Note that there are three main parts to this lab/homework. The first has us work with R code and the second asks us to evaluate different types of plots. The last is just a short part where I ask you to think about what you learned etc.

Section 1: Visual data in R

**Purpose:**

1. **Learn how to use ggplot to make different types of plots**
2. **Understand how to change different aspects and features of a plot**

**Tasks:**

1. **Read and follow along with the R code**
2. **Copy and past your R code into the space in this document and then post to AsULearn**

# Making plots with ggplot package

To do this part of the homework you can copy and past the code in the ‘code blocks’ into R and run it. Then when you figure out how to answer the questions just copy and paste your code from R into this document!

Rather than a normal handout/lab this one tells a story and has you learning ggplot while you do so. Be sure to check out the reading here: <https://r4ds.had.co.nz/data-visualisation.html> before doing this project.

If you need more background on how ggplot works check out chpt 3 of <https://socviz.co/makeplot.html#makeplot> “Data Visualization”

You might not know how to do the things being asked for right away. That is fine and expected. What i want you to learn is how to *find the answers* rather than what the answer is. In other words, I am more concerned with the process than the result. But if you are getting frustrated reach out and I’m always happy to help

## Story time

You are working on your Ph.D with a professor of anthropology at Fancypants University. She has just learned about a new species of monkey that she wants you to collect data on. So you are off to the island of Naboombu, where you will be given the task of watching these primates.



## Collecting your data, part 1

When you arrive on the western part of the island you notice that there are a whole lot of monkeys here and you need to find a way to get an idea of what they look like. So you and your team decide to start by measuring their tails (they aren’t apes!). You manage to measure 500 of these fellas (ok, this is a bit of a reach but lets say that you are *really* quick at measuring and the monkeys like having their tails measured….)

below is a very quick reminder of how R works

#first we make an object and assign something to it...  
  
the\_numbers <- c(4, 8, 15, 16, 23, 42)  
#once we have an object we can do things to that object.  
  
mean(the\_numbers)

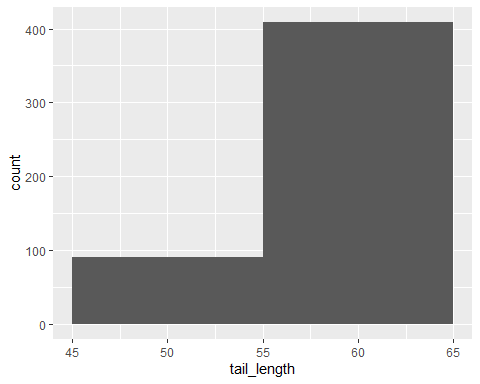
sum(the\_numbers)

sd(the\_numbers)

We are going to be simulating the data you collected rather than writing 500 entries..

Run the following R code in a new session to get the first monkey data! **Be sure to run the first line of code that loads in the tidyverse library.** I made some comments so you can follow along with what the code is doing:

library(tidyverse)  
set.seed(2020) #this sets the randomizer in R  
tail\_length <-rnorm(500,57,2) # create a set of 500 random numbers with a mean of 57 and a standard deviation of 2   
sample\_number <-1:500 # create a vector of numbers from 1 to 500  
monkey\_pop <- rep(letters[1], 500) # this makes a vector of 500 'a's  
monkey\_data\_1 <- tibble(sample\_number, tail\_length, monkey\_pop) #create a data table (a tibble is a function that makes nice data tables with a modern flair...)  
  
monkey\_data\_1 %>% ggplot(aes(tail\_length)) + geom\_histogram(binwidth = 10) # this takes the monkey data and plots a histogram of the tail length



### 

### Question 1: Changing the bin length:

You note that the plot you made with the code above doesn’t look very nice. After thinking for a bit you realize the bin length is too large and you can’t see enough of the data. What code would you use to change the binwidth to be 1. Copy the code from above and change it in R so the figure looks better. Then copy that code into the box below:

### 

#copy your answer to question 1 here:

### Question 2: Ok, but you know your professor wants the histogram to be blue. How can you change the code below to make the fill of the histogram be blue?

#### (n.b. for help see section **3.5 “Mapping aesthetics vs setting them” of** [**https://socviz.co/makeplot.html#makeplot**](https://socviz.co/makeplot.html#makeplot)**)**

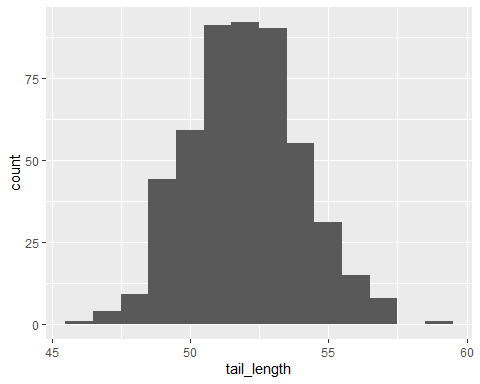
#copy your answer to question 2 here:

You send this lovely blue histogram off to your professor and go and take a nap (measuring 500 tails *plus* data entry is serious work!)

## Collecting your data, part 2

The next day you are talking to some local people and hear about another group of monkeys on a different part of the island. **First run the code below to get that data and make a histogram!**

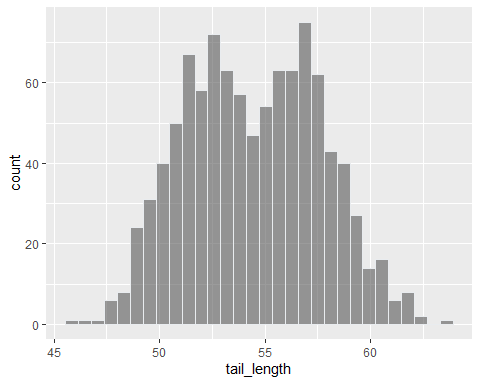
tail\_length <-rnorm(500,52,2)  
sample\_number <-501:1000  
monkey\_pop <- rep(letters[2], 500)  
monkey\_data\_2 <- tibble(sample\_number, tail\_length, monkey\_pop)  
  
monkey\_data\_2 %>% ggplot(aes(tail\_length)) + geom\_histogram(binwidth = 1)



Now you want to look at both of the populations at the same time! Thankfully you have taken this class and have some R code ready to do this. Run the code below in your R session to make a histogram of both datasets

all\_data <- bind\_rows(monkey\_data\_1, monkey\_data\_2) # here we are using the function bindrows to add the two datasets together!  
  
all\_data %>% ggplot(aes(tail\_length )) + geom\_histogram(color="#e9ecef", alpha=0.6, position = 'identity')

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Hmmm…that doesn’t look right since you can’t tell the different groups apart. But you think back to your stats class in college and remember that we need to set an aes(...) argument. The aesthetic specifies that a variable will be expressed by one of the available visual elements, such as size and **color**. You sorta did this above when we made the histogram blue but this is a bit trickier…

### Question 3: How can you change the code below to make the different monkey populations have different colors (note: you just need to add one argument to the aes() part. i put “\*\*\*\*\*\*\*\*\*\*\*\*\*” to show where to put this

#Question 3: edit the below code to color by the variable monkey\_pop  
  
all\_data %>% ggplot(aes(tail\_length, \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* )) + geom\_histogram(color="#e9ecef", alpha=0.6, position = 'identity')

So you look at this and think to yourself “Hey, they overlap a bit but there are difference at the tails….😀. I am going to send this to my adviser and she will be soooooo happy!”

A few hours later you get an email with these requests. Your adviser wants you to do all of the following to the figure you have made

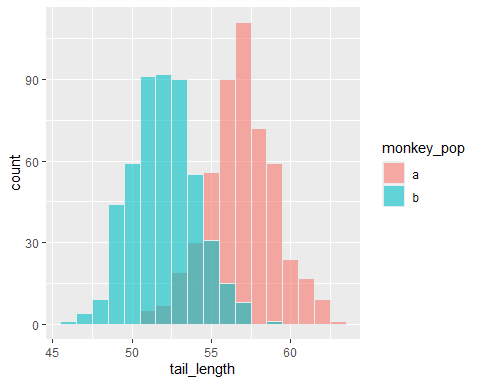
### Question 4: updates to the plot

Here is what she wants you to do:

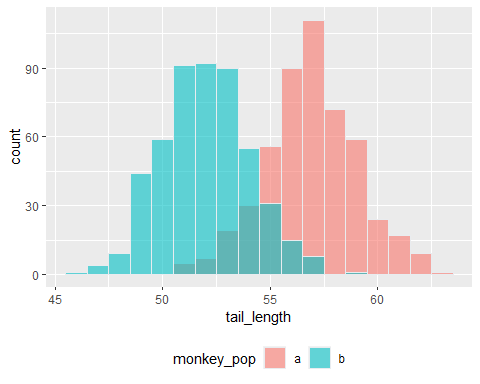
* Put the legend on the bottom rather than on the right
* Label the x axis “tail length (cm)”
* Label the y axis “number of observations”
* Give it a title that is meaningful
* Put a credit somewhere that says who made the figure
* Change the legend so that it says “Monkey groups”
* Change the labels from “a” and “b” to “Group one” and “Group 2”

Lets take this step by step. Copy the below R code and use each step to build the new graph. If you need help use the class readings or ask on Discord! Then copy the code you used below. For example, to get the first part of this we would update the code like this

#old code  
all\_data %>% ggplot(aes(tail\_length, fill=monkey\_pop )) + geom\_histogram(color="#e9ecef", alpha=0.6, position = 'identity', binwidth = 1)



# new code



To figure out how to do this we are going to need to check online. One good source is <https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>. For example if you look at the right hand column of page 2 you will see a box that talks about how to change the labels…another good source is this site: <https://ggplot2.tidyverse.org/reference/> which i use quite a bit to rejigger figures.

### 

#copy your answer to question 4 here:  
  
  
#plot a. here is my version to the professors first request.   
  
all\_data %>% ggplot(aes(tail\_length, fill=monkey\_pop )) + geom\_histogram(color="#e9ecef", alpha=0.6, position = 'identity', binwidth = 1) + theme(legend.position = "bottom")  
  
#plot b  
  
  
  
#plot c  
  
  
  
#plot d  
  
  
  
#plot e  
  
  
#plot f  
  
  
#plot g

## 

## Bonus

Now that you have played around with options on ggplot and read the sections about this function play around in a ‘sandbox’ and change whatever you want (background color, text type, etc)

#copy your answer to bonus question here:

## Part 3: New visualizations with new data

## 

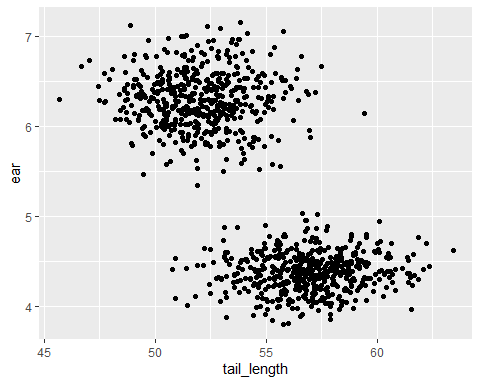
You and your professor submit your data paper to a journal. After a few months you get your peer review back and the reviewers say “wow, this is cool! But ya know what, we think you need more measurements before we can conclude these are 2 different populations. Can you go and measure ear length?”

copy and Run the following code to get the new\_data

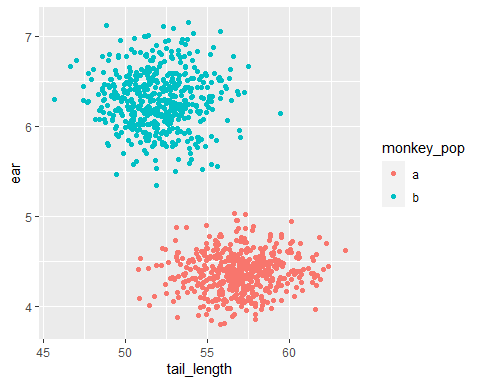
ear1 <- rnorm(500, 4.4, .22)  
ear2 <- rnorm(500, 6.3, .31)  
  
ear1 <-tibble(ear =rnorm(500, 4.4, .22), monkey\_pop = rep(letters[1], 500), sample\_number = 1:500)  
  
  
ear2 <-tibble(ear =rnorm(500, 6.3, .31), monkey\_pop = rep(letters[2], 500), sample\_number = 501:1000)  
  
ear\_all <- bind\_rows(ear1, ear2)  
  
new\_data <- all\_data %>% left\_join(ear\_all, by= c("monkey\_pop", "sample\_number"))

Since you know have two measurements on each monkey you need a new way to visualize these data. Remembering what you learned at AppState you go and make a scatterplot of these data. Copy the below code into R and see what happens!

new\_data %>% ggplot(aes(tail\_length, ear)) + geom\_point() # this makes the scatterplot geom



new\_data %>% ggplot(aes(tail\_length, ear, color = monkey\_pop)) + geom\_point()



You are thrilled and send it off to your adviser.

She comes back and asks for the following updates:

### Question 5: updating the scatteplot

* Get rid of the legend
* Add a regression line for each group
* Add a transparency or a jittter to deal with over plotting
* Make the 2 groups have a different shape
* Add an ellipse to the groups ( *see “stat\_ellipse()” for a hint*)

#write you answers to question 4 here  
  
  
#plot a  
  
  
#plot b  
  
  
  
#plot c  
  
  
# plot d

### 

### Question 6: taking it further

Your adviser now asks if you can make the plots with the new data for the following different geoms.

* make a violin plot
* make a boxplot
* make a ridgeline plot (n.b. to do this one we need to install a new package called “ggridges” (<https://www.r-graph-gallery.com/294-basic-ridgeline-plot.html>)

# write you answers to question 6 here. i put the first one here to help you out  
  
#plot a  
new\_data %>% ggplot(aes( monkey\_pop, tail\_length, fill = monkey\_pop)) + geom\_violin()

#plot b

#plot c

## export the data

she wants the whole dataset sent to her so she can take a look at it herself. You know that you can use write\_csv to do a basic export but she has some specific requests:

### Question 7

* change the name of the csv file so it is called “new\_primate\_data” rather than “my\_monkeydata.csv”
* change the format of the exported file to an Excel sheet (this is hard)

#write you answers to question 7 here. as a tip look at this code: write\_csv(new\_data, "my\_monkeydata.csv")  
  
  
#part a  
  
  
  
  
  
#part b

### 

# Once more, for the folks who like to dial it up to 11 (optional)

## 

Your paper has been accepted(!!!) but the editors want it selected for a special feature that requires an interactive component. To do this we need to use a few new packages: *plotly* and *glue*. Plotly lets us make interactive graphs fairly easily (it is one of my secrets that once you get the hang of it the code isn’t too hard to use and it makes fancy slides!). Glue lets us take different parts of a dataframe and paste (or glue) them together

install.packages("plotly")  
library(plotly)  
  
p <- new\_data %>% ggplot(aes(tail\_length, ear, color = monkey\_pop)) + geom\_point() # this is the basic plot we made before. I am just saving it to an object called p  
  
ggplotly(p) # just this simple function does so much!

add sample number to the hover:

p <- new\_data %>% ggplot(aes(tail\_length, ear, color = monkey\_pop, label= sample\_number)) + geom\_point() #here, we are adding the label argument to the aes function so that the ‘sample number’ variable shows up on when we put the mouse over the point.   
  
ggplotly(p)

make the tooltip look pretty

install.packages("glue")  
library(glue)  
  
  
  
p <- new\_data %>% ggplot(aes(tail\_length, ear, color = monkey\_pop, text = glue('sample is: {sample\_number}'))) + geom\_point() # here we use glue to add the "sample is:" text in front of the *sample number* variable.

ggplotly(p)

# What else can you do with R visualizations

Now that you have learned a lot about how to work with visualization data i want you to take a moment and think about how to approach another dataset

The palmerpenguins package contains two datasets. to install it copy the following code into R

install.library("palmerpenguins")

One is called penguins, and is a simplified version of the raw data; see [?penguins](https://allisonhorst.github.io/palmerpenguins/reference/penguins.html) for more info

The second dataset is penguins\_raw, and contains all the variables and original names as downloaded; see [?penguins\_raw](https://allisonhorst.github.io/palmerpenguins/reference/penguins_raw.html) for more info.

Both datasets contain data for 344 penguins. There are 3 different species of penguins in this dataset, collected from 3 islands in the Palmer Archipelago, Antarctica. The culmen is the upper ridge of a bird’s bill. In the simplified penguins data, culmen length and depth are renamed as variables bill\_length\_mm and bill\_depth\_mm to be more intuitive.

think about what the below code is doing. Once again we are using the pipe (%>%) to take the dataset and apply a function to it (in this case, the count function)

library(palmerpenguins)

glimpse(penguins)

penguins %>%  
 count(species)

## # A tibble: 3 x 2  
## species n  
## <fct> <int>  
## 1 Adelie 152  
## 2 Chinstrap 68  
## 3 Gentoo 124

penguins %>%  
 group\_by(species) %>%  
 summarize(across(where(is.numeric), mean, na.rm = TRUE))

Take a look at these data. what sort of figures might you want to make? Think about how you could tell a story about these penguins via an image. Then write some code that will let you do this. When you are done, take a moment to explain in prose why you made this figure and how you used R to do it

Section 2: comparing plots

**Purpose:**

1. **Understand what makes a plot good and how to ‘fix’ bad plots**

**Tasks:**

1. **For each figure, note 1) what is the figure trying to say and 2) what makes it problematic/how someone might fix it for it to be better**

**Figure A**

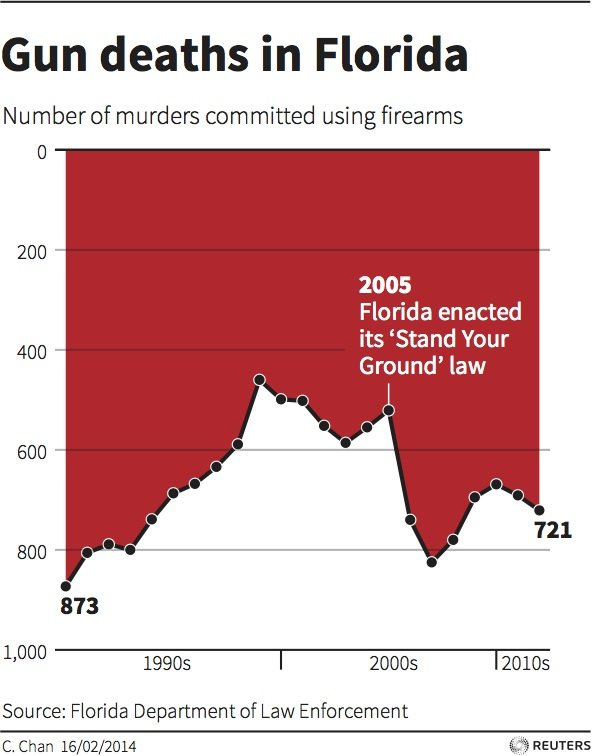


What is this figure trying to say?

What is problematic?

How might someone redo this better?

**Figure B**

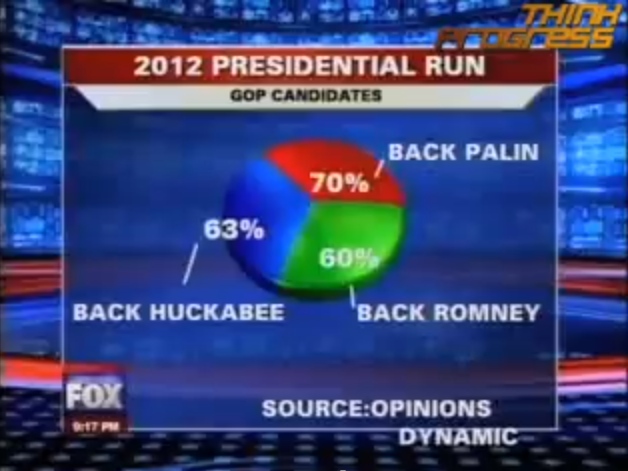


What is this figure trying to say?

What is problematic?

How might someone redo this better?

Figure C.

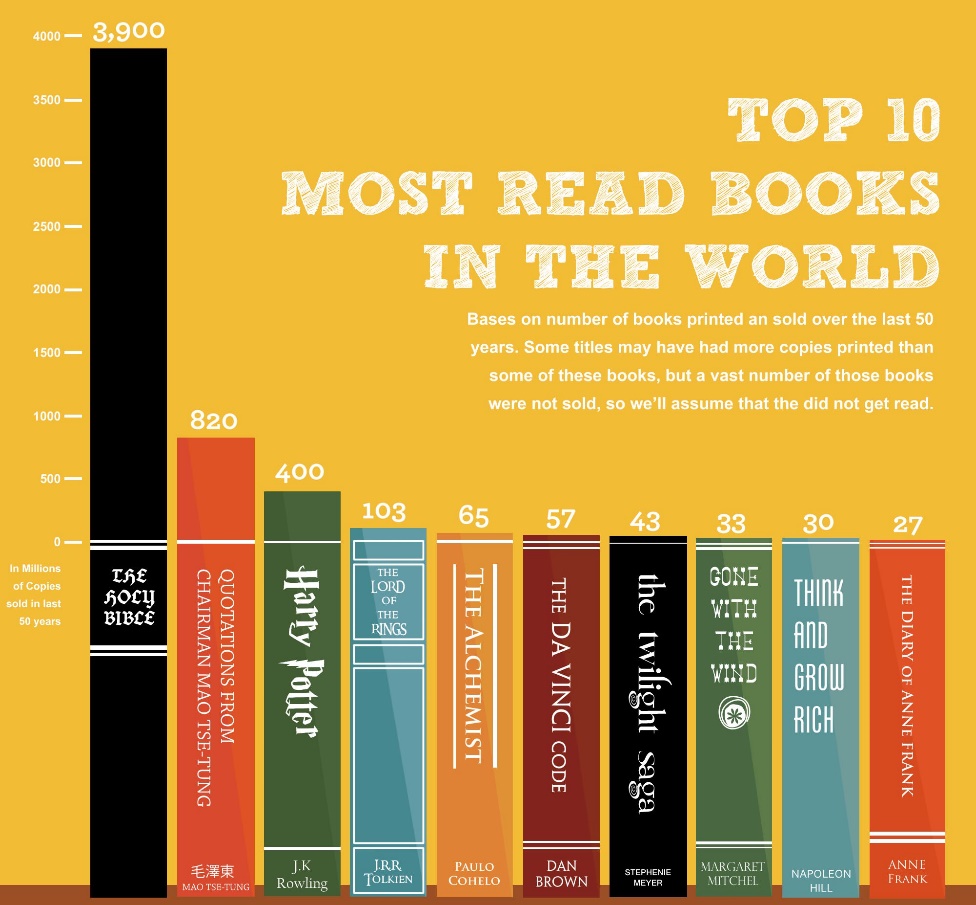


What is this figure trying to say?

What is problematic?

How might someone redo this better?

Figure D.

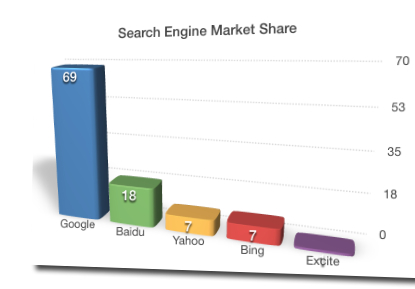


What is this figure trying to say?

What is problematic?

How might someone redo this better?

Figure E.

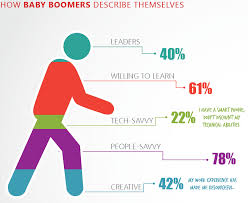


What is this figure trying to say?

What is problematic?

How might someone redo this better?

Figure F.

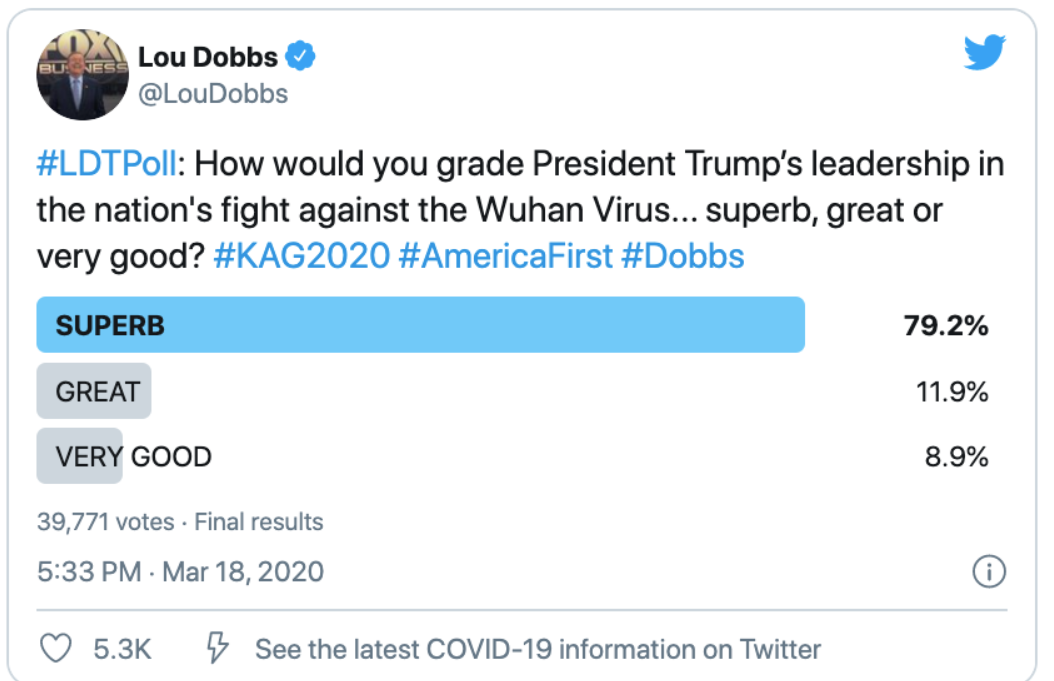


What is this figure trying to say?

What is problematic?

How might someone redo this better?

Figure G



What is this figure trying to say?

What is problematic?

How might someone redo this better?

**And finally…..the best pie chart:**



Section 3: Thinking about R

This is not easy stuff. Most folks don’t put much thought into what makes a good visualization nor do they take the time to keep these ideas in mind when creating them. This is true both in the media and in academia. I hope you have learned something from this project!

1. After thinking about these topics, what would you say makes a good data visualization stand out from a bad one?

2. ggplot is one of the most used packages in R (I saw a reference that it has been used to make over a million plots!). The ‘language’ is a bit hard to parse at first. Do you think you are getting a handle on how to make plots?

3. Now that you know something about data visualizations, what sorts of visualization might you make for your data analysis project?

4. for the project where we look at how stats are seen in the media, did the story you chose have a figure in it? If so, do you think it is a good one?