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 paste 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….)

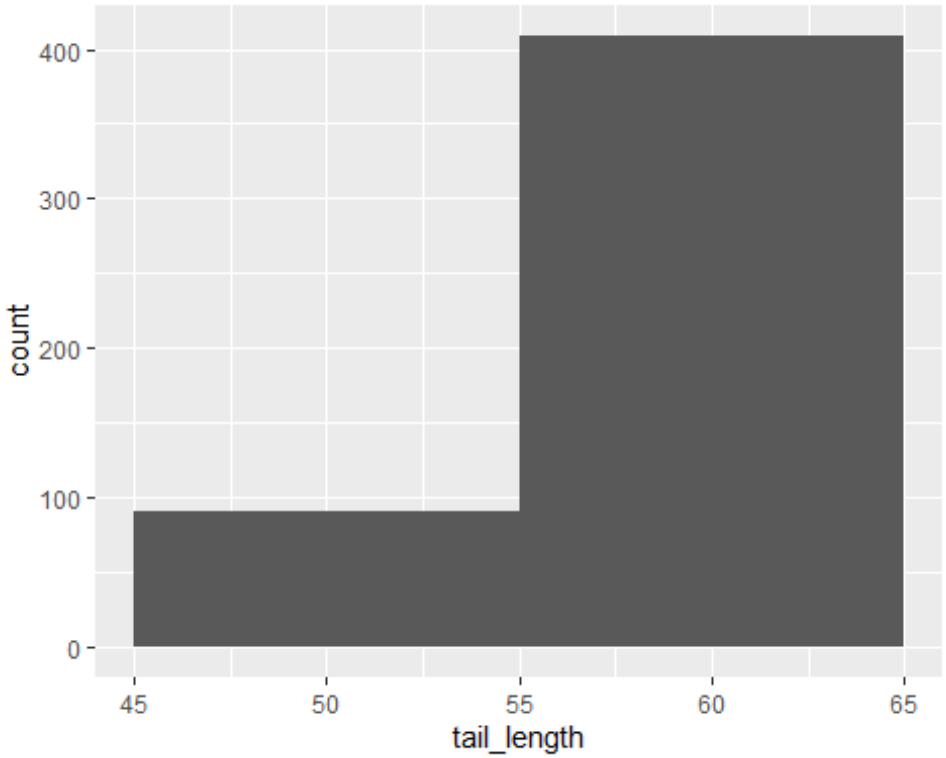
below is a very quick reminder of how R works

```
#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:
```

```
monkey_data_1 %>%  ggplot(aes(tail_length)) + geom_histogram(binwidth =  1)
```

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)***

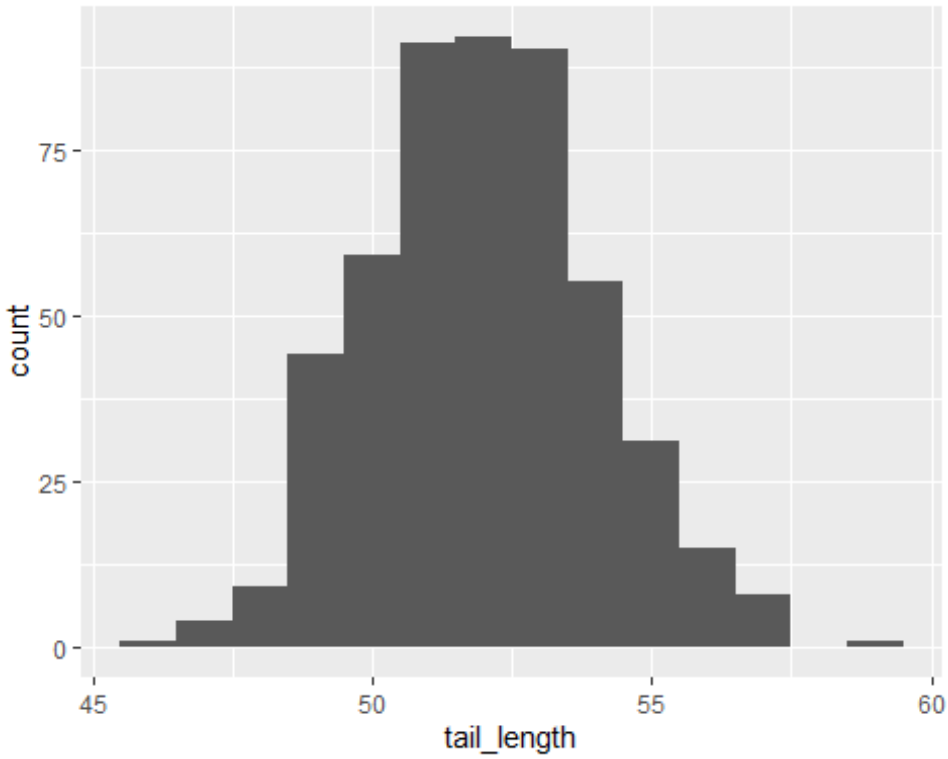
```
#copy your answer to question 2 here:

monkey_data_1 %>%  ggplot(aes(tail_length,)) + geom_histogram(binwidth =  1,
col="blue",fill="blue")
```

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

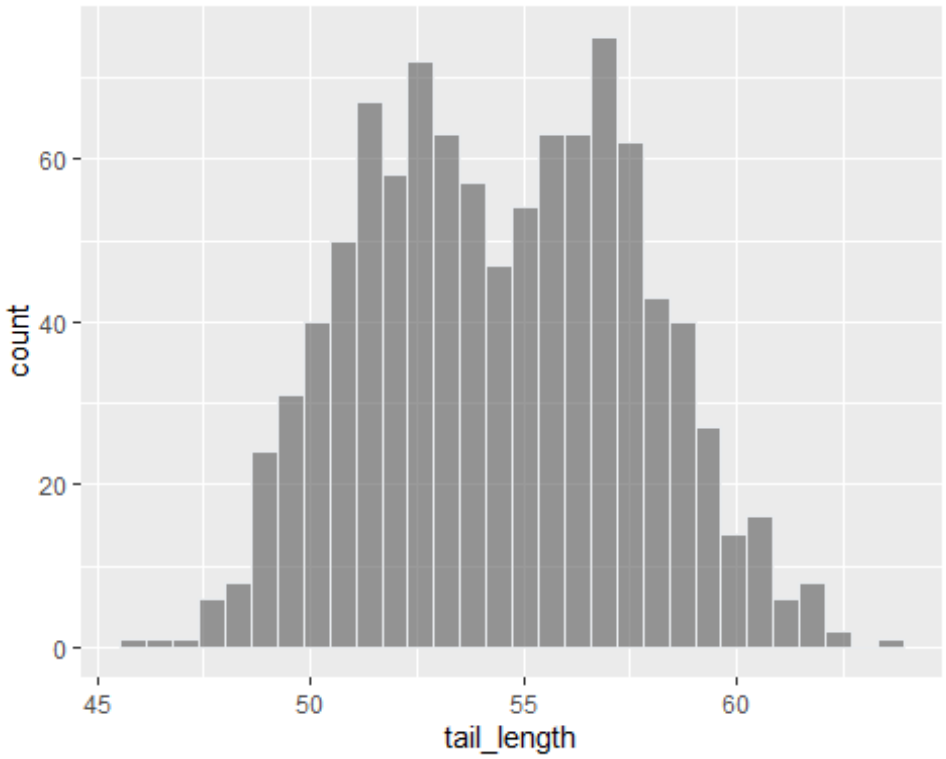
The next day you are talking to some local people and hear about another group of monkeys on a different part of the island. **First run the code below to get that data and make a histogram!**

```
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, fill=monkey_pop )) +
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 differences 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

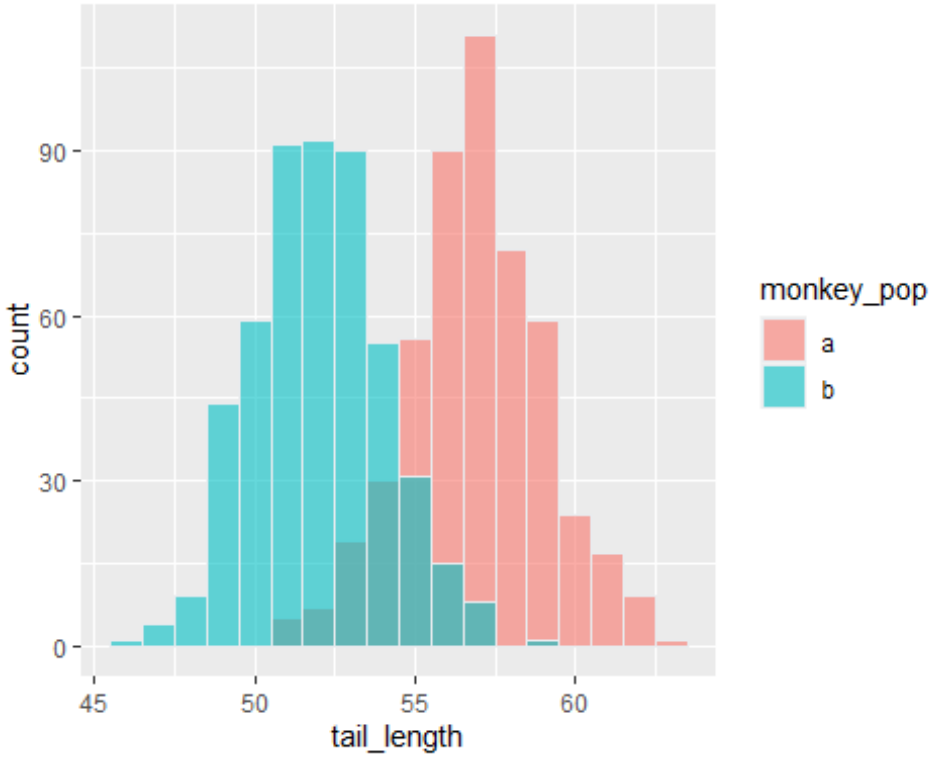
Here is what she wants you to do:

- Put the legend on the bottom rather than on the right

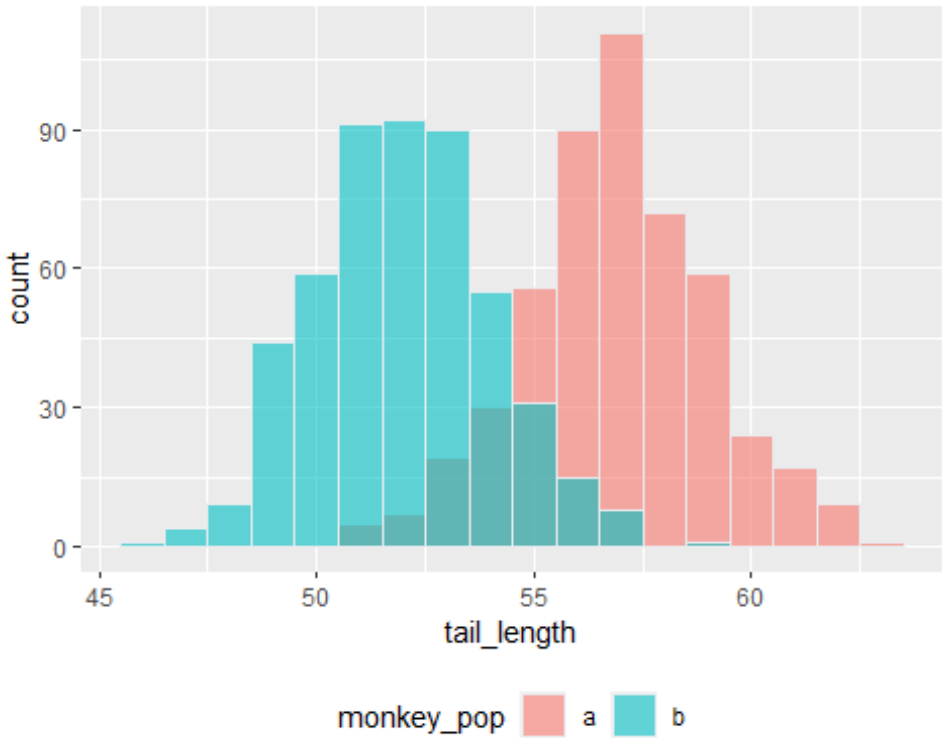
- Label the x axis “tail length (cm)”
- Label the y axis “number of observations”
- Give it a title that is meaningful
- Put a credit somewhere that says who made the figure
- Change the legend so that it says “Monkey groups”
- Change the labels from “a” and “b” to “Group one” and “Group 2”

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:

all_data %>% ggplot(aes(tail_length, fill=monkey_pop )) +

 geom_histogram(color="#e9ecef", alpha=0.6, position = 'identity', binwidth =  1) +

 theme(legend.position = "bottom")+

 labs(title="Island Monkey Characteristics",x="tail length (cm)",y="number of observations")+

 scale_fill_discrete(name="Monkey groups",labels=c("group one","group 2"))+

 annotate('text',x=62,y=110,label="by Ann-Marie Mignone", size=2)



#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:

 library(tidyverse)

all_data %>% ggplot(aes(tail_length, fill= monkey_pop)) +

 scale_fill_manual(values=c("green","orange"))+

 geom_histogram(color="white", alpha=0.6, position = 'identity', binwidth =  1) +

 theme_bw()+

 theme(legend.position = "bottom")+
```

```
 labs(title="Island Monkey Characteristics",x="tail length (cm)",y="number of observations")+

 annotate('text',x=62,y=110,label="by Ann-Marie Mignone", size=2)
```

Part 3: New visualizations with new data

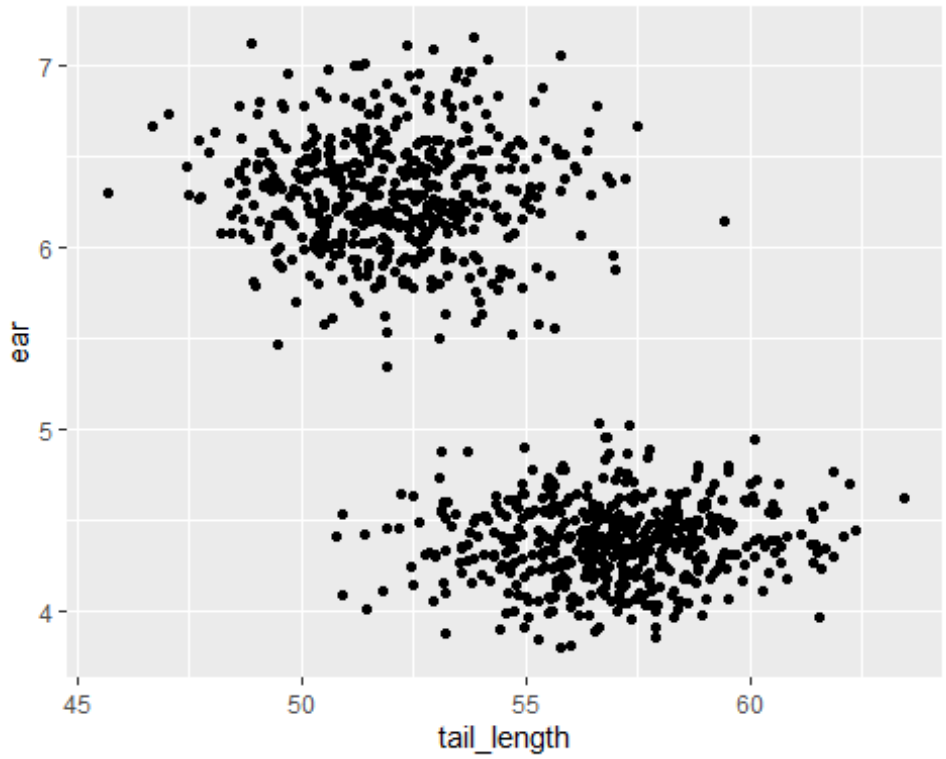
You and your professor submit your data paper to a journal. After a few months you get your peer review back and the reviewers say "wow, this is cool! But ya know what, we think you need more measurements before we can conclude these are 2 different populations. Can you go and measure ear length?"

copy and Run the following code to get the new_data

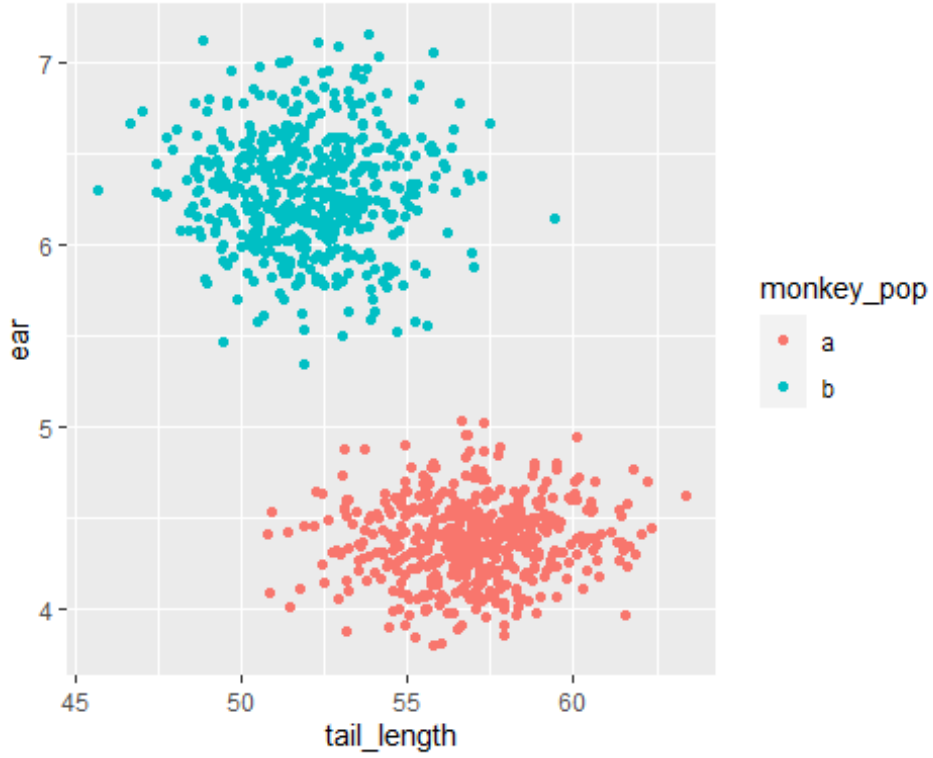
```
ear1 <- rnorm(500, 4.4, .22)
ear2 <- rnorm(500, 6.3, .31)

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 now 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
new_data %>%  ggplot(aes(tail_length, ear, color=monkey_pop,shape=monkey_pop)) +
geom_point()+
 stat_ellipse()+
 theme(legend.position="none")+
 stat_smooth()+
 geom_jitter(size=1.5)
```

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()

new_data %>%   ggplot(aes(tail_length, ear, color=monkey_pop,)) +
```

```
  geom_boxplot()


library(ggridges)
new_data %>%   ggplot(aes(x=tail_length, y=ear, height=0,color=monkey_pop)) +
  geom_ridgeline()
```

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
write_csv(new_data, file='new_primate_data')
#part b
install.packages("writexl")
library("writexl")
write_xlsx(new_data,"C:\\Users\\rebecca\\Documents\\new_primate_data.xlsx")
```

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` for more info

The second dataset is `penguins_raw`, and contains all the variables and original names as downloaded; see `?penguins_raw` 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

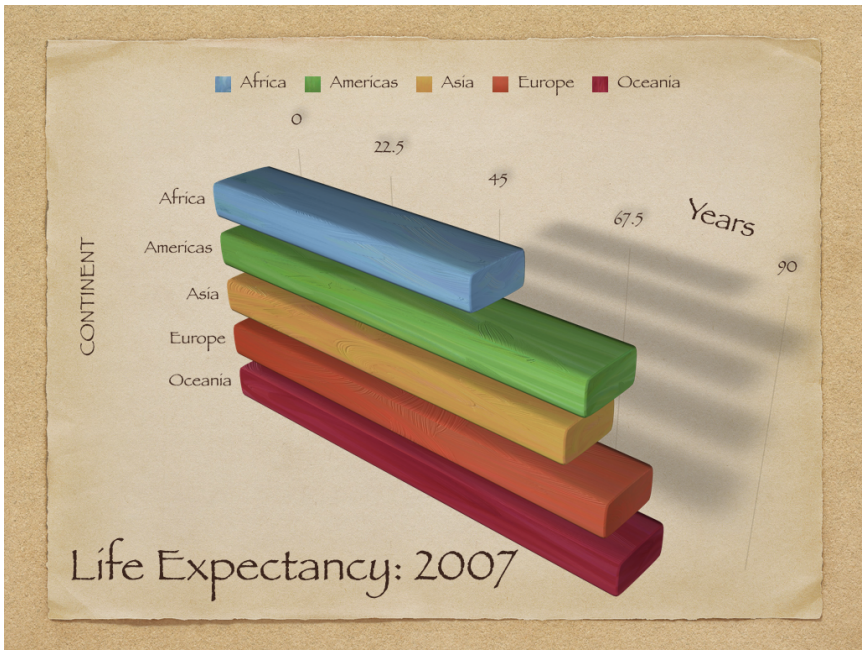
Purpose:

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

Tasks:

2. **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?

life expectancy for each continent

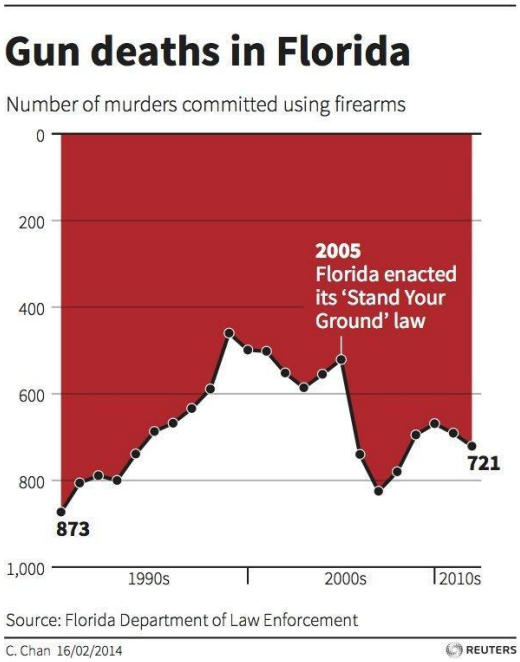
What is problematic?

The bars are coming towards the viewer, causing a skeed point of view and possibly altering how the data look compared to each other. The numerical values are tilted and hard to apply to the bars due to this depth issue.

How might someone redo this better?

Make the bars two-dimensional.

Figure B



What is this figure trying to say?

How many murders caused by gunshots occured in Florida by year.

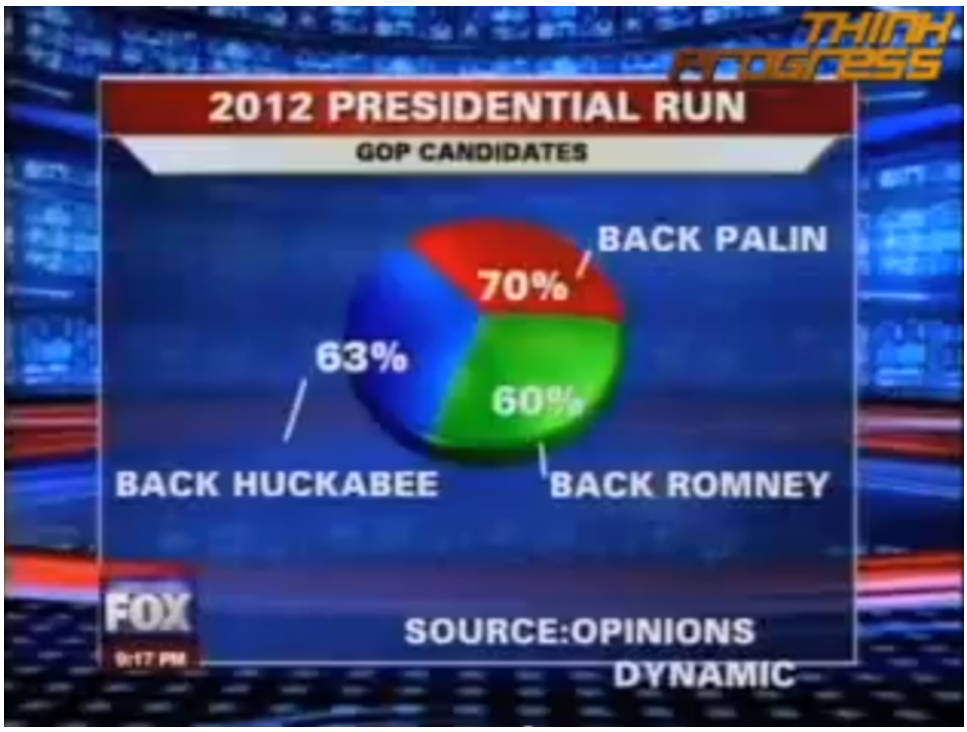
What is problematic?

The graph is? Upside down?? The y-axis variables go from high to low instead of low to high, making it easy to misinterpret the data.

How might someone redo this better?

Reverse the y-axis numerical so that at the intersection with the x-axis it starts at zero.

Figure C.



What is this figure trying to say?

What percentages of the answering population support which presidential candidate.

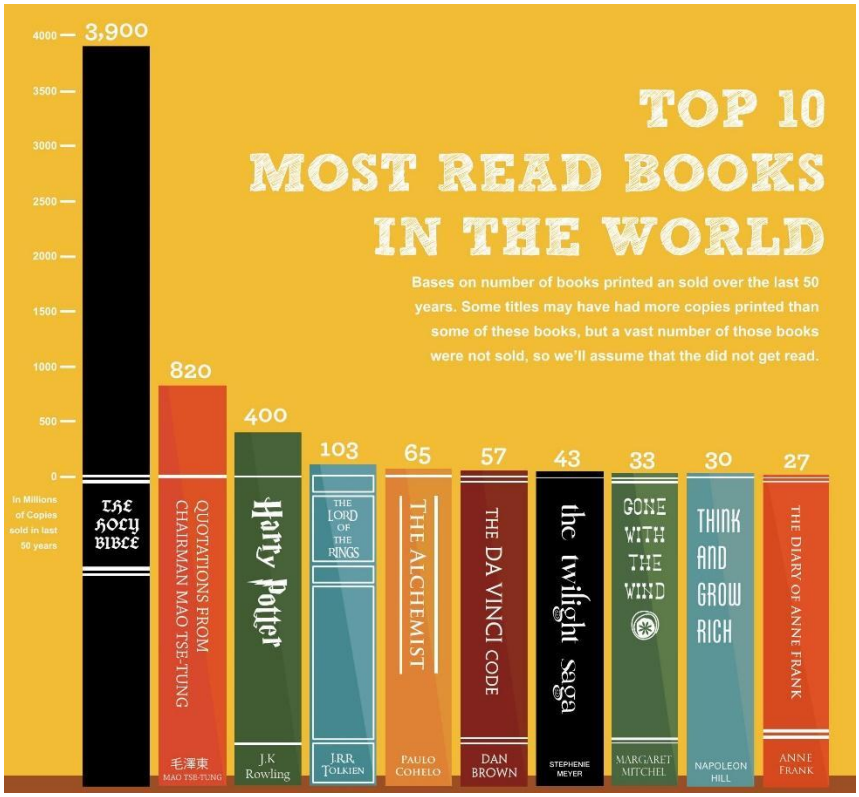
What is problematic?

The graph is tilted, causing an issue in depth perception. This makes the two smaller sections visually appear larger than the one which is actually larger by percentage. The percentages also add up to create a number greater than 100.

How might someone redo this better?

Make the graph flat to convey information more accurately. Also they should only allow one option for their test, or do some math to make the percentages out of one-hundred to make comparison easier.

Figure D.



What is this figure trying to say?

How much and in what order the most read book in the world are in ranks of reads/sells

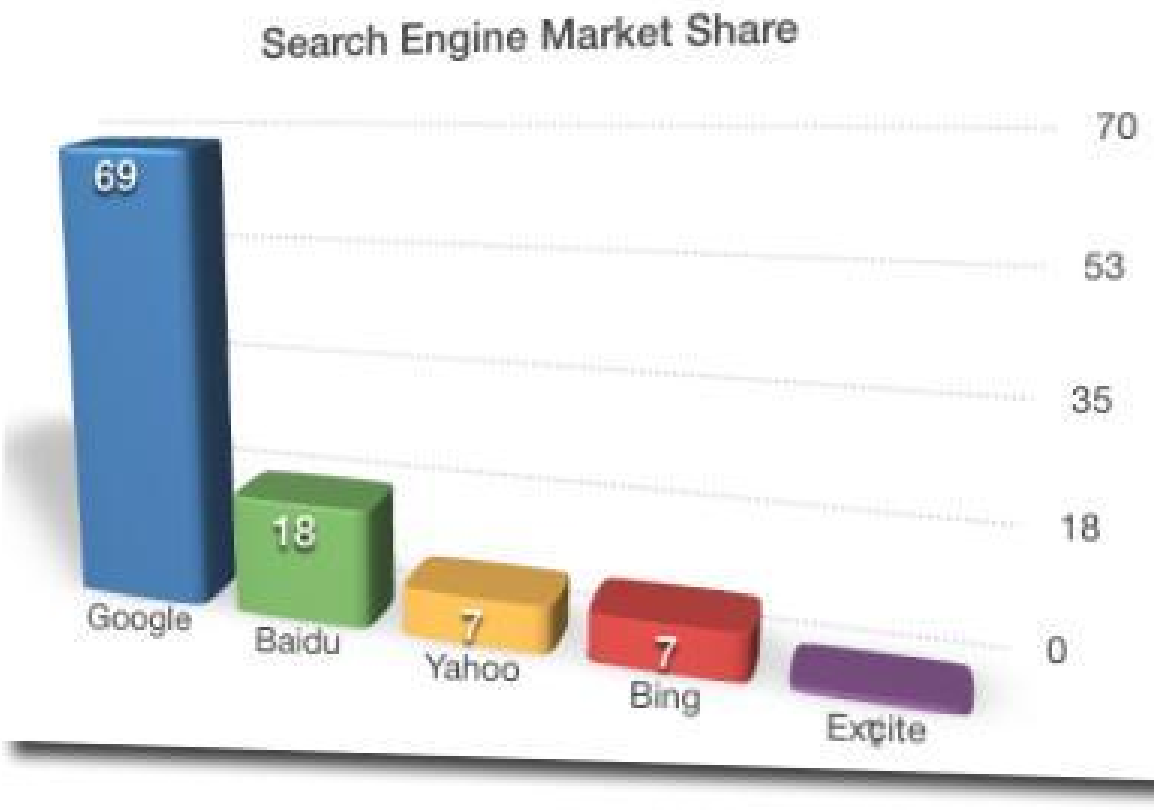
What is problematic?

Everything is really small. It's hard to easily look at the axises and compare. The graph is skewed, making it nearly impossible to visually compare the last 7 books.

How might someone redo this better?

Create a break in the y-axis to still portray the first book's numbers, but also allow a smaller unit y-axis so that the lower books are comparable to each other. Also make the axises and the text involved with them more clear.

Figure E.



What is this figure trying to say?

How much the search engines market share.

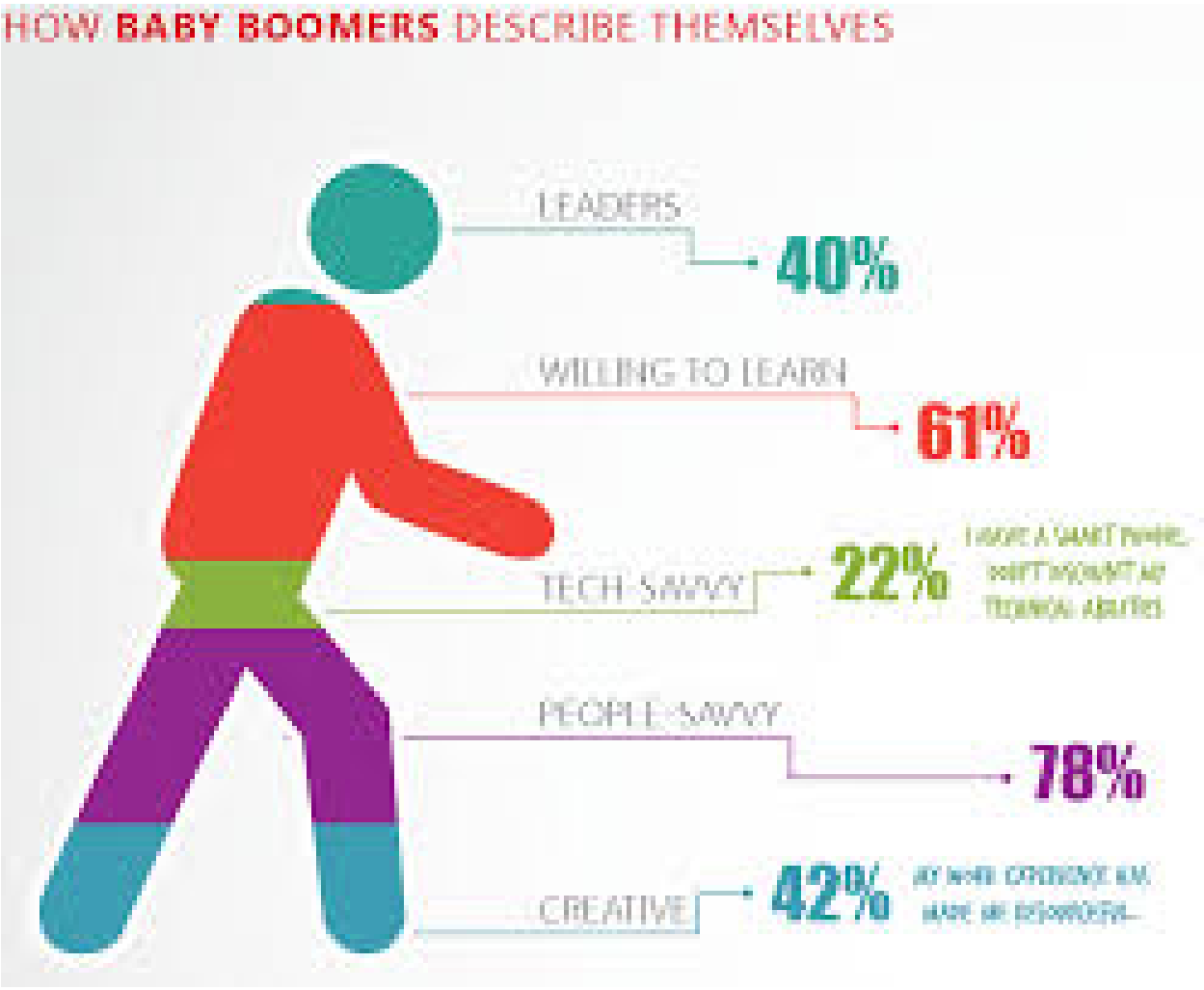
What is problematic?

NO AXIS LABELS! We don’t know the units or the value of the axises. The graph is also tilted, which as stated before causes impairment to the visual representation.

How might someone redo this better?

Supply labels, units, and values. Also make the graph flat or centred so that the visual representation is actually useful.

Figure F.



What is this figure trying to say?

What traits, and how much, baby-boomers relate with.

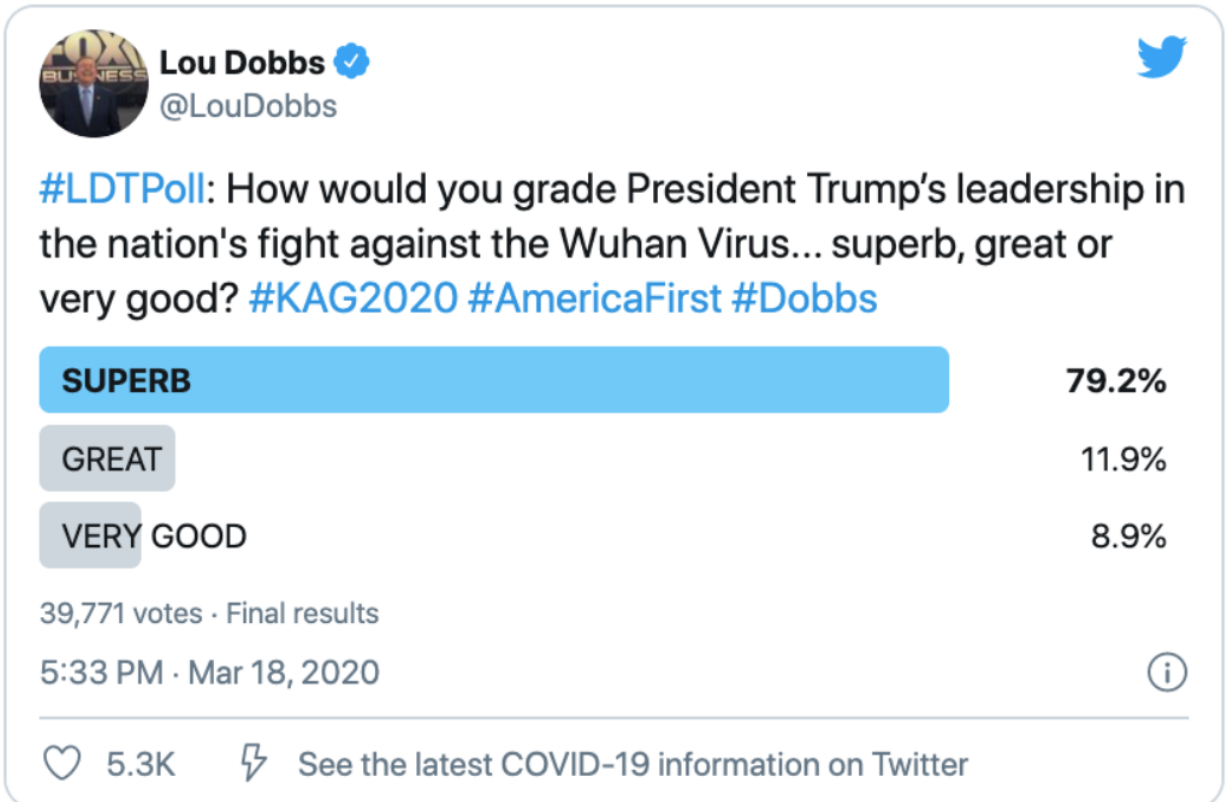
What is problematic?

Percentages are not accurately represented by size when compared to each other. The percentages don't add up to one-hundred percent. And the human silhouette disproportionally shows/cuts out the coloured slots.

How might someone redo this better?

Match the colours sizes to the percentage sizes (I'm looking at you, tiny sliver of 78%), also try to collect your data/manipulate your data to be out of 100%. Maybe use a more blocky/uniform human shape or a different shape (like a pie) all together to represent your data.

Figure G



What is this figure trying to say?

How good people think Trump in handling COVID-19.

What is problematic?

All of the answers are positive, only allowing people to rank their levels of positive feelings. Leaves out other possible choices and thus is only being answered by people who agree. Also doesn't provide definitions for the choices, so it is up to personal knowledge which can vary (some people may see VERY GOOD as better than SUPERB, or may not know what specific vocabulary words mean).

How might someone redo this better?

Provide varying levels of choices with straightforward definitions.

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?

Clear labels, contrasting colours/text/shapes (aka high readability), comparable data, straightforward/defined vocabulary

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?

A little bit, yeah!! I think I'm having a hard time understanding in which parts I can add new information.

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

I really enjoyed the scatterplots, but also highly value the histograms. My personal favourite thing is the gradient colour option to compare high and low counts. Now I just need to figure out how to manually apply it to two or more data sets!! >:)

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?

No mine did not.