HW2

Noah Love

1/29/2021

HW2 - Simple Data Mining

This homework is to meant for you to do some basic data mining tasks using statistics from your prior classes.

Context - US's winner takes all voting system

Most of the voting in the US takes the form of "winner-takes-all" which encourages the formation of a two party system. We will see if we can identify the party affiliations simply by looking at the voting patterns.

We've collected data from the senate.gov

Q1 Data wrangling of the Senate Voting In `votes.json`, you will find the Senate voting records from the 105th to 115th Senate. The file is organized by senators, each with an id encoded by `S` followed by a digit. For each senator, their voting record for the last version of the bill is recorded, `1` stands for "Yea", `-1` stands for "Nay", `0` stands for "Not voting", and `-9999` is used if there were issues during data collection. The bills are labeled with the congress term (e.g. `106`), the session number (e.g. `1` or `2`), then the issue ID.

Please wrangle this dataset into a matrix called `voting_matrix` so we can calculate the correlation between the senators' voting behavior, i.e. we will run `cor(voting_matrix, ...)` in Q3 to identify how senators' vote with one another.

Please report the dimensions of your matrix and what percentage of the matrix does not contain `-1`, `1`, or `0`. For example, if I only have 2 senators and 2 bills where one out of the four terms is not `-1`, `1`, or `0`, then I would report `25%`. - I'm intentionally not telling you what are the columns and what are the rows, this should be implied.

Small scale work

I got really stuck on the conversion. So this is a small scale version of the following.

```
votes_small_json <- fromJSON(paste(readLines('votes_small.json')))
votes_small_df <- ldply(votes_small_json, data.frame)
#swap
votes_small_fx <- data.frame(t(votes_small_df[-1]))
colnames(votes_small_fx) <- votes_small_df[,1]

votes_small_df
```

```
##    .id X105_1_H.R..1003_00044 X105_1_H.R..1119_00296 X105_1_H.R..1122_00071
## 1 S238                      1                      1                      1
## 2 S213                      1                      1                     -1
## 3 S250                      1                      1                      1
```

```
##   X105_1_H.R..1469_00095 X105_1_H.R..1757_00105 X105_1_H.R..1871_00100
## 1                      1                      1                     -1
## 2                     -1                      1                     NA
## 3                      1                      1                     -1
##   X105_1_H.R..2014_00211 X105_1_H.R..2015_00209
## 1                      1                      1
## 2                     NA                     NA
## 3                     NA                     NA
```

```
json_to_matrix <- function(nested_json){
  json_og <-   fromJSON(paste(readLines(nested_json)))
  json_df <- ldply(json_og, data.frame)
  swapped_df <- data.frame(t(json_df[-1]))
  colnames(swapped_df) <- json_df[,1]
  json_matrix <- as.matrix(swapped_df)
  return(json_matrix)
}

#Test
json_to_matrix('votes_small.json')
```

```
##                        S238 S213 S250
## X105_1_H.R..1003_00044    1    1    1
## X105_1_H.R..1119_00296    1    1    1
## X105_1_H.R..1122_00071    1   -1    1
## X105_1_H.R..1469_00095    1   -1    1
## X105_1_H.R..1757_00105    1    1    1
## X105_1_H.R..1871_00100   -1   NA   -1
## X105_1_H.R..2014_00211    1   NA   NA
## X105_1_H.R..2015_00209    1   NA   NA
```

Complete Dataset

```
#View(votes)
length(votes)
```

```
## [1] 231
```

There is a list of 231 different voting records

```
head(names(votes),18)
```

```
##  [1] "S238" "S213" "S250" "S239" "S127" "S231" "S010" "S167" "S200" "S223"
## [11] "S179" "S249" "S206" "S014" "S211" "S017" "S225" "S116"
```

This function shows all of the senators (based on their codes). We will use this to match up senators.

```
votes_matrix <- json_to_matrix('votes.json')
voters_matrix <- json_to_matrix('voters.json')
```

```
senate <- rbind(voters_matrix, votes_matrix)
```

RBind

```
dim(senate)
```

Dimensions

```
## [1] 807 231
```

```
#I used the votes instead of combined (senate) because there is no point in including names and parties
sum(is.na(votes_matrix))/prod(dim(votes_matrix))
```

```
## [1] 0.5672667
```

Q2 Expectations Please articulate your expectations for the correlation matrix given the 2 party system. - What's its dimension? - How many modes do you expect to see in the distribution of correlation values? - Where will the correlation values center?

I would expect strong correlation among members of the same party, meaning that about 50% of the senate will be correlated together. There are also a few members of both parties that will likely be on the extremes of that party and not that correlated. Therefore I am expected a two hump correlation plot, with the density at 0 probably being about the same as the two extremes (so 2 modes). I would expect the correlation values to be really close to 0. This should depend on time frame I would think and who controlled congress. Beings as it appears Republicans had it longer, maybe it is a little shifted to republicans.

```
ncol(votes_matrix)
```

```
## [1] 231
```

Then the correlation matrix should be 231 by 231.

Q3 Mining with correlations Please calculate the correlation value between the senators' voting pattern, please set `use=pairwise.complete.obs` in the function `cor()`.

```
cor_matrix <- cor(votes_matrix, use="pairwise.complete.obs")
```

```
## Warning in cor(votes_matrix, use = "pairwise.complete.obs"): the standard
## deviation is zero
```

```
#View(cor_matrix)
dim(cor_matrix)
```

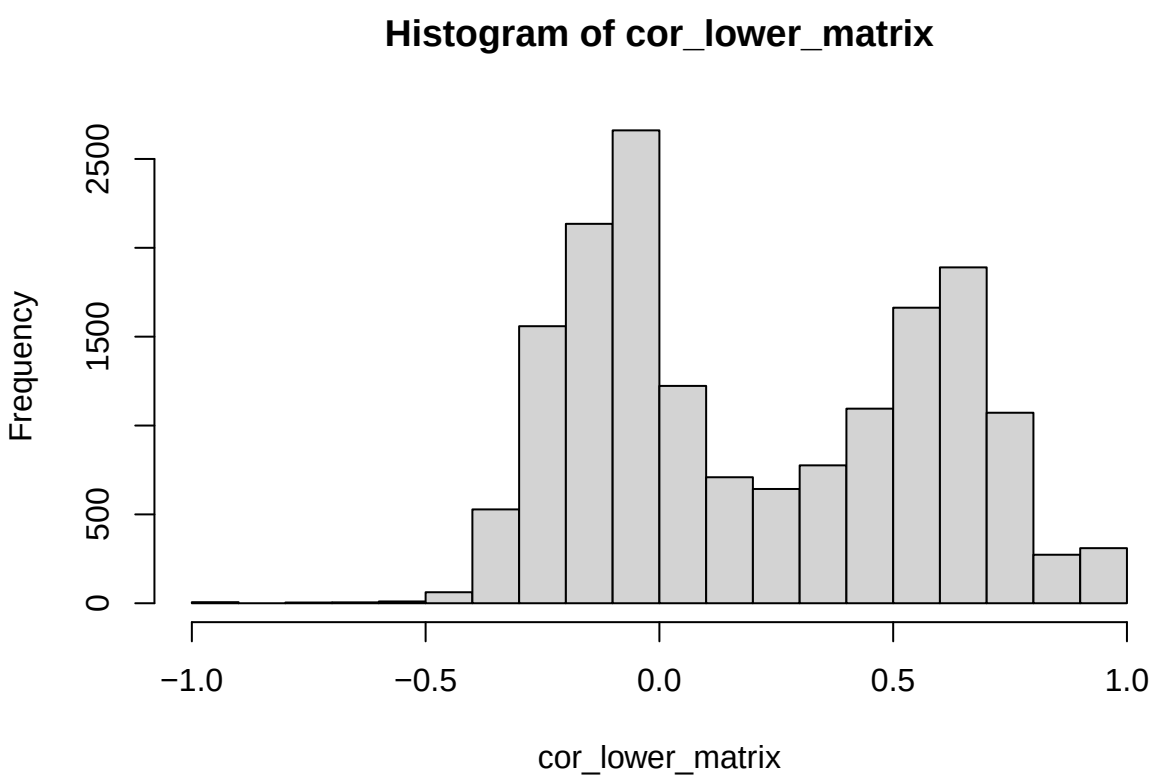
```
## [1] 231 231
```

** - Please explain why R will warn you that some correlations cannot be computed. **

Not all correlations can be computed because many senators were not a senator during the time when the bills were voted on, meaning that there is a NA value rather than a 0.

** - Please plot the histogram of the lower triangular values within the correlation matrix. Hint: `lower.tri` **

```
cor_lower_matrix <- cor_matrix

cor_lower_matrix[upper.tri(cor_lower_matrix)] <- NA

#View(cor_lower_matrix)

hist(cor_lower_matrix)
```



** - Please comment on this histogram relative to your expectations from Q2. Side comment: make sure this makes sense to you! **

Similar to what I expected, although more shifted. I expected to see a small shift from more republicans. Possibly a little bit more towards zero than I expected as well for the two modes of the distribution. I expected party politics to be more entrenched. Also it is interesting that there is more positive skew. It would make sense though as its unlikely anyone disagreed on everything, as there are bills that should pass unanimously (like medals or certain funding bills).

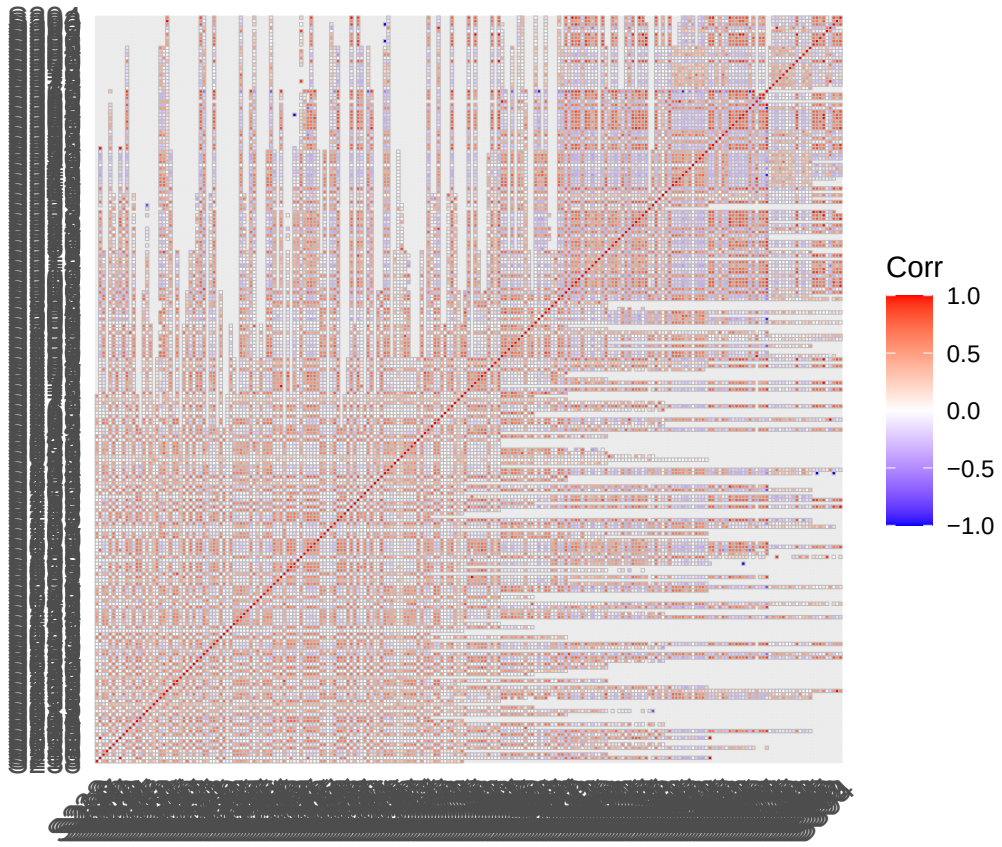
- Please re-order the correlataion matrix such that it starts with the highest positive correlation (1) to the strongest negative correlation (-1) with respect to Senator McConnell's votes, i.e. the previous Senate majority leader who has served since 1985 (id=`S174`, member of the Republican Party). Note that the majority leader is the party's spokesperson for the party's position on issues.
  - It would make sense to filter out senators who do not overlap Senator McConnell.

```
#Remove those that do not overlap (Cor = NA)

mitch_df <- as_tibble(cor_matrix, rownames = NA)
#View(mitch_df)

mitch_df <- mitch_df %>%
  rownames_to_column(var = "rowname") %>%
  filter(is.na(S174) ==FALSE) %>%
  arrange(desc(S174)) %>%
  relocate(S174) %>%
  column_to_rownames(var = "rowname") %>%
  as.matrix()
```

```
#View(mitch_df)
```

- Please try to visualize this re-ordered correlation matrix, hint: `image()`

```
#image(mitch_df)

#alternative
ggcorrplot(cor_matrix)
```



- Please list the senators, if any, who have a negative correlation with Senator McConnell's votes but an **average** greater than 0.2 correlation with the "possible Republicans".

```
mitch_df_2 <- as_tibble(mitch_df, rownames = NA)

possible_republicans <- mitch_df_2 %>%
  rownames_to_column() %>%
  filter(S174 >= .2) %>%
  column_to_rownames()

#now all the columns are still there, but only the rows are "possible republicans".

#Then we can compute the average correlation with possible republicans by just averaging each column.

#View(possible_republicans)

correlation_to_GOP <- possible_republicans %>%
  summarise_each(funs(mean(.,na.rm = TRUE)))
```

```
mitch_df_3 <- as_tibble(mitch_df, rownames = NA)

negative_with_mitch <- mitch_df_3 %>%
  rownames_to_column() %>%
  filter(S174 <0) %>%
  column_to_rownames() %>%
  select(S174) %>%
  as.data.frame() %>%
  tibble::rownames_to_column() %>%
  pivot_longer(-rowname) %>%
  pivot_wider(names_from=rowname, values_from=value) %>%
  select(-name)

voters_df <- as.data.frame(voters_matrix)

complete_tibble <- as_tibble(rbind.fill(voters_df, correlation_to_GOP, negative_with_mitch))

complete_df <- as.data.frame(complete_tibble)
rownames(complete_df)[1] <- "First Name"
rownames(complete_df)[2] <- "Last Name"
rownames(complete_df)[3] <- "Party Affiliation"
rownames(complete_df)[4] <- "State"
rownames(complete_df)[5] <- "Corr. To GOP"
rownames(complete_df)[6] <- "Negative cor. with Mitch"

finished_tibble <- complete_df[ , colSums(is.na(complete_tibble)) == 0]
finished_tibble <- finished_tibble %>%
  tibble::rownames_to_column() %>%
  pivot_longer(-rowname) %>%
  pivot_wider(names_from=rowname, values_from=value) %>%
  select(-name)

finished_tibble$`Corr. To GOP` <- as.numeric(finished_tibble$`Corr. To GOP`)

finished_tibble <- finished_tibble %>%
  filter(`Corr. To GOP`> .2)

colnames(finished_tibble)
```

```
## [1] "First Name"               "Last Name"
## [3] "Party Affiliation"        "State"
## [5] "Corr. To GOP"             "Negative cor. with Mitch"
```

```
finished_tibble
```

```
## # A tibble: 4 x 6
##   `First Name` `Last Name` `Party Affiliat~ State `Corr. To GOP`
##   <chr>        <chr>       <chr>            <chr>          <dbl>
## 1 Orrin        Hatch       R                UT             0.309
## 2 Richard      Burr        R                NC             0.215
## 3 Tom          Cotton      R                AR             0.313
## 4 Ben          Sasse       R                NE             0.290
## # ... with 1 more variable: `Negative cor. with Mitch` <chr>
```

- We will define a 'possible Republican' as someone with a correlation of 0.2 or higher voting record with

Senator McConnell.
- You will need the file `voters.json` to get the names of these senators.

Why might this group be interesting if they exist? This will not be graded by the grader but your projects will require you to make this kind of articulation. This should be done in 5 or fewer sentences.

These are the "counterintutive" people. They provide a counterexample to what we consider to be the two-party system. These individuals are negatively correlated with their party leader, and yet are correlated with the party meaning that their voting patterns are likely on the edge of the party, far enough away from the party leader.

Q4 Machine learning intuition with regression We will compare 2 feature selection methods for regression, with the goal of predicting a senator's voting pattern. In both methods, for any target senator, we will select 2 other senators to predict their vote on any issue. - Method 1: - Pick the senators with the strongest positive and strongest negative correlation with the chosen senator. - Run OLS against both senators simultaneously.

```
# #ols <- lm(S174 ~ TWO OTHER SENATORS, votes_df)
#
# #lm(S### ~ (sort(top 1), sort(bottom 1), df)
#
# #Create a function for this
#
# cor_matrix_senator <- as_tibble(cor_matrix, rownames = NA)
#
#
# method1 <- function(senID){
#   #find the top senator
#
#   cor_matrix_senator_up <- cor_matrix_senator %>%
#     select(senID) %>%
#     slice_max(n =1 )
#
#   #find the bottom senator
#   cor_matrix_senator_down <- cor_matrix_senator %>%
#     select(senID) %>%
#     slice_min(n =1 )
#
#   #run ols
#   ols <- lm(Y ~ cor_matrix_senator[])
# }
#
# senator_id <- "S000"
#
# test <- function(senator_id){
#   cor_matrix_senator_down <- cor_matrix_senator %>%
#   select(senator_id)
#   return()
# }
```

- Method 2:

```
#  - To identify the first senator, simply find the one senator with the strongest correlation.

# method2 <- function(senator_id){
```

```
#   #highest cor
#
#   cor_matrix_senator_up <- cor_matrix_senator %>%
#   select(senID) %>%
#   slice_max(n =1 )
#
#   #run ols
#   ols <- lm(cor_matrix_senator_up, senator_id)
#   resids <- ols$residuals

  #correlation with the above residual

#}




 # - Identify the second senator by identifying the senator whose voting record has the strongest corre



 # - Re-run OLS against both senator simultaneously.
```

Please answer the following: **- Why isn't choosing the 2 senators with the 2 strongest positive correlations a reasonable strategy? This should be 3 or fewer sentences.**

Just by seeing some people are more likely to vote with them than others doesn't mean that can help us predict the future vote. If this senator was a swing voter and has low correlations throughout, those would not help us.

What does `lm()` do when one senator has no voting record for a particular bill? Is the behavior different if the senator is the independent or dependent variable?

For each senator, please randomly choose 20% of their bills data into a test set, then use the remaining 80% of bills to run both methods above. Please report which method has a better prediction performance according to what you see? - Do not impute the missing values - You should choose a prediction metric - This is not a full cross validation because we are not rotating the test set. - Please visualize the prediction performance across senators for both methods.