Map Activity

Libraries Used

```r
library(colorspace)
library(broom)
library(tidyverse)
library(tmap)
library(maps)
library(sf)
library(knitr)
```

The Data

First, we will read in a pre-downloaded set of data from the Census API.

```r
census_data <- read_csv("census_data.csv")
```

Then, let's view the first 10 rows:

```r
kable(head(census_data, n = 10))
```

county_fips	state_abbv	county_name	county	state	rShift	popEstimate16	unemployment
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1
1001	AL	Autauga County	Autauga County	Alabama	0.8020531	55278	5.1

This dataset contains information on county (by state) population estimates, median income, unemployment, and political shifts to the right of the spectrum.

Publications and research reports that use this data must cite it appropriately by including the following information:

- Ella Foster-Molina and Ben Warren. Partisan Voting, County Demographics, and Deaths of Despair Data. February 2019.

The codebook for the dataset is:

- county_fips: Five digit Federal Information Processing Standards code that uniquely identifies counties and county equivalents in the United States
- state_abbv: State postal abbreviation
- county_name: County name, may be identical to county variable
- county: County name
- state: State full name
- rshift: Percentage difference between the Republican presidential vote in that county in 2016 and 2012. For example, 46.7955% of Kent County in Delaware (FIPS 20001) voted for Romney in 2012. In 2016, 49.81482% of that county voted for Trump. Therefore, the county shifted towards the Republican presidential candidate by 3.01325%. Positive value mean leaning more Republican; negative values mean leaning less Republican.
- popEstimate16: Population in the county in 2016
- unemployment: Unemployment rate in 2016
- medianIncome16: Median household income in the county in 2016.
- prcntGOP16: Percent of the county that voted for the Republican presidential candidate, Donald Trump, in 2016.

Graphics

Once we have the data set imported to R, we can begin to visualize some of the data.

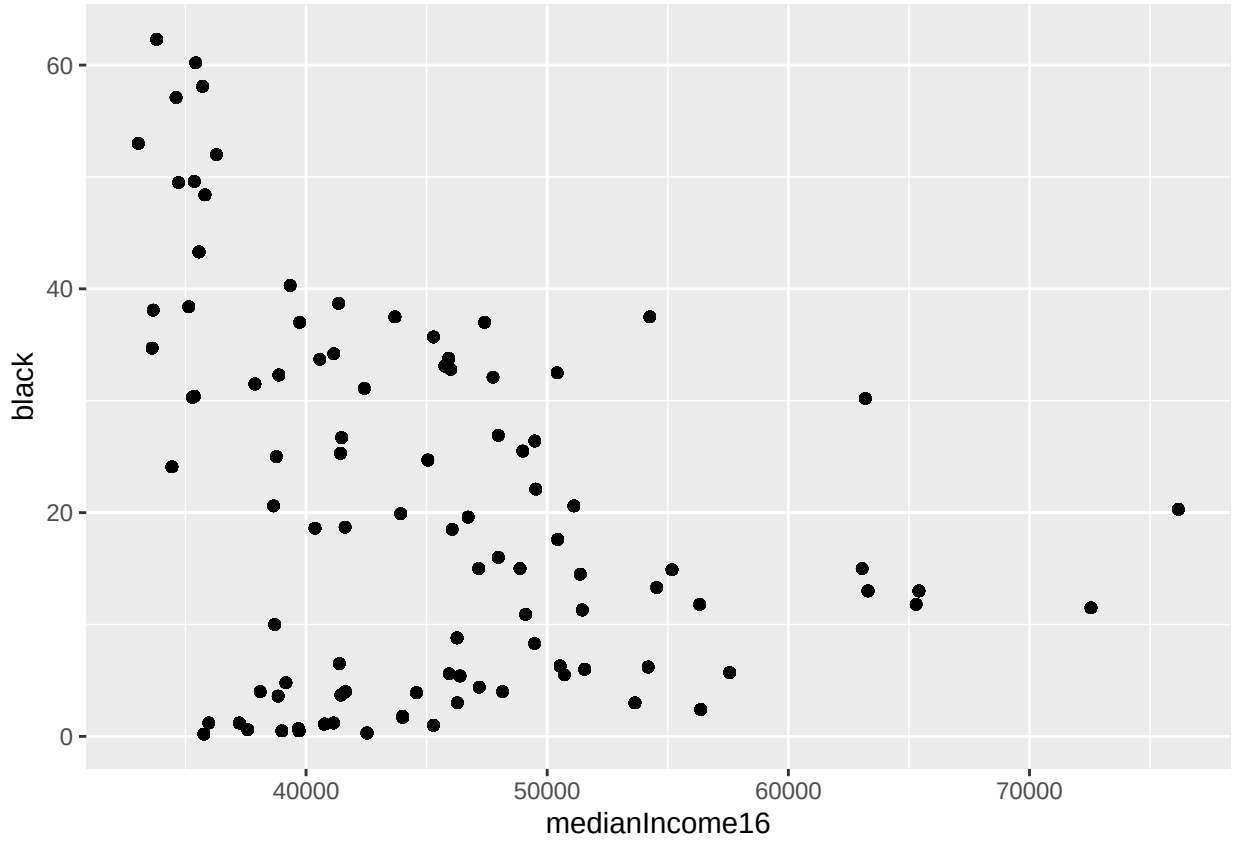
Creating a ggplot

Before we start looking at relationships between variables, let's take this very large dataset and only look at county information for North Carolina.

```
# elements selection from the data frame census_data
NC_data <- subset(census_data, subset = (state == "North Carolina"))
```

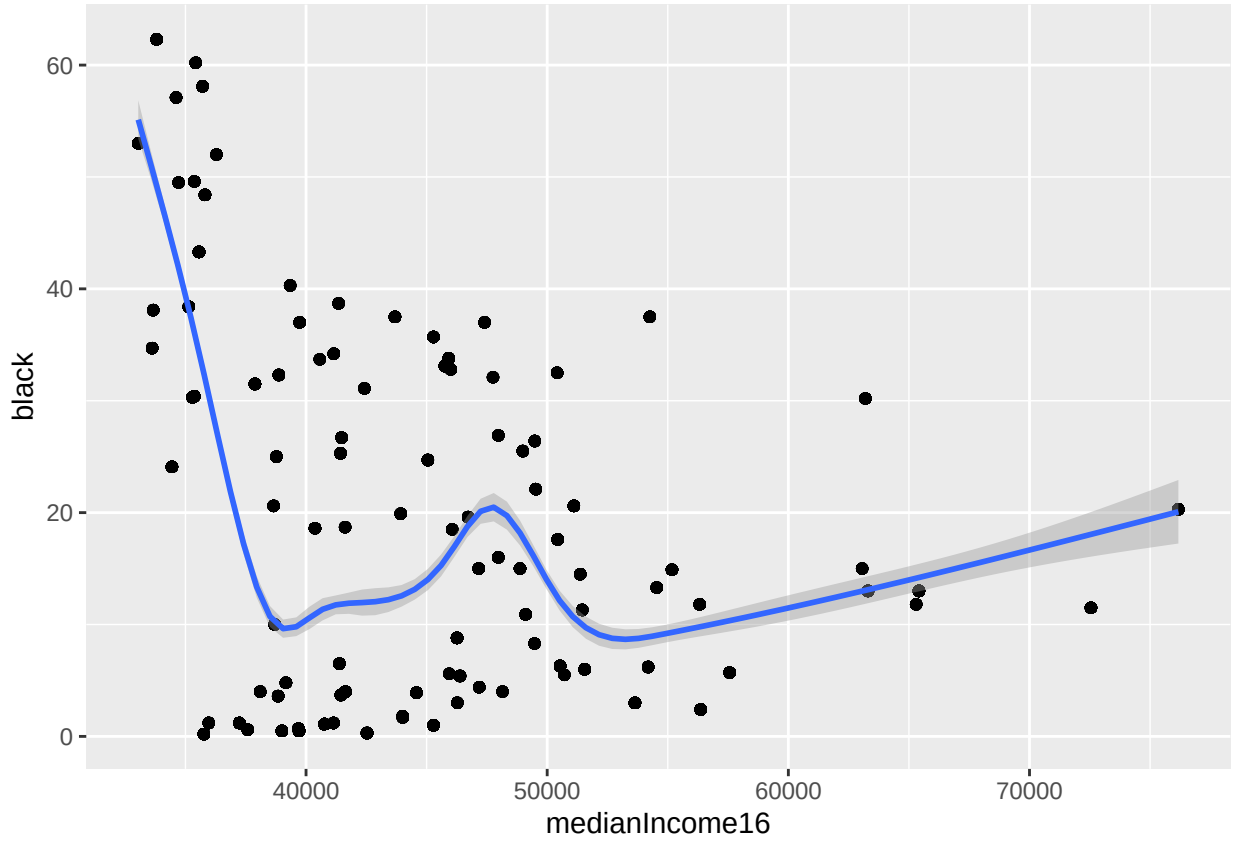
To plot the North Carolina census data and understand the relationship between two of the variables, run this code to put medianIncome16 on the x-axis and black on the y-axis:

```
#create a scatterplot with no trend line
ggplot(data = NC_data) +
  geom_point(mapping = aes(x = medianIncome16, y = black))
```



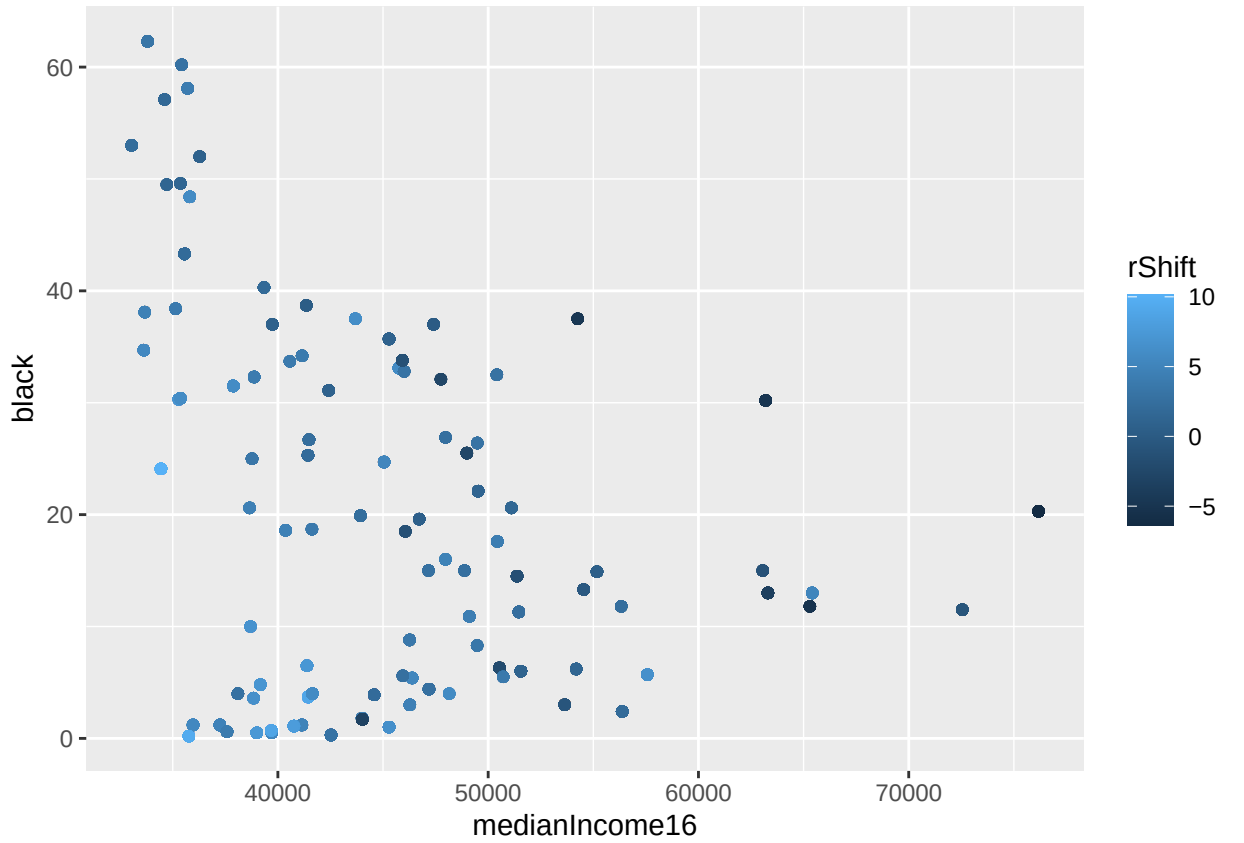

```
#create a scatterplot with a trend line
ggplot(data = NC_data) +
  geom_point(mapping = aes(x = medianIncome16, y = black)) +
  geom_smooth(mapping = aes(x = medianIncome16, y = black))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



The scatterplots here shows a slightly negative trend between median income and black ethnicity, suggesting that as percent black decreased, median income increased in 2016 by county. We can observe the same trend with an additional variable by using color to categorize. To use color, let's look at rShift by county in North Carolina with the following code:

```
ggplot(data = NC_data) +
  geom_point(mapping = aes(x = medianIncome16, y = black, color = rShift))
```



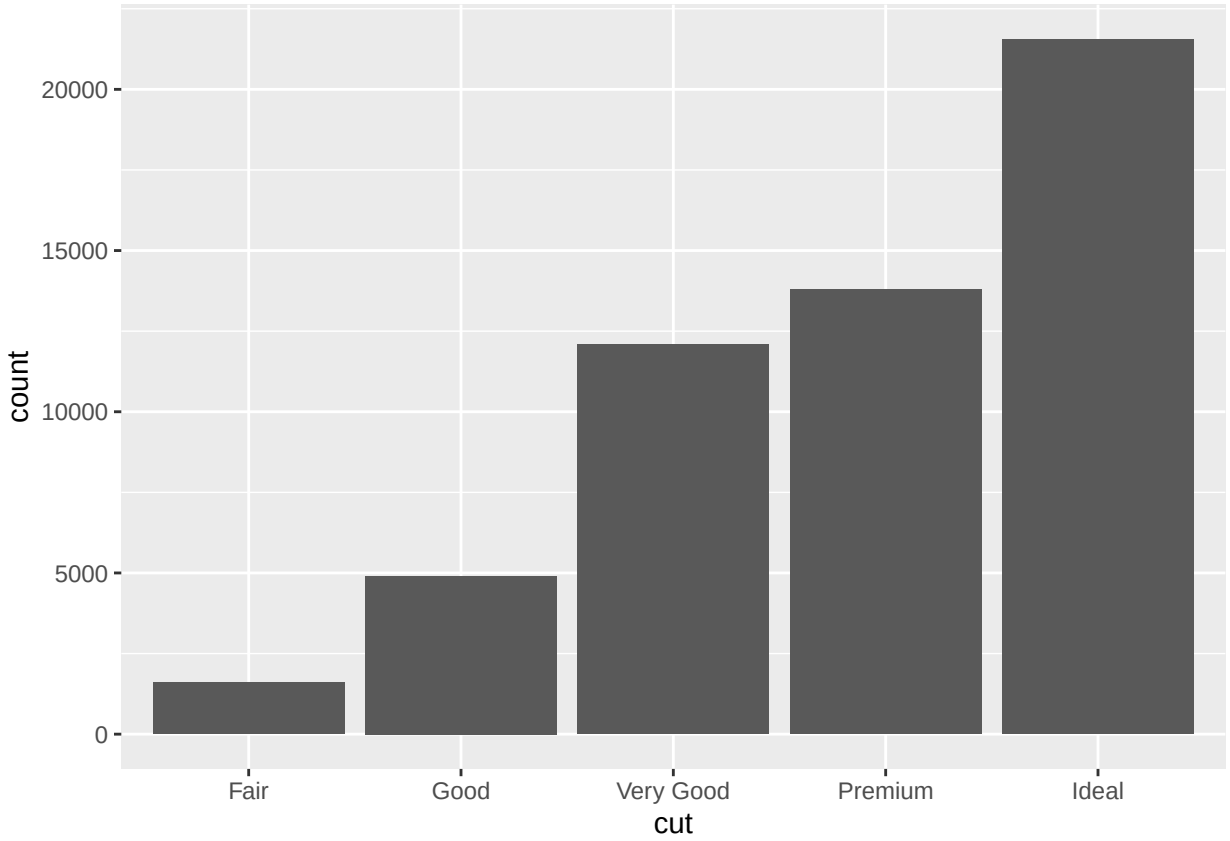
We now see that outlier counties - with median incomes over $60000 - are more likely to shift away from the republican party and favor the democratic party. This is seen with the darker blue representing more liberal counties and the lighter blue representing more conservative counties. Trends like this can be observed with any combination of variables, but some variables will have stronger relationships than others.

Bar Charts

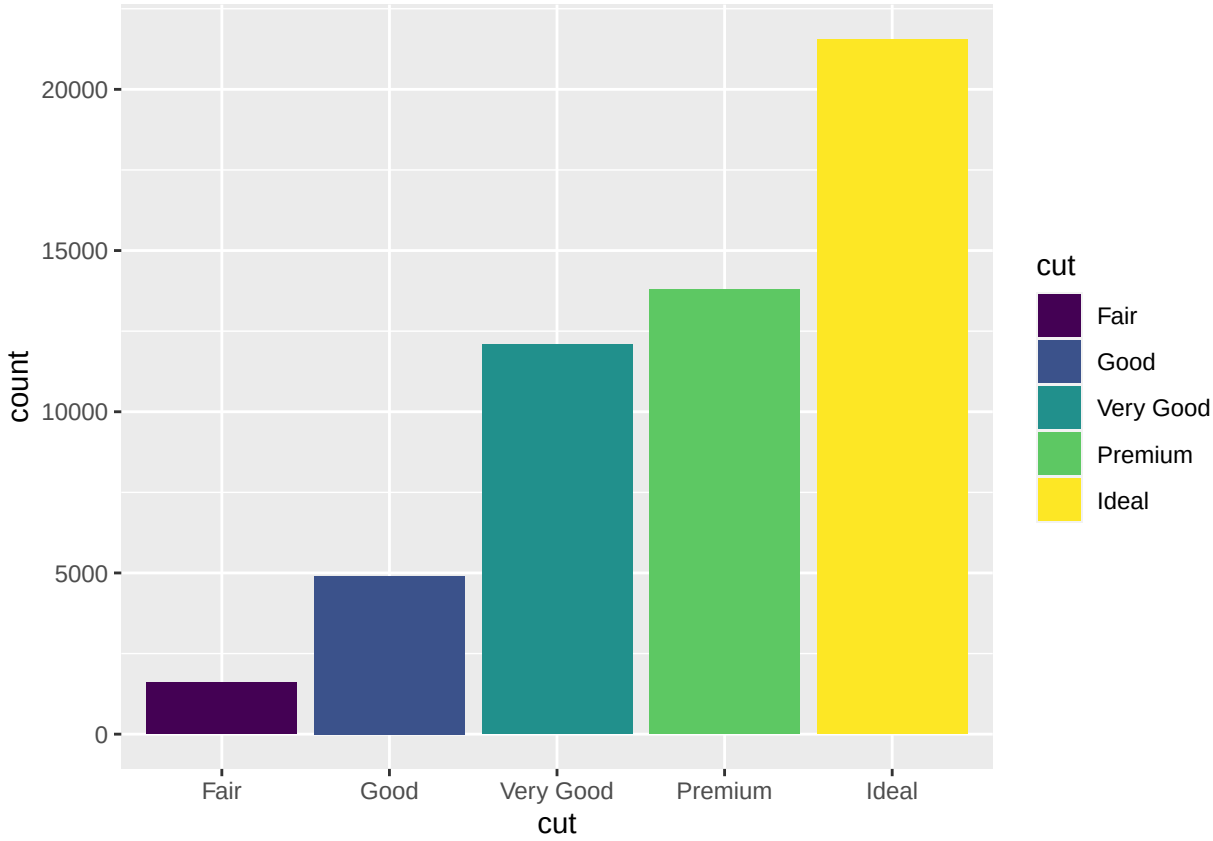
When visualizing data, histograms are used to show distributions of variables while bar charts are used to compare variables. Thus, histograms plot quantitative data while bar charts plot categorical data

Let's quickly go over how to make a bar chart with a different dataset called diamonds. The diamonds dataset comes in ggplot2 and contains information about ~54,000 diamonds, including the price, carat, color, clarity, and cut of each diamond. Follow the code below using geom_bar() to show the total number of diamonds in the diamonds dataset, grouped by cut:

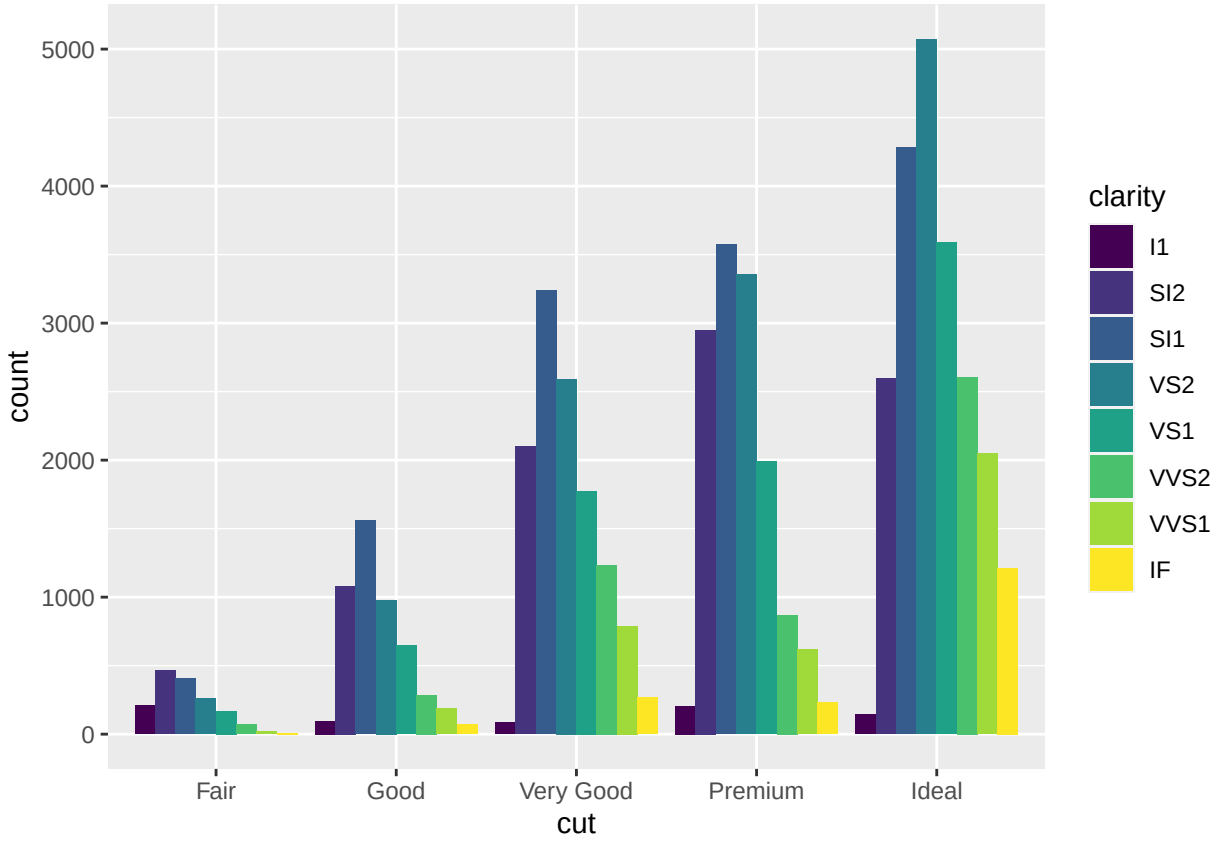
```
#basic bar chart
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```




```
#bar chart with cut color coded
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut))
```




```
#bar chart with third variable, clarity, compared within each cut type side-by-side
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge")
```



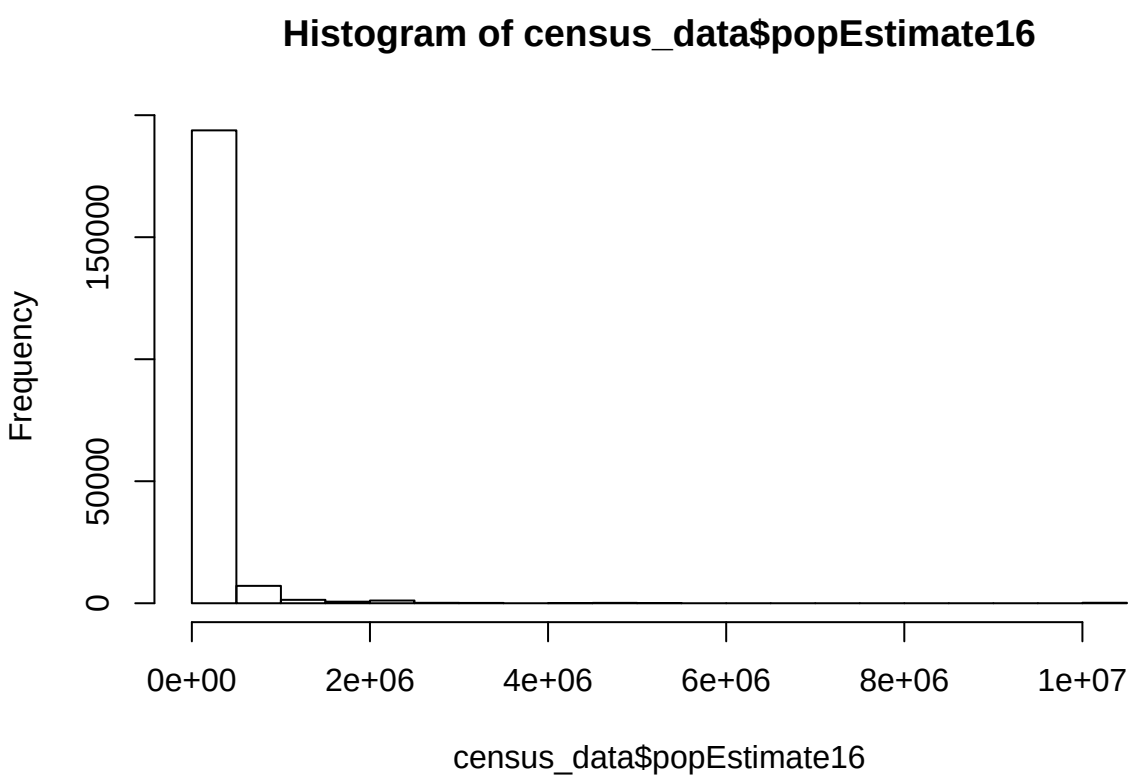

```
#ggplot(NC_data, aes(x=county, y=medianIncome16)) + theme(legend.position="none")

#barplot(census_data$state_abbv)
```

The bar charts show that more diamonds are available with high quality cuts than with low quality cuts (Hadley).

Histogram

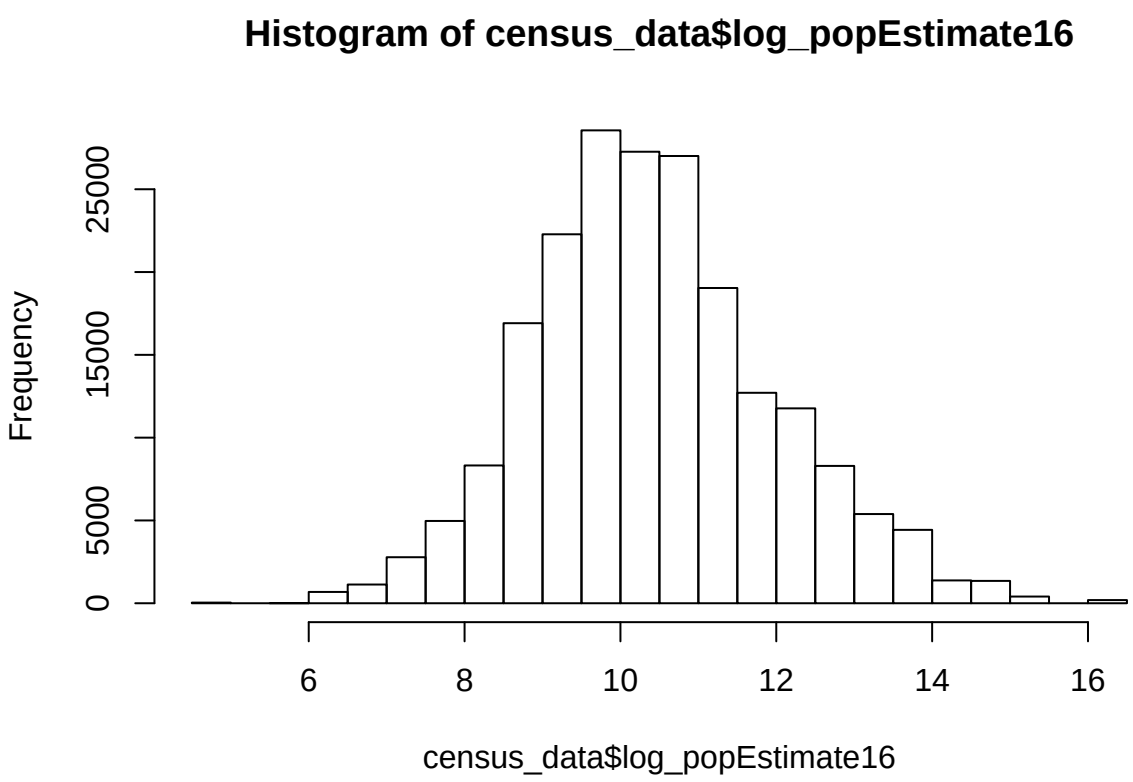
However, it is quite simple to make a histogram using the hist() function. Let's use our census_data dataset and take a look at the population estimates for 2016:

```
hist(census_data$popEstimate16)
```



Not much information can be gathered from a histogram at this scale. It might be more helpful to change the scale of popEstimate16. We can do that with an easy call to the log() function and save it as a new variable to the census_data dataset.

```
#create log_popEstimate16 variable
census_data$log_popEstimate16 <- log(census_data$popEstimate16)

#histogram
hist(census_data$log_popEstimate16)
```



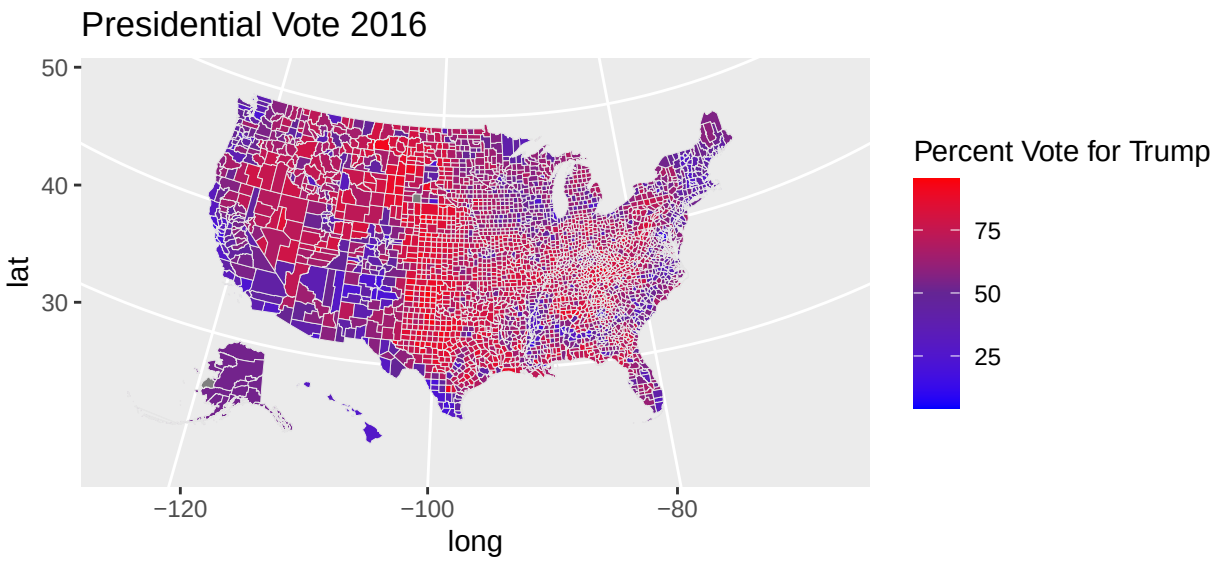
Much more information can be gathered from this histogram. We see a fairly normal distribution in the estimated population for 2016.

Choropleth Map

Using ggplot2, we can create an aesthetically pleasing and information filled map using census_data:

```
ggplot(
  data = census_data
  , aes(
    x = long
    , y = lat
    , group = group
    , fill = prcntGOP16
  )
) +
  geom_polygon(
    color = "gray90"
    , size = 0.05
  ) +
  coord_map(
    projection = "albers"
    , lat0 = 39
    , lat1 = 45
  ) +
  scale_fill_gradient2(
    low = "blue"
```

```
    , mid = scales::muted("purple")
    , high = "red"
    , midpoint = 50
  ) + labs(
    title = "Presidential Vote 2016"
    , fill = "Percent Vote for Trump"
  )
```



Using colors, we can see the counties across the country that had a large or small percentage vote for Trump in the 2016 election. This shows a massive amount of data in a visual format that is easy to understand. Ultimately, visualizing large amounts of data using charts or graphs is more effective and efficient than trying to see trends from raw numbers in spreadsheets. Visualizing data with graphs and charts can draw attention to data points worth continued exploration.

Independent Activities

1. Use the provided data set (which can be downloaded from github) or diamonds and create some graphics of your own!

- Use a histogram, scatter plot, or bar chart to explore different variables in the data set. Once you have found a variable that you are interested in, create a map of the entire US as well as a map of a particular state using it following the code used.

2. Review the tidyCensus package documentation to pull data of your own.

- Explore the possible variables (there are many!) and think of a question that you can answer using data visualization. Once you have a question to try to answer, pull the variables of interest. Remember, you may have to change the scale of some variables or mutate them in another way. Once your data is ready for graphing, have at it! Try out different graphs and see which ones lead you to an insightful conclusion.