Ggplot Homework

Simeon Markind

2017-03-07

Date Assigned: March 10, 2017
Date Due: April 6, 2017 by 11:59 pm

Introduction

For this homework you are going to use ggplot to reproduce the following plots. For each question I have generated what the answer should look like, you have to figure out how to use ggplot to make it!

For questions that ask you for commentary in addition to or in place of a plot please write comments in the code right below your code for the corresponding plot.

For each plot you produce assign the plot object to a variable associated with the corresponding questions. i.e:

question1a <- ggplot(...) + ...

Additionally, be sure to include in the subtitle of the chart the question that the plot is created for. Your charts must all be appropriately titled, (chart titles must be descriptive and accurate for the data displayed and no longer than 8 words), and axes/guides must be labeled appropriately as well.

You should use dplyr for your data manipulation and must use ggplot for creating your plots.

Data

For this homework you will need to read in the treasuries, national_election_results, the oh_election_results, and stock_closings csv files in the ggplot Homework folder.

Layering

We will start by reviewing how to add chart elements to a graphic by using ggplots layering grammar. Let's take a look at the unemployment data in the treasuries.csv file.

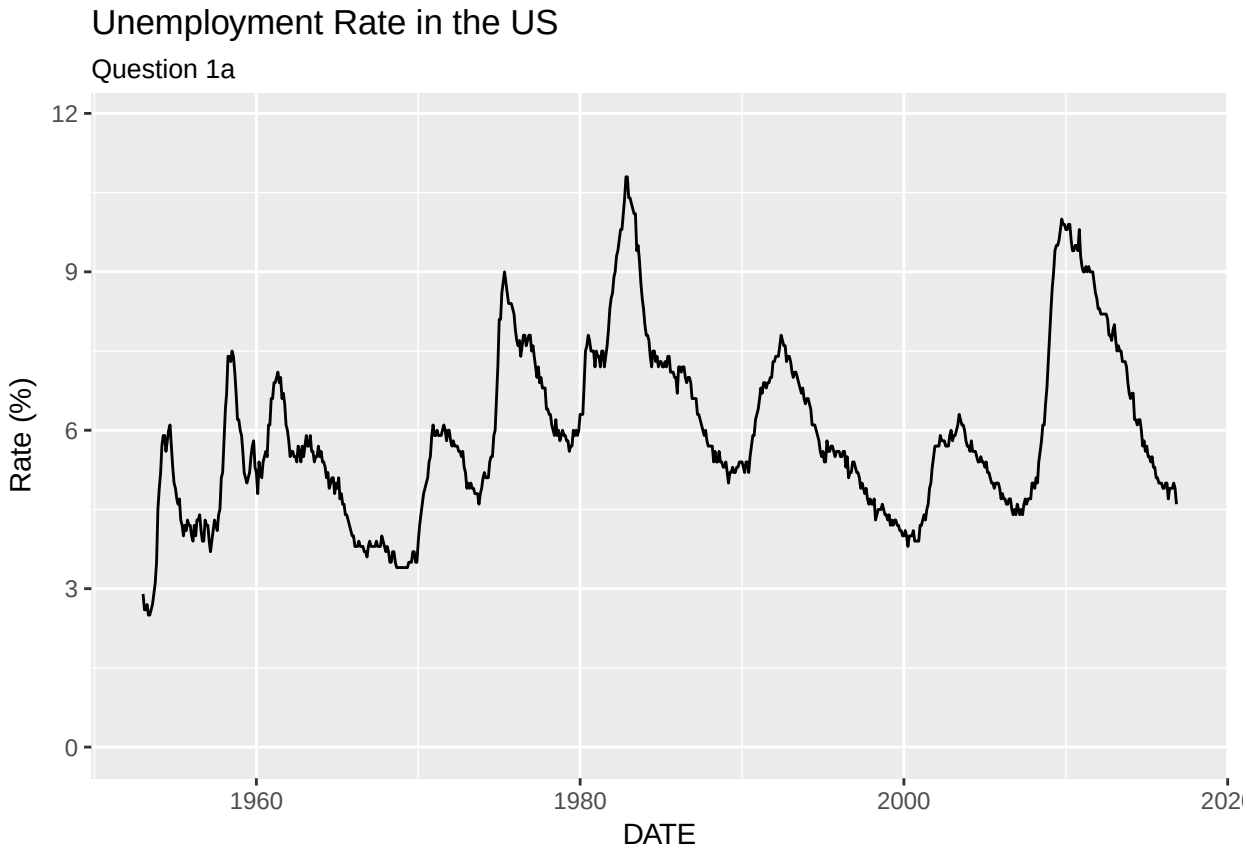
Question 1a.

Using the UNRATE data in the treasuries file create an appropriately labeled graph of the unemployment rate over time from 1953 onward. What is the value of this chart? Does the chart point to anything about the cyclicality of employment? What further questions do you have upon viewing this chart? Answer these questions in text below the graphic.

I am including the chart/code below as a template for what your answers should look like throughout this assignment.

```
## Set the theme to theme_gray here as when we change it later we don't want the different themes to ov
theme_set(theme_grey())

## basic lineplot of unemployment rate data from the treasuries file
```

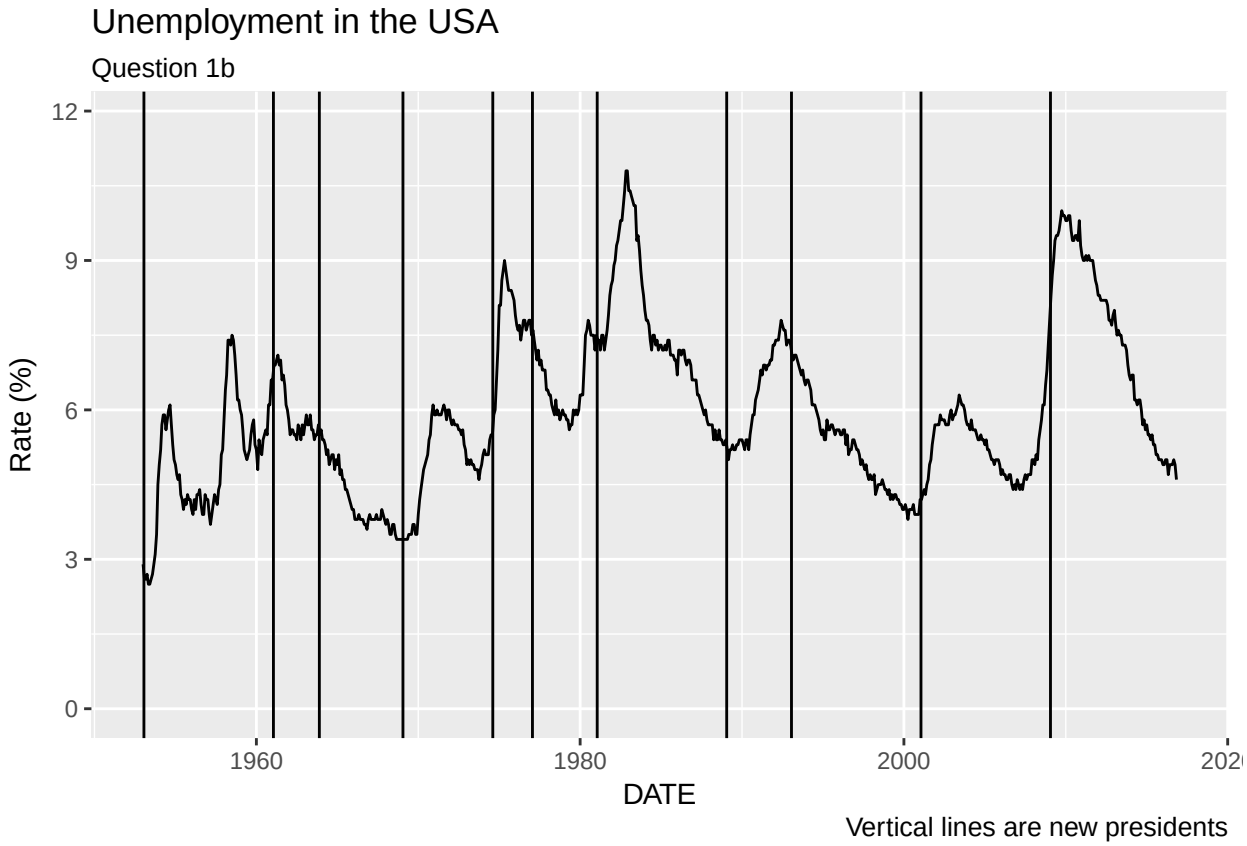
```
treasuries <- as.tbl(treasuries)
plot.data <- treasuries %>% filter(DATE >= as.Date("1953-01-01")) %>%
    select(DATE, UNRATE)

question1a <- ggplot(plot.data, aes(x = DATE, y = UNRATE)) +
    geom_line() +
    scale_y_continuous(name = "Rate (%)",
                       limits = c(0, max(treasuries$UNRATE) + 1)) +
    ggtitle("Unemployment Rate in the US",
            subtitle = "Question 1a")

question1a
```



The chart seems to hint at underlying cyclicality in employment cycles but without context the value of the chart is weak. For example, how does this US employment chart compare with other similar countries over time.

Question 1b.

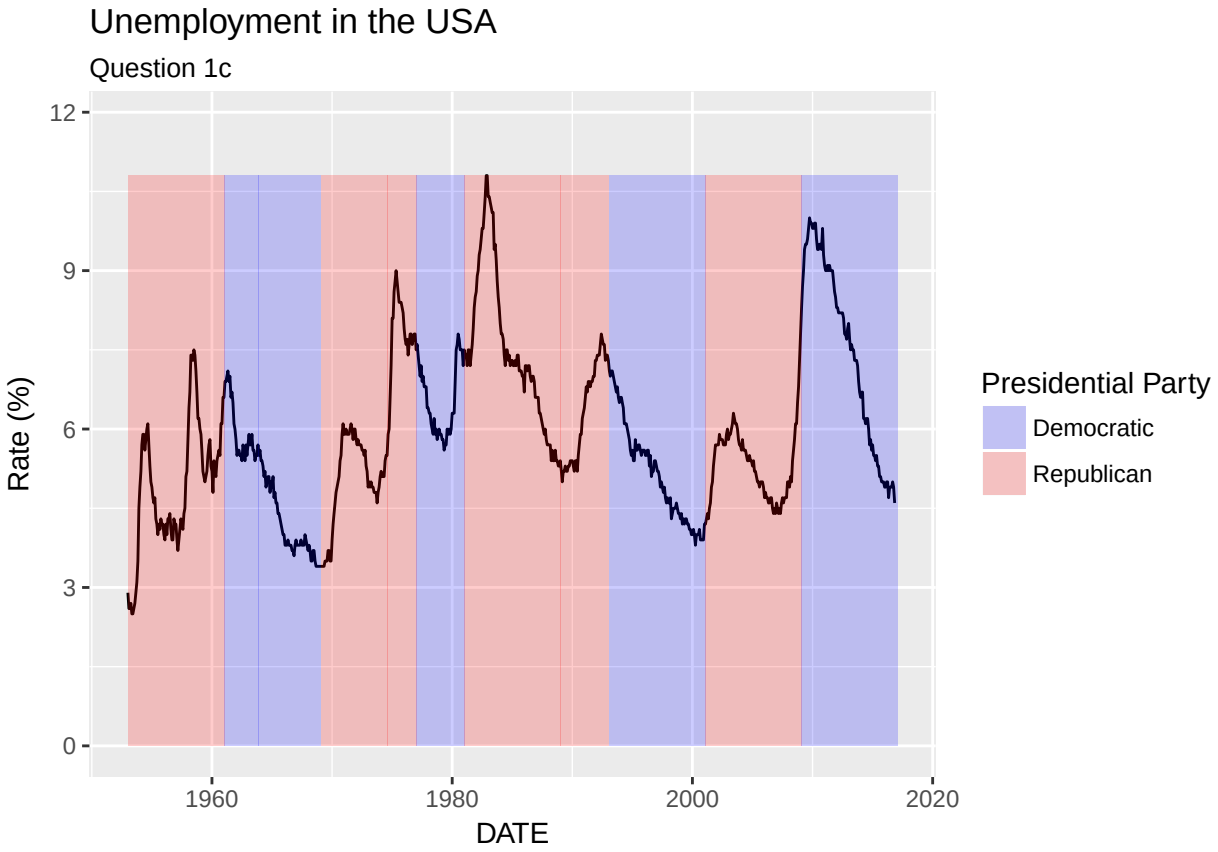
Let's see how the full range of unemployment data looks by presidential cycle. Using the "presidential" dataset included with ggplot2, graph vertical lines for the start of each president's time in office. You will want to use the geom_vline function with the xintercept argument to add the vertical lines. (You will need to use the as.numeric function around the start dates to graph them without throwing an error). Be sure to add a caption explaining what the lines are and add text below the chart discussing the value of the chart as well as further questions.



Question 1c.

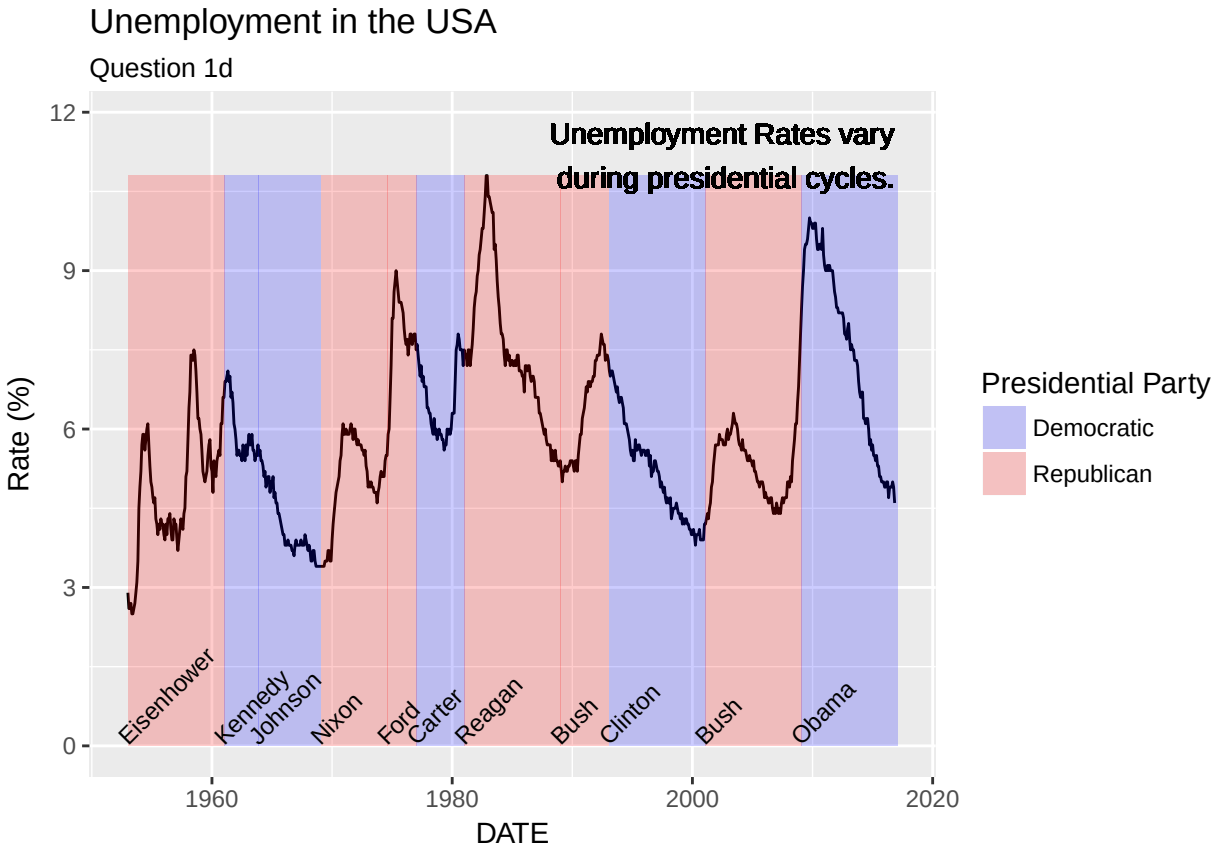
Now let's instead take a look at the unemployment rate when different parties controlled the white house instead of different individuals again using the presidential dataset. You will want to use the geom_rect function in ggplot. For the geom_rect aes call, your xmin values should be "start" values and your xmax values should be "end" values. Your y-values should be from the bottom of the plot to the top of the plot, I leave it up to you to decide how to code that in. Your fill values should correspond to the presidential "party." Additionally add the following scale_fill_manual call to your plot: `scale_fill_manual(values = alpha(c("blue", "red"), 0.2))`

Be sure to answer what you think of this plot, do you see any evidence for employment corresponding with the party in power? What further information would you like?



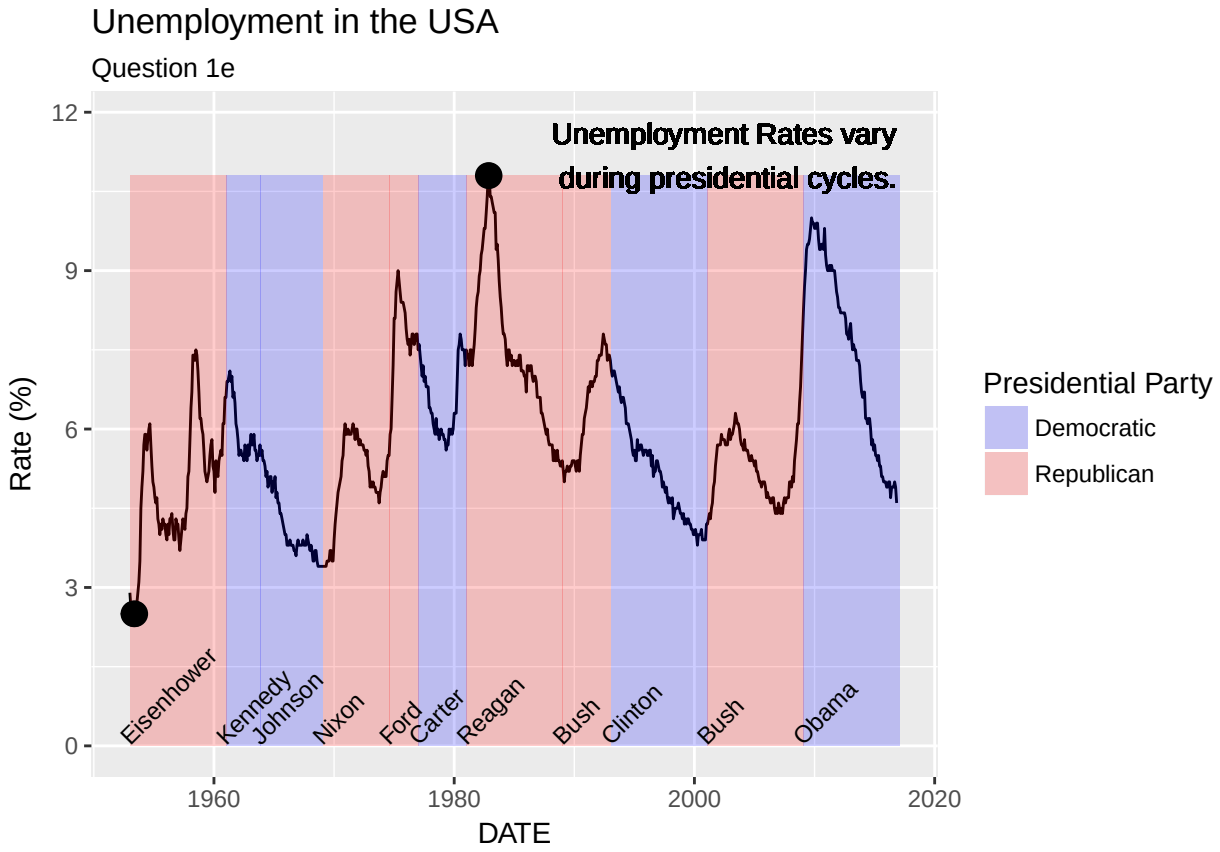
Question 1d.

We just filled in the background using the geom_rect. Now use geom_text to fill in the name of the president at the bottom of the chart at the appropriate date. Make sure to angle your text so that each name can be clearly seen. Additionally, add a caption in the top right of your chart which says "Unemployment rates vary during presidential cycles." Your caption should take up two lines. Use the \n symbol in your text string to insert a line break.



Question 1e.

For the last addition to this chart, add a point at the highest and lowest unemployment levels to chart 1d. What do you think of this final chart? What message, if any does the chart convey? Do you think this message could be misleading and our chart points to a conclusion that the data do not support?



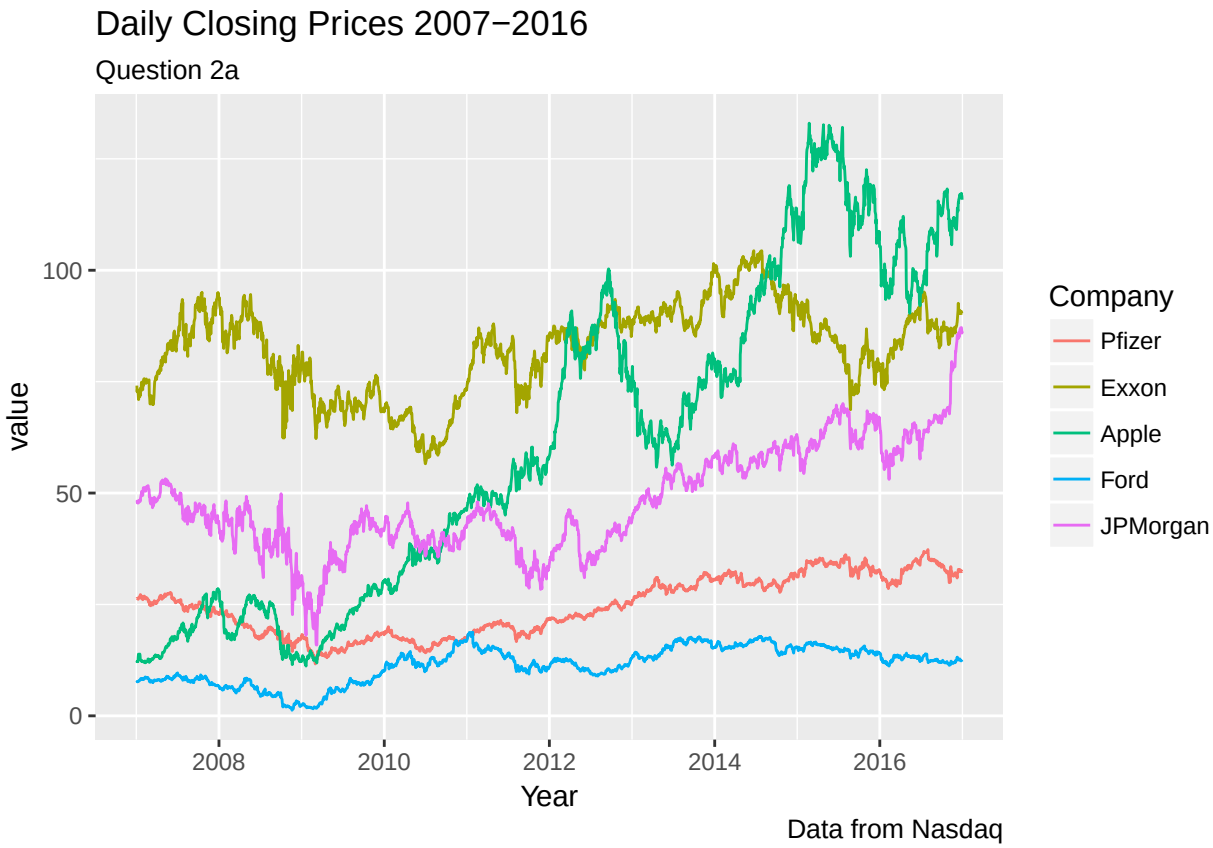
Stock Portfolio Analysis

For this next section we will concentrate on learning about different types of geoms available in ggplot as well as how to deal with overlapping data. For this segment we will need the stock_closings and treasury csv files.

Question 2a.

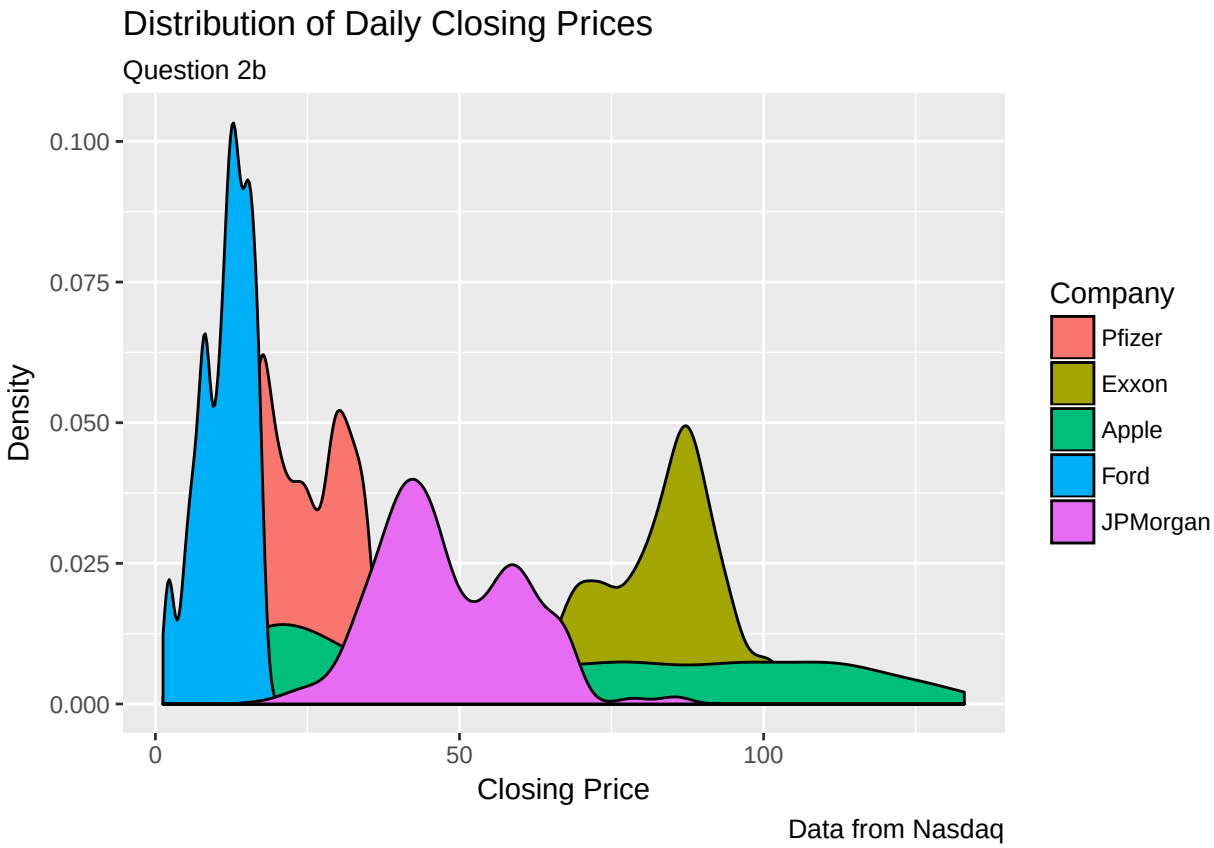
Let's start by just looking at the stock_closings csv to get a sense of the data. Print out the first 3 rows of the dataset and remark on the data. (Frequency, is it all the same format, how long is the series, is it sorted the way we want, what type of work will we need to do on the data in order to use it?). Construct an appropriately labelled line plot of the daily closing prices for each company differing color by company. Be sure to include a caption indicating data source (stock data source is Nasdaq). In a comment below the chart discuss your takeaways.

date	Pfizer	Exxon	Apple	Ford	JPMorgan
2016-12-30	32.48	90.26	115.82	12.13	86.29
2016-12-29	32.49	90.35	116.73	12.23	85.89
2016-12-28	32.35	90.30	116.76	12.25	86.50



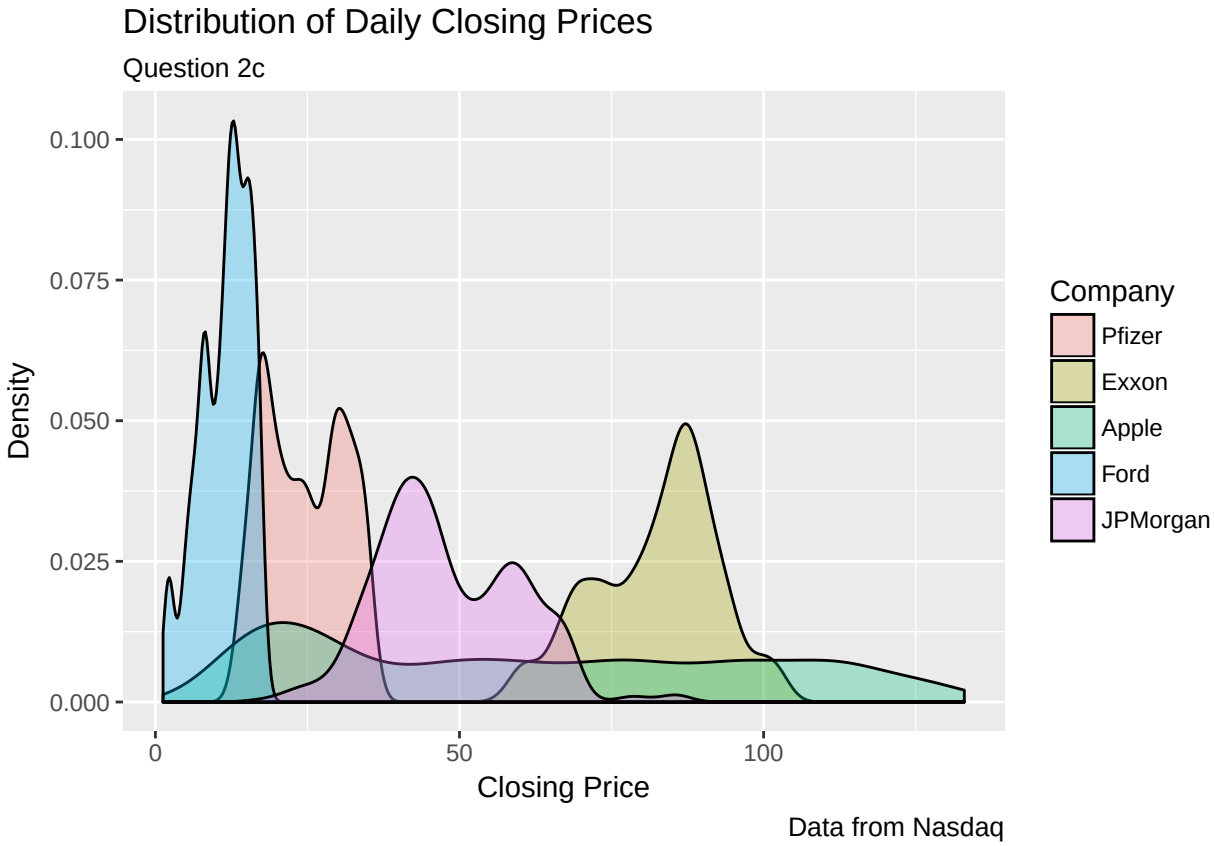
Question 2b

So let's look at the same data in another way. Now let's look at the distribution of closing prices for each company over the period. Instead of regular line plots, we will make smoothed histograms using geom_density. Set the fill for each density distribution to be the different company. Take a look at the chart, what does this chart show? What are some problems with this display?



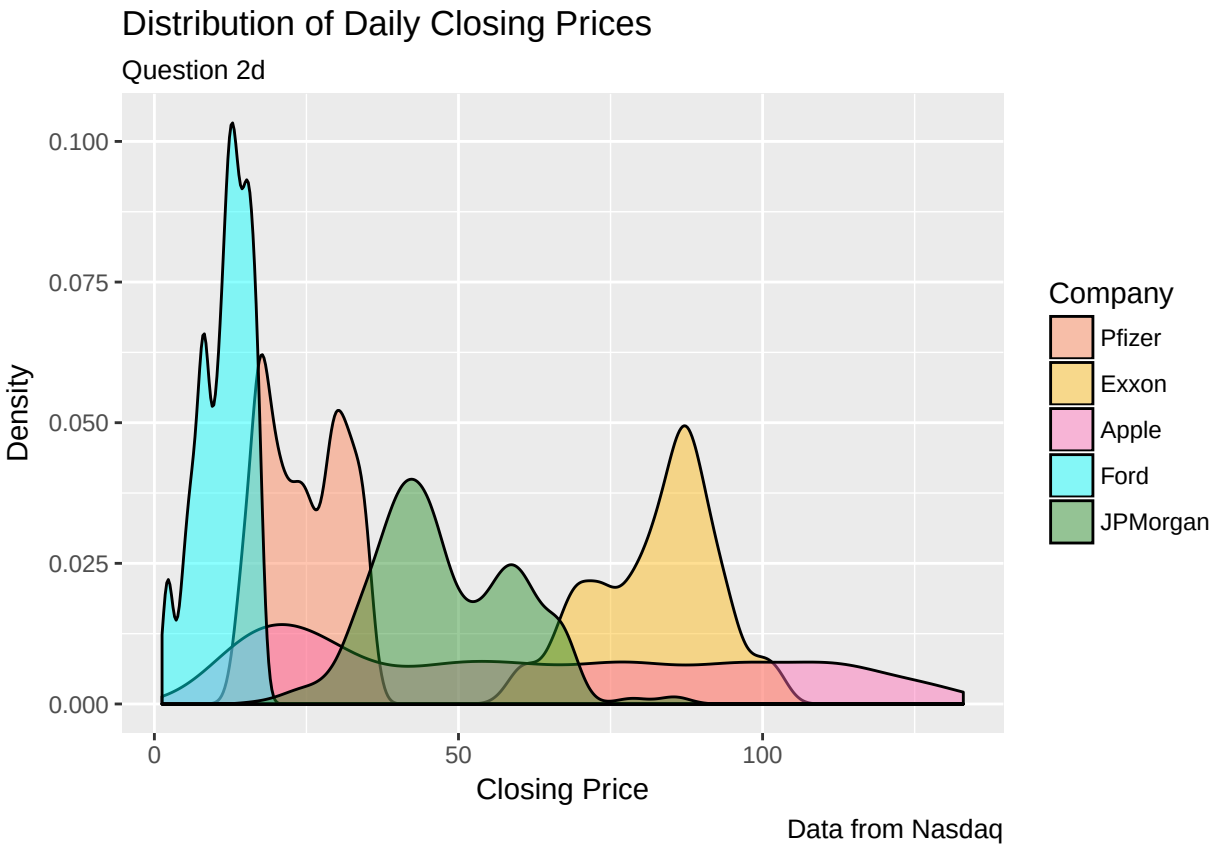
Question 2c.

All the density distributions overlap! We need a way to make each curve somewhat transparent so that we can see all the curves beneath it. This is where the "alpha blending" feature of ggplot comes in. Check out `?alpha` for more information on the alpha function in the scales package. One way to use alpha is to just specify `alpha = number` in the geom call of the ggplot. Try that now, add an alpha argument to the geom_density call. Alpha can take a number of 0 (clear) to 1 (opaque). Make sure that your value allows you to see each density distribution distinctly.



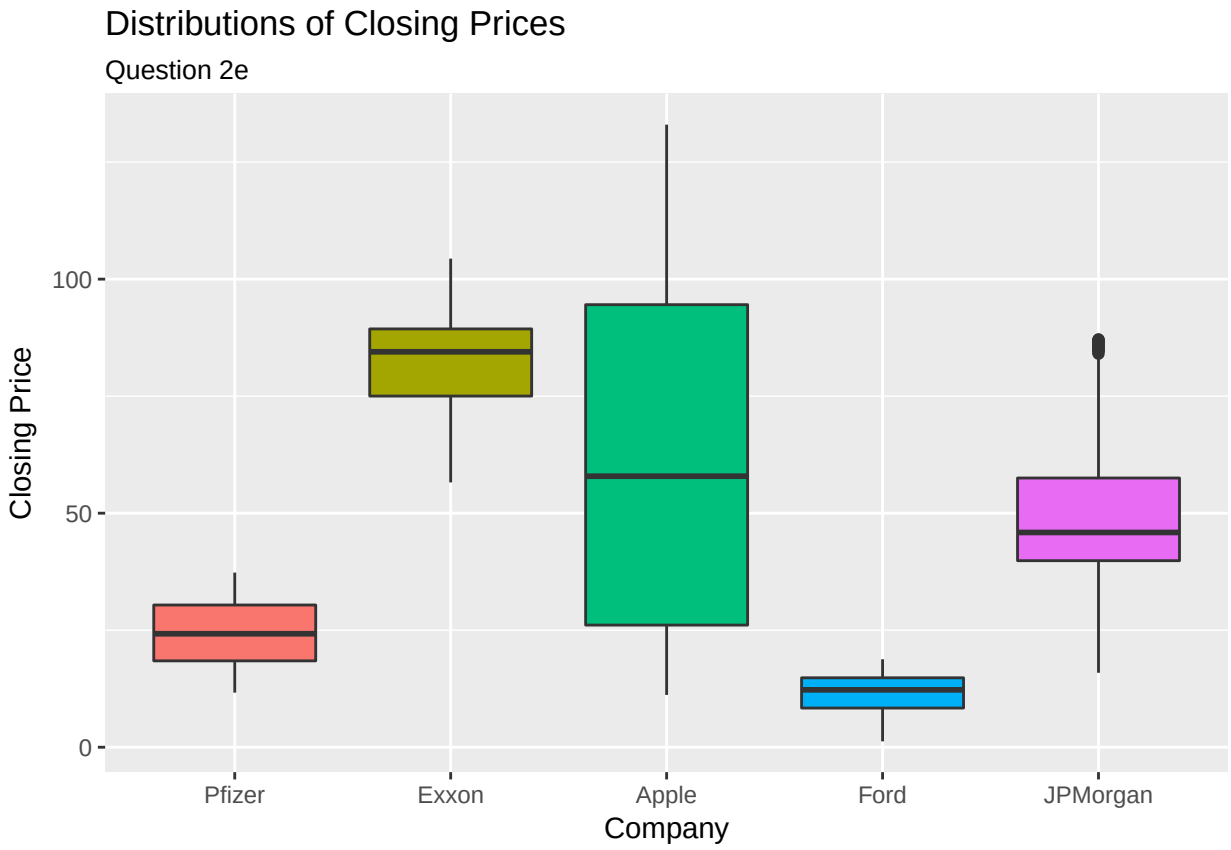
Question 2d.

We can also specify alpha in a scale_fill_manual call in the values argument. Here we will instead call alpha in the scale function and we will also specify the colors we want. alpha takes arguments of the form `alpha(colour, value)`. If we do not specify the `colour` argument the defaults are used. This time specify a vector of colors so that our chart has the following: coral, darkgoldenrod1, cyan, forestgreen, hotpink. Provide an appropriate alpha value. Below write your takeaways from the plot, does this distribution tell you anything about the quality of investment each stock would be? In evaluating a stock, what do you think an "ideal" distribution would be if one exists?



Question 2e.

So kernel density distributions are one way of looking at differentiation among different categories. Another way to show how the stocks differ in their spread would be to plot boxplots for the closing price of each stock over the period. Make a boxplot below with a differently filled box for each company. Do not include a guide for the different fills since it will be obvious from the x-axis which box is for which company. (Look up how to turn off the guide.) What are some of the strengths and weaknesses of this visualization?



Question 2f.

We need a way to compare growth among the companies instead of looking at raw closing price values. One way to do this is to find the relative closing price of each company relative to itself at the start of the period. I.E. the closing price on day one would be 1.0, `stocks$Apple[1]/stocks$Apple[1]` and each other day would be a percentage of the first day. `stocks$Apple[n]/stocks$Apple[1]`. Create a new data frame of the relative closing prices for each stock. Be sure that the observation you divide each price by is chronologically the first price to appear. Make a line-graph of these relative closing prices similar to question 2a. What are your takeaways from this chart? What further questions do you have?

Question 2g.

Now with our relative price data we can look at our density plots again but this time use the relative data. Show the new graph of the density distributions for the relative closing prices. Is this a valuable chart? What does it show? What are the weaknesses of the chart.

Question 2h.

Now make the barcharts for the relative price data. Again do not show the fill scale. Make sure that there is a mark at the value of 1.0 on the y-axis. Is this a valuable chart, what does this chart show us that the barplots of the raw closing price do not? What do you take away from the data displayed in this chart, what investment decisions, if any do you think this chart points to?

Question 2i.

What if there was way we could combine a boxplot with a density distribution. Great news! There is, it's called a violin plot. Using the relative closing price data from the boxplots above, make a violin plot of the data. Below discuss what a violin plot is able to show that a traditional boxplot does not, do you prefer this plot over the boxplot? Why or why not? Does this plot lead you to draw differnt conclusions than the boxplot?

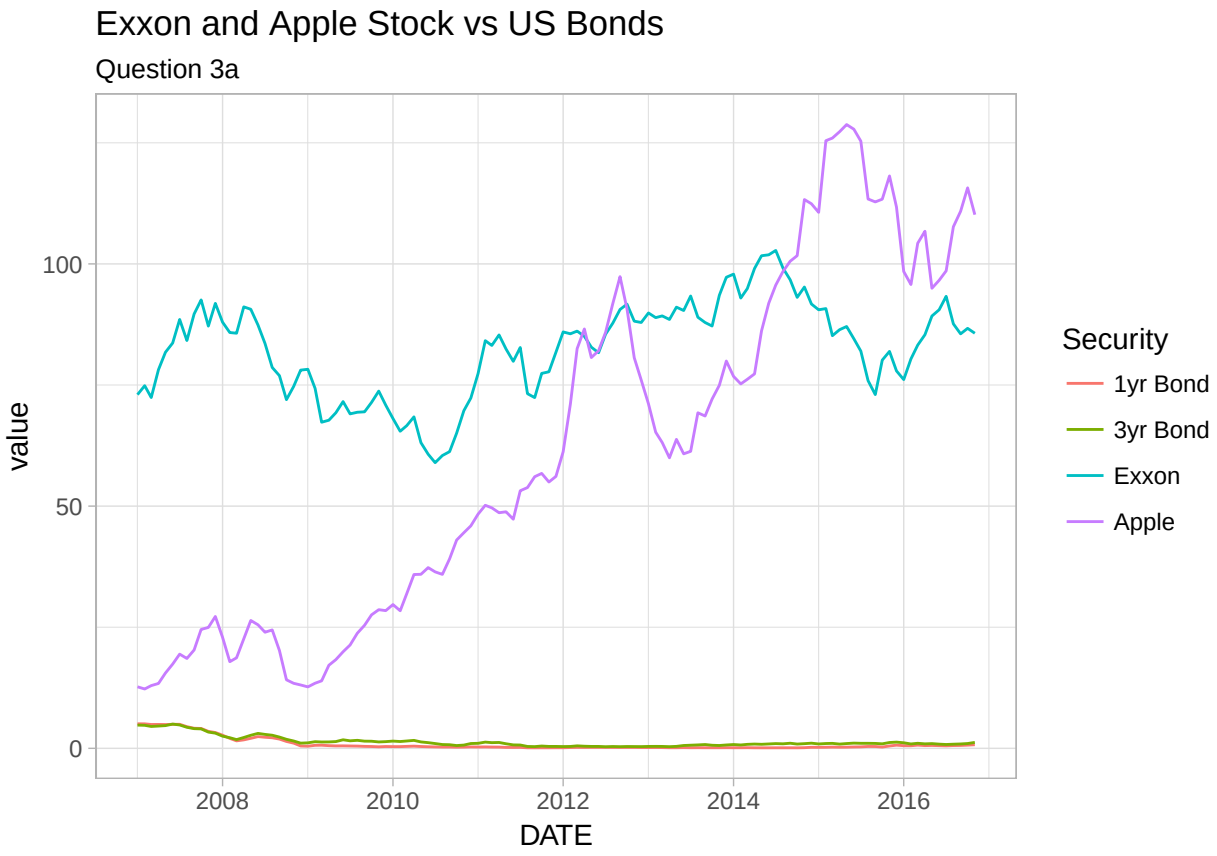
Theming

In lecture we discussed the theming mechanism of ggplot and I showed you that we can customize themes to make for individualized presentations. Over the course of question 3 you will build your own customized theme and use it to display plots for the rest of the homework. As you build your custom theme be sure that your displays are easily readable and understandable.

For this section we will look at how the stock data trends with the treasury data. You will need to create a dataset that merges the stock data with the treasury data by finding an average monthly closing price for each stock and merging the monthly data with the treasury data. (Hint, you will want to create a month column in the stock data of the format "%Y-%m-01" so that it is easy to merge these dates with the treasuries data).

Question 3a.

We will start with a simple theme change. Change the theme to one of dark, light, or classic for the following plot. Pick two treasuries and two stocks and create an appropriately labeled simple time series lineplot of their values. What do you think of this plot? Does this plot make a valid comparison? How would you accurately label the y-axis?



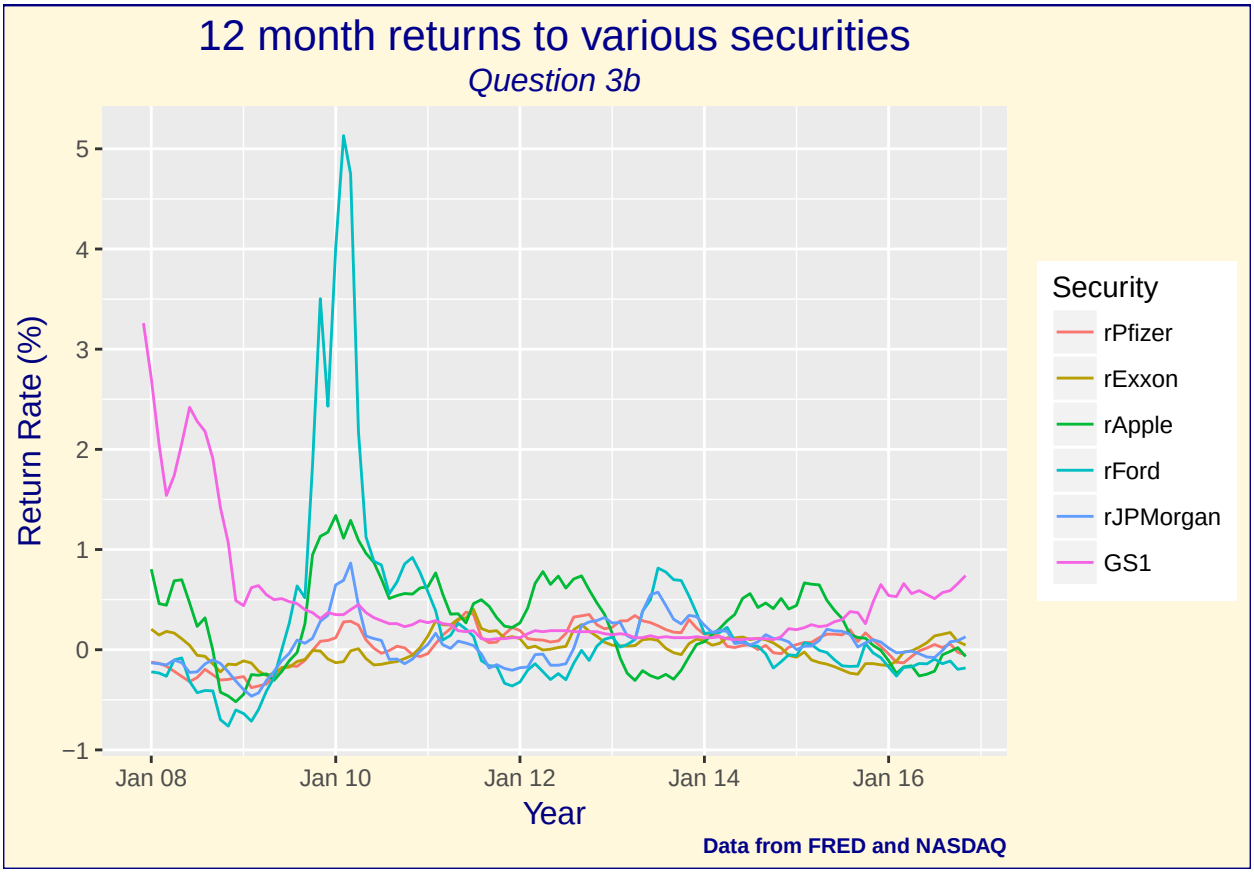
Question 3b.

Ok, so the above chart is no good. (You should have described why in your explanation of chart 3a). Instead we will plot the average monthly returns to each stock vs. the monthly interest rates on the bond. Calculate the monthly one-year returns to each stock as the percent change in closing price of monthA with the closing price for the same stock 12 months earlier. You will want to make use of the `shift` function. After calculating the yearly percent change in closing price merge this stock data with the 1 year treasury bond data. (Note, this is not necessarily the only or best way to compare stocks with bonds as an investment but it will allow us to compare rates.)

For the following plot we will begin updating the theme. Create your own custom.theme built off of the basic theme_gray() that comes with ggplot. For your theme change the following aspects of your plot. You must change the specified aspects and may change others if you wish:

- plot title - center, change size and color
- axis title - change size, color and font
- subtitle - center, change size, color, font face
- caption - change size, color, and face
- background - change fill and add a border line

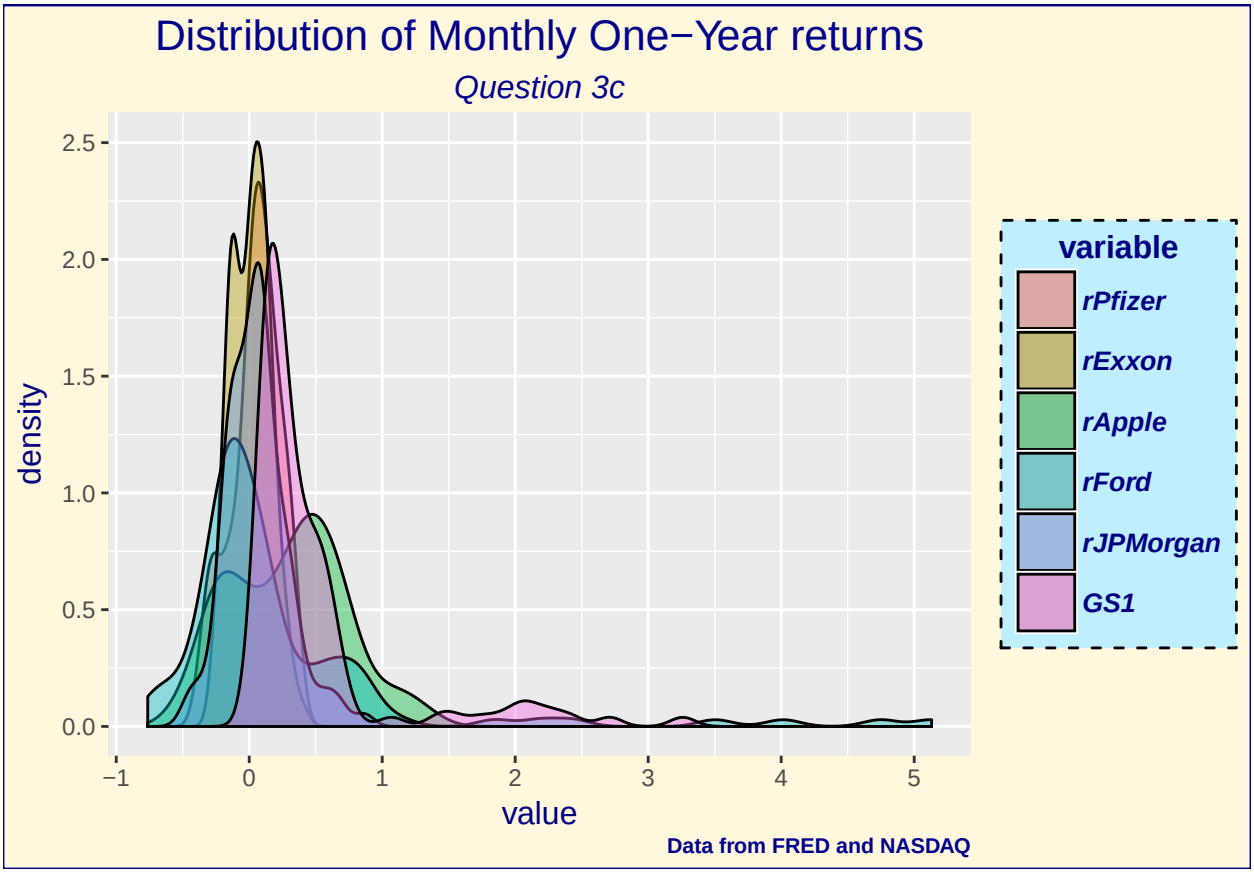
Beneath your plot describe your takeaways from the chart.



Question 3c.

Similar to our stock analysis above, make kernel density plots of the return rate data. We will also continue updating our custom theme. To your custom theme from question 3b update the **legend** as follows:

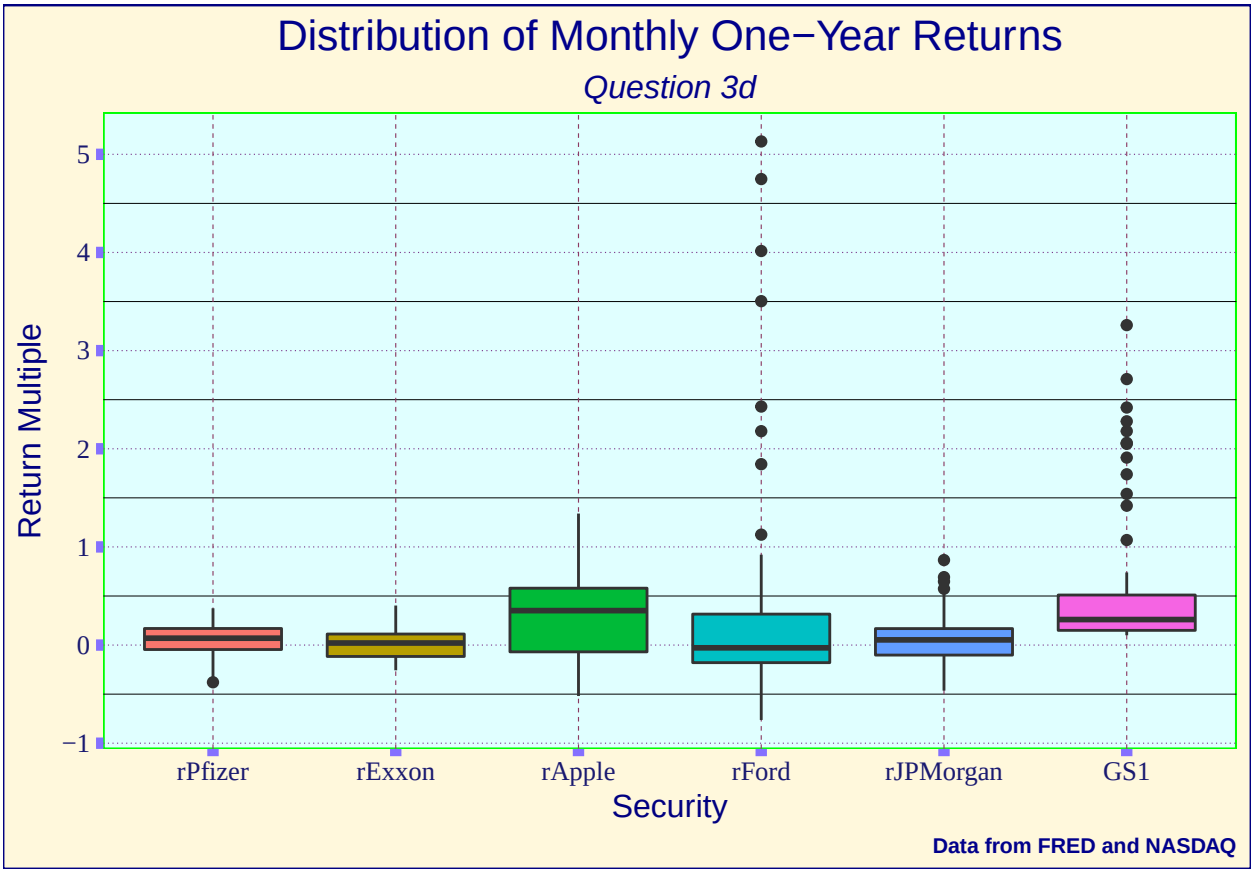
- Title - center, size, color, font
- key text - font, color, face
- key box - size, background color
- legend box - add a border to the entire legend and change the background color



Question 3d.

Now make a boxplot for our return data with the fill color differing for each security. Below the plot explain your takeaways, is the plot meaningul, is it misleading, does it give us a better understanding than the density plot? For this last theming question we will change aspects of the **panel** on which our data is plotted itself. Update the following aspects:

- Background color
- Major and minor x axis breaks
- Major and minor y axis breaks
- tick marks for x and y axis
- text for x and y axis labels



Adding Trend Lines to Data

For this analysis we will need the gdp, stock_closings, and treasuries datasets. Our first step will be to read in the raw data and examine it.

```
gdp <- read.csv(paste0(path,"gdp.csv"), stringsAsFactors = FALSE)
head(gdp,3)
stock_closings <- read.csv(paste0(path,"stock_closings.csv"), stringsAsFactors = FALSE)
head(stock_closings,3)
treasuries <- read.csv(paste0(path,"treasuries.csv"), stringsAsFactors = FALSE)
kable(head(treasuries,3))
```

We see that each of these series are time series but at different frequencies. Before we can merge them we will need to aggregate them all to the lowest frequency data (quarterly).

Question 4a.

(i) Standardize the date variables for each of the data sets and make sure they are Date type objects and create a new variable in each series that is the quarter (Q1 - Jan,Feb,Mar Q2 - Apr,May,Jun Q3 - Jul,Aug,Sep Q4 - Oct,Nov,Dec). Create a new variable in each series that is the year

(ii) Then use dplyr's group_by() and summarise() functions to average over all the observations in the quarter. If you prefer to use data.table feel free to use it instead to perform the same analysis. HINT: Do not forget to use the na.rm option when averaging. Also exclude the GS30 variable since it has a lot

of missing values.

(iii) Finally merge the data sets by year and quarter

Your resulting data should look something like this:

Its expected that we would have so many NAs since gdp has been around a lot longer than Pfizer. We can cleanup the data to only be data points that are complete using the na.omit() function

```
economic_data <- na.omit(economic_data)
head(economic_data)
```

We see that we have data from 2007 on-wards. Now lets create an aggregate measure of the stock market. For now just sum the stock values for the five different companies.

Question 4b

Add a some new columns to the economic_data data frame:
(i) An index of the five stocks (just sum them)
(ii) The change in the stock market index between subsequent quarters
(iii) The change in the 1 year treasury bond (gs1) between subsequent quarters
HINT: You may want to use the diff() function. You will also have to deal with the fact the diff() will not return an object that is the same length as the input.

The resulting data frame should look something like:

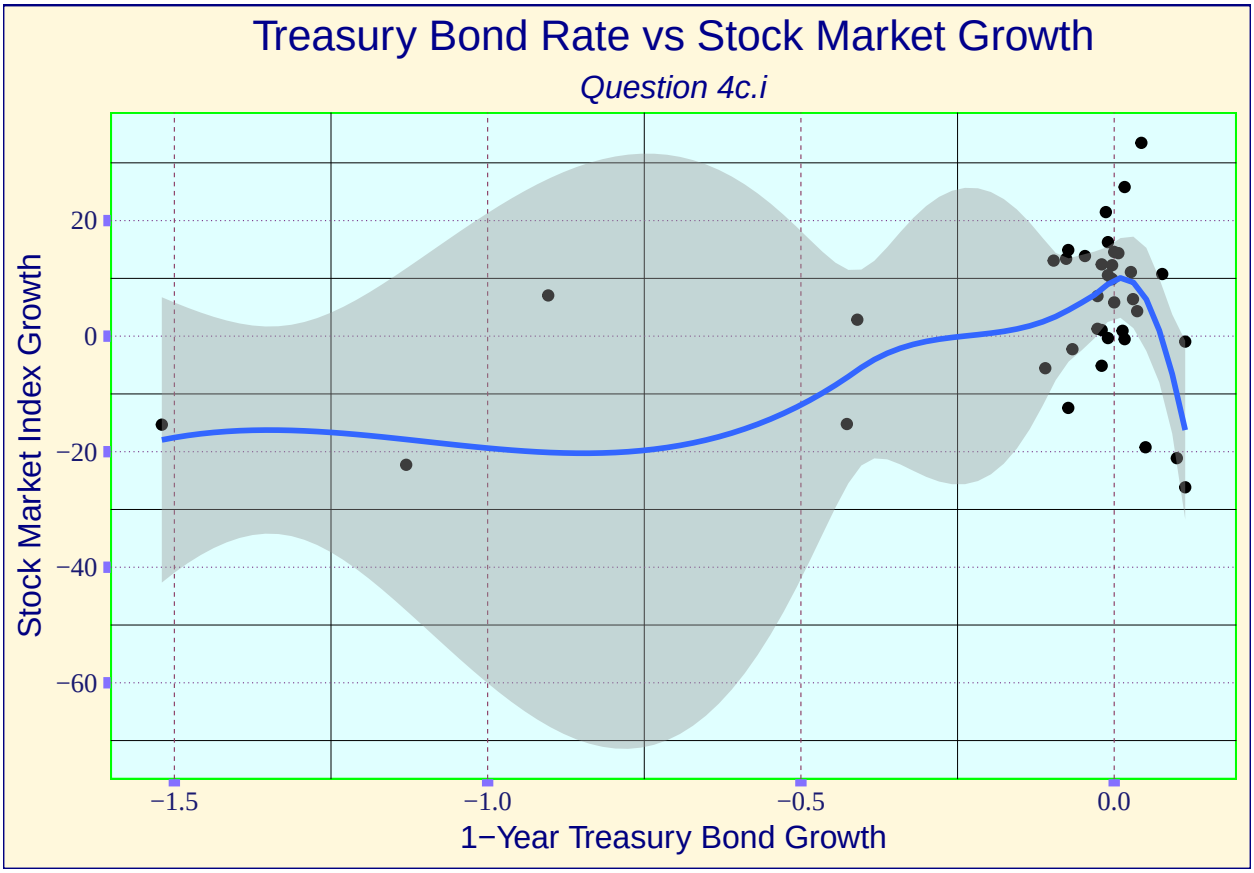
Now that we have the data all set up we can move on to charting.

Question 4c

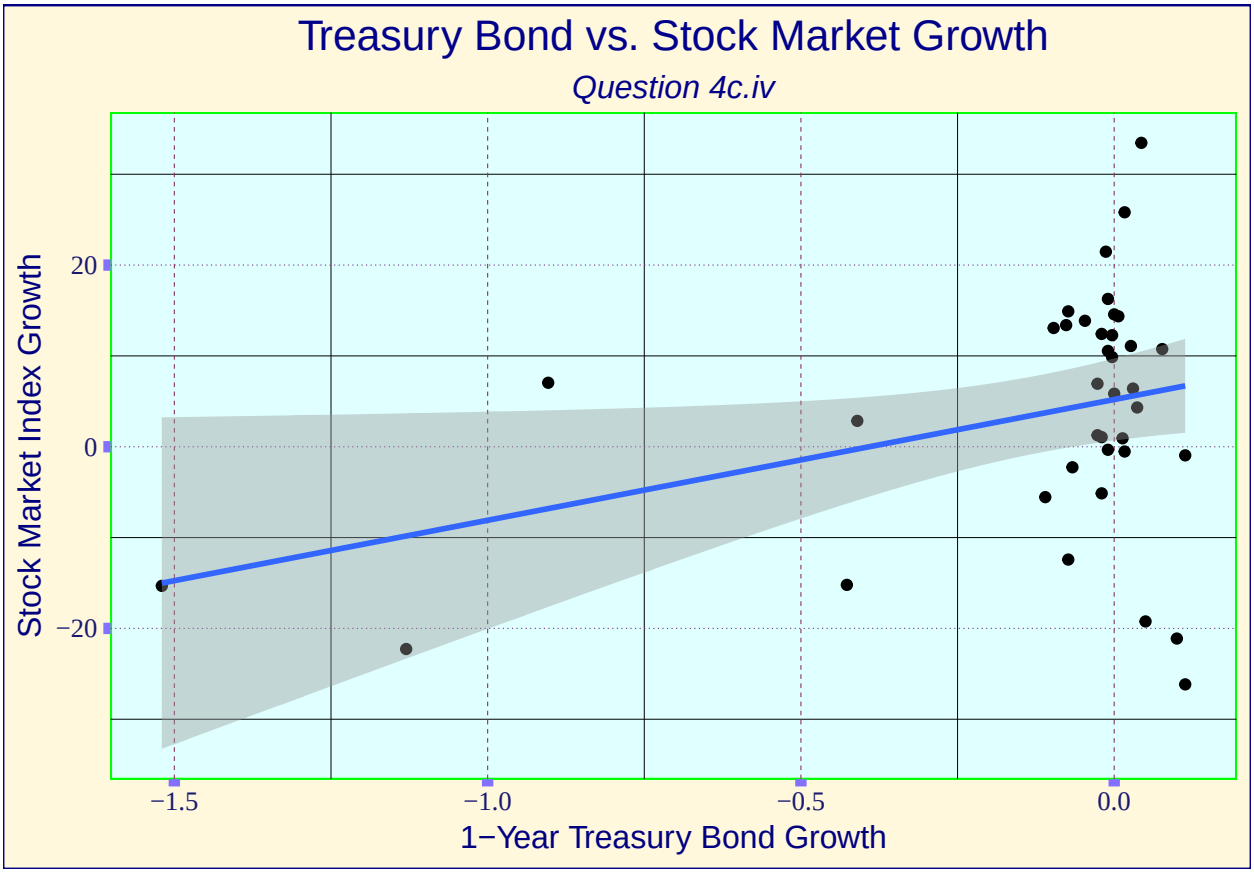
(i) Make a scatter plot of the growth in the stock market index vs. the growth in the 1 year bond with a default call to geom_smooth. Make sure your chart has appropriate titles and labels.

(ii) What information does this chart present? What does the shaded region represent?

(iii) What is the default method used by geom_smooth? Is this a method commonly used in econometrics?

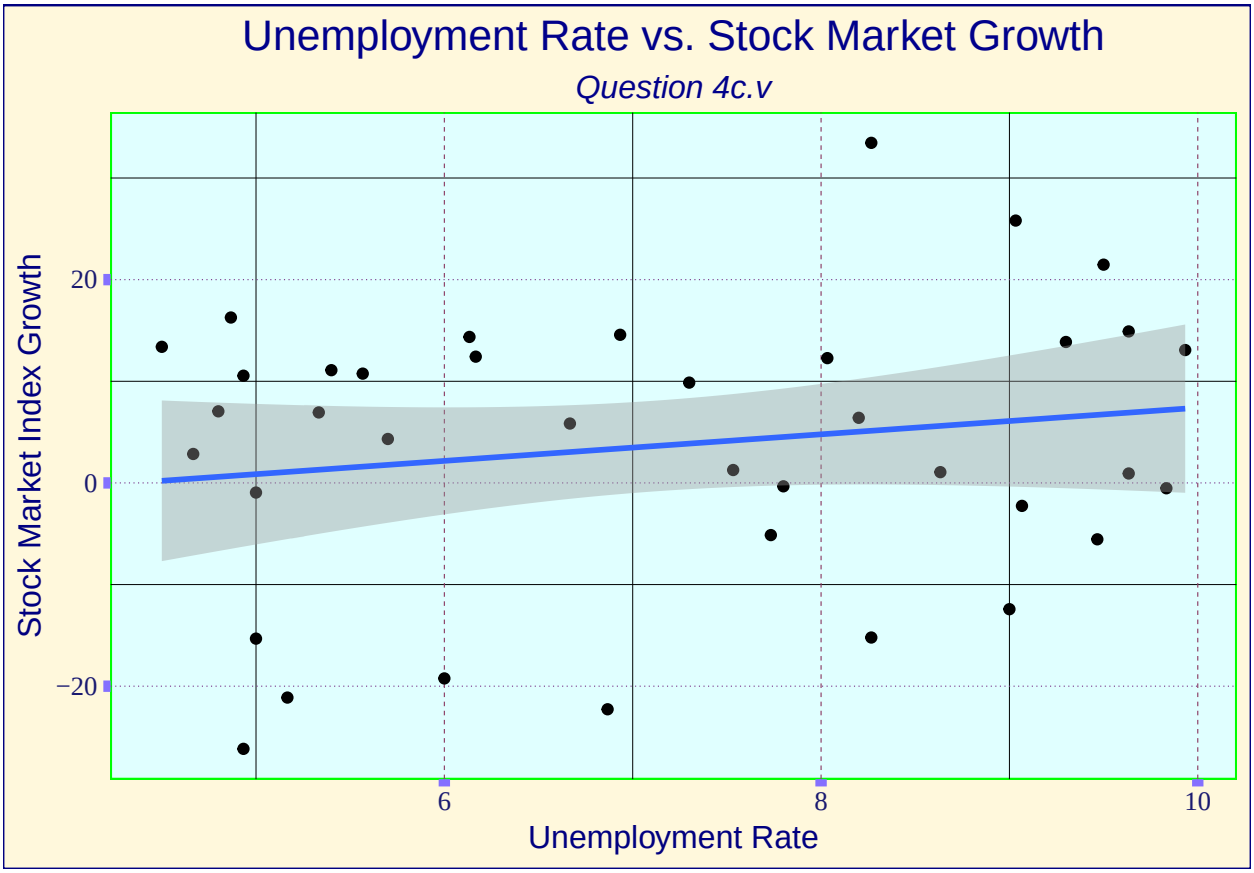


(iv) Now make the same chart but instead use the method "lm"" for geom_smooth(). What does the "lm" method do? What do we learn about the relationship between 1-Year Treasury bond growth and our stock market index growth?

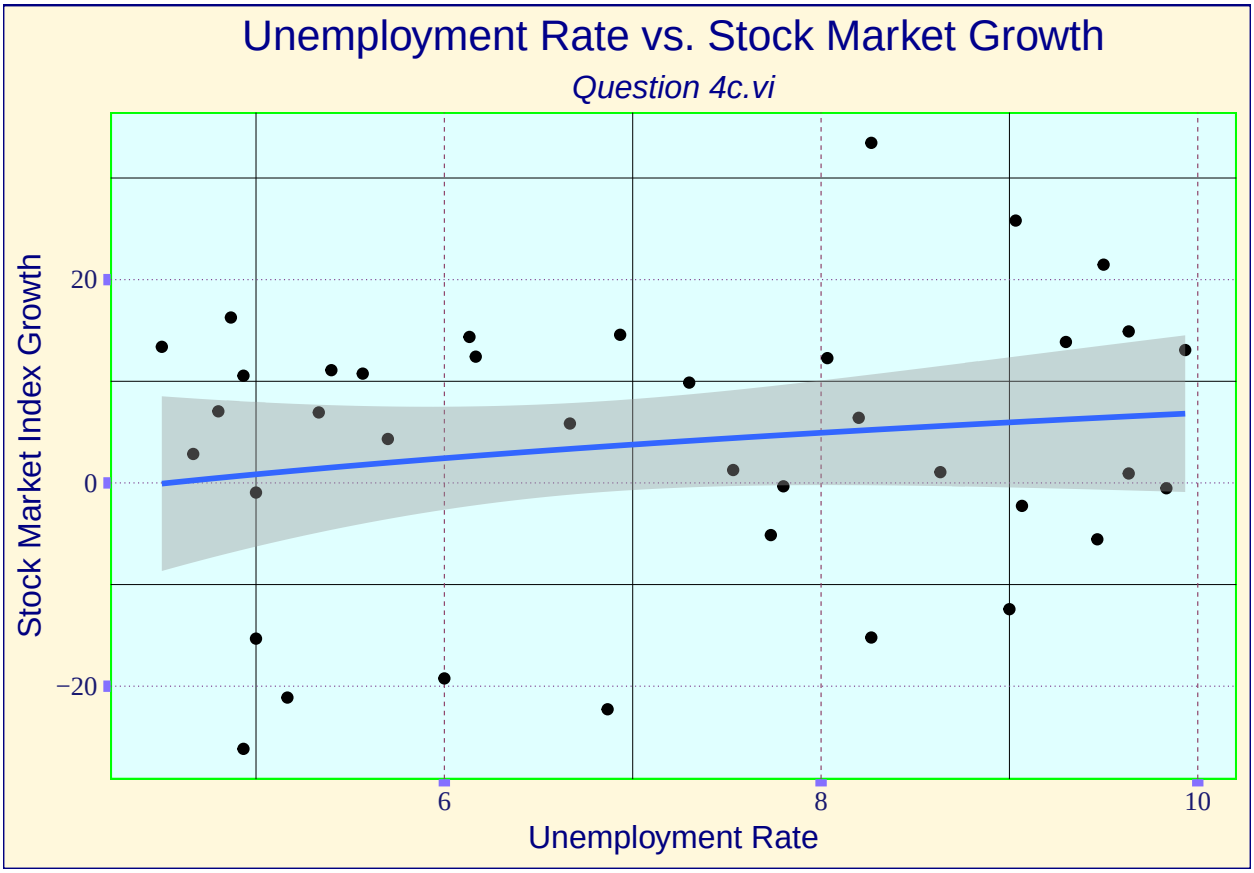


It would seem that the relationship between the growth in the 1-year treasury bond and the stock market index is more complicated than a linear relationship.

(v) Now plot stock market growth vs. unemployment rate with a geom_smooth() call using "lm".



(vi) Same chart as 4c.v but now specify a formula for "lm". When specifying a formula you can only use y and x. Variables which do not appear on the chart cannot be in the formula. For now test out stocks = log(unemployment rate).
HINT: You will want to look at the documentations for geom_smooth.



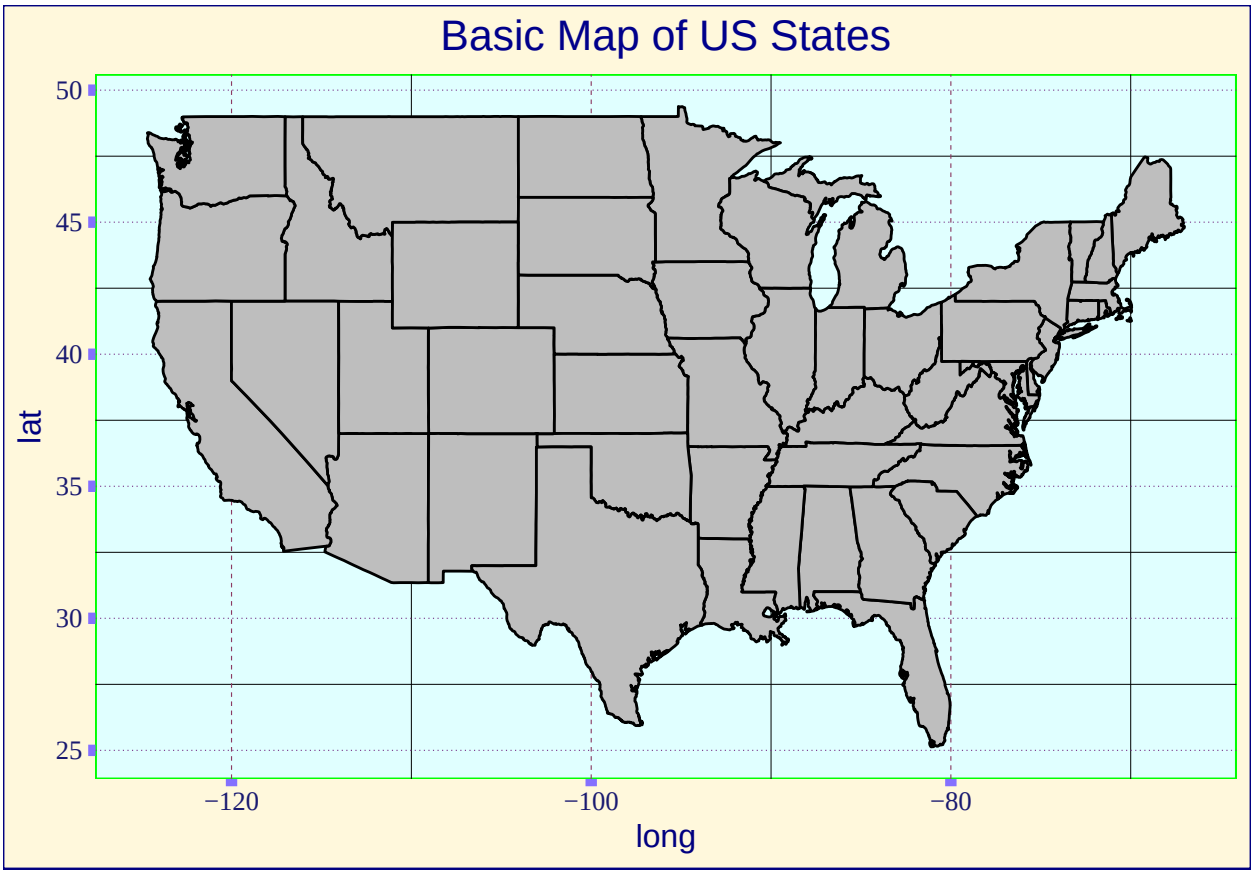
(vii) Free form chart: Investigate the relationship between two variable we have not already made a chart for. Make a scatter plot of the two and add a smoothing curve (you decide what is the most appropriate formula) Make sure the chart is appropriately labeled. Discuss what, if anything, the chart shows? A null result is equally important to understand.

Mapping

Maps are a great way to visualize data with a spatial component. One of the most common types of plots for visualizing spatial data is a choropleth map (some examples here and here). A choropleth map is map which shows some form of geographic boundaries and color the distinct regions based on some metric (for example unemployment). We can make these maps using the very versatile ggplot package. The package "maps" has actual maps of us states and counties that we can use with ggplot.

```
# install.packages("maps")
library(maps)

# let's start out looking at state level granualrity
states <- map_data("state")
head(states,3)
# create the ggplot object
state_map <- ggplot(data = states, mapping =
                        aes(x = long, y = lat, group = group)) +
             geom_polygon(color = "black", fill = "grey") +
    ggtitle(label = "Basic Map of US States")
# display the map
```

```
state_map
```



We see we have successfully created a very boring map of the U.S. states where the boundary for each state is outlined in black. Note that in lecture we used the `borders()` function for map-making but here use `geom_polygon()`. Now we need some actual data. Let's pull in 2016 election results and merge it with our map data frame.

```
election_results <- read_csv(paste0(path, "national_election_results.csv"))
```

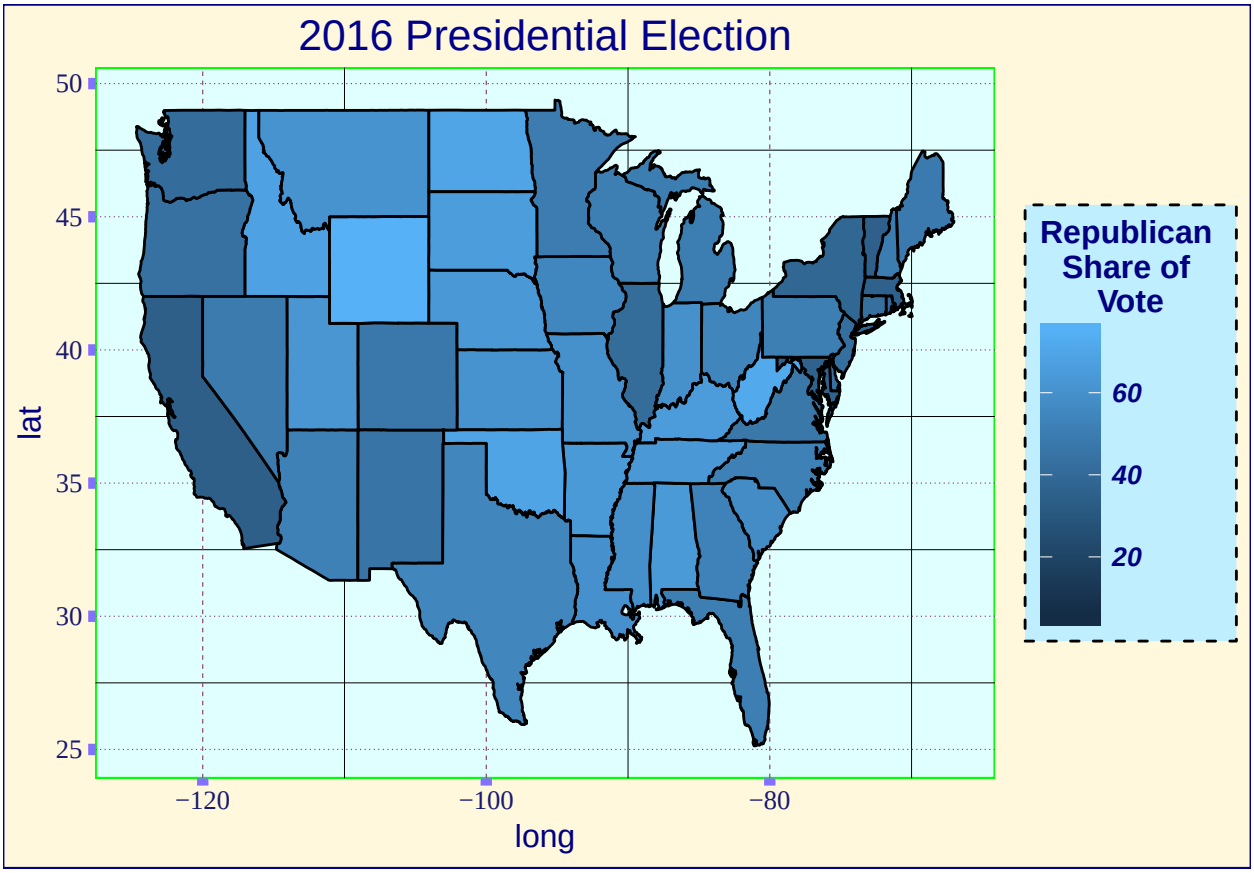
```
## Parsed with column specification:
## cols(
##   state_name = col_character(),
##   state_code = col_character(),
##   dem_votes_2016 = col_integer(),
##   rep_votes_2016 = col_integer()
## )
```

```
head(election_results)
# create a new variable that is the lower case state name
# needs to be lower case to match states dataframe
election_results <- election_results %>%
    mutate(region = tolower(state_name))

## Create column for election results fraction
election_results <- election_results %>%
    mutate(rep_frac_2016 =
                    (rep_votes_2016)/(dem_votes_2016 + rep_votes_2016))
```

```
election_results <- election_results %>%
    mutate(dem_frac_2016 =
                      (dem_votes_2016)/(dem_votes_2016 + rep_votes_2016))
# now lets merge the map data with the election results data
election_results <- merge(election_results, states, by="region")
```

We see that this data is structured so that we have a column for states name and state code as well as columns for the number of votes for the democratic and republican candidates as well as their vote shares. Let's start by making a choropleth map of the republican vote share by state.

```
election_results_map <- ggplot(election_results,
                               aes(x = long, y = lat, group = group)) +
                        geom_polygon(aes(fill = rep_frac_2016 * 100),
                                     color = "black") +
                        expand_limits(x = states$long, y = states$lat) +
    scale_fill_continuous(name = "Republican \nShare of \nVote") +
    ggtitle(label = "2016 Presidential Election")

election_results_map
```

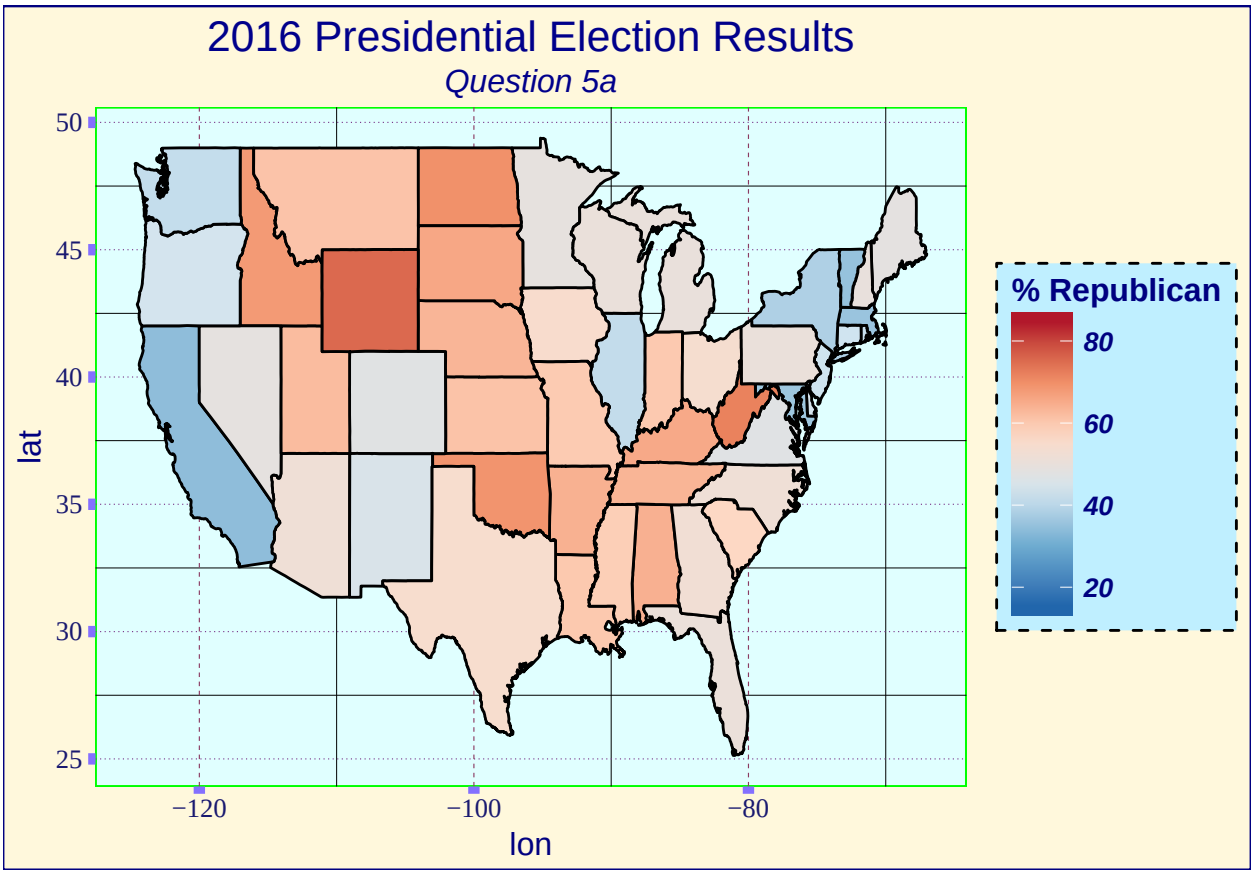


This is not a super useful color scheme for us to use to visualize the data.

Question 5a

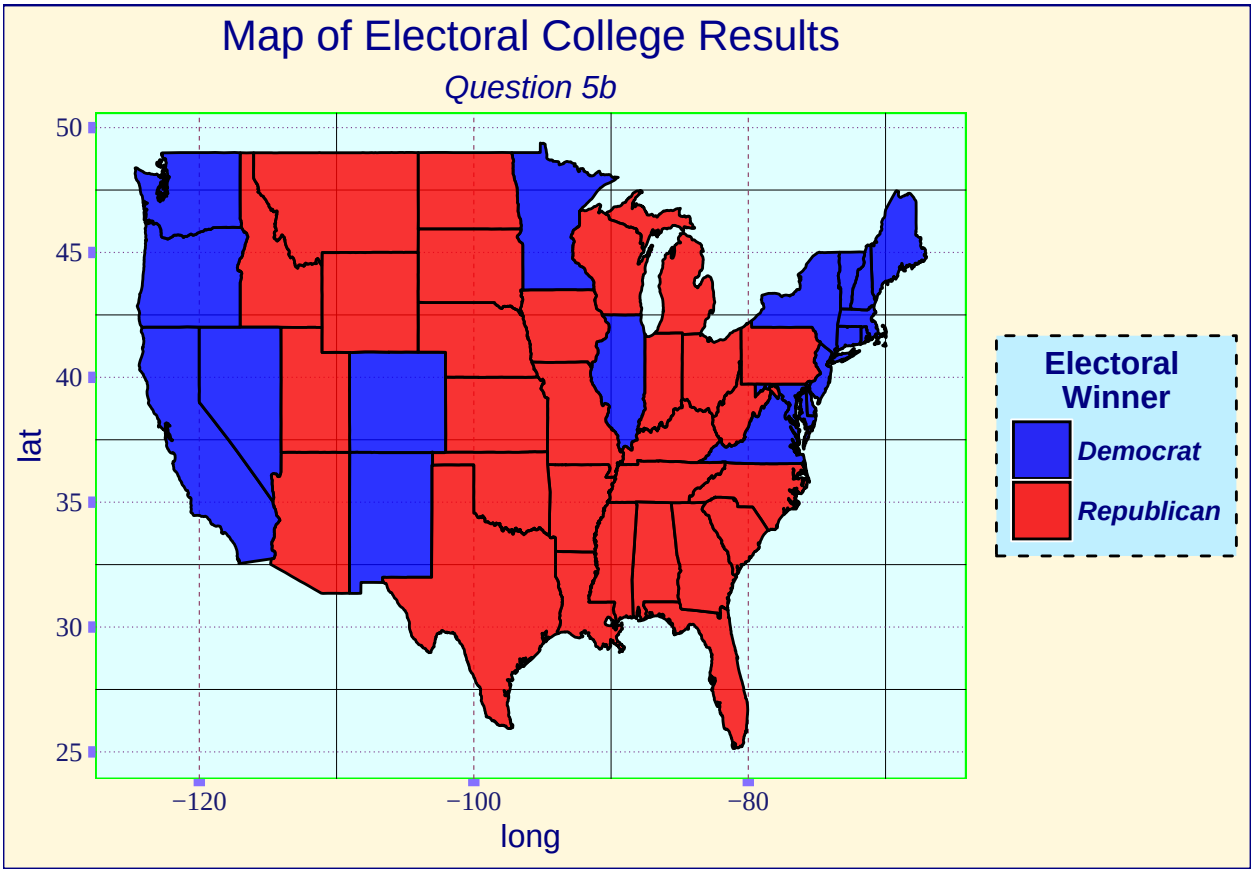
Change the color scale to be the familiar red for more republican and blue for more democratic states using the scale_fill_distiller() function and the palette argument. You can decide what you want 50% to be (purple

or white are common options). You should also ensure that 50% falls in the middle of your color scale. Add a useful title and color bar label and relabel the x and y axis for lat and lon. Your result should look something like:



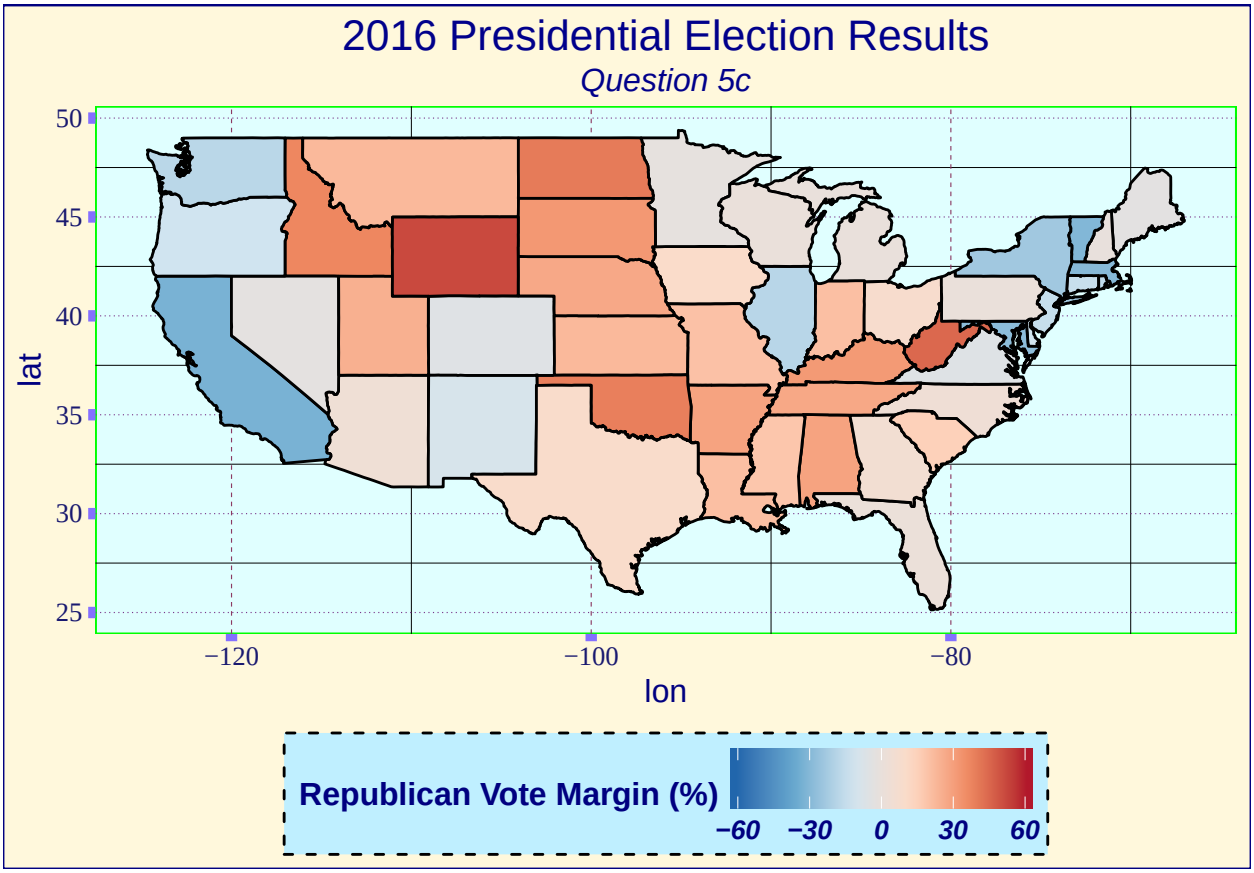
Question 5b

Now let's take a look at the simple question of who won each state. Create a binary variable for each state for whether Trump or Clinton won and map the United States.



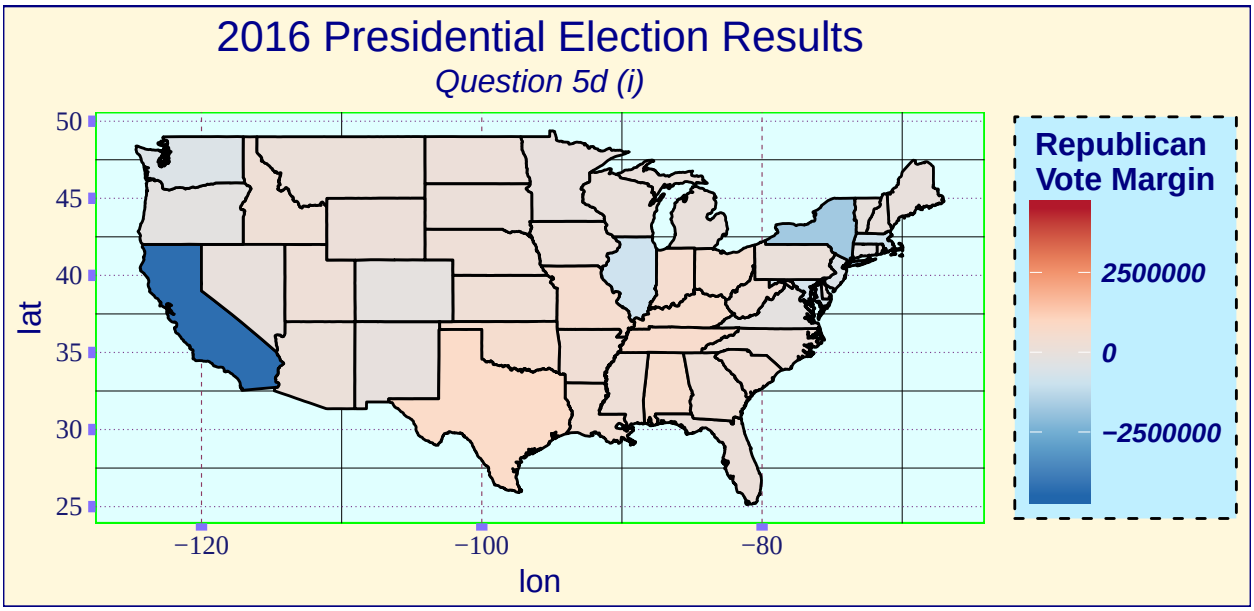
Question 5c

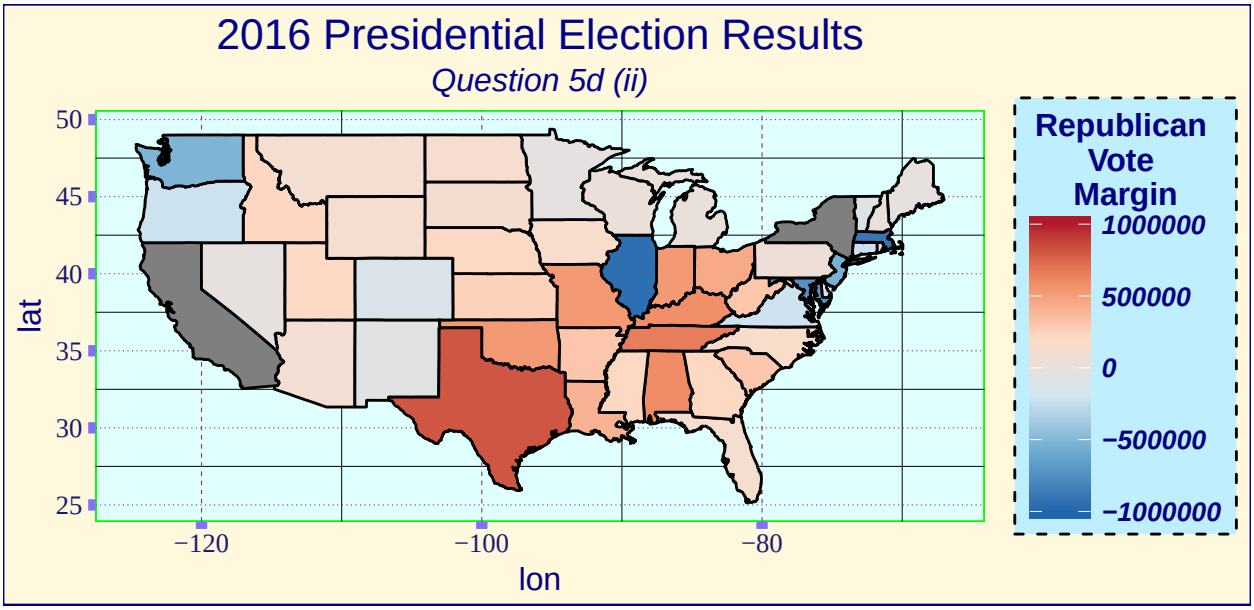
Instead of plotting the fraction of the vote the Republican candidate received, plot the difference in the vote share between the republican and democratic candidate. Be sure to use an appropriate color scale, and title.



Question 5d.i

Almost the same as in the previous part but this time plot the difference in the number of votes between the republican and democratic candidate. Be sure to center your scale around 0. Secondly make a plot excluding California and New York (HINT:, you will want to manipulate the limit argument for your fill scale). Compare 5d.ii with 5d.i, how much more vibrant are the colors from one to the other, what does this tell you about CA and NY?





Question 5e

What do we learn from each of these plots? What information is conveyed by each plot. Are certain plots more informative than others? In what situations would if be more informative to look at vote shares vs. vote counts?

Now let's drill down on the results for Ohio. First we need to subset the states data to be only for Ohio.

Mapping at County Level

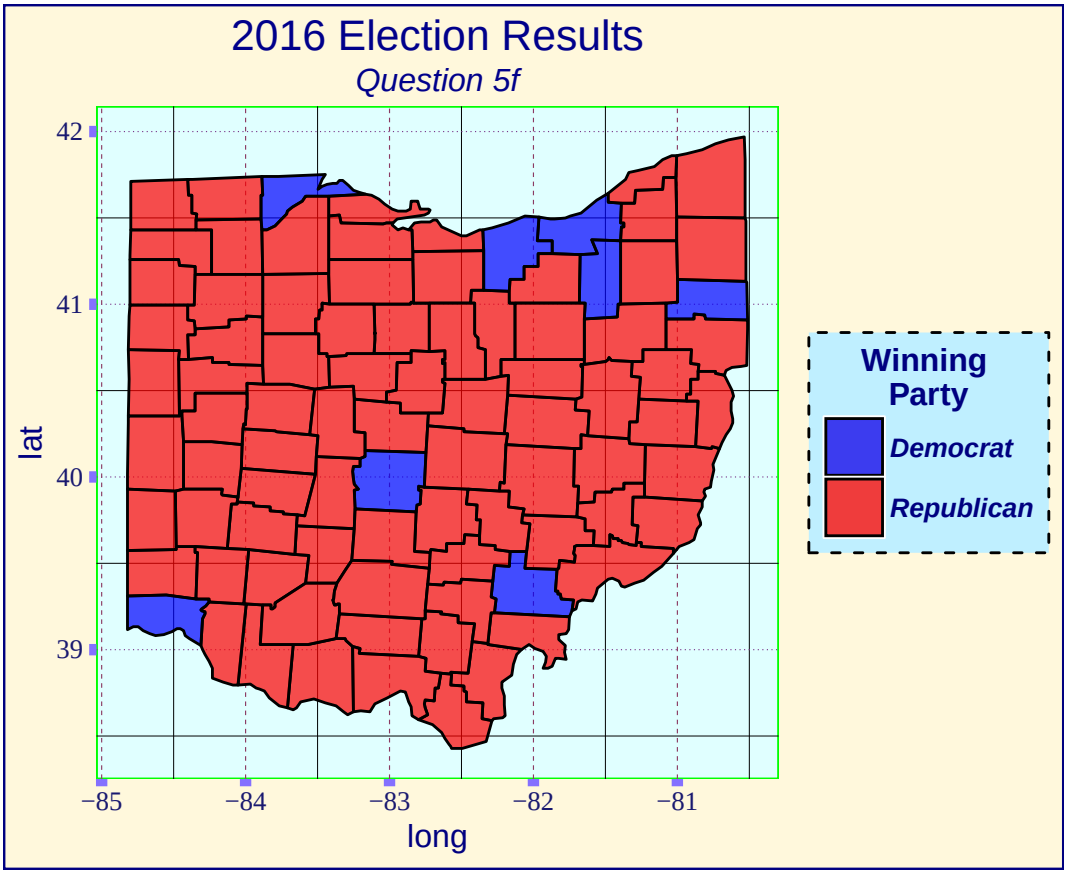
Subset the state and county map_data to be only for Ohio. Try and plot the Ohio counties similar to our plot of the states. First make a layer that is the state data and then add a layer for the county data.

You will want to start with:

```
ohio          <- subset(map_data("state"), region == "ohio")
ohio_counties <- subset(map_data("county"), region == "ohio")
head(ohio_counties)
```

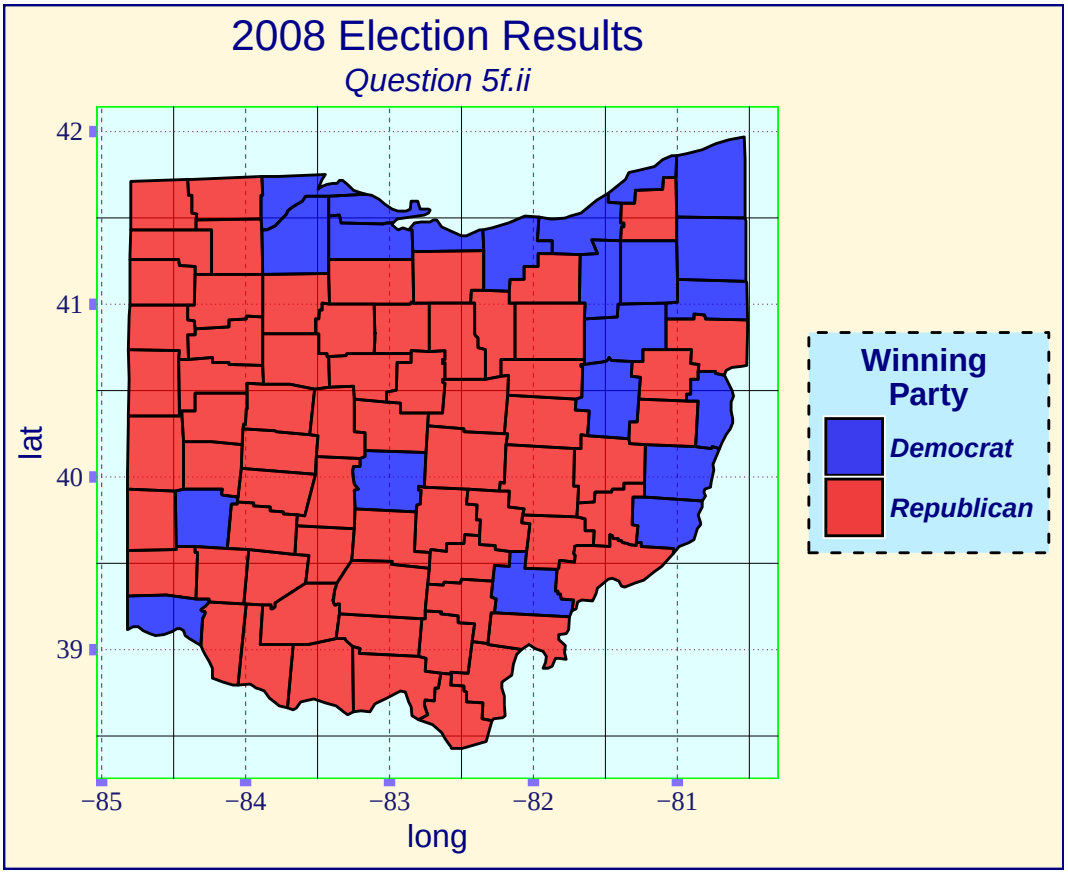
Question 5f.i

Let's now create a county level map looking at which party won each county using the oh_election_results csv.



Question 5f.ii

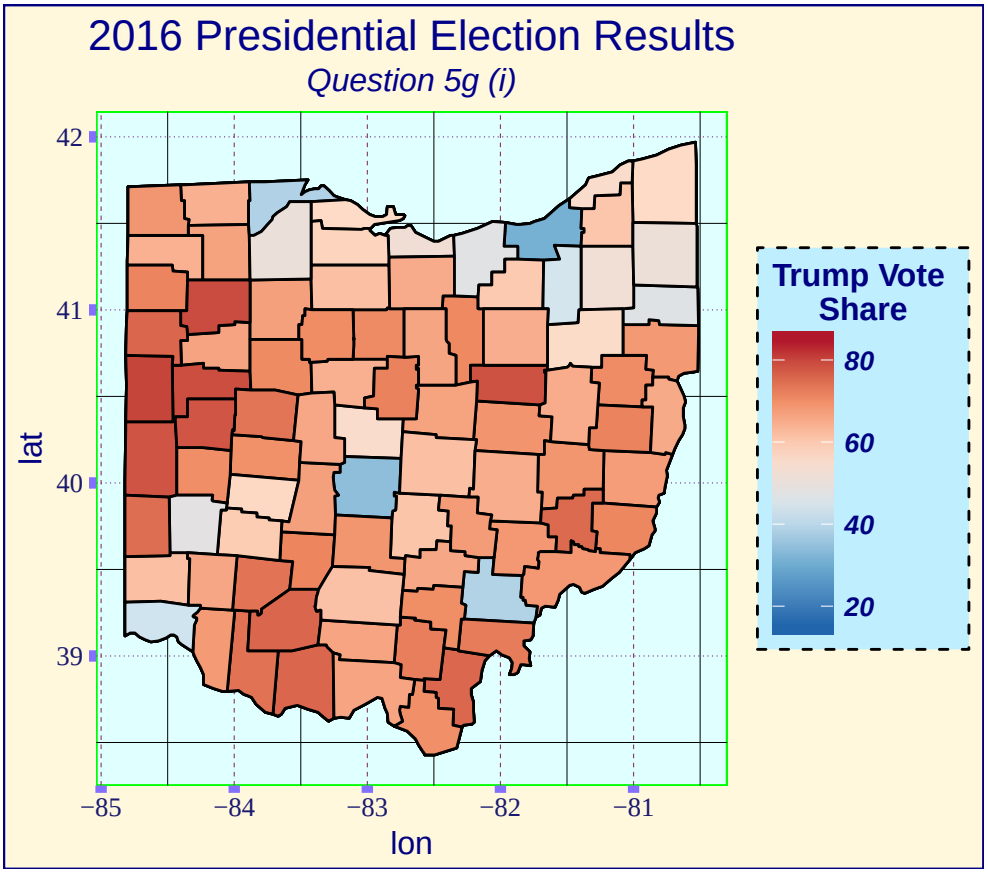
Now make the same county level plot looking at the 2008 election to see how the county votes differed with 2016.

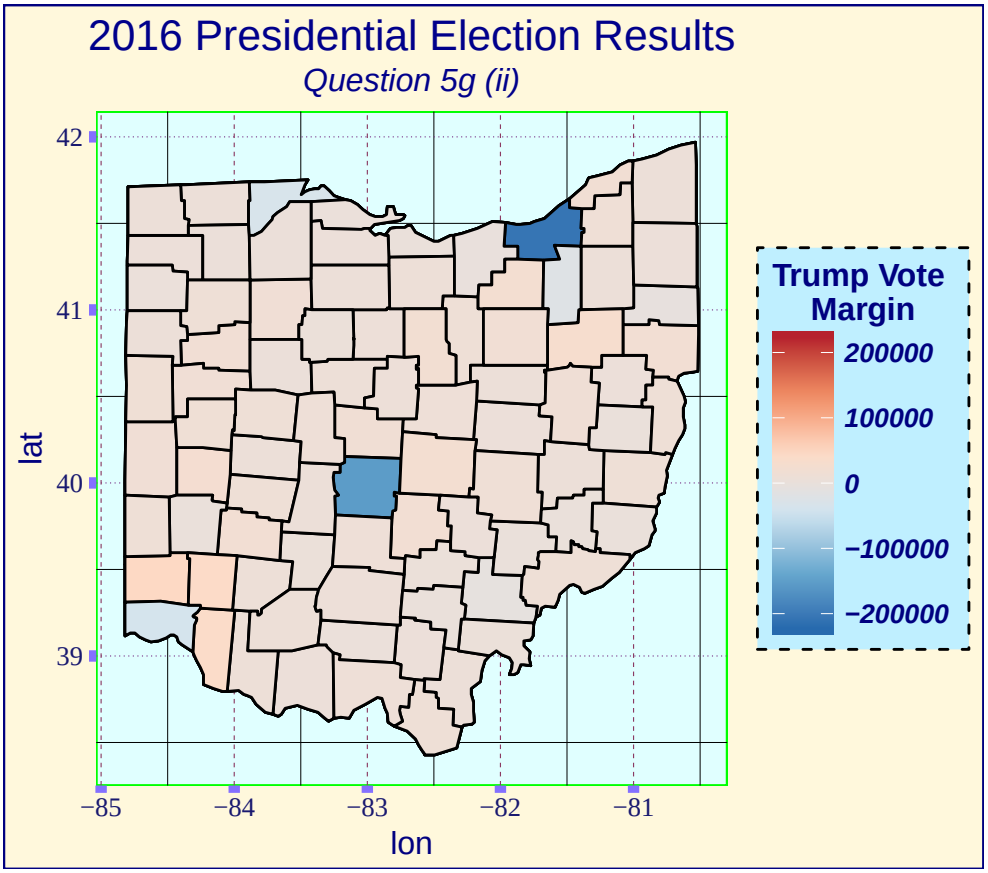


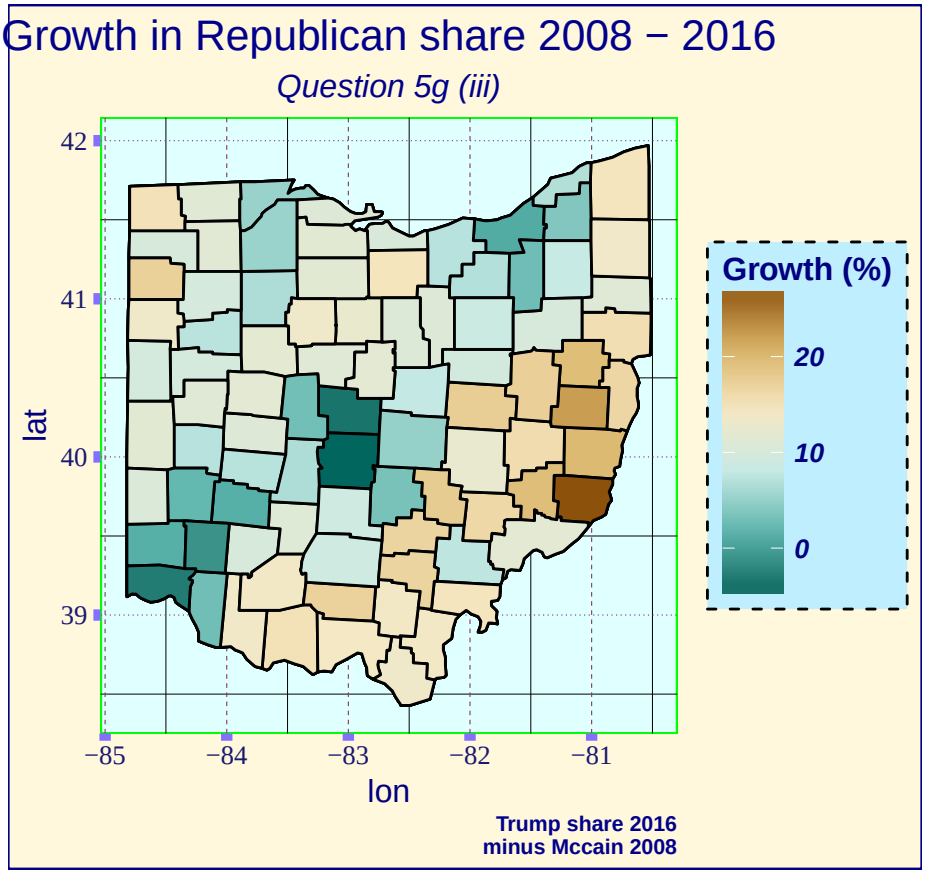
Question 5g

Using the data the file make the following plots with appropriate color scales, labels, and titles:
(i) Fraction of the vote for Trump in 2016
(ii) Difference in the vote count for Trump and Clinton in 2016
(iii) Growth in the fraction of the vote for Republicans in 2008 vs 2016.

HINT: After you merge you will want to ensure that you have preserved the ordering of the counties.

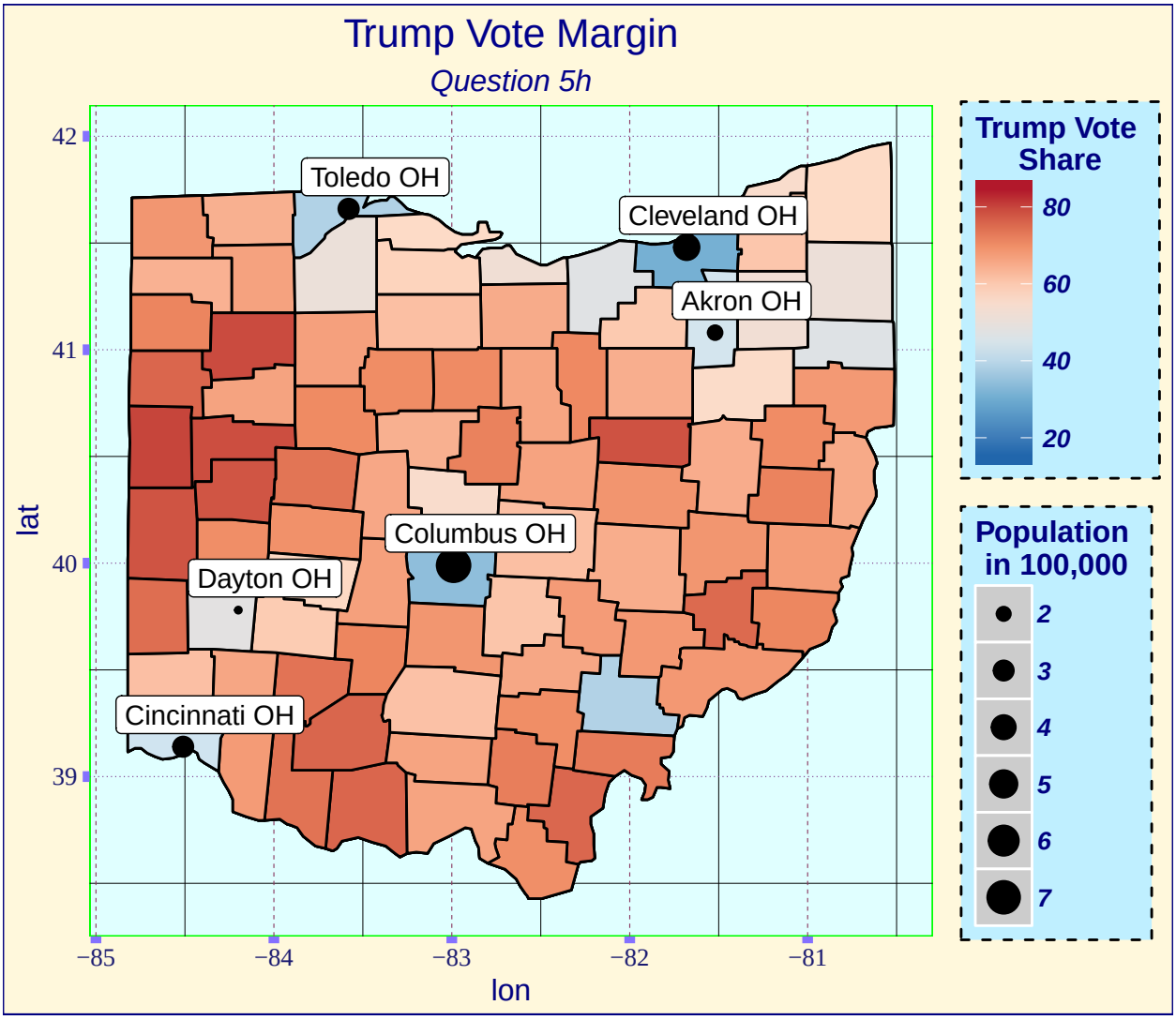

2016 Presidential Election Results
Question 5g (i)
42
41
40
39
lat
−85
−84
−83
−82
−81
lon
Trump Vote Share
80
60
40
20


2016 Presidential Election Results
Question 5g (ii)
42
41
40
39
lat
−85
−84
−83
−82
−81
lon
Trump Vote Margin
200000
100000
0
−100000
−200000



Question 5h

let's add a layer to the map from question 5g.i showing the location of the major cities of Ohio. Please plot the cities with a marker proportional to the population and label them with their name. Use the us.cities table to find all cities in Ohio with over 100,000 people. Use geom_label to add the city names above each point where the city is plotted. Be sure to offset the label so that each point is visible.



Question 5i

What does these maps show us about the voting divide between urban and rural counties? When comparing maps 5g.i and 5g.ii what different story does each map tell? Do you think that using one of these maps instead of the other is misleading, why or why not? What other variables might be highly correlated with urban/rural?