–Introduction

In this report, I will first discuss some major difficulties I met during the process of cleaning the data, and also how I solved the problem. I will also discuss some findings based on exploration on data from the US Federal Election Committee which include individual contributions based on date, state and state per capita.

–Reading the Data Sets

Consider the facts that this dataset dataset(itcont20) is very large, I read with fread function to speed up the whole reading process and also avoding the warning because of sepeartor, quotation and comment character.

Then, when I moved on to count by date, I found the column that indicates the date contains two different formats, with one 7 digits and one 8 digits. After examination, I found that it is because for a month with one digit, like January, the data record as 1. But for a month with two digits, like October, the data record is 10. Thus, to solve this, I used the paste0 function to paste a 0 in front of every number that has only 7 digits.

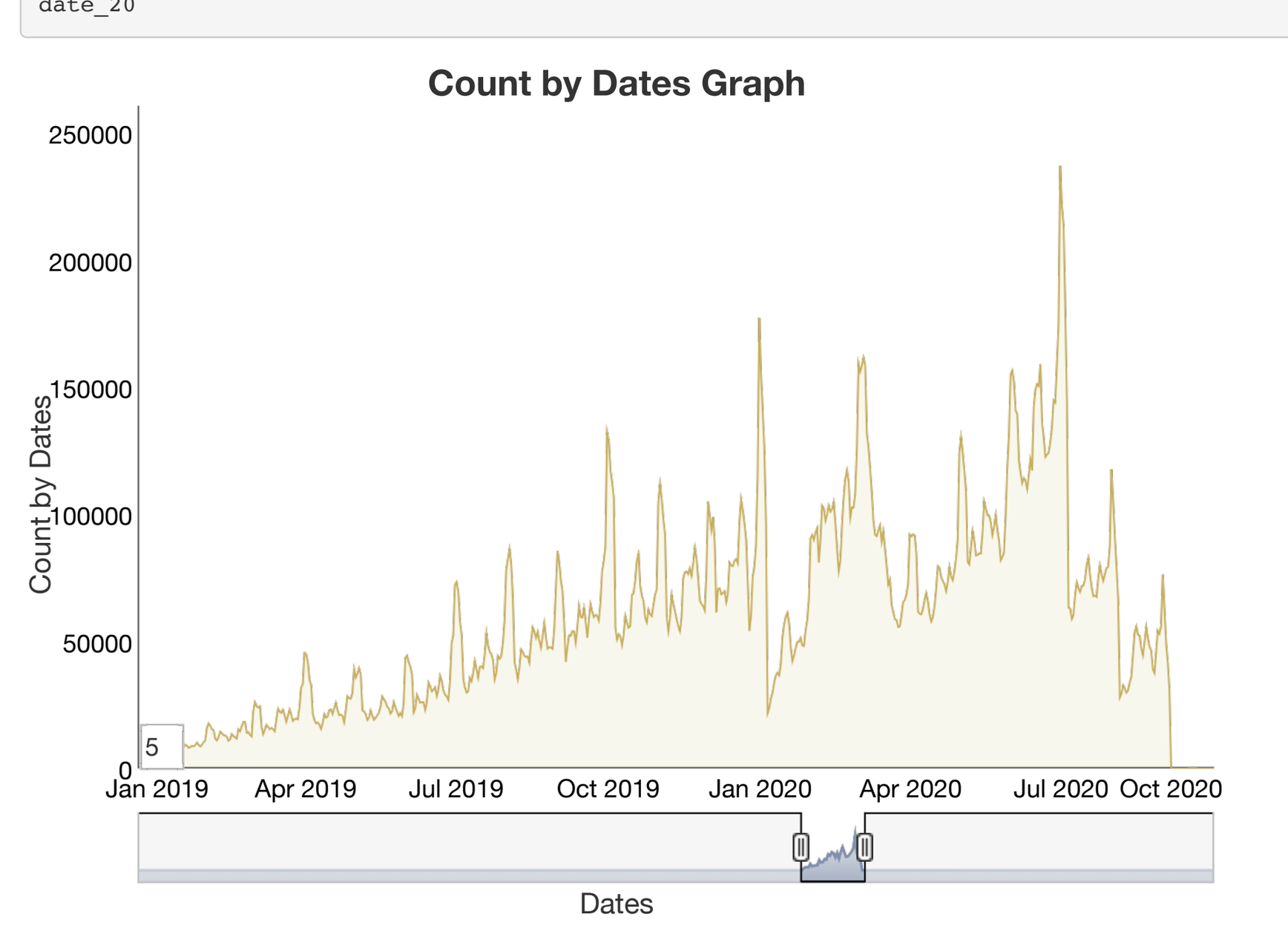
Next, when coding the count states part, I found the state abbreviation is far more than the actual states that United States actually have, I tried to match the state abbreviation with state full name to figure out the problem. I found out this is because the data content individuals outside the united states. So I subset into two groups, one within the states, one outside the states.

When it comes to calculating state per capita, I first download the state population of 2019 from the census government website. I read the csv table into the r, and figure out the correct column should be POPESTIMATE2019. I subset the data into a frame which only has a state name and POPESTIMATE2019. To make sure I can combine this new dataset with the previous dataset I had. I sort both datasets in with state in ascending order, and I cbind two datasets. Then, I used a mutate function to create a new column which indicates states by per capita. However, I found this new column is full of numbers less than one, to make it more readable, I time 100 to each number to make it become a percentage. Then, I used the same method I used for graphing the date graph to graph this state per capita graph.

– Exploring and Interpreting the Data

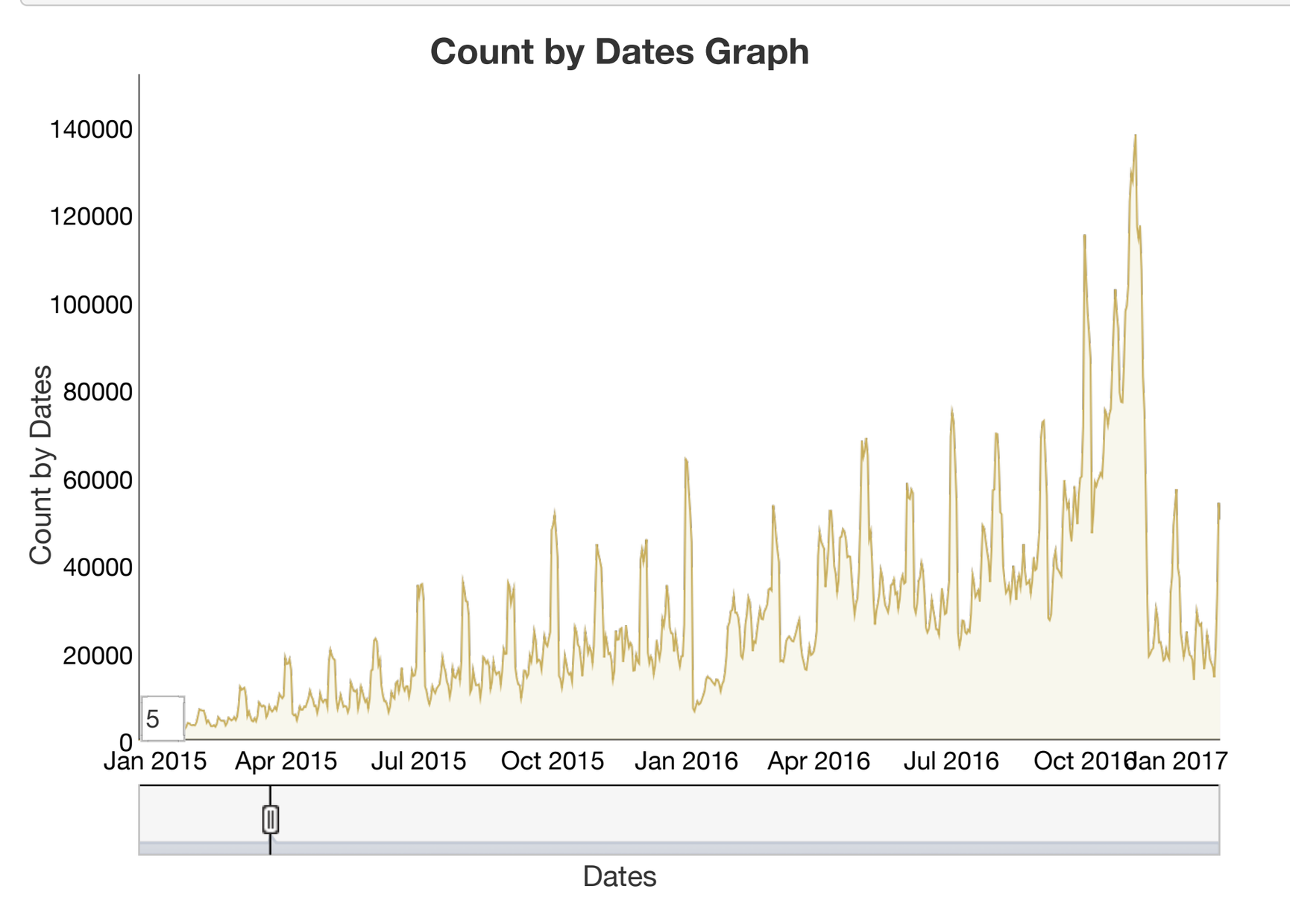
Individuals are more likely to make contributions on certain specific dates, like the end or beginning of each month, for example, December 31st and June 30. It turns out that, by checking the memo text from these dates, for example December 31st, many individuals set bi-weekly or semi-monthly or monthly pay schedule. Thus, it is reasonable that at the end or beginning of each month, the number of contributions are higher compared to mid of a month.

```
date_20
```



This is a graph of numbers of contributions on different dates between 2019 to 2020. It turns out that at the end/beginning of each month, the number of contributions is considerably higher than in the middle of each month, with the peak at June 30, about 0.2 million people made contributions on that single date. The dataset contends some error dates which larger the range of plot, but the majority of the data is focused on the range between 01-01-2019 and 12-31-2020.

```
date_16
```



This is a graph of numbers of contributions on different dates between 2015 to 2016. It turns out that at the end/beginning of each month, the number of contributions is considerably higher than in the middle of each month. This follows the same pattern as I mentioned in the 2019-2020 data.However,this graph with a peak at November 4, which was four days before presidential election About 0.1 million contributions made on that day which means people care more about candidates involved in presidential election.The dataset contends some error dates which larger the range of plot, but the majority of the data is focused on the range between 01-01-2015 and 12-31-2017.

By comparing 2016 contribution data with 2020 contribution data, the number of contribution is significantly increase, contributions made in 2020 almost equal to double amount of contributions made in 2016. More and more individuals are making contributions to support their preferred candidate's political activities.

–Count State

The data consist of both contributions made within the states and out of the states. With the fact that the majority of the contributions made within the states, I focused on just analyzing the contributions made within the states. It turns out that the total number of contributions made in each state are highly associated with the population of the states. With a higher population, the number of contributions will tend to be higher.

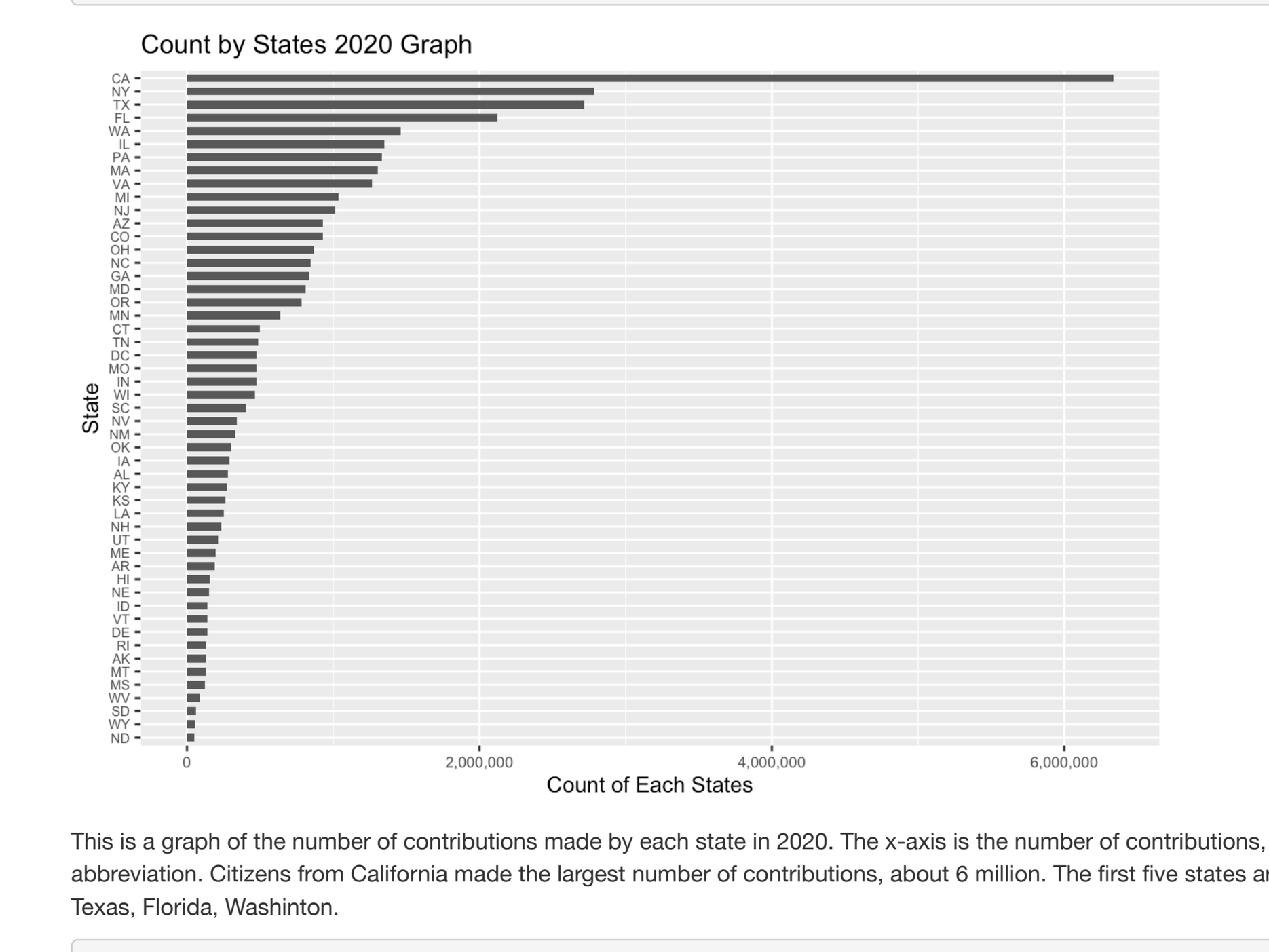
```
sum(America$count)
```

```
## [1] 37285922
```

```
sum(Missing$count)
```

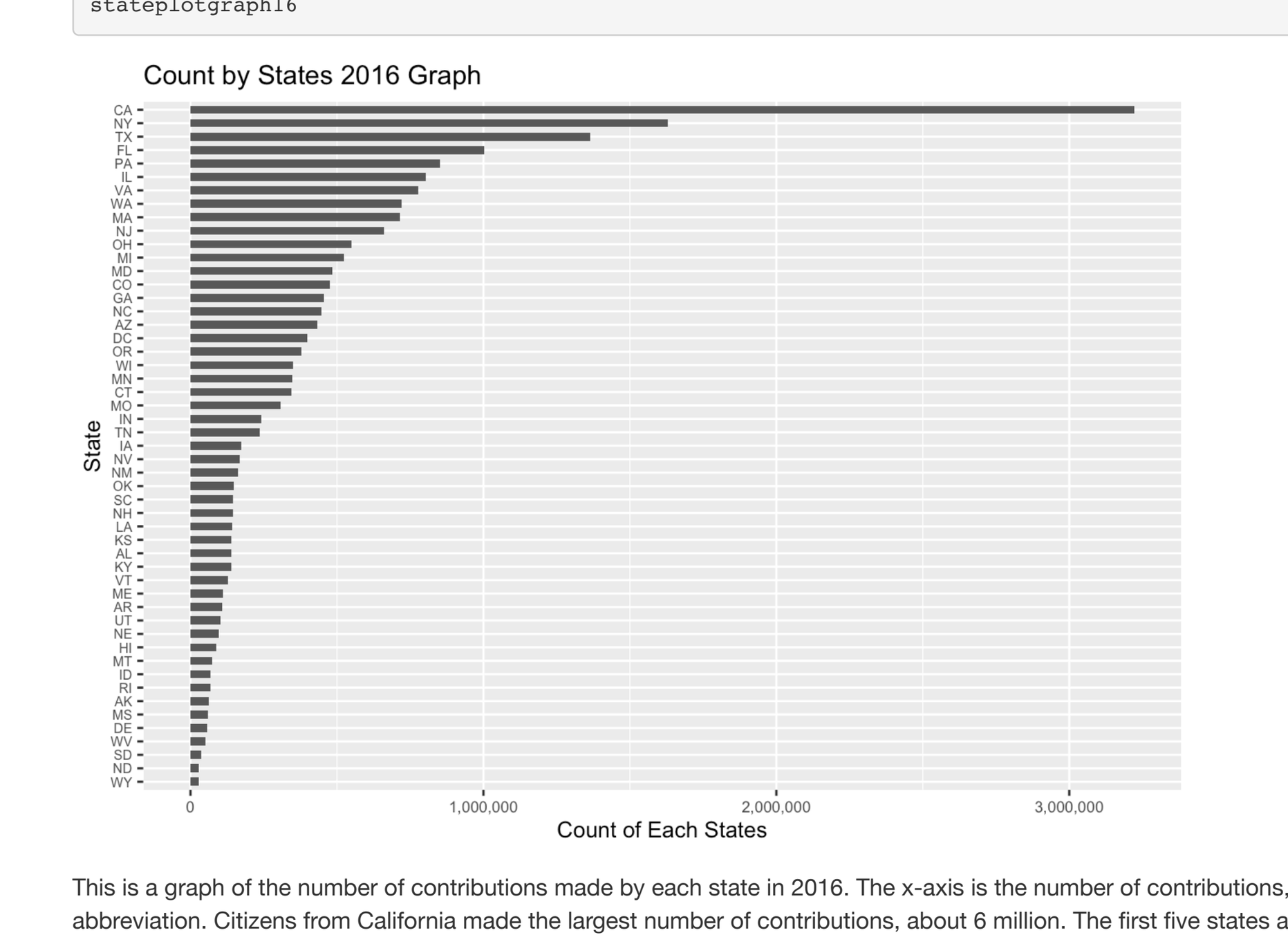
```
## [1] 102039
```

```
stateplotgraph
```



This is a graph of the number of contributions made by each state in 2020. The x-axis is the number of contributions, and y-axis is the state's abbreviation. Citizens from California made the largest number of contributions, about 6 million. The first five states are California, New York, Texas, Florida, Washinton.

```
stateplotgraph16
```



This is a graph of the number of contributions made by each state in 2016. The x-axis is the number of contributions, and y-axis is the state's abbreviation. Citizens from California made the largest number of contributions, about 6 million. The first five states are California, New York, Texas, Florida, Pennsylvania.

Comparing the graph from two time periods, the first five contribution states changed a little bit with the fifth being Pennsylvania instead of Washington. However, in terms of actual numbers, almost all the states double the number of contributions in 2020 compared to 2016. However, surprisingly, it is not because there are about two times that individuals become making contributions. It is actually due to the reason that the frequency of contribution one individual makes within a year increases. This is proved by applying finding the length of a unique individual's name.

```
###Determine how many unique individuals are made contributions
CA2020 <- filter(itcont20,STATE == 'CA')
length(unique(CA2020$NAME))
```

```
## [1] 452022
```

```
CA2016 <- filter(itcont16,STATE == 'CA')
length(unique(CA2016$NAME))
```

```
## [1] 363013
```

```
NY2020 <- filter(itcont20,STATE == 'NY')
length(unique(NY2020$NAME))
```

```
## [1] 221621
```

```
NY2016 <- filter(itcont16,STATE == 'NY')
length(unique(NY2016$NAME))
```

```
## [1] 201101
```

```
TX2020 <- filter(itcont20,STATE == 'TX')
length(unique(TX2020$NAME))
```

```
## [1] 242431
```

```
TX2016 <- filter(itcont16,STATE == 'TX')
length(unique(TX2016$NAME))
```

```
## [1] 199830
```

```
FL2020 <- filter(itcont20,STATE == 'FL')
length(unique(FL2020$NAME))
```

```
## [1] 197264
```

```
FL2016 <- filter(itcont16,STATE == 'FL')
length(unique(FL2016$NAME))
```

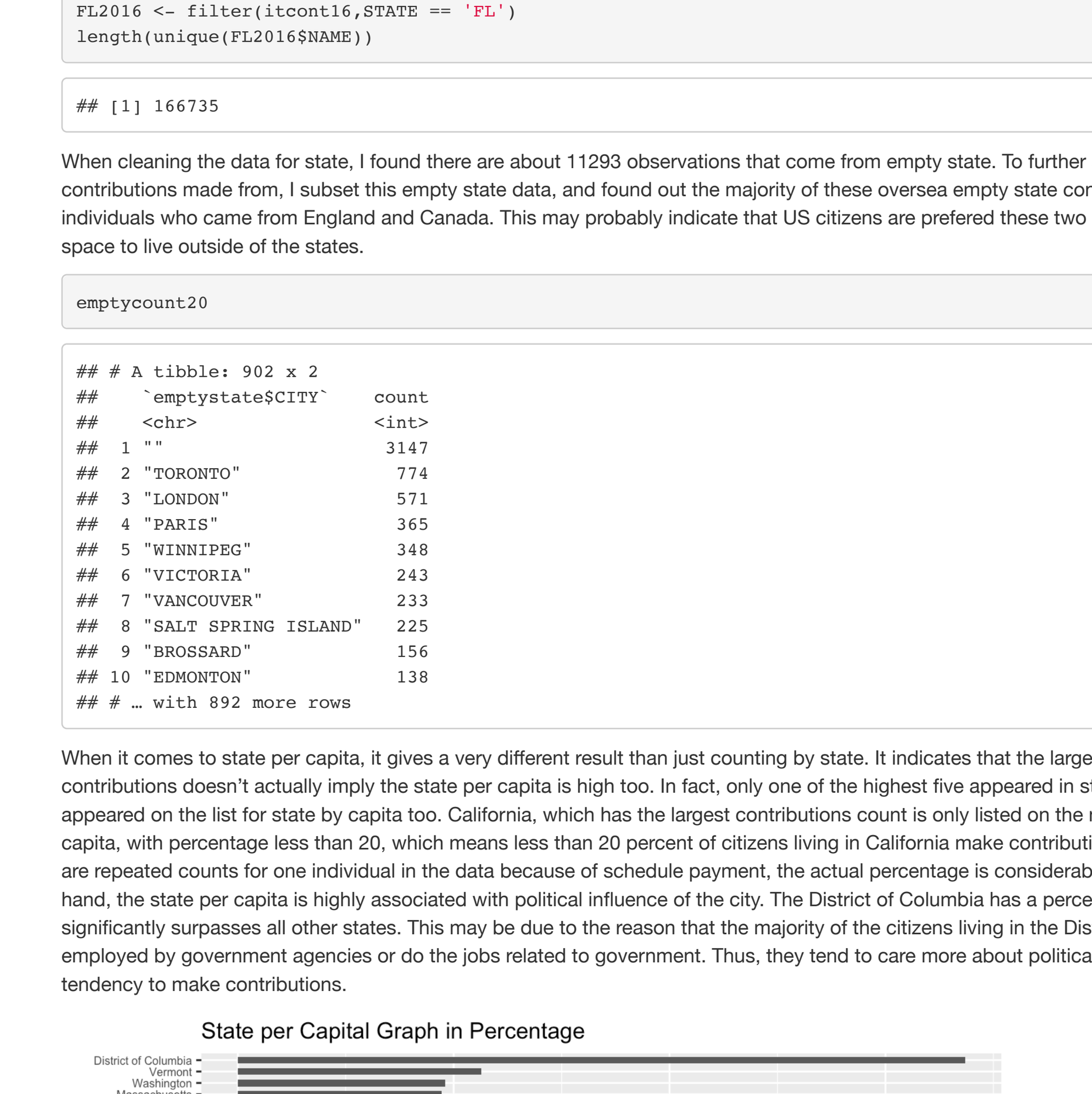
```
## [1] 166735
```

When cleaning the data for state, I found there are about 11293 observations that come from empty state. To further analyze where those contributions made from, I subset this empty state data, and found out the majority of these oversea empty state contributions are made from individuals who came from England and Canada. This may probably indicate that US citizens are prefered these two countries when choosing a space to live outside of the states.

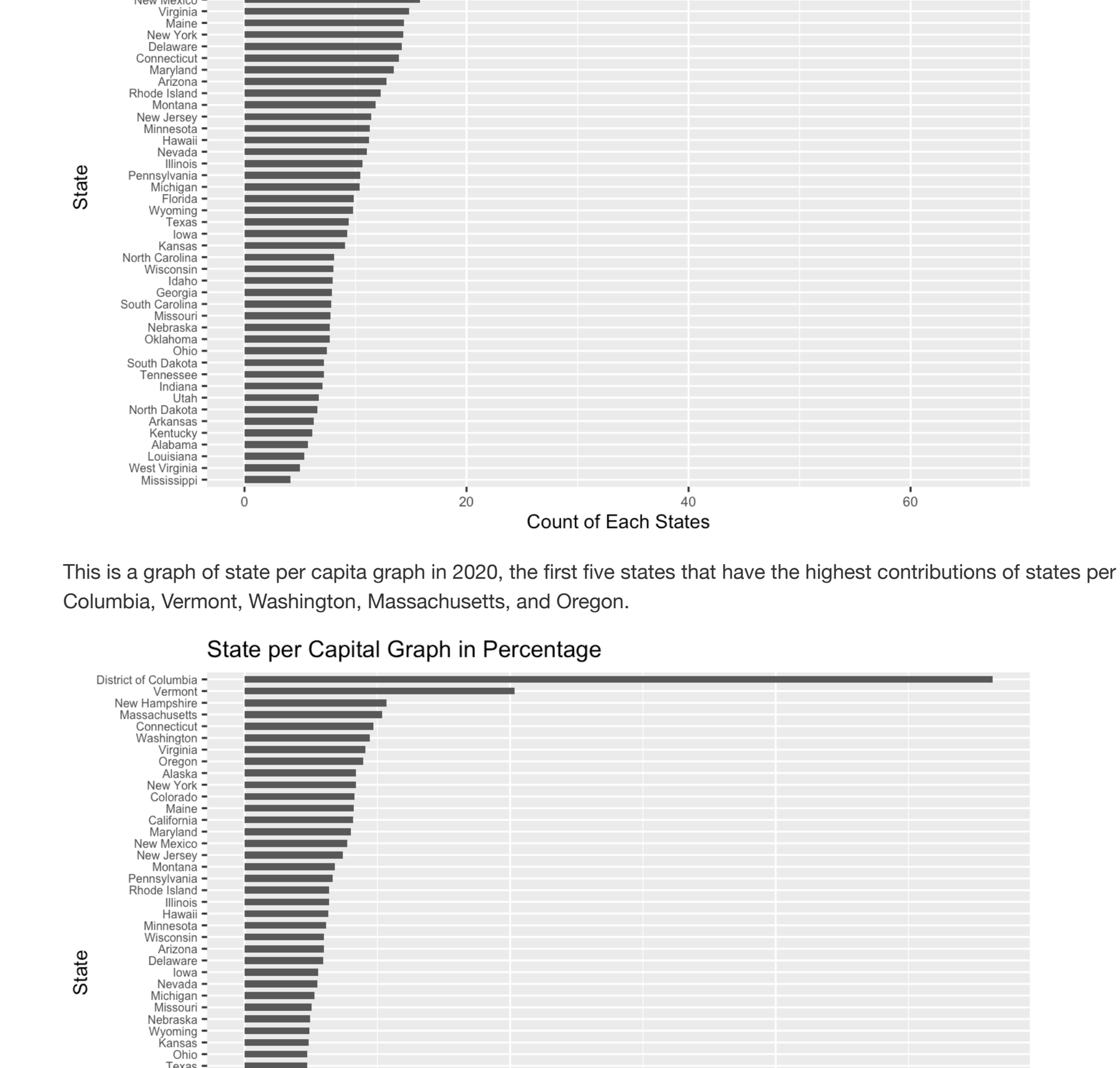
```
emptycount20
```

```
## # A tibble: 902 x 2
##    `emptystate$CITY`    count
##    <chr>                <int>
##  1 ""                    3147
##  2 "TORONTO"              774
##  3 "LONDON"               571
##  4 "PARIS"                365
##  5 "WINNIPEG"             348
##  6 "VICTORIA"             243
##  7 "VANCOUVER"            233
##  8 "SALT SPRING ISLAND"   225
##  9 "BROSSARD"             156
## 10 "EDMONTON"             138
## # … with 892 more rows
```

When it comes to state per capita, it gives a very different result than just counting by state. It indicates that the large number of total contributions doesn't actually imply the state per capita is high too. In fact, only one of the highest five appeared in state count, Washington appeared on the list for state by capita too. California, which has the largest contributions count is only listed on the ninth position in state per capita, with percentage less than 20, which means less than 20 percent of citizens living in California make contributions. Consider the fact there are repeated counts for one individual in the data because of schedule payment, the actual percentage is considerably much lower. On the other hand, the state per capita is highly associated with political influence of the city. The District of Columbia has a percentage over 60 which significantly surpasses all other states. This may be due to the reason that the majority of the citizens living in the District of Columbia are employed by government agencies or do the jobs related to government. Thus, they tend to care more about political activities and have the tendency to make contributions.



This is a graph of state per capita graph in 2020, the first five states that have the highest contributions of states per capita are District of Columbia, Vermont, Washington, Massachusetts, and Oregon.



This is a graph of state per capita graph in 2020, the first five states that have the highest contributions of states per capita are District of Columbia, Vermont, New Hampshire,Massachusetts, and Connecticut. Compared to 2020, the top five states is a little bit different, but it actually just varied by one or two ranks. Still, the District of Columbia remains in first place.

```
setwd('/Users/ssta/Desktop/indiv20')
Header <- read.csv("/Users/ssta/Desktop/indiv_header_file.csv")
library(tidyverse)
library(lubridate)
library(data.table)
library(stringr)
library(ggplot2)
library(xts)
library(dygraphs)
###Read Data
###Consider the facts that this dataset dataset(itcont20) is very large, I read only first 10 columns at the beginning and determine the structure of the data, after I found each cell is divided by character '|', I used the read.table function to read the whole dataset with sepearter = '|', but later, I found there was an error, and I fixed the code with quotation = ' '. The second time I tried to read the whole datasets, I facing another error message and the reading speed is considerably slow, so I change the code to fread function, and this function allows my computer to read the whole datasets much faster and without needs of specification like separator = '|' or quotation = ' '.
itcont20 <- fread('itcont.txt',sep = '|',quote = '')
###I read the header csv file and extract the header from the header files and fix the missing header for the dataset(itcont20).
names <- names(Header)
oldnames = c('V1','V2','V3','V4','V5','V6','V7','V8','V9','V10',
'V11','V12','V13','V14','V15','V16','V17','V18','V19','V20','V21')
newnames = c( "CMTE_ID","AMNDT_IND","RPT_TP","TRANSACTION_PGI","IMAGE_NUM","TRANSACTION_TP","ENTITY_TP","NAME",
"CITY","STATE","ZIP_CODE","EMPLOYER","OCCUPATION","TRANSACTION_DT","TRANSACTION_AMT","OTHER_ID",
"TRAN_ID","FILE_NUM","MEMO_CD","MEMO_TEXT","SUB_ID" )
setnames(itcont20, oldnames, newnames)
itcont16 <- fread('itcont16.txt',sep = '|',quote = '')
setnames(itcont16, oldnames, newnames)
###Count Date
###1. prefix value to make them all 8 digits by pasting 0 in front of 7 digits number
###2. using pipeline function avoid creating multiple variable names, it allow me to futher code based on the results above code by using %>% command.
###3.applying the lubridate function to fix the format of date into yyyy-mm-dd format.
###4.omit NA data since it is relatively small compared to the size of the dataset

numberfixed <- itcont20$TRANSACTION_DT
numberfixed <- str_pad(numberfixed, 8, pad = "0")### prefix value to make them all 8 digits
#data_itcont20copy<-subsetitcont20
itcont20$numberfix <- numberfixed
datecount <- itcont20 %>%
group_by(itcont20$numberfix) %>%
summarise(count=n())
names(datecount)[1] <- 'Numberfix'
Numberfix<-lubridate::mdy(datecount$Numberfix) ### change to form like yyyy-mm-dd
datecount$numberdate <- Numberfix
newdatecount <- na.omit(datecount)
datecount
numberfixed16 <- itcont16$TRANSACTION_DT
numberfixed16 <- str_pad(numberfixed16, 8, pad = "0")### prefix value to make them all 8 digits
#data_itcont20copy<-subsetitcont20
itcont16$numberfix16<-numberfixed16
datecount16 <- itcont16 %>%
group_by(itcont16$numberfix16) %>%
summarise(count=n())
names(datecount16)[1] <- 'Numberfix16'
Numberfix16 <- lubridate::mdy(datecount16$Numberfix16) ### change to form like yyyy-mm-dd
datecount16$numberdate16 <- Numberfix16
newdatecount16 <- na.omit(datecount16)
datecount16
###Why data clustered on certain dates
December3119 <- filter(itcont20, numberfixed ==12312019)
June3120 <- filter(itcont20, TRANSACTION_DT ==6302020)
###Date Graph
###When it comes to graphing the date graph, I first used ggplot, but I found the range is too wide. To make the graph make sense but at the same time, don't lose any information, I switched to using the dygraph package which allows me to create an interaction graph.
###1.I used xts function to create extensible time series graph. After graphing it out, I found most data cludtered around 2018-12-20 to 2020-10-01.
###2.In order to make the graph that makes sense, I set the range of the graph between 2018-12-20 to 2020-10-01 to capture the most gathered data. But, still the lower range selector is up to change with any range.
dategraph <- xts(x = newdatecount$count, order.by = newdatecount$numberdate)
date_20 <- dygraph(dategraph,main='Count by Dates Graph') %>%
dyOptions(labelsUTC = TRUE, fillGraph = TRUE, fillAlpha = 0.1, drawGrid = FALSE, colors = "#C8AE5C") %>%
dyRangeSelector() %>%
dyCrosshair(direction = "vertical") %>%
dyHighlight(highlightCircleSize = 10, highlightSeriesBackgroundAlpha = 1, hideOnMouseOut = TRUE)  %>%
dyRoller(rollPeriod = 5)%>%
dyRangeSelector(dateWindow = c("2018-12-20", "2020-10-01"))%>%
dyAxis("y", label = "Count by Dates")%>%
dyAxis("x", label = "Dates")
date_20
###Date Graph 2016
dategraph16 <- xts(x = newdatecount16$count, order.by = newdatecount16$numberdate16)
date_16 <- dygraph(dategraph16,main='Count by Dates Graph') %>%
dyOptions(labelsUTC = TRUE, fillGraph = TRUE, fillAlpha = 0.1, drawGrid = FALSE, colors = "#C8AE5C") %>%
dyRangeSelector() %>%
dyCrosshair(direction = "vertical") %>%
dyHighlight(highlightCircleSize = 10, highlightSeriesBackgroundAlpha = 1, hideOnMouseOut = TRUE)  %>%
dyRoller(rollPeriod = 5)%>%
dyRangeSelector(dateWindow = c("2014-12-20", "2017-01-01"))%>%
dyAxis("y", label = "Count by Dates")%>%
dyAxis("x", label = "Dates")
date_16
###Count State--Within the State
###1.using the group by function to make a summarise with count function
###2.use the build-in state function to convert state abbreivation into state full name to figure out the reason why there is much more state abbreviation than actual states in united states
###3.subset the dataset into within the state and outside of the state.
statecount<-itcont20 %>%#1
group_by(itcont20$STATE) %>%
summarise(count=n())
names(statecount)[1] <- 'state_ab'
x<-statecount$state_ab
statecount$state_full_name <- state.name[match(x,state.abb)]#2
statecount[16, 3] <- 'District of Columbia'
America <- subset(statecount, (!is.na(statecount[,3]))) #3
Missing <- statecount[rowSums(is.na(statecount)) > 0,]
sum(America$count)
sum(Missing$count)
###Count State 2016--Within the State
statecount16 <- itcont16 %>%#1
group_by(itcont16$STATE) %>%
summarise(count=n())
names(statecount16)[1]<- 'state_ab'
x <- statecount16$state_ab
statecount16$state_full_name <- state.name[match(x,state.abb)]#2
statecount16[31, 3] <- 'District of Columbia'
America16 <- subset(statecount16, (!is.na(statecount16[,3]))) #3
###State Graph--Within the State
###I had some difficulties when graphing the counts by states since there are too many states, all the indicators are squeezed together and the count becomes scientific notation by ggplot default. To fix this problem, I first rotate the whole graph by applying coord_flip code and using the scale function from scale packages to change the scientific notation backs to numbers. Then, I changed the font size of the indicators to make it more readable.

library(scales)
stateplot <- ggplot(America,aes(x=reorder(state_ab,count),y = count))
stateplotgraph <- stateplot+geom_bar(stat = 'identity',width = 0.6, position = position_dodge(width=10))+
  coord_flip()+#flip the whole graph to make it readable
  scale_y_continuous(name="Count of Each States", labels = comma) + # fix the scientific notation
   theme(text = element_text(size = 10))+
  theme(axis.text.y = element_text(size = 6))+
  theme(axis.text.x = element_text(size = 7))+
  xlab("State") +
  ylab("State") +
  ggtitle('Count by States 2020 Graph')# adding xlab,ylab and title'
stateplotgraph
library(scales)
stateplot16 <- ggplot(America16,aes(x=reorder(state_ab,count),y = count))
stateplotgraph16 <- stateplot16 + geom_bar(stat = 'identity',width = 0.6, position = position_dodge(width=10))+
  coord_flip()+#flip the whole graph to make it readable
  scale_y_continuous(name="Count of Each States", labels = comma) + # fix the scientific notation
   theme(text = element_text(size = 10))+
  theme(axis.text.y = element_text(size = 6))+
  theme(axis.text.x = element_text(size = 7))+
  xlab("State") +
  ylab("State") +
  ggtitle('Count by States 2016 Graph ')# adding xlab,ylab and title')
stateplotgraph16
###Determine how many unique individuals are made contributions
CA2020 <- filter(itcont20,STATE == 'CA')
length(unique(CA2020$NAME))
CA2016 <- filter(itcont16,STATE == 'CA')
length(unique(CA2016$NAME))
NY2020 <- filter(itcont20,STATE == 'NY')
length(unique(NY2020$NAME))
NY2016 <- filter(itcont16,STATE == 'NY')
length(unique(NY2016$NAME))
TX2020 <- filter(itcont20,STATE == 'TX')
length(unique(TX2020$NAME))
TX2016 <- filter(itcont16,STATE == 'TX')
length(unique(TX2016$NAME))
FL2020 <- filter(itcont20,STATE == 'FL')
length(unique(FL2020$NAME))
FL2016 <- filter(itcont16,STATE == 'FL')
length(unique(FL2016$NAME))
### Count State--Outside of the State
Missing <- statecount[rowSums(is.na(statecount)) > 0,]
statecount[2, 3] <- 'Armed Forces Americas (except Canada)'
statecount[7, 3] <-'Armed Forces Pacific'
emptystate <- filter(itcont20, STATE =='')
emptycount20 <- emptystate %>%
group_by(emptystate$CITY) %>%
summarise(count=n())%>%
arrange(desc(count))
emptycount20
###Calculate state per capital
###1. download the state population of 2019 from the census government website and read it by read.csv.
###2. subset the data into a frame which only has a state name and POPESTIMATE2019 to make the cleaning easier.
###3. sort the state count within the states in assending order to make the cbine possible.
###4. use pipeline and mutate to create new variable based on manipulation on exsiting variables.
###5. new column is full of numbers less than one, to make it more readable, and easy to understand, I time 100 to each number to make it become a percentage.

population <- read.csv("/Users/ssta/Desktop/2019populationestimate.csv")
populationstate <- subset(population,select=c(NAME,POPESTIMATE2019))
populationstaterevision<-populationstate[-c(1, 53), ]

sort.America <- with(America,  America[order(state_full_name) , ])
rownames(populationstaterevision) <-  1:nrow(populationstaterevision)
populationstaterevisionorder <- order(populationstaterevision$NAME)
combinedstatepop <- cbind(sort.America,populationstaterevision)
rcombinedstatepop <- combinedstatepop%>%
  select(state_full_name, count, POPESTIMATE2019) %>%
  mutate(state_per_capital_percentage = round(100*count/POPESTIMATE2019,2))
###Draw percapitalplot for 2020
###1. draw the plot in the assending order
###2. rotate the whole graph by applying coord_flip code and changed the font size of the indicators to make it more readable.
statepercapitalplot <- ggplot(rcombinedstatepop,aes(x=reorder(state_full_name,state_per_capital_percentage),y = state_per_capital_percentage))
percapitalplot <- statepercapitalplot+geom_bar(stat = 'identity',width=0.6, position = position_dodge(width=10))+
  coord_flip()+
  scale_y_continuous(name="Count of Each States", labels = comma) +
  theme(text = element_text(size=10))+
  theme(axis.text.y = element_text(size=6))+
  theme(axis.text.x = element_text(size=7))+
  xlab("State") +
  ylab("State per Capital") +
  ggtitle('State per Capital Graph in Percentage')
percapitalplot
###Draw percapitalplot for 2016
sort.America16 <- with(America16,  America16[order(state_full_name) , ])
combinedstatepop16 <- cbind(sort.America16,populationstaterevision)
rcombinedstatepop16 <- combinedstatepop16%>%
  select(state_full_name, count, POPESTIMATE2019) %>%
  mutate(state_per_capital_percentage = round(100*count/POPESTIMATE2019,2))

statepercapitalplot16 <- ggplot(rcombinedstatepop16,aes(x=reorder(state_full_name,state_per_capital_percentage),y = state_per_capital_percentage))
percapitalplot16<-statepercapitalplot16+geom_bar(stat = 'identity',width=0.6, position = position_dodge(width=10))+
  coord_flip()+
  scale_y_continuous(name="Count of Each States", labels = comma) +
  theme(text = element_text(size=10))+
  theme(axis.text.y = element_text(size=6))+
  theme(axis.text.x = element_text(size=7))+
  xlab("State") +
  ylab("State per Capital") +
  ggtitle('State per Capital Graph in Percentage')
percapitalplot16
```