Project 17

Name: Elizabeth McGuckin

Recall the 2018 election data, available here: /depot/statclass/data/election2018/itcont.txt and the data dictionary for this data, which is available here: https://www.fec.gov/campaign-finance-data/contributions-individuals- file-description/

1a. Use the system command in R to read the data for the first 100,000 donations and store this data into a file called: shortfile.txt (We use .txt instead of .csv because the file is not comma delimited.)

In [1]:

```
system("head -100000 /depot/statclass/data/election2018/itcont.txt > shortfile.txt")
```

In this line of code we are pulling data from a specific txt file. head-100000 shows the first 100,000 lines of data

1b. Use the read.csv command to read this data into a data frame in R, called: myDF (Hint: check the help for read.csv: ?read.csv to remind yourself about the "sep" and the "header" parameters for read.csv. In particular, this data has "|" as the separator between the data elements, and it does not have a header.)

In [2]:

```
myDF = read.csv("shortfile.txt",header=FALSE, sep="|")
```

Since the data does not have a header we do "header=FALSE" so that we do not give the data a false header. "Sep" separates the columns of data.

1c. Check the dimension of the resulting data frame. It should be 100,000 rows and 21 columns.

In [3]:

```
dim(myDF)
```

1. 100000
2. 21

dim stands for dimension and checks how many row and columns there are.

2a. Split the data for these 100,000 donations according to the State from which the donation was given. Store the resulting data in a list called: myresult (Hint: Check the data dictionary for the meanings of the columns, since we do not have column headers.) (Another hint: Remember that we can refer to a column of data in a data frame by its number, for instance, myDF[[8]] is the name of the donor.)

In [4]:

```
myResults=split(myDF[[15]],myDF[[10]])
```

"Split" lets us separate the State from donation amount.

2b. Check the names of myresult: names(myresult) We see the the first element of the list does not have a name. This is a pain! To solve this, you can give it a name, for instance, by writing: names(myresult)[1] <- "unknown" (or any other kind of name that you want, to indicate that the name is unknown)

In [7]:

```
names(myResults)[1]<-"unknown"
```

We are labeling the missing mumber unknown here.

3a. Find the mean donation amount, according to each state.

In [9]:

```
myMean<-sapply(myResults, mean)
```

Here we are finding the mean donation amount

3a. Find the mean donation amount, according to each state.

In [9]:

```
myMean<-sapply(myResults, mean)
```

Here we are finding the mean donation amount

3b. What is the mean donation from Hoosiers (i.e., for people from Indiana)?

In [10]:

```
myMean["IN"]
```

IN: 367.914678899083

Here we are finding the mean for Indiana.

3c. Find the standard deviation of the donation amount, according to each state.

In [11]:

```
mySD<-sapply(myResults, sd)
```

Here we are finding the standard deviation

3d. Find the number of donations, according to each state.

In [14]:

```
myDon<-sapply(myResults, length)
```

Length allows us to see all of the donations for all of the states

3e. For a sanity check, make sure that the number of donations in 3d adds up to 100,000 altogether.

In [13]:

```
sum(myDon)
```

100000

"Sum" allows us to see donations for all of the states