Problem Set 1

Sebastian Urbina

28 September, 2023

From the files section on CourseWorks, download the file fec22.txt, which contains data for candidate political action committees for the 2022 elections in the U.S. Use the file fec.codebook.txt to see the values for the fields. Write R code to do the following.

1.Read the data into a data object called fec22.df using the appropriate command. Report the number of records/observations in the data.

```
#Importing data fec22.txt from local disc
# Setting the working directory
setwd(path)
```

We will set the working directory in the folder where the data in located locally.

```
#Loading data
fec22.df <- read.csv("fec22.txt", header=FALSE, sep = "|")
#Names of variables
names(fec22.df)
```

```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24"
## [25] "V25" "V26" "V27" "V28" "V29" "V30"
```

Using the read.csv function we are capable of importing the data into R as a data frame. We are using as separetor the "|" statement and indicating that the data set not have a header. We can see that the names of the variables are not includes, and their respective names go between V1 to V30. Because of this we will use the codebook **fec.codebook.txt** to guide us in our analysis.

```
# Number of records/observations in the data
print(dim(fec22.df))
```

```
## [1] 4027   30
```

```
########################### Data set characteristics ############################
#V21-v25 variables with constant NA values
print(str(fec22.df))
```

```
## 'data.frame':    4027 obs. of  30 variables:
##  $ V1 : chr  "H2AK00200" "H2AK01158" "H2AK01240" "H2AK00218" ...
##  $ V2 : chr  "CONSTANT,CHRISTOPHER" "PELTOLA,MARY" "WOOL, ADAM L" "REVAK, JOSHUA CARL" ...
##  $ V3 : chr  "C" "I" "O" "O" ...
##  $ V4 : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ V5 : chr  "DEM" "DEM" "DEM" "REP" ...
##  $ V6 : num  164638 7751293 16217 121841 1971161 ...
##  $ V7 : num  0 186868 0 0 112963 ...
##  $ V8 : num  164638 7060033 16217 121841 1924781 ...
##  $ V9 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ V10: num  0 0 0 0 0 ...
##  $ V11: num  0 691260 0 0 46380 ...
##  $ V12: num  615 25 1100 0 0 ...
##  $ V13: num  0 0 0 0 0 0 650000 0 0 0 ...
##  $ V14: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ V15: num  0e+00 0e+00 0e+00 0e+00 0e+00 0e+00 2e+05 0e+00 0e+00 0e+00 ...
##  $ V16: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ V17: num  143180 0 0 0 2525 ...
##  $ V18: num  158023 7149826 15117 116666 1770698 ...
##  $ V19: chr  "AK" "AK" "AK" "AK" ...
##  $ V20: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ V21: logi  NA NA NA NA NA NA ...
##  $ V22: logi  NA NA NA NA NA NA ...
##  $ V23: logi  NA NA NA NA NA NA ...
##  $ V24: logi  NA NA NA NA NA NA ...
##  $ V25: logi  NA NA NA NA NA NA ...
##  $ V26: num  1000 384021 0 5000 81305 ...
##  $ V27: num  5000 10000 0 0 0 0 0 0 0 0 ...
##  $ V28: chr  "12/31/2022" "12/31/2022" "07/15/2022" "09/16/2022" ...
##  $ V29: num  8300 136658 0 14600 43128 ...
##  $ V30: num  0 3913 0 0 1000 ...
## NULL
```

```
glimpse(fec22.df)
```

```
## Rows: 4,027
## Columns: 30
## $ V1  <chr> "H2AK00200", "H2AK01158", "H2AK01240", "H2AK00218", "H2AK00226", "~
## $ V2  <chr> "CONSTANT,CHRISTOPHER", "PELTOLA,MARY", "WOOL, ADAM L", "REVAK, JO~
## $ V3  <chr> "C", "I", "O", "O", "O", "C", "C", "O", "O", "I", "C", "O", "C", "~
## $ V4  <int> 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 2, 2, 1, 1, 2, ~
## $ V5  <chr> "DEM", "DEM", "DEM", "REP", "REP", "REP", "REP", "REP", "REP", "RE~
## $ V6  <dbl> 164637.90, 7751293.39, 16217.07, 121841.00, 1971160.93, 1548.51, 1~
## $ V7  <dbl> 0.00, 186868.19, 0.00, 0.00, 112963.43, 0.00, 0.00, 0.00, 0.00, 17~
## $ V8  <dbl> 164637.90, 7060033.09, 16217.07, 121841.00, 1924781.35, 5621.60, 1~
## $ V9  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ V10 <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 140.00, 0.00, 0.00, 0.00, 249173.19,~
## $ V11 <dbl> 0.00, 691260.30, 0.00, 0.00, 46379.58, 0.00, 41233.99, 0.00, 0.00,~
## $ V12 <dbl> 614.85, 25.00, 1100.00, 0.00, 0.00, 0.00, 23971.16, 0.00, 0.00, 0.~
## $ V13 <dbl> 0, 0, 0, 0, 0, 0, 650000, 0, 0, 0, 27000, 0, 0, 0, 5000, 100000, 0~
## $ V14 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ V15 <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 200000.00, 0.00, 0.00, 0.00, 1~
## $ V16 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ V17 <dbl> 143180.09, 0.00, 0.00, 0.00, 2525.05, 0.00, 450000.00, 0.00, 0.00,~
```

```
## $ V18 <dbl> 158023.05, 7149826.02, 15117.00, 116666.00, 1770697.90, 1548.51, 9~
## $ V19 <chr> "AK", "AK", "AK", "AK", "AK", "AK", "AK", "AK", "AK", "AK", "AK", ~
## $ V20 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, ~
## $ V21 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ V22 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ V23 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ V24 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ V25 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ V26 <dbl> 1000.00, 384020.59, 0.00, 5000.00, 81305.00, 0.00, 36853.74, 0.00,~
## $ V27 <dbl> 5000.00, 10000.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
## $ V28 <chr> "12/31/2022", "12/31/2022", "07/15/2022", "09/16/2022", "12/31/202~
## $ V29 <dbl> 8300.00, 136657.70, 0.00, 14600.00, 43128.37, 0.00, 650.00, 0.00, ~
## $ V30 <dbl> 0.00, 3912.66, 0.00, 0.00, 1000.00, 0.00, 0.00, 0.00, 0.00, 110603~
```

```
# first observations
head(fec22.df)
```

```
##          V1                    V2 V3 V4  V5         V6       V7         V8 V9
## 1 H2AK00200 CONSTANT,CHRISTOPHER  C  1 DEM  164637.90      0.0  164637.90  0
## 2 H2AK01158         PELTOLA,MARY  I  1 DEM 7751293.39 186868.2 7060033.09  0
## 3 H2AK01240         WOOL, ADAM L  O  1 DEM   16217.07      0.0   16217.07  0
## 4 H2AK00218   REVAK, JOSHUA CARL  O  2 REP  121841.00      0.0  121841.00  0
## 5 H2AK00226         PALIN, SARAH  O  2 REP 1971160.93 112963.4 1924781.35  0
## 6 H2AK01059        PURHAM, RANDY  C  2 REP    1548.51      0.0    5621.60  0
##   V10       V11     V12 V13 V14 V15 V16       V17        V18 V19 V20 V21 V22
## 1   0      0.00  614.85   0   0   0   0 143180.09  158023.05  AK   1  NA  NA
## 2   0 691260.30   25.00   0   0   0   0      0.00 7149826.02  AK   1  NA  NA
## 3   0      0.00 1100.00   0   0   0   0      0.00   15117.00  AK   1  NA  NA
## 4   0      0.00    0.00   0   0   0   0      0.00  116666.00  AK   1  NA  NA
## 5   0  46379.58    0.00   0   0   0   0   2525.05 1770697.90  AK   1  NA  NA
## 6 140      0.00    0.00   0   0   0   0      0.00    1548.51  AK   1  NA  NA
##   V23 V24 V25      V26   V27        V28       V29     V30
## 1  NA  NA  NA   1000.0  5000 12/31/2022   8300.00    0.00
## 2  NA  NA  NA 384020.6 10000 12/31/2022 136657.70 3912.66
## 3  NA  NA  NA      0.0     0 07/15/2022      0.00    0.00
## 4  NA  NA  NA   5000.0     0 09/16/2022  14600.00    0.00
## 5  NA  NA  NA  81305.0     0 12/31/2022  43128.37 1000.00
## 6  NA  NA  NA      0.0     0 07/27/2022      0.00    0.00
```

```
#Changing character to date variable
fec22.df$V28 <- mdy(fec22.df$V28)
class(fec22.df$V28)
```

```
## [1] "Date"
```

From this exploratory analysis, we can see that the data has 4027 observations and 30 variables. The variables V21 to V25 have constant NA. Using the glimpse function we can observe in much detail each variable. We can see that the varaible V28 which is a variable of class date its not recognized that way and instead is imported as a character variable. We will correct this using the lubridate package.

2.Report any variables that are missing values systematically. Is this what you expect? Why or why not?

Employing the "str" function, we can first see that V21 to V25 may have systematic NA cases in their observations.

```
#V21-v25 variables with constant NA values
str(fec22.df[,21:25])
```

```
## 'data.frame':    4027 obs. of  5 variables:
##  $ V21: logi  NA NA NA NA NA NA ...
##  $ V22: logi  NA NA NA NA NA NA ...
##  $ V23: logi  NA NA NA NA NA NA ...
##  $ V24: logi  NA NA NA NA NA NA ...
##  $ V25: logi  NA NA NA NA NA NA ...
```

To verify our hypothesis, we will independently sum the NA cases in each column. If, respectively, they have a total of 4027 NA, it would mean that they have a systematic NA problem in that column. For this task, we will apply the "**sum()**" and "**is.na()**" function for each column, giving us the outcome required.

```
#Creating matrix for puttinf results
results1 <- matrix(NA, nrow = 5, ncol = 1 )
#Calculating the number of NA of each variable
results1[1,1] <-sum(is.na(fec22.df$V21))
results1[2,1] <-sum(is.na(fec22.df$V22))
results1[3,1] <-sum(is.na(fec22.df$V23))
results1[4,1] <-sum(is.na(fec22.df$V24))
results1[5,1] <-sum(is.na(fec22.df$V25))
# Naming the columns
colnames(results1) <- c("NA")
#Naming the rows
rownames(results1) <- c("V21","V22","V23","V24","V25")
# Printing the results in a tibble format
print(results1)
```

```
##       NA
## V21 4027
## V22 4027
## V23 4027
## V24 4027
## V25 4027
```

```
#Describe the variables using the describe function from the psych package
print(describe(fec22.df[,c(21:25)]))
```

```
##      vars n mean sd median trimmed mad min  max range skew kurtosis se
## V21*    1 0  NaN NA     NA     NaN  NA Inf -Inf  -Inf   NA       NA NA
## V22*    2 0  NaN NA     NA     NaN  NA Inf -Inf  -Inf   NA       NA NA
## V23*    3 0  NaN NA     NA     NaN  NA Inf -Inf  -Inf   NA       NA NA
## V24*    4 0  NaN NA     NA     NaN  NA Inf -Inf  -Inf   NA       NA NA
## V25*    5 0  NaN NA     NA     NaN  NA Inf -Inf  -Inf   NA       NA NA
```

As expected, variables **V21 to V25** have systematically missing values. This problem can occur because these variables come from another data set merged with the original one. We can hypothesize that the merged process was not correctly done or that the "key" variables for the merging at the moment to perform the left_joint function didn't match the other data set, resulting in an import of the new variables with NA cases. We can hypothesize this because variables v21 to V25 came from the Election result data included in the 1996-2006 files.

3. Subset the data to produce two different data objects—one for Senate candidates and one for House candidates (the variable *CAND OFFICE DISTRICT* equals 0 for Senate candidates, is greater than 0 for House candidates). Do a check that will give you a sense that the subsetting worked correctly.

First, utilizing the case_when and mutate functions, we will create a new variable named "cand" to divide the candidates for each electoral race (Senate and House). This variable will help review if the subsetting data sets worked correctly.

```
#Creating new variable for Sen and Hou candidates, we will use this later to
#check if the subsetting worked us we wanted.
fec22.df <- fec22.df %>%
mutate(cand = case_when(V20 == 0 ~ "Senate",
                        V20 > 0 ~ "House",
                        TRUE ~ NA))

print(table(fec22.df$cand))
```

```
##
##  House Senate
##   3406    617
```

In the table above, it is appreciated that there are 3406 candidates for the House and 617 for the Senate election. We should be capable of performing these same results for each subset created and storing them in their respective objects. To do so, we will use the filter command.

```
# Creating subset for Senate candidates
sen_df <- fec22.df %>%
filter(V20 == 0)

# Creating subset for House candidates
hou_df <-fec22.df %>%
filter(V20 > 0)

#Cheking the subsets for correct outcome Senate
describe(sen_df$V20)
```

```
##    vars   n mean sd median trimmed mad min max range skew kurtosis se
## X1    1 617    0  0      0       0   0   0   0     0  NaN      NaN  0
```

```
summary(sen_df$V20)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

```r
nrow(sen_df)
```

```
## [1] 617
```

```r
#Cheking the subsets for correct outcome House
describe(hou_df$V20)
```

```
##    vars    n  mean    sd median trimmed  mad min max range skew kurtosis   se
## X1    1 3406 10.28 10.58      6    8.26 5.93   1  53    52 1.73     2.79 0.18
```

```r
summary(hou_df$V20)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    3.00    6.00   10.28   14.00   53.00
```

```r
nrow(hou_df)
```

```
## [1] 3406
```

Once the data was subsetted, we used three methods to verify that our subsets correctly separated the data. We first obtain a description of the specific variable in each new subset, where we get the number of observations, the mean, median, sd, min and max, etc. In the case of the Senate, we obtained 617 observations with a Mean, Median, and SD of 0, meaning that all the observations are equal to 0. On the other hand, in the case of the House, we obtained 3406 observations with a minimum value of 1 for all the observations. These results confirm that the subset was divided correctly from the original dataset.

Searching for more validation, we performed two more tests, a summary of statistics and a count of total rows per dataset. These tests corroborated our results.

4. Calculate and report the mean, median, and standard deviation for total receipts (variable name TTL RECEIPTS) for races for each chamber. Do this for the subsets produced in the previous step without using dplyr. Also do this on the original data that you read in (i.e., fec22.df) using dplyr and compare the results from the two approaches.

We will use the base r commands to perform the mean, median and sd for the variable total receipts for each subset. We will put the results in a 2 x 3 matrix, where the rows will be the Senate and House candidates, and the columns will be the mean, median, and sd.

```r
# Creating a matrix for putting the results
results <- matrix(NA, nrow = 2, ncol = 3 )
# Calculating the mean, median and sd for Senate
results[1,1] <- round(mean(sen_df$V6),0)
results[1,2] <- round(median(sen_df$V6),0)
results[1,3] <- round(sd(sen_df$V6),0)
# Calculating the mean, median and sd for House
results[2,1] <- round(mean(hou_df$V6),0)
results[2,2] <- round(median(hou_df$V6),0)
results[2,3] <- round(sd(hou_df$V6),0)
```

```r
# Naming the columns and rows
colnames(results) <- c("Mean","Median","SD")
#Naming the rows
rownames(results) <- c("Senate candidates"," House candidates")
# Printing the results into the matrix created
print(results)
```

```
##                       Mean Median       SD
## Senate candidates 3018274  24984 12378427
##  House candidates  671722  33824  2212341
```

In this case, we will use the dplyr packages, which allow us to use pipelines (>%>), divide the data into groups (group_by), and summarize the statistics (mean, median, sd), all of this in a few lines of code.

```r
# Calculating the mean, median and sd for Senate and House using dplyr and the
#original data, for this we need to group the data by V20 == 0, which is the
#variable that indicates the chamber and all the observations that not equal to
# 0 are House candidates. We will use the summarize finction contain in dplyr.
print(fec22.df %>%
group_by(V20==0) %>%
summarize(mean = mean(V6, na.rm = TRUE),
          median = median(V6, na.rm = TRUE),
          sd = sd(V6, na.rm = TRUE)))
```

```
## # A tibble: 3 x 4
##   `V20 == 0`     mean median        sd
##   <lgl>         <dbl>  <dbl>     <dbl>
## 1 FALSE       671722. 33824.  2212341.
## 2 TRUE       3018274. 24984  12378427.
## 3 NA            5997.  4782.     6224.
```

The results are the same, but using the dplyr package allows us to perform the same result using fewer lines of code, making it preferable.

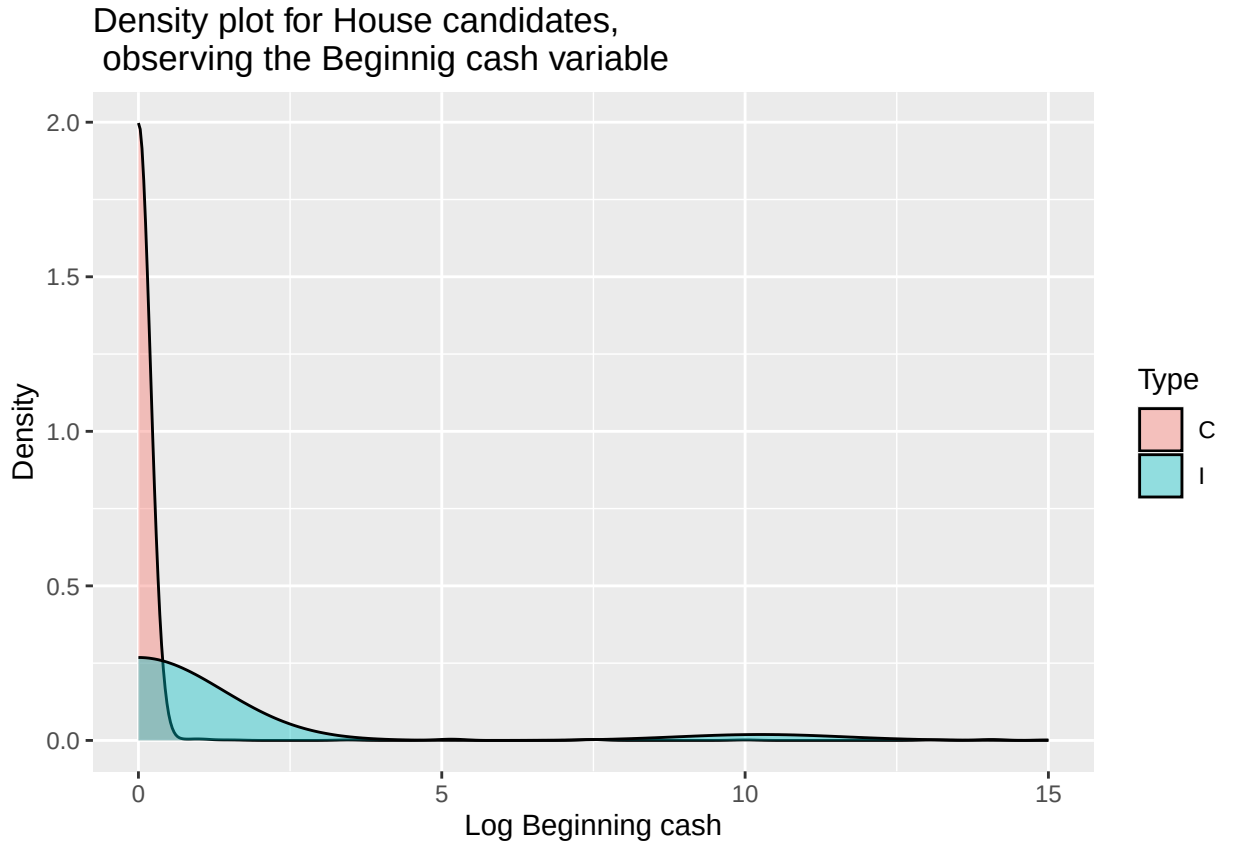
5. For the data that includes only House candidates, produce density plots that shows two distributions—one for candidates who are incumbents and one for candidates who are challengers. The variable CAND ICI equals "I" for incumbents, equals "C" for challengers, and equals "O" for candidates in open seat races. Write a sentence that summarizes what you see.

We will show the density distribution of the variable COH_BOP, which contains the initial cash that candidates have at their disposal. We will scale the x-axis by a logarithm of ten for a more comprehensive graph. Also, we will use the ggplot2 packages to create this figure because it gives us more tools when constructing our representation.

```r
#Density plot for House candidates, incumbents observing the variable V10,
#which is the Beginning cash.
hou_df2 <- hou_df %>%
filter(V3 == "I" |  V3 == "C")
```

```
p <- ggplot(hou_df2, aes(x=V10, y = ..density.., fill= V3)) +
  geom_density(alpha=0.4) +
  scale_x_log10()+
  labs(title = "Density plot for House candidates,\n observing the Beginnig cash variable",
       x = "Log Beginning cash", y = "Density",
       fill = "Type") +
       scale_x_continuous(limits = c(0, 15))

p
```



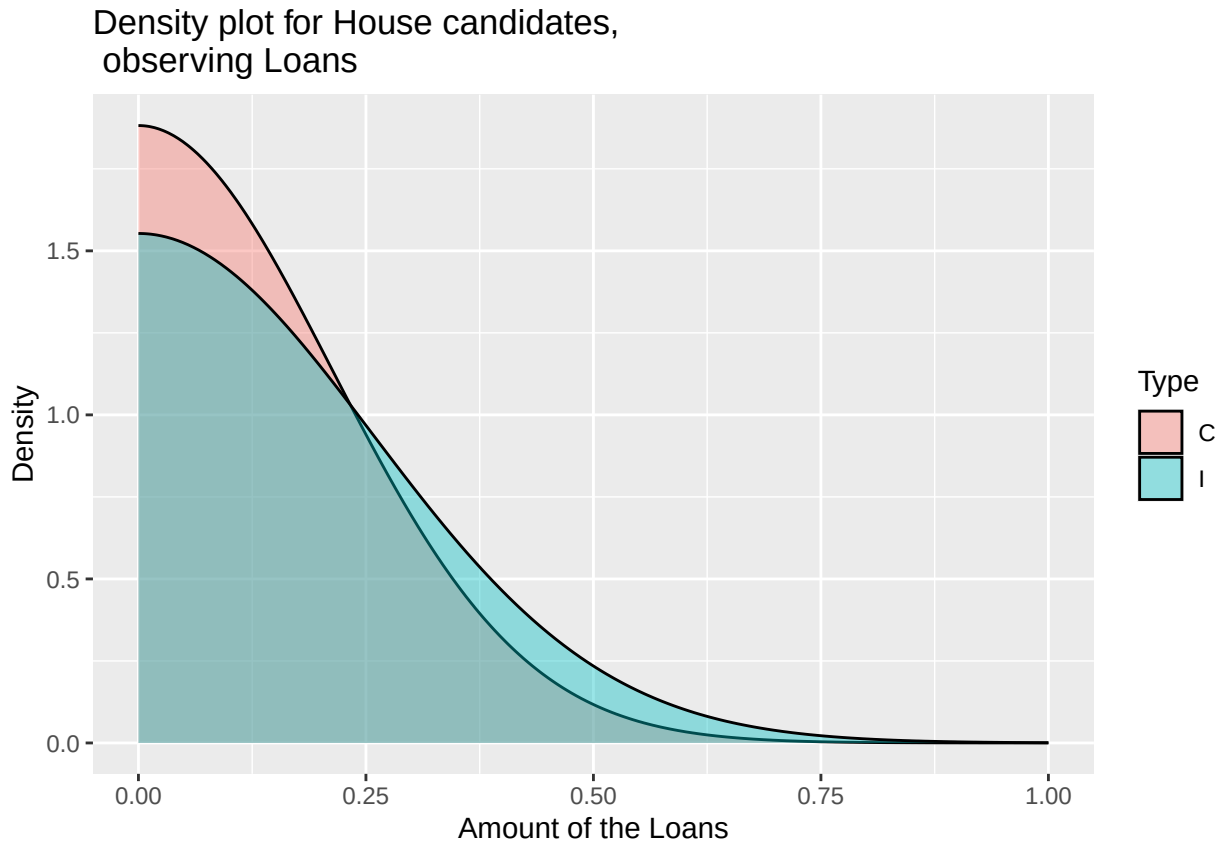
This graph exhibits the vast difference between incumbents and challengers in initial campaign cash. We can see that challengers in their totality remain at the lower part of the x_axis, meaning lower amounts of money. On the other hand, the distribution of incumbents is more uniform, having a variance in the relationship of initial capital. Still, it portrays that a large number of incumbents initiated the electoral campaign with large sums of money. This can be seen in the right part of the x-axis.

It's interesting that in the incumbent's group, extreme cases can be seen at the right part of the graph. Finally, it is essential to mention that in this case, having less than 0 dollars at the beginning of the campaign is impossible, so the density curve starts at 0, which is why the representation is of a half curve.

```
#Density plot for House candidates, incumbents observing the variable V10,
#which is the Loans from candidate.
hou_df2 <- hou_df %>%
filter(V3 == "I" |  V3 == "C")
p1 <- ggplot(hou_df2, aes(x=V13, y = ..density.., fill= V3)) +
  geom_density(alpha=0.4)+
```

```
  scale_x_log10()+
  labs(title = "Density plot for House candidates,\n observing Loans",
       x = "Amount of the Loans", y = "Density",
       fill = "Type")


p1 <- p1 + scale_x_continuous(limits = c(0, 1))
p1
```



This final graph demonstrates the amount of cash in loans of the candidates. We can see that challengers generally ask for more loans and that incumbents have loans bigger than challengers.