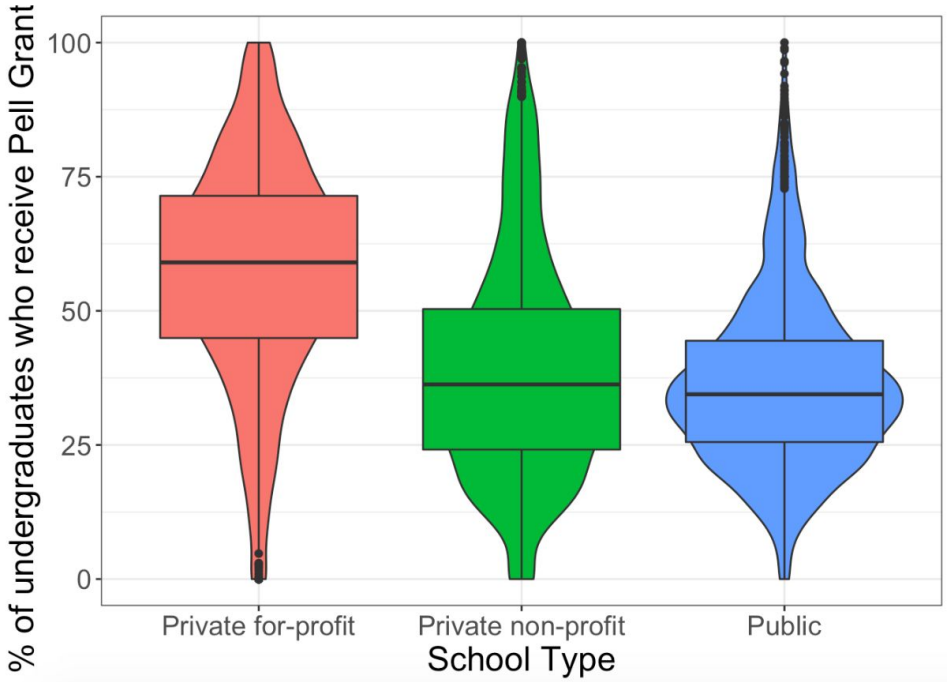
Increasing font size

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
  guides(fill=FALSE) +
  theme_bw() +
  stat_summary(fun.y="median", geom="point") +
  theme(text=element_text(size=18))
```



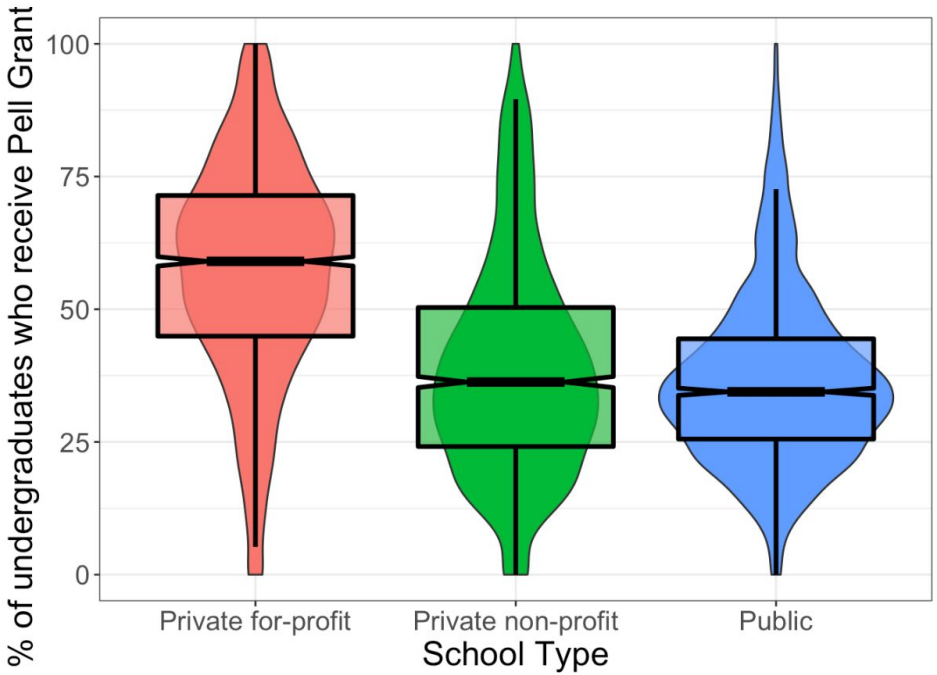
Adding boxplot on top

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
  guides(fill=FALSE) +
  theme_bw() +
  stat_summary(fun.y="median", geom="point") +
  theme(text=element_text(size=18)) +
 geom_boxplot()
```



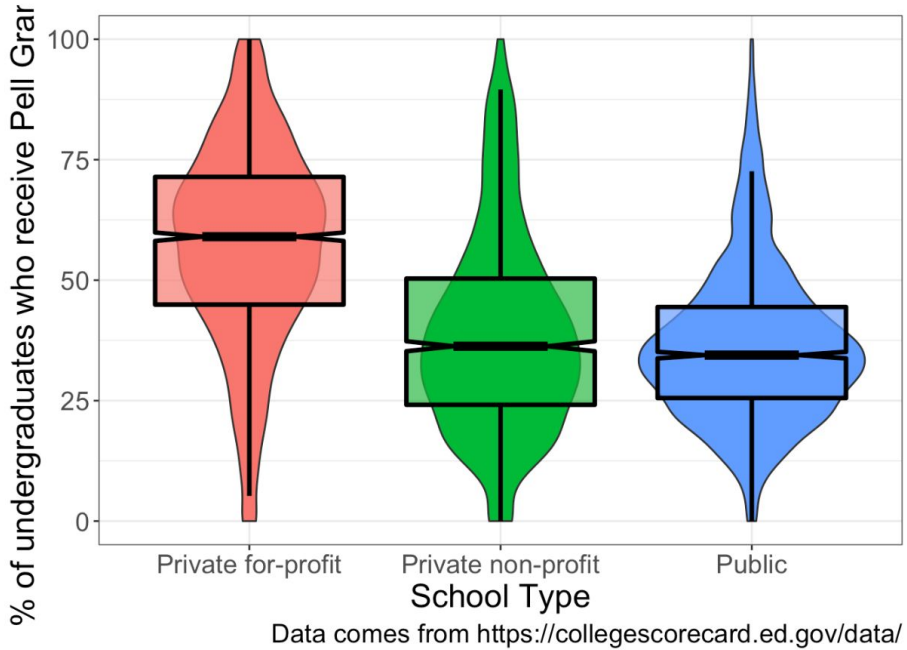
Stylizing boxplot

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
  geom_violin() +
  labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
  guides(fill=FALSE) +
  theme_bw() +
  stat_summary(fun.y="median", geom="point") +
  theme(text=element_text(size=18)) +
 geom_boxplot(notch = TRUE,  outlier.size = -1, color="black",lwd = 1.2,  alpha = 0.7)
```



Adding sources

```
ggplot(data=college_board_data, mapping = aes(x=funding, y=percent_of_students_with_pell_grants, fill=funding)) +
 geom_violin() +
 labs(x="School Type", y="% of undergraduates who receive Pell Grant aid", fill="School Type") +
 guides(fill=FALSE) +
 theme_bw() +
 stat_summary(fun.y="median", geom="point") +
 theme(text=element_text(size=18)) +
geom_boxplot(notch = TRUE,  outlier.size = -1, color="black",lwd = 1.2,  alpha = 0.7) +
labs(caption = "Data comes from https://collegescorecard.ed.gov/data/")
```



Patterns and Trends.

Exploring your data

- Now that you have learned how to make some simple plots with ggplot, you can begin to explore you data

Exploring your data

- Now that you have learned how to make some simple plots with ggplot, you can begin to explore you data
- It is important to explore/visualize your data before you apply any sort of statistical analyses

Exploring your data

- Now that you have learned how to make some simple plots with ggplot, you can begin to explore you data
- It is important to explore/visualize your data before you apply any sort of statistical analyses
- Some nice functions you can use for quick data exploration
  - summary()
  - cor()
  - table()

How I explore data, a (brief) introduction to dplyr

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

How I explore data, a (brief) introduction to dplyr

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

How I explore data, a (brief) introduction to dplyr

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

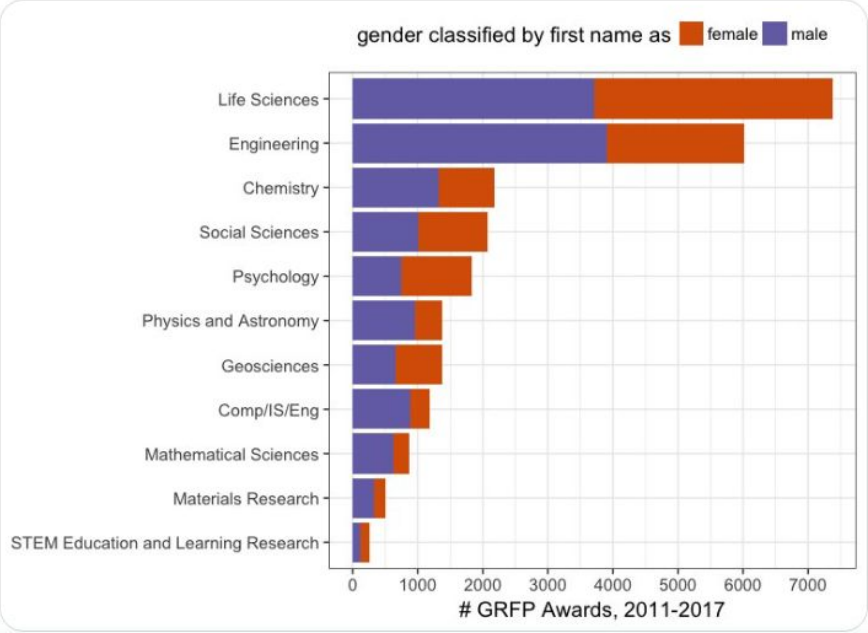
- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

These all combine naturally with `group_by()` which allows you to perform any operation “by group”.

Exploring previous NSF GRFP data

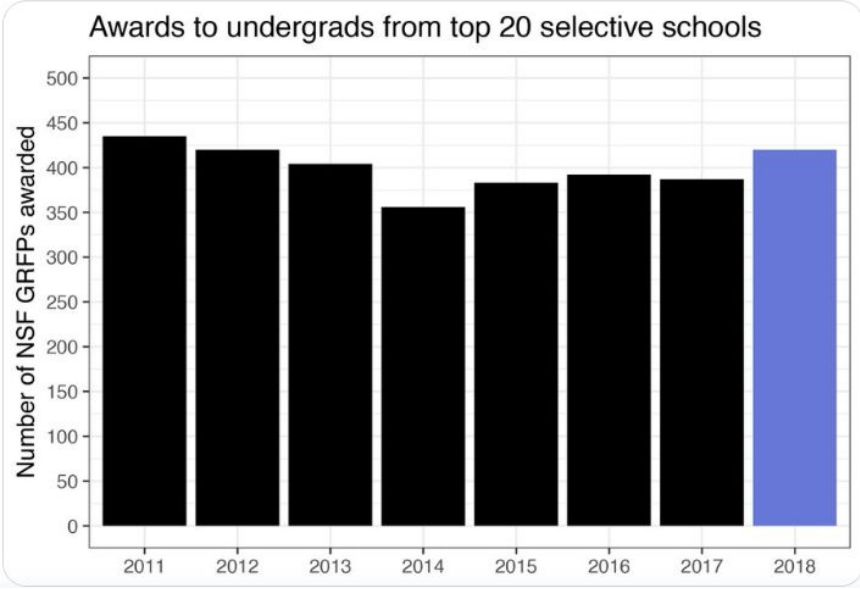
@NatalieTelis

Want to look at the demographics of the #GRFP? I do!! I classify all award winners by first name. First off: life science and engineering far dominate the awards year to year. Also, though the NSF overall is near 50/50 MF, big subfield variance... sound familiar?



@NatalieTelis

Couldn't resist the @NSF #GRFP recipient data. Where do they come from? And how has that changed since this Dear Colleague Letter? nsf.gov/pubs/2016/nsf1... Check out this plot, read my blog post, telis.blog/2018/04/03/the... or go straight to the shiny app: nsf-grfp.shinyapps.io/shiny/ 1/n



Manipulating our NSF data

```
#Make data frames
#read in NSF data and count awards per school per year
NSFData = read_csv("~/SACNAS/all_nsf_hon_and_reg_clean.csv")
```

	last_name	first_names	bs_school	field	curr_school	award_type
1	Dekarske	Madeline M	Agnes Scott College	Chemistry - Chemical Catalysis	Agnes Scott College	2017_hon
2	Hutcheson	Melissa Anne	Agnes Scott College	Physics and Astronomy - Particle Physics	Agnes Scott College	2015_reg
3	Brown	Erin	Allegheny College	Mathematical Sciences - Computational and D...	Allegheny College	2015_hon
4	Cusanno	Brianna	Allegheny College	Social Sciences - Communications	Allegheny College	2017_reg
5	Calamari	Zachary Thomas	University of Michigan Ann Arbor	Geosciences - Paleontology and Paleobiology	American Museum Natural History	2015_hon
6	Ingala	Melissa Robin	Fordham University	Life Sciences - Ecology	American Museum Natural History	2017_hon
7	Amarante	Linda M.	Long Island University C W Post Center	Life Sciences - Neurosciences	AMERICAN UNIVERSITY	2015_hon
8	Horin	Adam Patrick	Purdue University	Psychology - Developmental Psychology	AMERICAN UNIVERSITY	2017_hon
9	Amarante	Linda M.	Long Island University C W Post Center	Life Sciences - Neurosciences	AMERICAN UNIVERSITY	2016_reg
10	Hamel	Brian Thomas	American University	Social Sciences - Political Science	AMERICAN UNIVERSITY	2016_reg
11	Moncrieff	Andre Eugene	Andrews University	Life Sciences - Ecology	Andrews University	2015_reg
12	Hoffman	Devin	Appalachian State University	Geosciences - Paleontology and Paleobiology	Appalachian State University	2017_reg
13	Anderson	Alyssa Jordan	Middlebury College	Geosciences - Geochemistry	Arizona State University	2015_hon
14	Anglin	Julia Mae	Arizona State University	Life Sciences - Neurosciences	Arizona State University	2015_hon
15	Atwater	Chloe Elizabeth	University of California-Davis	Social Sciences - Archaeology	Arizona State University	2015_hon
16	Bookman	Rebecca Mary	Texas Christian Univeristy	Social Sciences - Archaeology	Arizona State University	2015_hon

Manipulating our NSF data

```
#Make data frames
#read in NSF data and count awards per school per year
NSFData = read_csv("~/SACNAS/all_nsf_hon_and_reg_clean.csv") %>%
  group_by(bs_school, award_type) %>%
  count() %>%
  ungroup() %>%
  mutate(year = gsub('\\D+','', award_type)) %>% #make a column with just year
```

```
> head(NSFData)
                bs_school award_type n year
1 Abia State University   2016_reg 1 2016
2   Agnes Scott College   2015_reg 2 2015
3   Agnes Scott College   2017_hon 1 2017
4        Albion College   2017_hon 1 2017
5      Albright College   2015_reg 1 2015
6     Alfred University   2016_hon 1 2016
```

Manipulating our NSF data

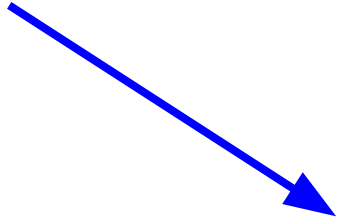
```
#count number of hon/awarded applications per year
perYear = NSFData %>%
  group_by(year) %>%
  count()
```

```
> head(NSFData)
                bs_school award_type n year
1 Abia State University    2016_reg 1 2016
2   Agnes Scott College    2015_reg 2 2015
3   Agnes Scott College    2017_hon 1 2017
4         Albion College    2017_hon 1 2017
5      Albright College    2015_reg 1 2015
6      Alfred University    2016_hon 1 2016
```

```
> head(perYear)
# A tibble: 3 x 2
# Groups:   year [3]
  year      n
  <chr> <int>
1 2015    924
2 2016   1013
3 2017    876
```

```r
#make counts proportions
countApps = NSFData %>%
  group_by(award_type, year) %>%
  count() %>%
```

```
> head(NSFData)
              bs_school award_type n year
1 Abia State University   2016_reg 1 2016
2   Agnes Scott College   2015_reg 2 2015
3   Agnes Scott College   2017_hon 1 2017
4        Albion College   2017_hon 1 2017
5     Albright College   2015_reg 1 2015
6     Alfred University   2016_hon 1 2016
```

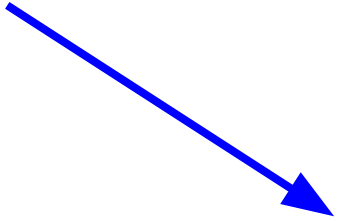


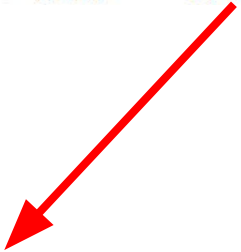
```
> head(countApps)
# A tibble: 6 x 4
# Groups:   award_type, year [6]
  award_type year      n
  <chr>      <chr> <int>
1 2015_hon   2015    453
2 2015_reg   2015    471
3 2016_hon   2016    533
4 2016_reg   2016    480
5 2017_hon   2017    413
6 2017_reg   2017    463
```

```r
#make counts proportions
countApps = NSFData %>%
  group_by(award_type, year) %>%
  count() %>%
  mutate(propAwards = n/perYear$n[match(year, perYear$year)])
```

```
> head(NSFData)
                bs_school award_type n year
1 Abia State University   2016_reg 1 2016
2   Agnes Scott College   2015_reg 2 2015
3   Agnes Scott College   2017_hon 1 2017
4        Albion College   2017_hon 1 2017
5     Albright College   2015_reg 1 2015
6     Alfred University   2016_hon 1 2016
```

```
> head(perYear)
# A tibble: 3 x 2
# Groups:   year [3]
  year      n
  <chr> <int>
1 2015    924
2 2016   1013
3 2017    876
```

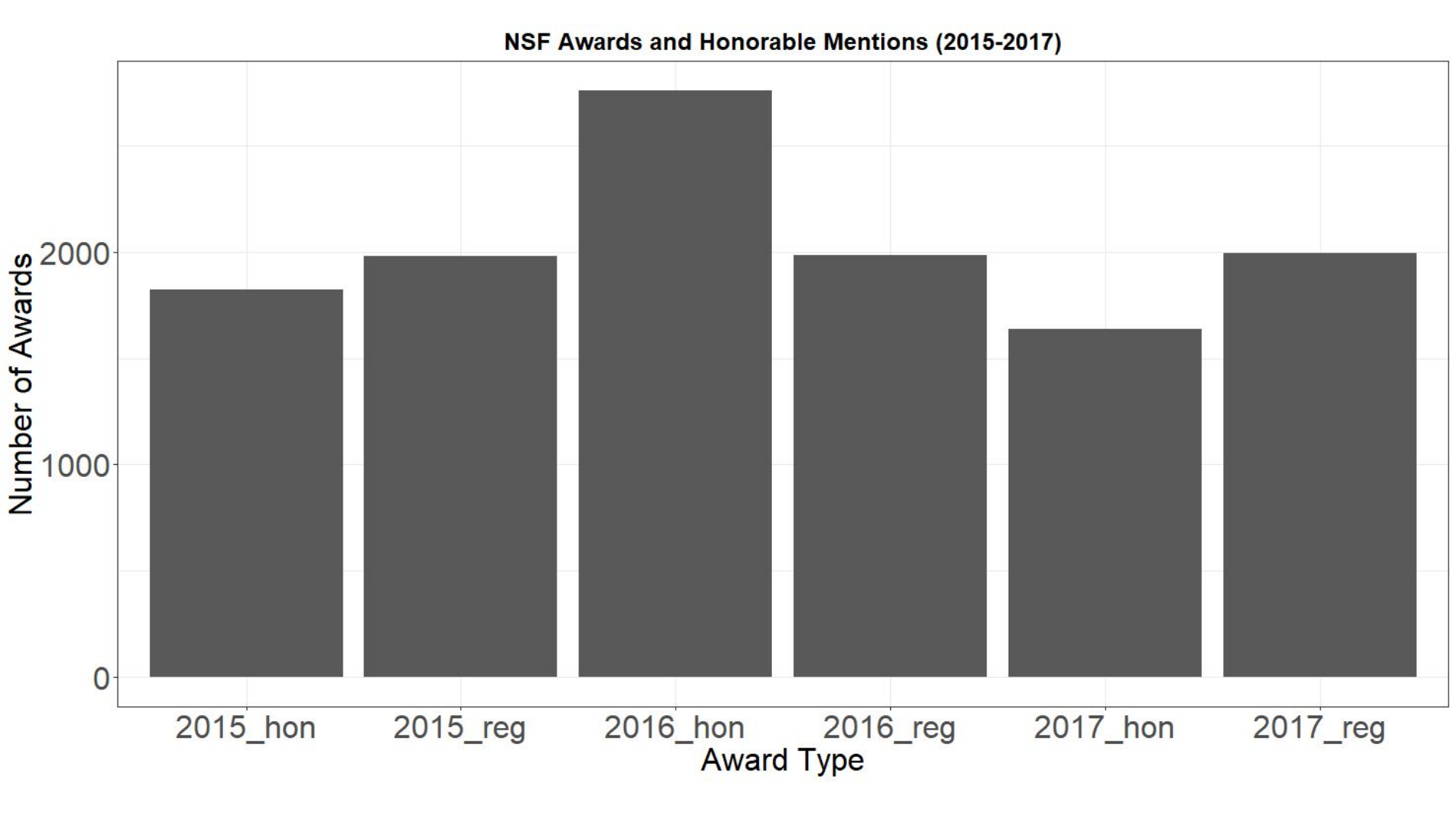




```
> head(countApps)
# A tibble: 6 x 4
# Groups:   award_type, year [6]
  award_type year      n propAwards
  <chr>      <chr> <int>      <dbl>
1 2015_hon   2015    453      0.490
2 2015_reg   2015    471      0.510
3 2016_hon   2016    533      0.526
4 2016_reg   2016    480      0.474
5 2017_hon   2017    413      0.471
6 2017_reg   2017    463      0.529
```

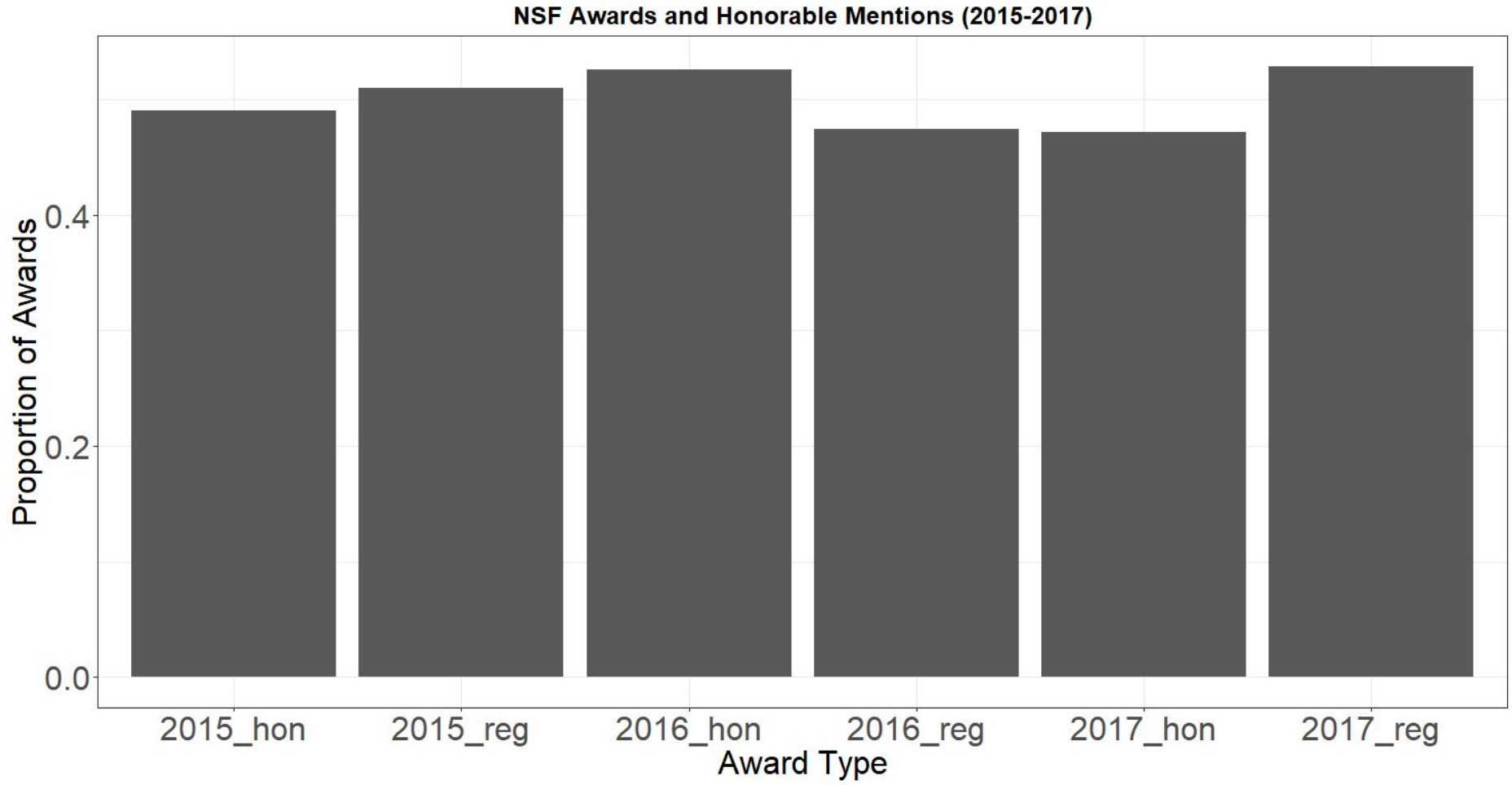
Visualize data

```
####Visualizing our new data frame
ggplot(NSFData, aes(x=award_type, y=n)) +
  geom_bar(stat = "identity") +
  labs(x = "Award Type", y = "Number of Awards", title = "NSF Awards and
Honorable Mentions (2015-2017)") +
  theme_bw() +
  theme(plot.title=element_text(size =18, face = "bold", hjust=0.5),
        axis.text.x = element_text(size  = 24, vjust=1, hjust=0.5),
        axis.text.y = element_text(size  = 24),
        axis.title=element_text(size=24),
        legend.title=element_text(size=24),
        legend.text=element_text(size=18),
        legend.position = "bottom")
```


NSF Awards and Honorable Mentions (2015-2017)
Number of Awards
2000
1000
0
2015_hon
2015_reg
2016_hon
2016_reg
2017_hon
2017_reg
Award Type

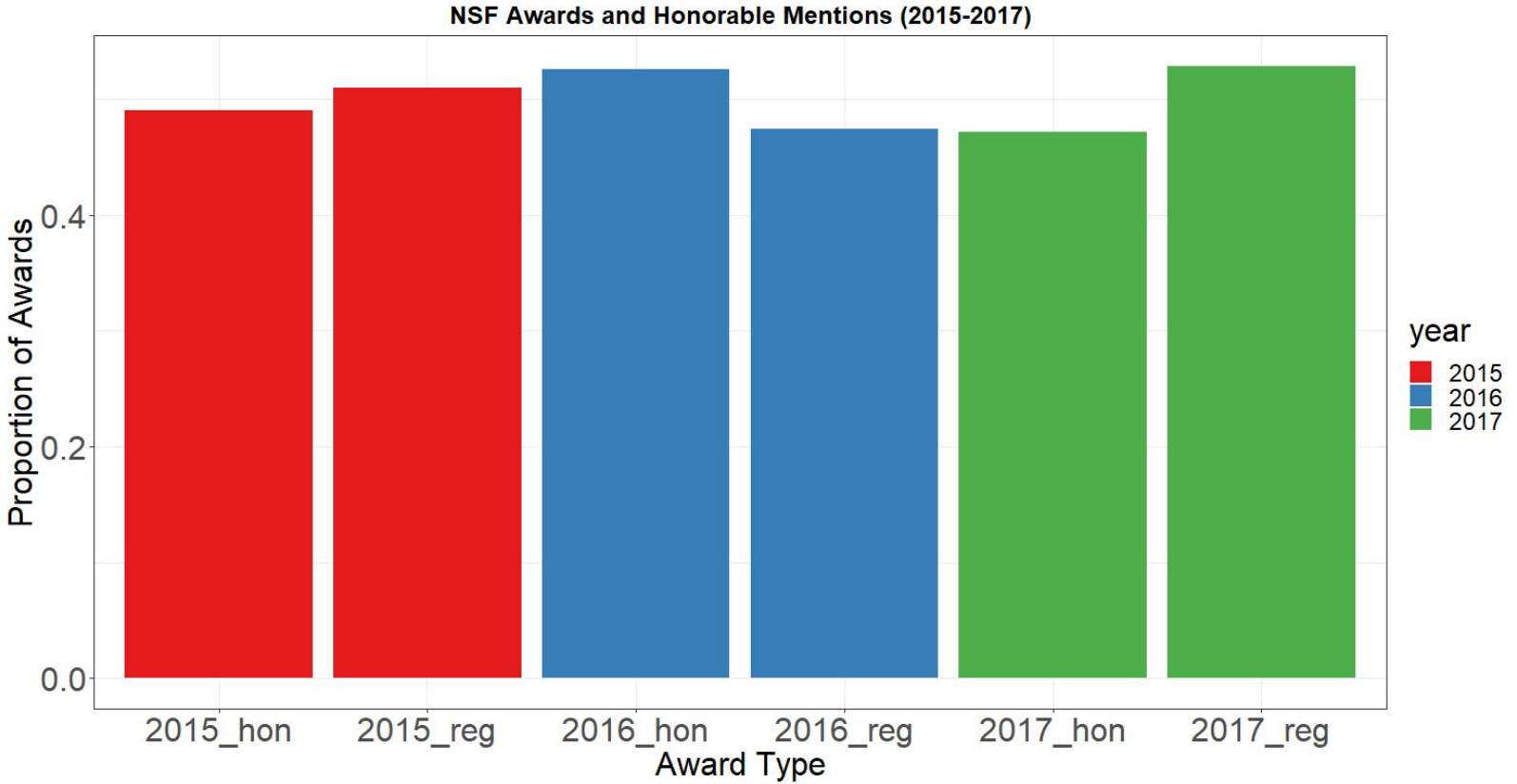
Visualize data

```
#Plot the proportional data
ggplot(data = countApps, aes(x=award_type, y=propAwards)) +
  geom_bar(stat = "identity") +
  labs(x = "Award Type", y = "Proportion of Awards", title = "NSF Awards and
Honorable Mentions (2015-2017)") +
  theme_bw() +
  theme(plot.title=element_text(size =18, face = "bold", hjust=0.5),
        axis.text.x = element_text(size  = 24, vjust=1, hjust=0.5),
        axis.text.y = element_text(size  = 24),
        axis.title=element_text(size=24),
        legend.title=element_text(size=24),
        legend.text=element_text(size=18),
        legend.position = "bottom")
```

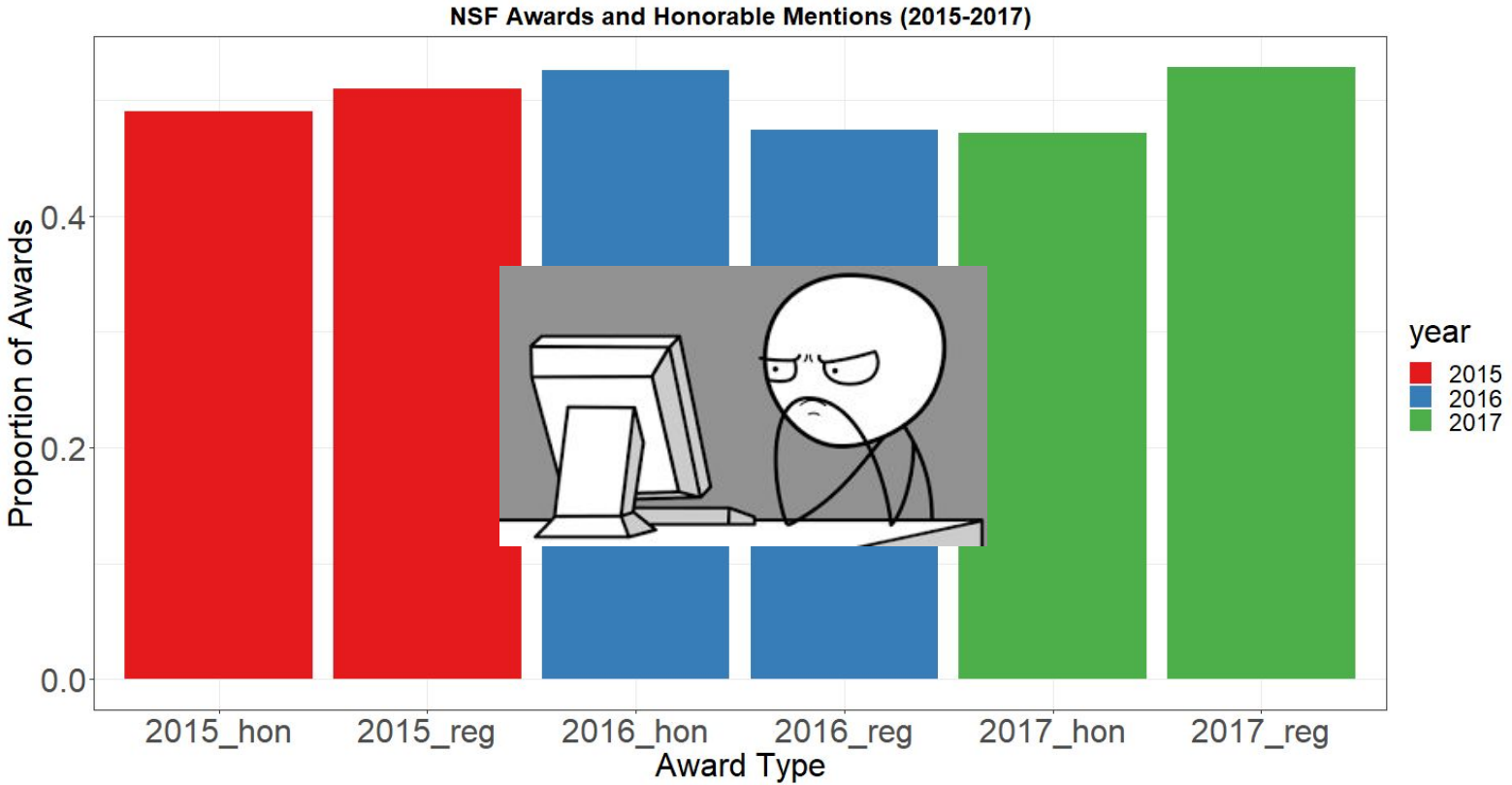

NSF Awards and Honorable Mentions (2015-2017)
Proportion of Awards
0.0
0.2
0.4
2015_hon
2015_reg
2016_hon
2016_reg
2017_hon
2017_reg
Award Type

Question: How many NSF awards were given vs. how many honorable mentions were given?

Add some color



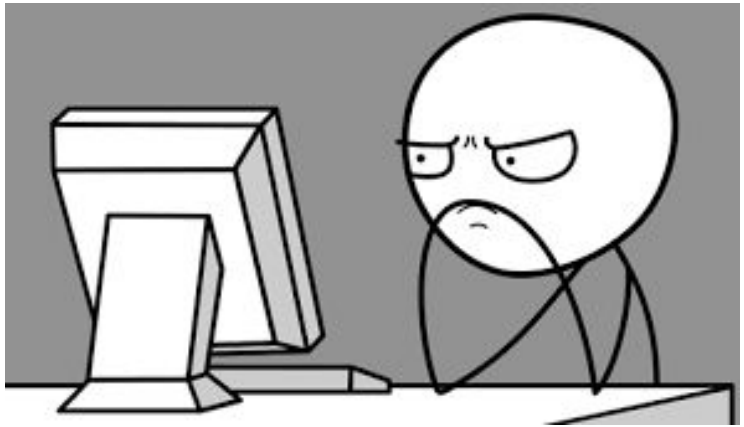
Is there a significant difference in the number of honorable mentions vs awards in 2016?



Deciding which statistical method to use

Questions to ask yourself:

- What is your data?
  - Continuous, Binary, Censored
- What is your dependent variable?
- What do you want to do?
  - Compare groups?
  - Evaluate effect?



Deciding which statistical method to use

Data Type	Compare Groups	Evaluate Effect
Continuous	2-way ANOVA, T-test	Linear regression or Multiple linear regression
Binary	Fisher’s Exact Test, Chi-Square	Logistic Regression

Deciding which statistical method to use

Questions to ask yourself:

- What is your data?
  - Continuous, Binary, Censored
- What do you want to do?
  - Compare groups
  - Evaluate effect?

Comparing the mean of two samples

Data Type	Compare Groups	Evaluate Effect
Continuous	2-way ANOVA, T-test	Linear regression or Multiple linear regression
Binary	Fisher's Exact Test, Chi-Square	Logistic Regression

Running a t-test in R

Step 1:

- Filter to year of interest
- Select column of interest

```
####Running t-test

#select columns of interest
honMen = NSFData %>%
  filter(award_type== "2016_hon") %>%
  select(n) %>%
  unlist()
honAward = NSFData %>%
  filter(award_type== "2016_reg") %>%
  select(n) %>%
  unlist()
```

Running a t-test in R

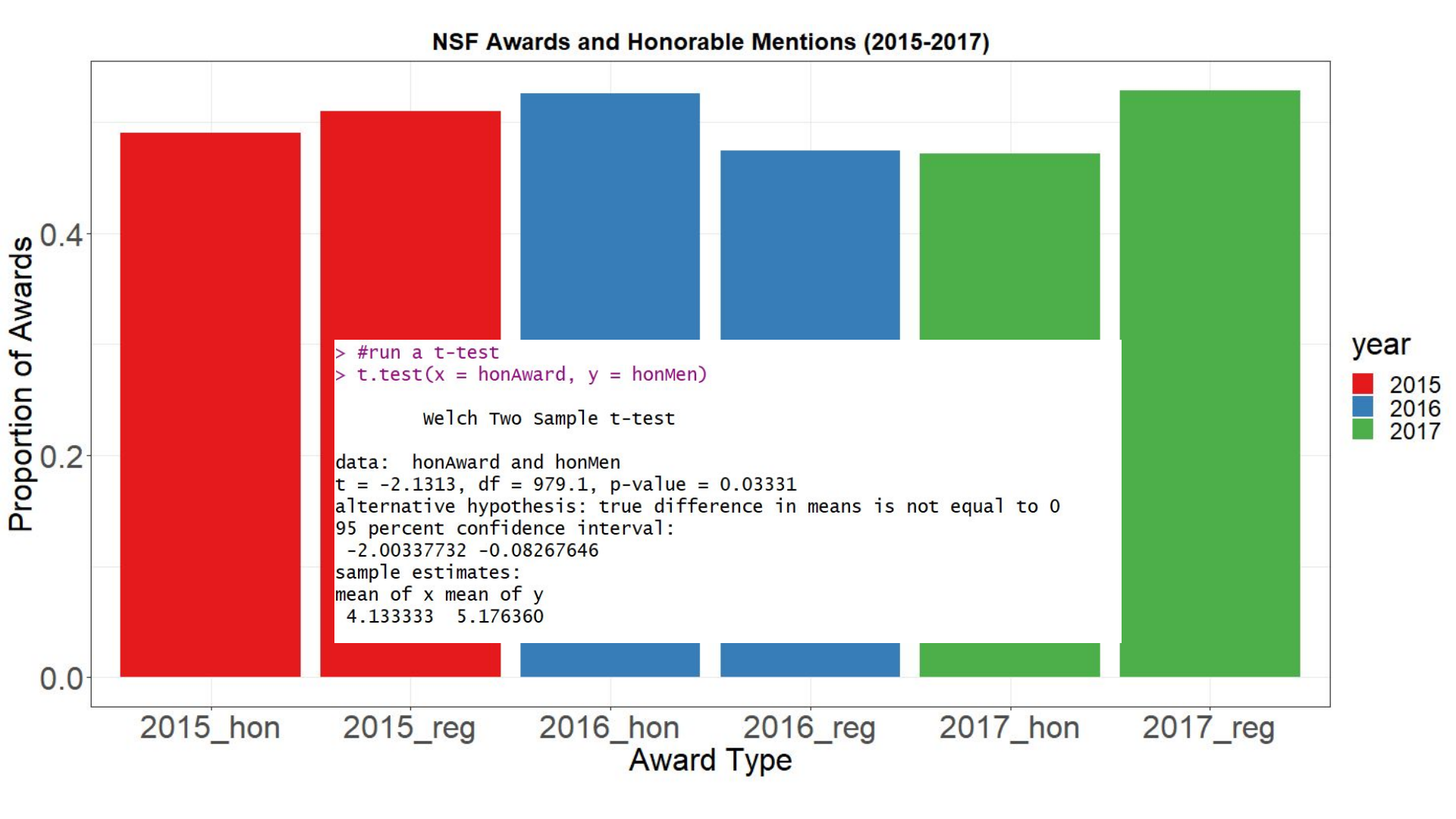
Step 1:

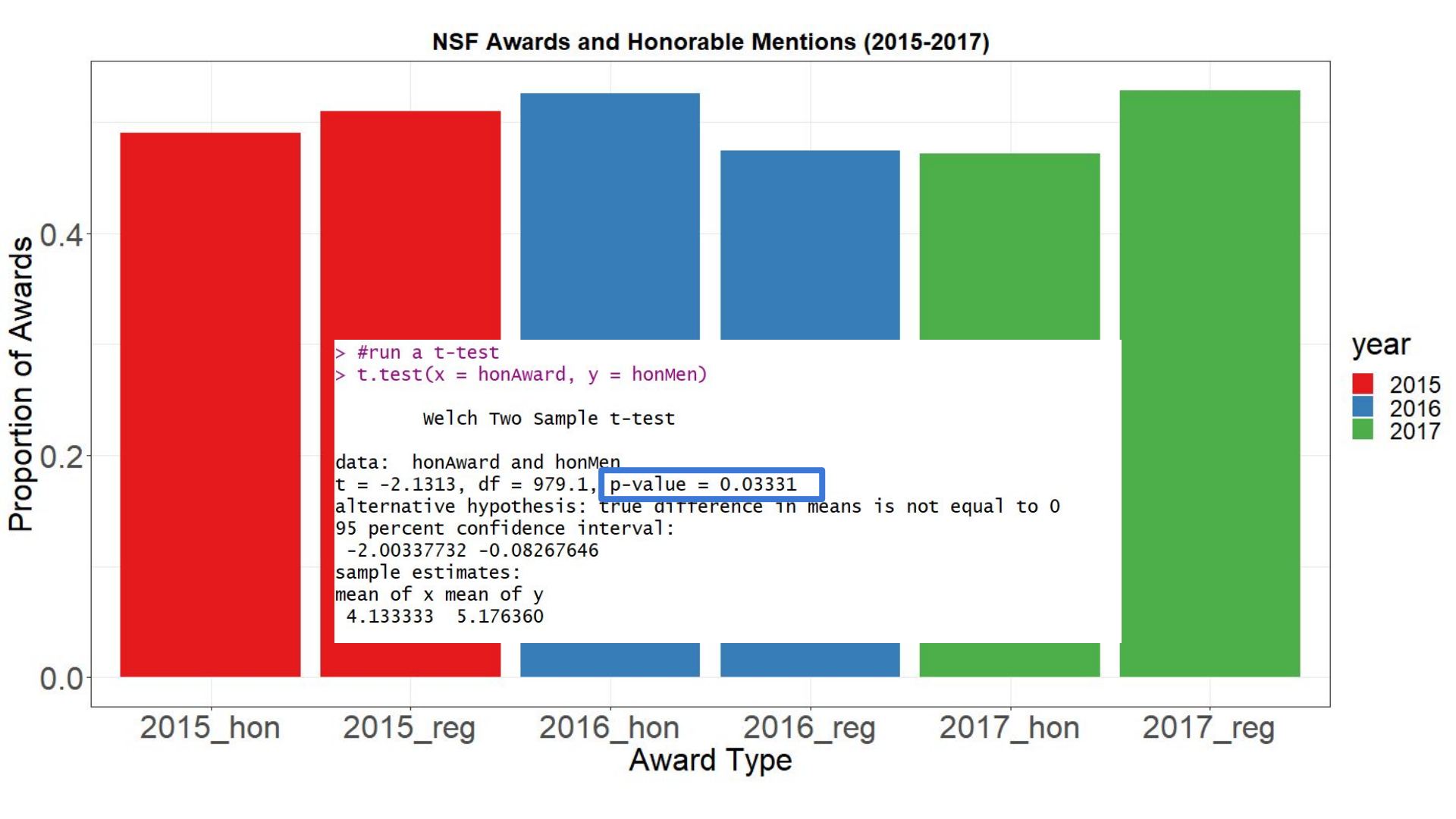
- Filter to year of interest
- Select column of interest

Step 2:

- Run your t-test

```
####Running t-test

#select columns of interest
honMen = NSFData %>%
  filter(award_type== "2016_hon") %>%
  select(n) %>%
  unlist()
honAward = NSFData %>%
  filter(award_type== "2016_reg") %>%
  select(n) %>%
  unlist()

#run a t-test
t.test(x = honAward, y = honMen)
```


NSF Awards and Honorable Mentions (2015-2017)
Proportion of Awards
0.0
0.2
0.4
2015_hon
2015_reg
2016_hon
2016_reg
2017_hon
2017_reg
Award Type
year
2015
2016
2017
> #run a t-test
> t.test(x = honAward, y = honMen)
Welch Two Sample t-test
data: honAward and honMen
t = -2.1313, df = 979.1, p-value = 0.03331
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.00337732 -0.08267646
sample estimates:
mean of x mean of y
4.133333 5.176360


NSF Awards and Honorable Mentions (2015-2017)
Proportion of Awards
0.0
0.2
0.4
2015_hon
2015_reg
2016_hon
2016_reg
2017_hon
2017_reg
Award Type
year
2015
2016
2017
> #run a t-test
> t.test(x = honAward, y = honMen)
Welch Two Sample t-test
data: honAward and honMen
t = -2.1313, df = 979.1, p-value = 0.03331
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.00337732 -0.08267646
sample estimates:
mean of x mean of y
4.133333 5.176360

Question: Can I predict the number of NSF awards given to a school if I know how many were received the previous year?

Deciding which statistical method to use

Questions to ask yourself:

- What is your data?
  - Continuous
- What is your dependent variable?
  - Number of awards in 2017
- What do you want to do?
  - Evaluate effect

Predicting an outcome given previous observations

Data Type	Compare Groups	Evaluate Effect
Continuous	2-way ANOVA, T-test	Linear regression or Multiple linear regression
Binary	Fisher's Exact Test, Chi-Square	Logistic Regression

Visualizing patterns

Step 1:

```
####Visualizing our data and making predictions

#reshape our data frame to turn award types into columns
reshapeNSFData = NSFData %>%
  select(-c(year)) %>%
  pivot_wider(names_from = award_type, values_from = n, values_fill = list(n=0)) %>%
  as.data.frame()
```

	last_name	first_names	bs_school	field	curr_school	award_type
1	Dekarske	Madeline M	Agnes Scott College	Chemistry - Chemical Catalysis	Agnes Scott College	2017_hon
2	Hutcheson	Melissa Anne	Agnes Scott College	Physics and Astronomy - Particle Physics	Agnes Scott College	2015_reg
3	Brown	Erin	Allegheny College	Mathematical Sciences - Computational and D...	Allegheny College	2015_hon
4	Cusanno	Brianna	Allegheny College	Social Sciences - Communications	Allegheny College	2017_reg
5	Calamari	Zachary Thomas	University of Michigan Ann Arbor	Geosciences - Paleontology and Paleobiology	American Museum Natural History	2015_hon
6	Ingala	Melissa Robin	Fordham University	Life Sciences - Ecology	American Museum Natural History	2017_hon
7	Amarante	Linda M.	Long Island University C W Post Center	Life Sciences - Neurosciences	AMERICAN UNIVERSITY	2015_hon
8	Horin	Adam Patrick	Purdue University	Psychology - Developmental Psychology	AMERICAN UNIVERSITY	2017_hon
9	Amarante	Linda M.	Long Island University C W Post Center	Life Sciences - Neurosciences	AMERICAN UNIVERSITY	2016_reg
10	Hamel	Brian Thomas	American University	Social Sciences - Political Science	AMERICAN UNIVERSITY	2016_reg

Visualizing patterns

Step 1:

```
####Visualizing our data and making predictions

#reshape our data frame to turn award types into columns
reshapeNSFData = NSFData %>%
  select(-c(year)) %>%
  pivot_wider(names_from = award_type, values_from = n, values_fill = list(n=0)) %>%
  as.data.frame()
```

```
> head(reshapeNSFData)
                bs_school 2016_reg 2015_reg 2017_hon 2016_hon 2015_hon 2017_reg
1 Abia State University         1         0         0         0         0         0
2   Agnes Scott College         0         2         1         0         0         0
3         Albion College        0         0         1         0         0         0
4       Albright College        0         1         0         0         0         0
5      Alfred University        0         0         0         1         0         0
6      Allegheny College        0         1         1         1         3         3
```

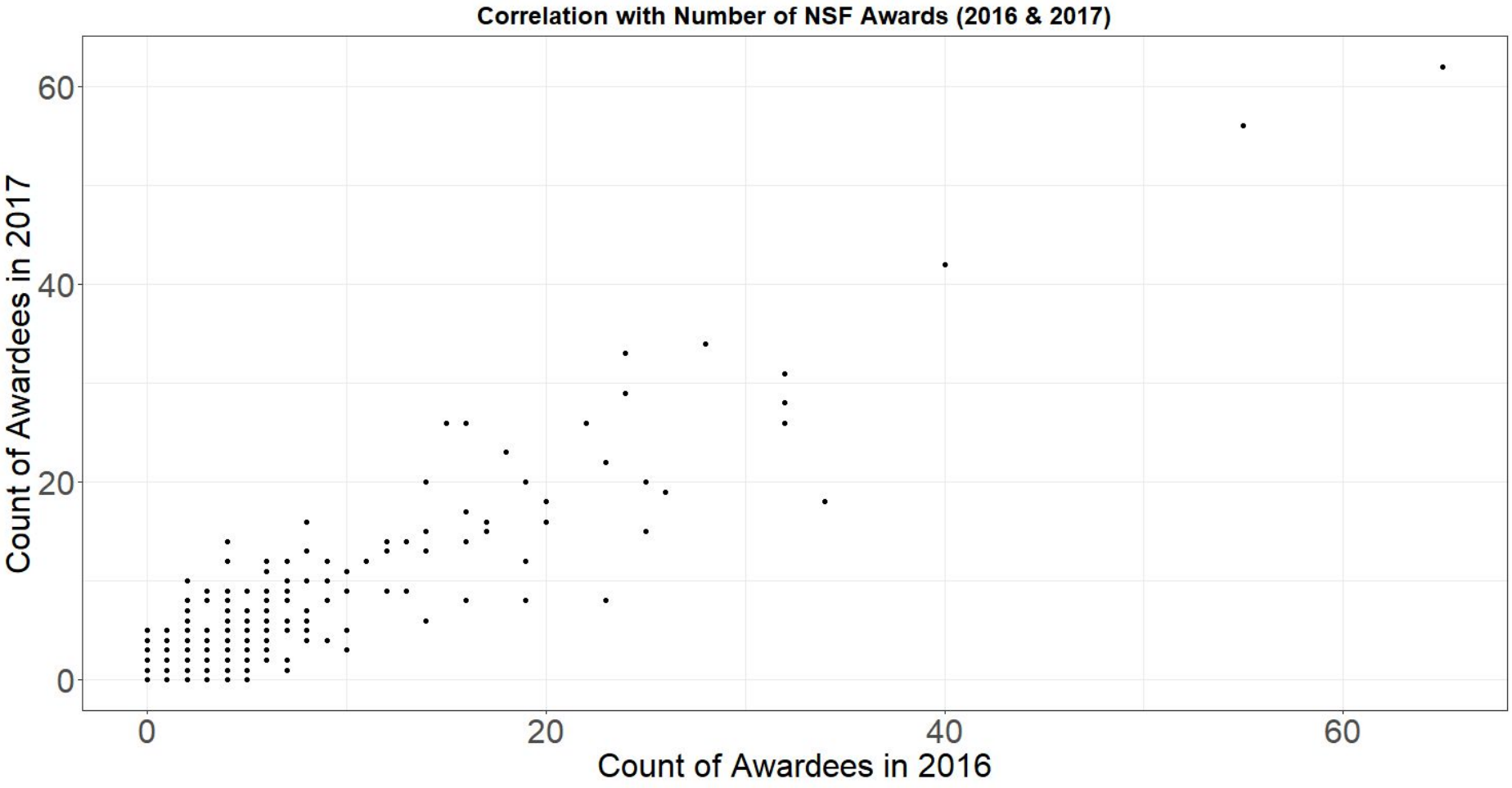
Visualizing patterns

Step 1:

```
####Visualizing our data and making predictions

#reshape our data frame to turn award types into columns
reshapeNSFData = NSFData %>%
  select(-c(year)) %>%
  pivot_wider(names_from = award_type, values_from = n, values_fill = list(n=0)) %>%
  as.data.frame()
```

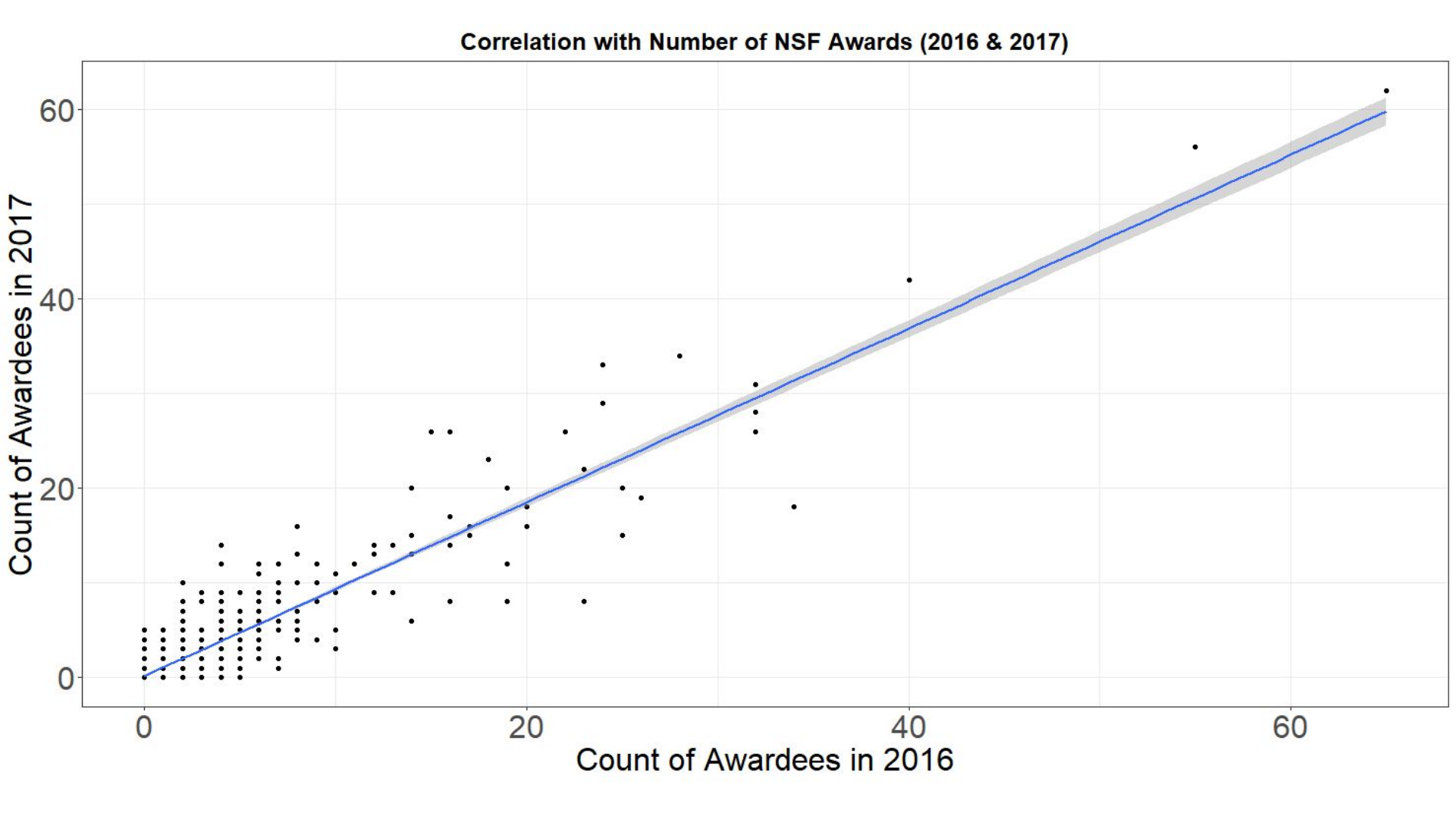
Step 2:

```
#Without the regression line
ggplot(reshapeNSFData, aes(x=reshapeNSFData$`2016_reg`, y=reshapeNSFData$`2017_reg`)) +
  geom_point() +
  theme_bw() +
  labs(x = "Count of Awardees in 2016", y = "Count of Awardees in 2017", title = "Correlation with Number of NSF Awards (2016 & 2017)") +
  theme_bw() +
  theme(plot.title=element_text(size =18, face = "bold", hjust=0.5),
        axis.text.x = element_text(size  = 24, vjust=1, hjust=0.5),
        axis.text.y = element_text(size  = 24),
        axis.title=element_text(size=24),
        legend.title=element_text(size=24),
        legend.text=element_text(size=18),
        legend.position = "bottom")
```


Correlation with Number of NSF Awards (2016 & 2017)
Count of Awardees in 2017
Count of Awardees in 2016
0
20
40
60
0
20
40
60

Visualizing patterns

```
#Without the regression line
ggplot(reshapeNSFData, aes(x=reshapeNSFData$`2016_reg`, y=reshapeNSFData$`2017_reg`)) +
  geom_point() +
  theme_bw() +
  labs(x = "Count of Awardees in 2016", y = "Count of Awardees in 2017", title = "Correlation with Number of NSF Awards (2016 & 2017)") +
  theme_bw() +
  theme(plot.title=element_text(size =18, face = "bold", hjust=0.5),
        axis.text.x = element_text(size  = 24, vjust=1, hjust=0.5),
        axis.text.y = element_text(size  = 24),
        axis.title=element_text(size=24),
        legend.title=element_text(size=24),
        legend.text=element_text(size=18),
        legend.position = "bottom")

#With the regression line
ggplot(reshapeNSFData, aes(x=reshapeNSFData$`2016_reg`, y=reshapeNSFData$`2017_reg`)) +
  geom_point() +
  theme_bw() +
  geom_smooth(method="lm") +
  labs(x = "Count of Awardees in 2016", y = "Count of Awardees in 2017", title = "Correlation with Number of NSF Awards (2016 & 2017)") +
  theme_bw() +
  theme(plot.title=element_text(size =18, face = "bold", hjust=0.5),
        axis.text.x = element_text(size  = 24, vjust=1, hjust=0.5),
        axis.text.y = element_text(size  = 24),
        axis.title=element_text(size=24),
        legend.title=element_text(size=24),
        legend.text=element_text(size=18),
        legend.position = "bottom")
```


Correlation with Number of NSF Awards (2016 & 2017)
Count of Awardees in 2017
Count of Awardees in 2016
0
20
40
60

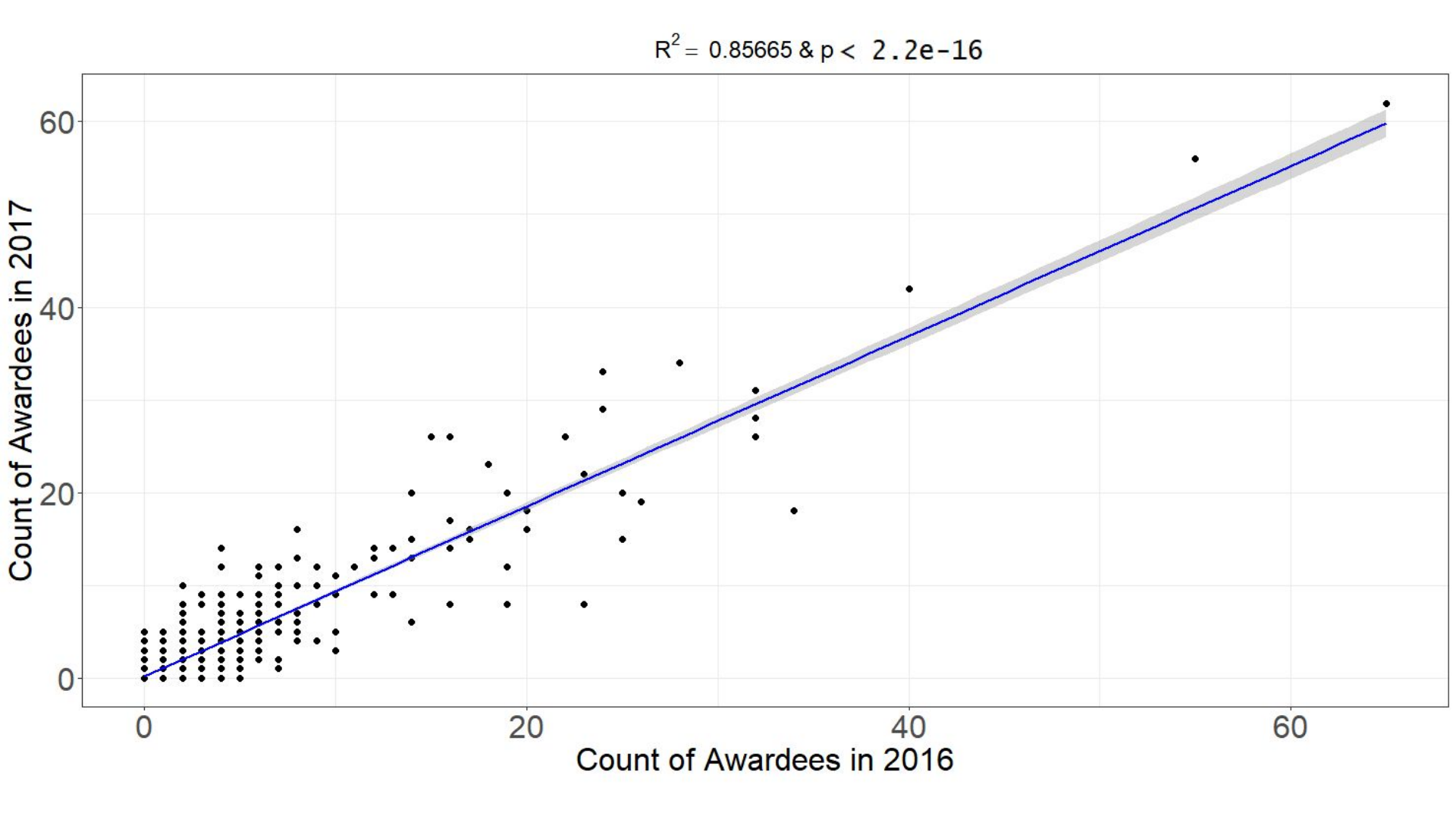
Running Linear regression in R

```
#linear regression with 2016 as predictor and 2017 outcome/response
model2017 = lm(data = reshapeNSFData, formula =  reshapeNSFData$`2017_reg` ~ reshapeNSFData$`2016_reg`)
```

Running Linear regression in R

```r
#linear regression with 2016 as predictor and 2017 outcome/response
model2017 = lm(data = reshapeNSFData, formula =  reshapeNSFData$`2017_reg` ~ reshapeNSFData$`2016_reg`)

#Use summary to check whether correlation is significant
summary(model2017)
```

```
Call:
lm(formula = reshapeNSFData$`2017_reg` ~ reshapeNSFData$`2016_reg`,
    data = reshapeNSFData)

Residuals:
     Min       1Q   Median       3Q      Max
-13.3732  -1.0114  -0.1763   0.8237  12.0604

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                  0.17627    0.06568   2.684  0.00741 **
reshapeNSFData$`2016_reg`  0.91756    0.01200  76.455  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.909 on 977 degrees of freedom
Multiple R-squared:  0.8568,    Adjusted R-squared:  0.8566
F-statistic:  5845 on 1 and 977 DF,  p-value: < 2.2e-16
```


R^2 = 0.85665 & p < 2.2e-16
Count of Awardees in 2017
Count of Awardees in 2016
0
20
40
60

Can anyone think of potential problems with our model?

Potential Problems

- Don’t know total number of applicants…
  - If we do get this information we can add this into our model

Slides available

https://mooney-lab.github.io/#

A couple nice resources for Rstats and Tidyverse

Rstats: https://rafalab.github.io/dsbook/

Tidyverse:https://moderndive.com/4-tidy.html

More general information about R:

Resources for beginners to self-learn

Quick R: free online tutorial

http://tryr.codeschool.com/

Swirl : Offline Interactive learning. Please see FAQ section for details.

Coursera: R Programming course by Johns Hopkins

Ebooks: Introduction to Statistical Machine Learning

Acknowledgements

Natalie Telis for NSF data below is the article and shiny app

https://www.science.org/content/article/nsf-graduate-fellowships-disproportionately-go-students-few-top-schools

https://nsf-grfp.shinyapps.io/shiny/

Recruiting potential computational PhD students

Ph.D. in Computational Biology and Bioinformatics

University of Southern California
Los Angeles, CA

The University of Southern California (USC) is an international leading institution in Computational Biology and Bioinformatics for more than 35 years. Since Michael Waterman joined USC in 1982, the group has grown to include a large number of core and affiliated faculty members with Nobel and Dan David prize laureates, members of the US National Academy of Sciences and the US National Academy of Engineering, and members of the Royal Society. The program has more than 60 doctoral students and postdoctoral associates. Our research is supported by grants from the National Institutes of Health (NIH) and the National Science Foundation (NSF), as well as private foundations. We were awarded Center of Excellence in Genome Science grants by NIH for two funding periods to develop novel and innovative genomic research projects.

Students in our program may choose from a broad set of research topics, with the following areas receiving particular emphasis:

- Computational structural biology
- Quantitative genetics, population genetics, and evolutionary biology
- Genomics, epigenomics and metagenomics
- Molecular dynamics, networks, and systems biology
- Big data, machine learning, and precision medicine
- Neuro and molecular imaging
- Sequence analysis, genome assembly

Fall 2022 Application Deadline:

December 15, 2021

Prepare for Various Careers

- Academic and research institutions
- Corporations
- Government Organizations
- Non-profit companies

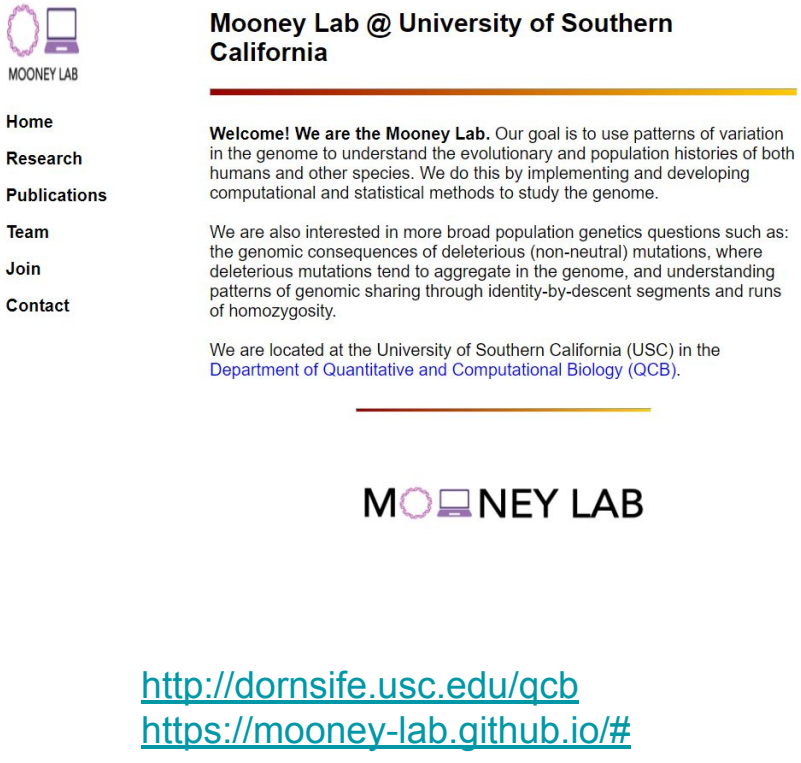
Financial Support

- Monthly stipend
- Paid tuition
- Health and dental insurance
- Minimum 5 years of support

Our graduates have an exceptional placement rate, with some opting for highly desirable industry positions, and others continuing successfully in academia and research. Southern California is one of the premier cultural hubs in the country, and we have nearly perfect weather year-round.

We hope you will consider applying to our Ph.D. program!

Please visit our web page for additional information: http://dornsife.usc.edu/qcb



MOONEY LAB

Home

Research

Publications

Team

Join

Contact

Mooney Lab @ University of Southern California

Welcome! We are the Mooney Lab. Our goal is to use patterns of variation in the genome to understand the evolutionary and population histories of both humans and other species. We do this by implementing and developing computational and statistical methods to study the genome.

We are also interested in more broad population genetics questions such as: the genomic consequences of deleterious (non-neutral) mutations, where deleterious mutations tend to aggregate in the genome, and understanding patterns of genomic sharing through identity-by-descent segments and runs of homozygosity.

We are located at the University of Southern California (USC) in the Department of Quantitative and Computational Biology (QCB).

MOONEY LAB

http://dornsife.usc.edu/qcb
https://mooney-lab.github.io/#