Homework 6

The purpose of this homework is to gain more practice with dplyr and ggplot2, to learn how to create choropleth maps, and to practice using simple linear regression to explore relationships. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through Gradescope by 11:59pm on Sunday October 18th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Additional practice visualizing COVID-19 data

To gain more practice using ggplot and dplyr, we will visualize COVID-19 cases in the United States as a function of the date. The results from these exercises could be useful for knowing where one needs to be careful when traveling (e.g., if you're planning on taking a flight for the Thanksgiving break).

As discussed on homework 5, data on all COVID-19 cases is being compiled by the New York Times and has been made publicly available on the New York Times GitHub repository. The data compiled by the Times is listed at the county level, where a county is a sub-region of a state. For example New Haven County is the county that contains the cities of New Haven and Waterbury.

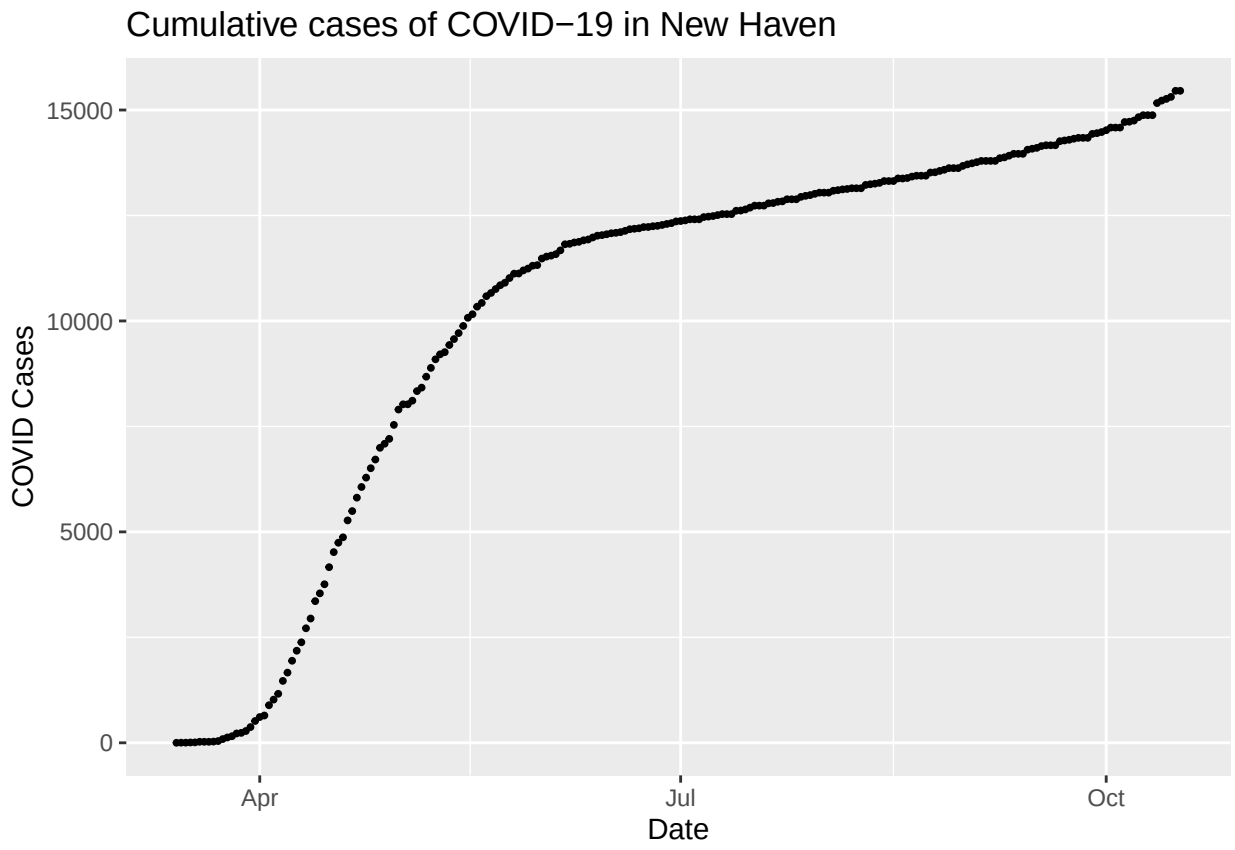
The code below loads the latest data from the New York Times GitHub repository into an object called `county_covid_cases` and it also uses dplyr to convert the variable `date` into an object that is of the class type Date. We will use this data in the exercises below.

```
# load covid county data and convert dates to the date type
county_covid_cases <- read.csv('https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counti
  mutate(date = as.Date(date))


dim(county_covid_cases)
```

```
## [1] 641431      6
```

Part 1.1 (10 points): Let's start by visualizing the cumulative number of cases in New Haven County as a function of the date using ggplot. The cumulative number of cases is in the variable called `cases`, so filter and plot the data appropriately to create this visualization.

```
new_haven_cases<-county_covid_cases%>%filter(county == "New Haven")%>%
  ggplot(aes(date, cases))+
  xlab("Date")+
  ylab("COVID Cases")+
  ggtitle("Cumulative cases of COVID-19 in New Haven")+
  geom_point(size = 0.7)

new_haven_cases
```



Part 1.2 (10 points): To understand the progression of COVID-19 cases, it is much more informative to visualize the number of *new cases* each day rather than the total cumulative number of cases. Let's do this visualization first using base R graphics and then we will create the visualization using ggplot.

To begin, we first need to calculate the number of new cases. To do this we can use the `diff()` function which takes the difference in values between successive numbers in a vector. For example, if we had a vector `vec <-c(2, 10, 3)`, using `diff(vec)` will return a vector with the values `8 -7`. Please complete the following steps to calculate the number of new cases:

1. Use dplyr to create a data frame object called `new_haven_sorted` which has the data from only New Haven sorted by the date.
2. Create a vector object called `new_haven_new_cases` which contains the *new cases* on each date based on the data in the `new_haven_sorted` data frame using the `diff()` function.

3. Create a vector object called `new_haven_date` which contains the dates from the `new_haven_sorted`.
4. Report the lengths of `new_haven_date` vector and the `new_haven_new_cases` vector.

From step 4 you should note that the `new_cases` vector has one less element that the `date` vector. This makes sense because we can't calculate the number of new cases on the very first day. Thus we should append a `NA` value as the first element of the `new_haven_new_cases` vector using the `c()` function to indicate that we don't know how many new cases there were on the first day. Please do this, and then use the base R `plot()` function to plot the number of new cases as function of the date for New Haven.

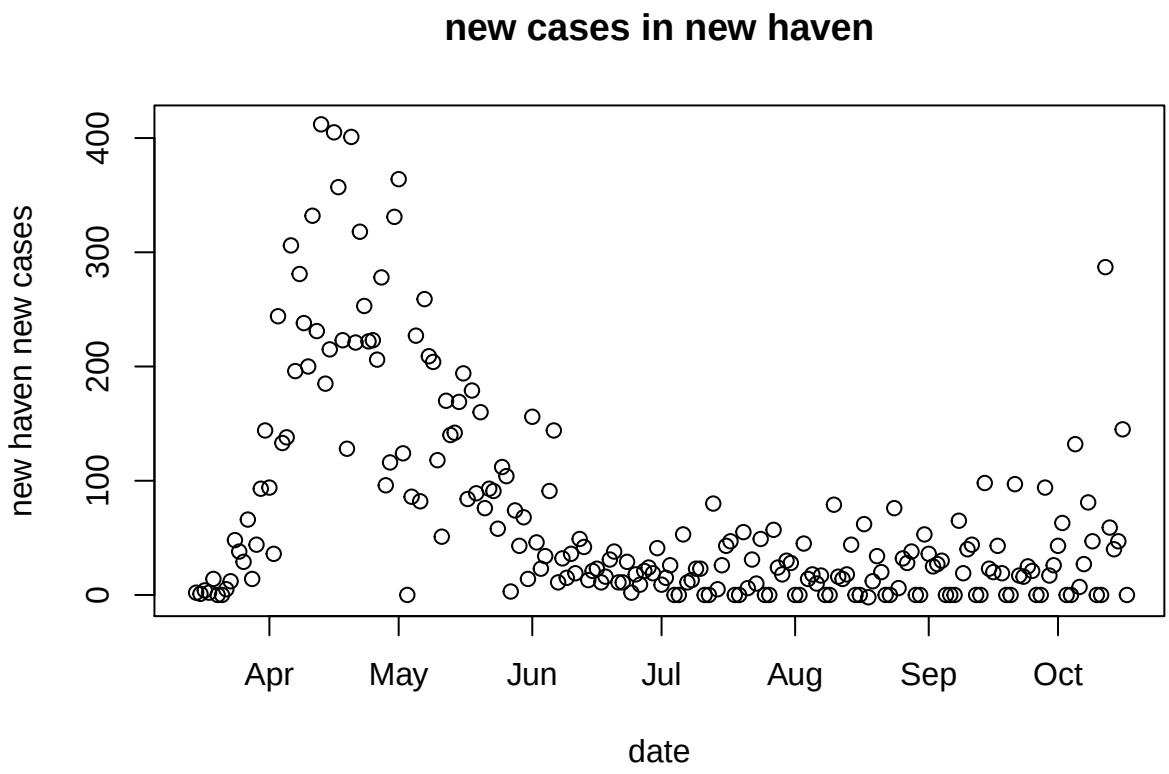
```
new_haven_sorted<-county_covid_cases%>%filter(county == "New Haven")%>%arrange(date)

new_haven_new_cases<-diff(new_haven_sorted$cases)

new_haven_date<-new_haven_sorted$date

length(new_haven_date)
```

```
## [1] 218
```

```
length(new_haven_new_cases)
```

```
## [1] 217
```

```
new_haven_new_cases<-c(NA, new_haven_new_cases)


plot(new_haven_date,
     new_haven_new_cases,
     xlab = "date",
     ylab = "new haven new cases",
     main = "new cases in new haven")
```

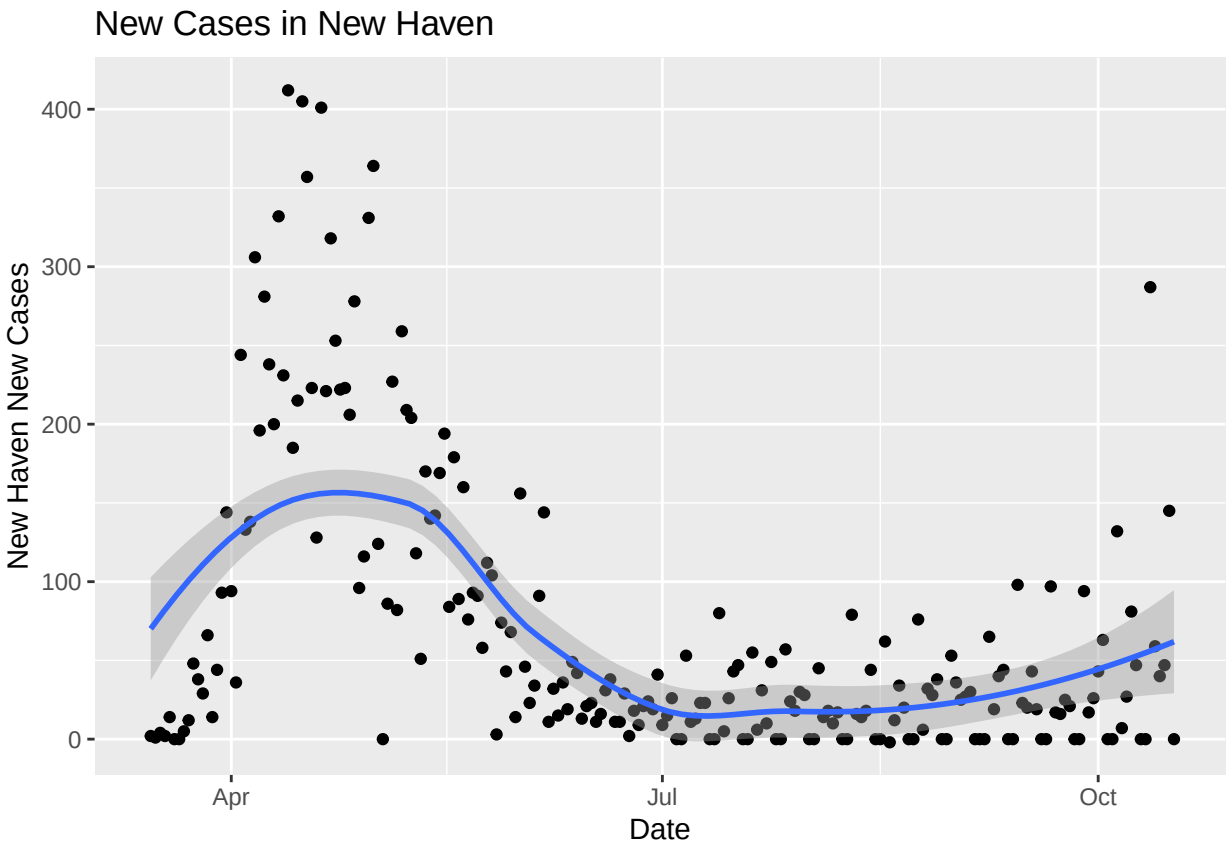


Part 1.3 (10 points):

Now let's recreate the plot in part 1.2 but using ggplot. To do this, recreate your `new_haven_cases` data frame, but now use the `mutate()` function to add a new variable called `new_cases` to this data frame. Once you have done this, use ggplot to visualize the data with a point for each day. Also add other layer showing the smoothed trends using the `geom_smooth()` **(for all future plots, be sure to also add a layer that has the smoothed results)**.

```
(new_haven_cases<-county_covid_cases%>%
  filter(county == "New Haven")%>%
  arrange(date)%>%
  mutate(new_cases = c(NA, diff(new_haven_sorted$cases)))%>%

           ggplot(aes(date, new_cases))+
           geom_point()+
           geom_smooth()+
           xlab("Date")+
           ylab("New Haven New Cases")+
           ggtitle("New Cases in New Haven"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Part 1.4 (10 points): Now let's now visualize the data from Connecticut as a whole. To do this, create a data frame called `conn_cases` that has data only from Connecticut summed over all the counties. Then add a variable to this data frame called `new_cases` which has the new cases for each date. Once you have created the `conn_cases` data frame, please visualize the number of new cases as a function of the date using ggplot.

```
conn_cases<-county_covid_cases%>%
    filter(state == "Connecticut")%>%
    group_by(date)%>%
    summarise(total_cases = sum(cases))%>%
    arrange(date)%>%
    mutate(new_cases = c(NA, diff(total_cases)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

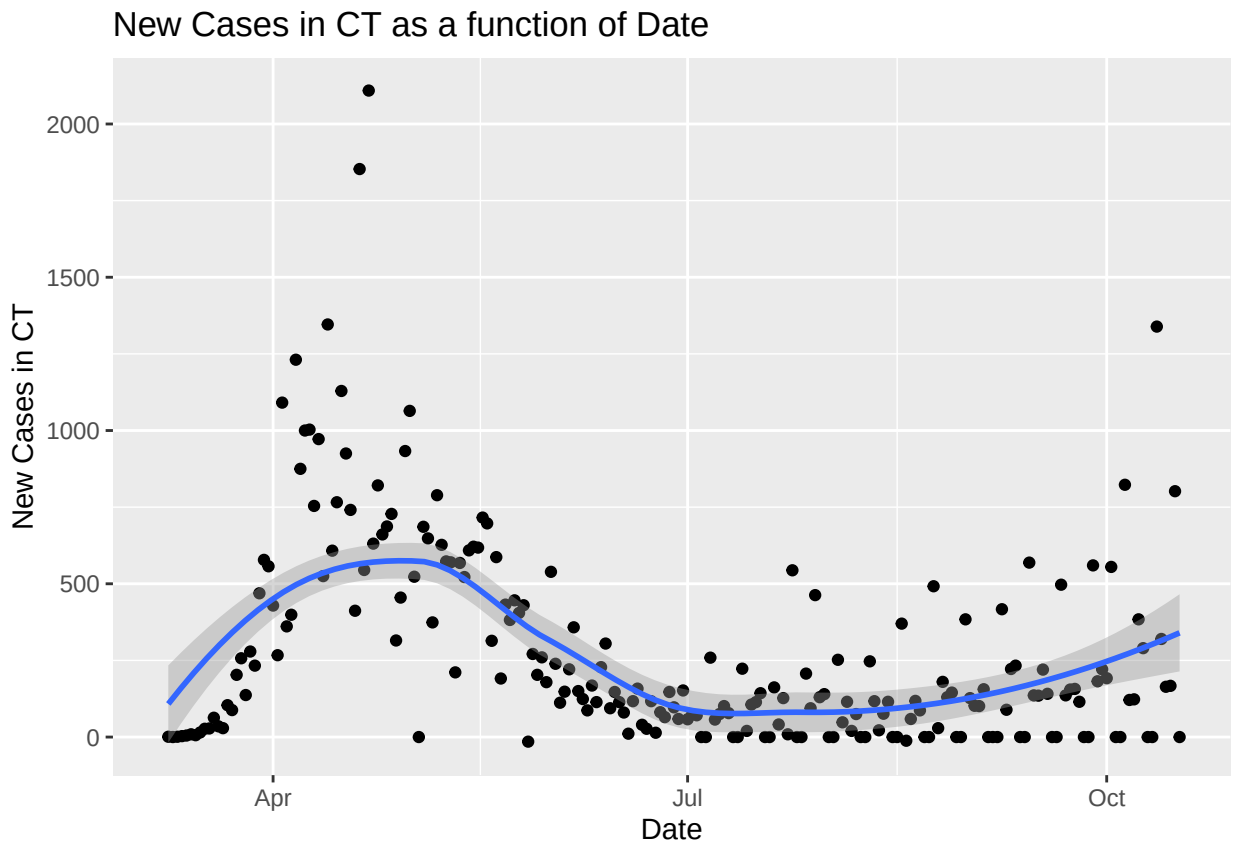
```
conn_cases
```

```
## # A tibble: 224 x 3
##    date       total_cases new_cases
##    <date>           <int>     <int>
##  1 2020-03-08           1        NA
##  2 2020-03-09           2         1
##  3 2020-03-10           2         0
```

```
##  4 2020-03-11           3         1
##  5 2020-03-12           6         3
##  6 2020-03-13          11         5
##  7 2020-03-14          20         9
##  8 2020-03-15          26         6
##  9 2020-03-16          41        15
## 10 2020-03-17          68        27
## # ... with 214 more rows
```

```
ggplot(conn_cases, aes(x = date, y = new_cases))+
  geom_point()+
  geom_smooth()+
  xlab("Date")+
  ylab("New Cases in CT")+
  ggtitle("New Cases in CT as a function of Date")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Part 1.5 (15 points): Now let's plot the new cases as a function of the date for all states. To do this create a data frame called `state_covid_cases` that has the number of new cases in each state for each date. Hint: you should repeat the processes done in part 1.4 by first getting the totals for all states (i.e., summed over counties), and then adding the variable new cases for all states. Using the `group_by()` function

appropriately in conjunction with `summarize()` and then with `mutate()` will be useful. Also to remove any possible errors, filter this data frame so that the `new_cases` variable is always greater than 0.

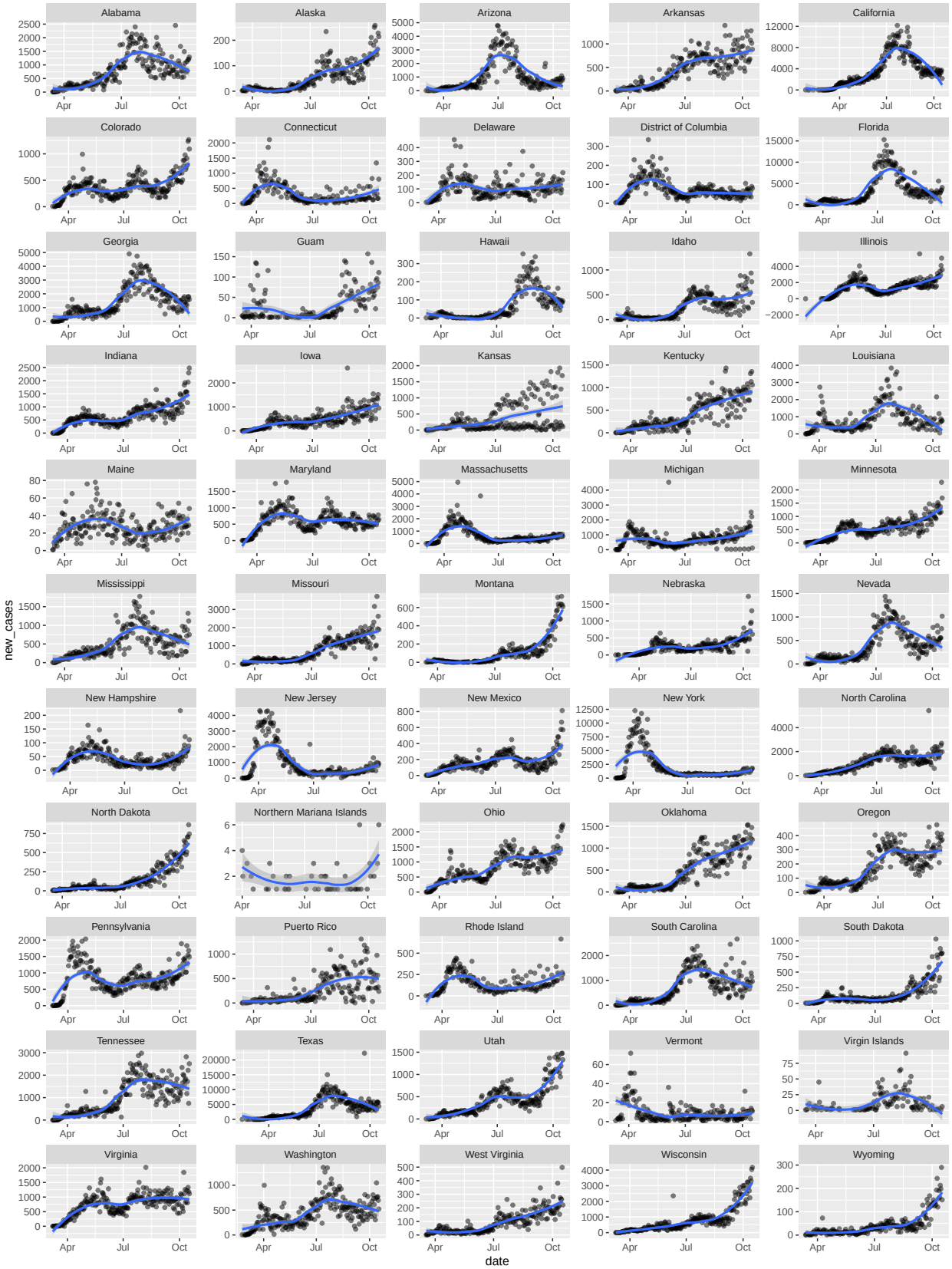
Once you have created `state_covid_cases` data frame, plot the new cases as a function of the date, and use a `facet_wrap()` to create the plot for all states. In the `facet_wrap()`, set the arguments `scales = "free"` to allow the different plots to have different y-axis scales, and use the argument `ncol = 5` to have 5 columns in the plot. In the answer section below, make a few comments on anything interesting you see.

```
state_covid_cases<-county_covid_cases%>%
  group_by(date, state)%>%
  summarize(cases = sum(cases))%>%
  arrange(state, date)%>%
  group_by(state)%>%
  mutate(new_cases = c(NA, diff(cases)))%>%
  filter(new_cases>0)
```

```
## `summarise()` regrouping output by 'date' (override with `.groups` argument)
```

```
state_covid_cases%>%
ggplot(aes(date, new_cases))+
  geom_point(alpha = .5)+
  facet_wrap(~state, scales = "free", ncol = 5)+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
District of Columbia
Florida
Georgia
Guam
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
Missouri
Montana
Nebraska
Nevada
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Northern Mariana Islands
Ohio
Oklahoma
Oregon
Pennsylvania
Puerto Rico
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virgin Islands
Virginia
Washington
West Virginia
Wisconsin
Wyoming
Apr
Jul
Oct
new_cases
date

Answers:

While New York started out with the most new cases at the beginning of the pandemic in the US, it spent a few months after on the decline. While my home state of California initially had a slower increase than New York in terms of new cases, around July the number of new cases increased heavily. During the summer season, it seems that new casees in Florida peaked but now during the fall months new cases are on the decline, likely because not as many people are vacationing in Florida now that school has started.

Part 2: Creating maps of the COVID-19 data

In part 3 of homework 5 you created a bar chart of the number of COVID-19 cases per capita for each state. While this plot might have "spoken to your senses without fatiguing your mind" this figure makes it hard to see spatial relationships between the levels of outbreaks in different states. Let's redo this now using a map which should make the relationships between different states easier to see.

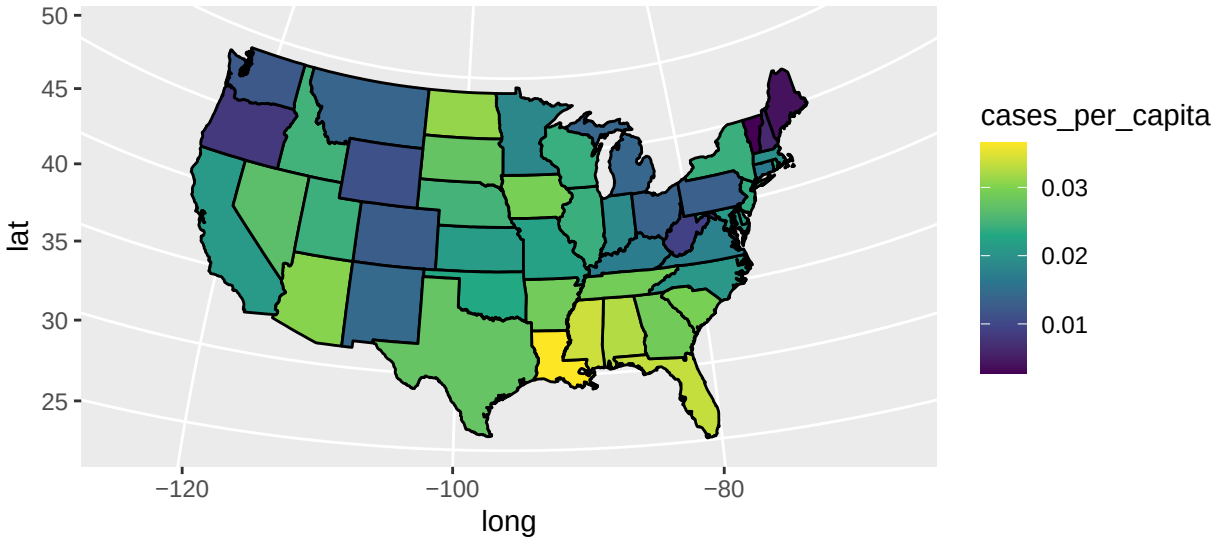
Part 2.1 (6 points): The code chunk below loads a data frame called `state_cases_per_capita` that has the total number of COVID-19 cases per capita for each state as of October 4th 2020 (this is the same data frame you created on homework 5 part 3 where it was called `inner_joined_cases_pop`). It also loads the package `maps` which contains map coordinate information. Once the maps package is loaded, we can use ggplot's `map_data("state")` function to get a data frame that contains coordinates of the borders of all states in the United States. Please use a left join to join the `state_cases_per_capita` information on to the state map data frame, and arrange the resulting data frame by variables `group` and `order`. Save the resulting data frame in an object called `states_covid_map`.

Note: If you forget to arrange the data after joining the data tables you might end up with a strange looking map).

```
library('maps')

load("state_cases_per_capita.rda")


state_covid_map<-map_data("state")%>%
  left_join(state_cases_per_capita, by = c("region" = "state"))%>%
  arrange(group, order)
```

Part 2.2 (6 points): Now create a choropleth map of the cases per capita for the continental United States. Adjust the color scale of this map to better show the differences between the states ("viridis" could be a good choice for the color palette). Does this speak to your senses more than the plot you created in Part 3.5 in homework 5?

```
ggplot(state_covid_map, aes(x= long, y = lat, group = group, fill = cases_per_capita))+
  geom_polygon(color = "black")+
  coord_map("polyconic")+
  scale_fill_continuous(type = "viridis")
```



Answers:

Yes this speaks to my senses much more and I prefer the chloropleth since I can visually locate based upon a color scale where exactly hot spots of corona-virus cases per capita are in relation to other states and with a picture of surrounding geography. The other graph was too lengthy and did not depict geography at all.

Rhetorical question (0 points): which type of map projection do you prefer??

Part 3: Simple linear regression

The 2020 election is coming up on November 3rd. To gain practice creating simple linear regression models, and to see if we can learn from the past, let's analyze data from the 2000 election!

In 2000, the United States presidential election was between a Yale alumnus, George W. Bush who was the Republican candidate, and a Harvard alumnus Al Gore who was the Democratic candidate. There were also a number of "third-party" candidates such as Princeton alumnus Ralph Nader who was the Green Party candidate, and Georgetown alumnus Pat Buchanan who was the Reform Party candidate.

The code chunk below contains data from the 2000 election for the state of Florida in a data frame called `florida data`. Each observational unit in this data frame contains information from the 67 counties in Florida including demographic information on each county as well as the votes received by each candidate in each county.

In the exercises below, you will use linear regression to look at the relationship between the votes that the Republican candidate *George W. Bush* received and the votes that the Reform candidate *Patrick Buchanan* received.

```
load('florida_vote_data_2000.Rda')

dim(florida_data)
```

```
## [1] 67 22
```

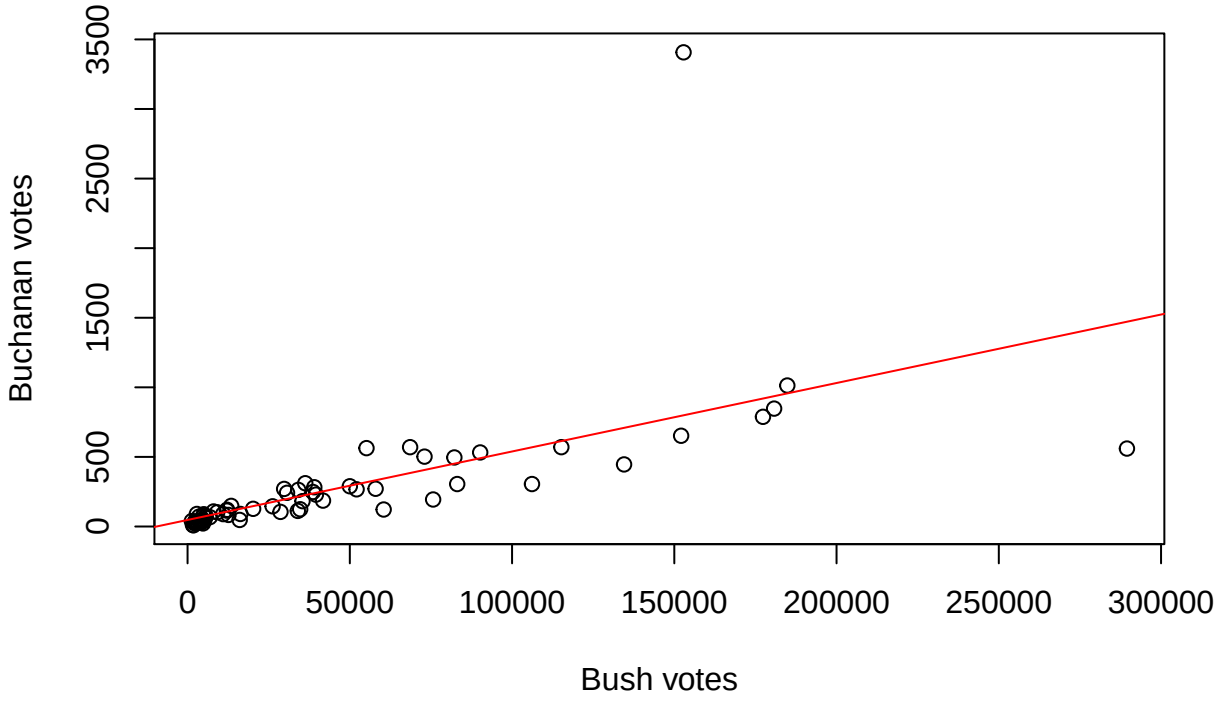
Part 3.1 (6 points):

Start the analysis by creating a scatter plot of the number of votes that Pat Buchanan received as a function of the votes that George Bush received *using base R graphics*. Then fit a linear model that can predict the number of votes Buchanan should receive given the number of votes that Bush received, and add the regression line to this plot in red. In the answer section below describe notabsxle features of this graph (i.e., trends and unusual points).

```
plot(florida_data$Bush, florida_data$Buchanan,
     xlab = "Bush votes",
     ylab = "Buchanan votes")

cor(florida_data$Bush, florida_data$Buchanan)
```

```
## [1] 0.6250012
```

```
lm_fit2<-lm(Buchanan ~ Bush, data = florida_data)
abline(lm_fit2, col = "red")
```



Answers:

The line is positive and relatively shallow and the data points seem to be linear. There seem to be two outliers around (150000, 3500) and (300000, 250).

3

Part 3.2 (5 points):

Now extract the coefficients from the linear model and print them. In the answer section below, report how many votes Buchanan is expected to get for every 1,000 votes Bush received, and how many votes the model predicts that Buchanan would have gotten if Bush had received 0 votes. Also, write an equation that predicts the number of votes Buchanan should get as a function of the number of votes Bush received (make sure to use use $LaTeX$ for the proper notation).

```
print(coef(lm_fit2))
```

```
##  (Intercept)         Bush
## 46.972816323  0.004920082
```

Answers:

$$\hat{y} = 46.97 + 0.004920082x$$

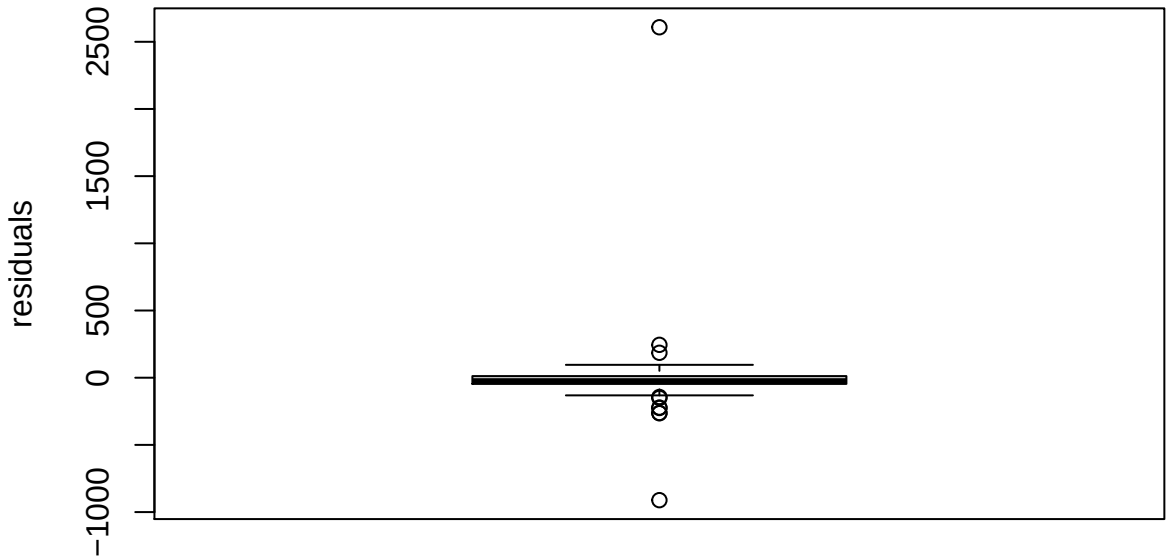
Part 3.3 (12 points):

From looking at the plot above, it should be clear that there is one extreme outlier. To see this more clearly, create a box plot of the residuals of the model below. Then report:

1. What is the county that the outlier corresponds to.
2. How many votes Buchanan actually received in that county.
3. The predicted number of votes that Buchanan should have received for this county based on the regression model fit above for that county.
4. The value of the residual for this county.

Be sure to use the appropriate notation when reporting these numbers. Finally, use the Internet to come up with a reasonable explanation that could have led to this outlier (embedding images in the markdown document could be useful here).

```
# boxplot of the residuals
boxplot(lm_fit2$residuals,
        ylab = "residuals")
```




```
# county that is the outlier
(index_max<-which.max(lm_fit2$residuals))
```

```
## 50
## 50
```

```r
(location<-florida_data[index_max, 2])
```

```
## [1]   Palm Beach
## 67 Levels:   Alachua              ...   Washington
```

```r
# actual number of Buchanan votes
(actual_number_Buchanan<-florida_data$Buchanan[index_max])
```

```
## [1] 3407
```

```r
actual_number_Buchanan
```

```
## [1] 3407
```

```r
# predicted number of Buchanan votes
predicted_votes<-lm_fit2$fitted.values[index_max]
predicted_votes
```

```
##       50
## 798.9876
```

```r
# residual value
residual<-max(lm_fit2$residuals)
residual
```

```
## [1] 2608.012
```

Answers

1. Palm Beach
2. $$y = 3407$$
3. $$\hat{y} = 798.9875$$
4. $$e = 2608.012$$

Part 3.4 (5 points):

Suppose that Buchanan received exactly the number of votes predicted by the regression model, and the residual number of votes he received were intended to be votes for Al Gore. To examine the consequences of this, start by calculating the total number of votes Bush received and the total number of votes Gore received. Then add the residual number of Buchanan votes from the outlier county to the total number of votes that Gore received. Fill in these values in the R Markdown table below to report these numbers. If the residual votes Buchanan received had indeed been intended for Gore, would this have changed who got the majority number of votes in Florida (and hence who would have won Florida)?

```
total_bush<-sum(florida_data$Bush)
total_bush
```

```
## [1] 2910078
```

```
total_gore<-sum(florida_data$Gore)
total_gore
```

```
## [1] 2909117
```

```
residual
```

```
## [1] 2608.012
```

```
total_gore_with_residuals<-total_gore+ residual
total_gore_with_residuals
```

```
## [1] 2911725
```

Answers

Yes. Had the residual votes that Buchanan received gone to Gore, Gore would have had more votes than Bush in Florida (2911725 > 2910078) and won Florida.

Bush votes	Gore + res	Gore votes
2910078	2911725	2608.012

Part 3.5 (2 points):

The United States uses the Electoral College system. In this system, the candidate who got the majority of the vote in a state wins all the Electoral College votes for that state (at least for most of the states in the US including Florida). Use the Internet to find the number of votes that Bush won the Electoral College system by in 2000. Based on the number of Electoral College votes that Florida had, would the outcome of the election changed if Gore have won Florida?

Bonus (0 points), report any possible policy differences that might have been enacted had Gore been elected.

Answers Bush won 271 electoral college votes, which was one more than the majority. Florida had 21 electoral votes, so if Gore had won he would have received 261 + 21 = 282 electoral votes and won the election.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 6.