HW2

Nikhil Gopal

2/4/2021

Question 1

```
setwd("C:/Users/d/Google Drive/Notability/Data Mining/psets/HW2")

#Question 1


library(jsonlite)
```

```
## Warning: package 'jsonlite' was built under R version 4.0.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.3

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
votes_json <- fromJSON("votes.json")

#convert to data frame
votes <- data.frame(bind_rows(votes_json, .id = "senator"), check.names = F)

#make each row a new senator
votes_id_num <- data.frame(bind_rows(votes_json), check.names = F)

#create a matrix using senators id num
voting_matrix <- data.matrix(votes_id_num, rownames.force = NA)

#name the rows in the matrix
rownames(voting_matrix) <- votes$senator

#flip the matrix so that senators become columns
voting_matrix <- t(voting_matrix)


#dimensions of matrix
dim(voting_matrix)
```

```
## [1] 803 231
```

```
#what percentage of matrix doesn't contain -1, or 0

counter <- 0
did_vote <- 0


for(cell in voting_matrix){
    if(is.na(cell) == TRUE){
    }else if(cell == "-9999"){
    }
  else{
      did_vote = did_vote + 1
    }
  counter = counter + 1
  }

did_vote/counter
```

```
## [1] 0.4326255
```

The dimensions of the matrix were 803x231 and approximately 43.26255$ did not contain -1,-1 or 0.

Question 2

The correlation matrix should be 231x231, since there are 231 senators in this dataset, and they will all have some (or no) correlation with each other.

There will likely only be 1 mode. Every senator will have a perfect correlation with themself, value of 1.000. I do not predict that any value will appear more frequently than 1.0 perfect correlations of the senators with themselves.

Given the 2 party system, the values will center around 0. Democrats and Republicans have nearly equal representation, and vote inversely. Thus, the correlations should center around 0.

Question 3

Some values will not be able to be calculated since some senators were not serving at the same time. Thus they will not have any overlapping votes and there will be values of NA in the dataset.

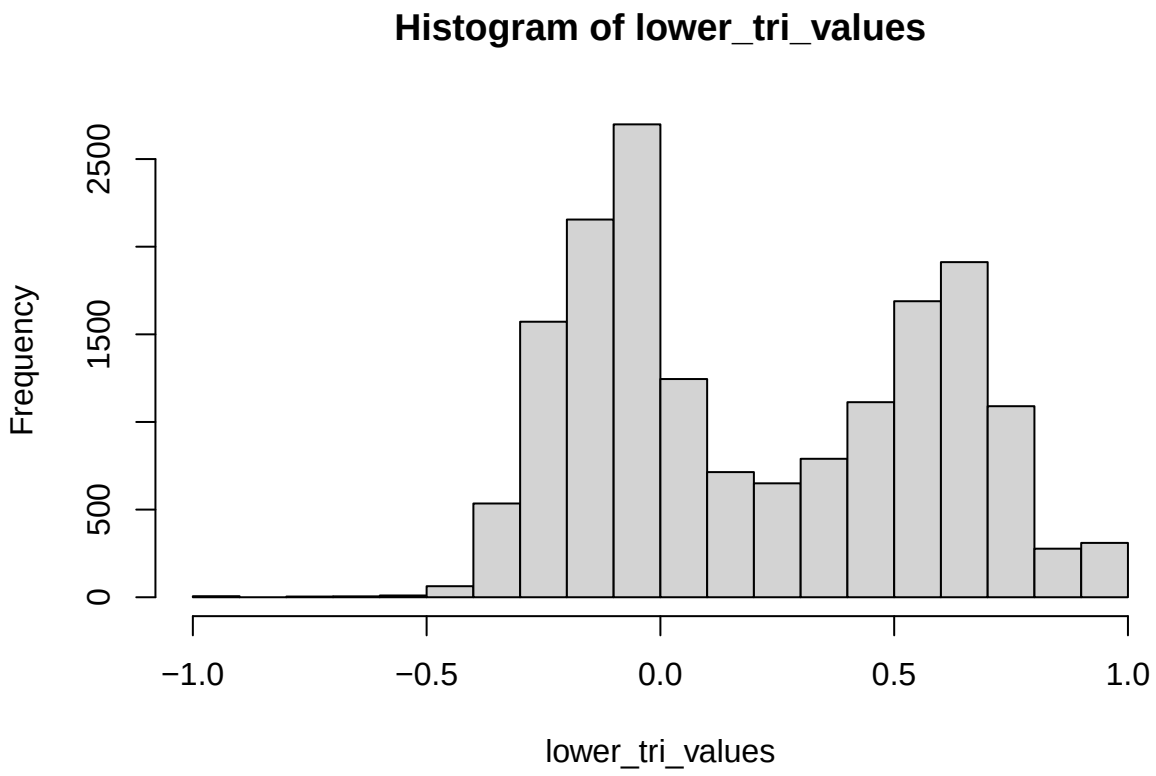
Correlation calcuation:

```
#Q3

#create the correlation matrix
correlation_matrix<- data.matrix(cor(voting_matrix, y=NULL, use = "pairwise.complete.obs"))
```

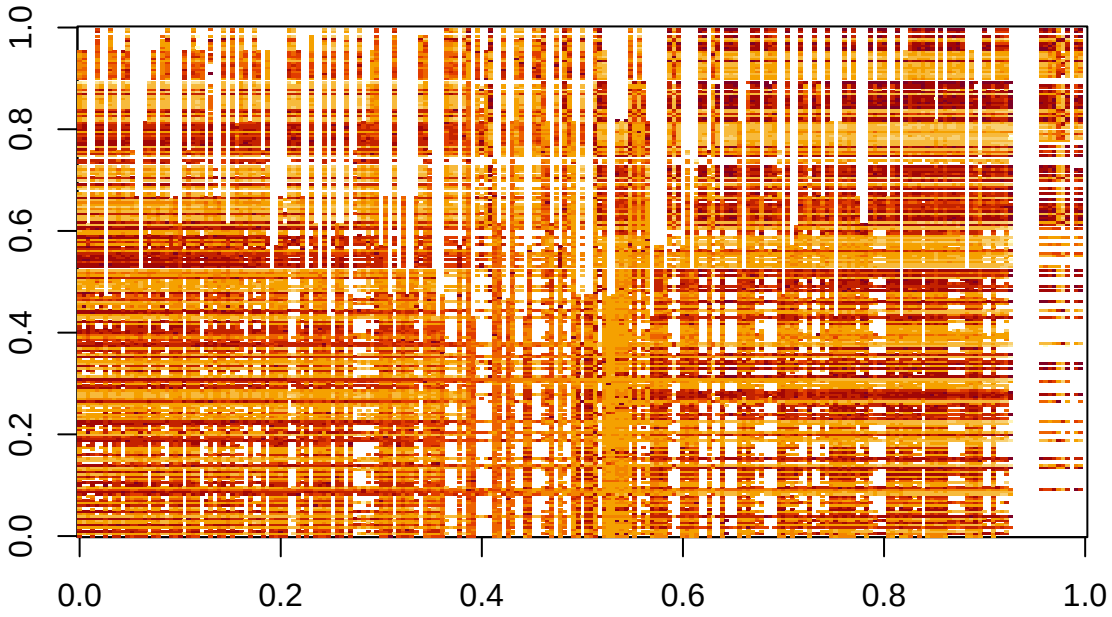
```
## Warning in cor(voting_matrix, y = NULL, use = "pairwise.complete.obs"): the
## standard deviation is zero
```

Lower tri histogram, reordered matrix/ visualization:

```
#lower tri
lower_tri <- data.matrix(lower.tri(correlation_matrix, diag = TRUE))

#iterate thru the correlation matrix and save the lower tri values to a vector
counter <- 0
lower_tri_values = c()
for(cell in lower_tri){
  if(cell==TRUE){
    lower_tri_values <- c(lower_tri_values, correlation_matrix[counter])
  }
  counter = counter + 1
}

#histogram of the lower tri values
hist(lower_tri_values)


#histogram of the lower tri values
hist(lower_tri_values)
```




```
#Mitch = row #72, idno: S174
mcconnell_matrix = correlation_matrix[order(correlation_matrix[,"S174"],decreasing=TRUE),]

#visualization of McConnel Matrix
image(mcconnell_matrix)
```



Correlations with possible republicans:

```
#find the senators with negative correlation with Mitch
library(schoolmath)
```

```
## Warning: package 'schoolmath' was built under R version 4.0.3
```

```
negative_cor_with_mitch = c()

index <- 0
for (senator in mcconnell_matrix[,"S174"]) {
  if(is.negative(senator) == TRUE){
    #print(paste(senator, rownames(mcconnell_matrix)[index]))
    negative_cor_with_mitch <- c(negative_cor_with_mitch, rownames(mcconnell_matrix)[index])
  }
  index = index + 1
}
```

```
## Error in if (y < 0) {: missing value where TRUE/FALSE needed
```

```
negative_cor_with_mitch
```

```
##   [1] "S375" "S245" "S113" "S374" "S207" "S208" "S118" "S363" "S198" "S205"
##  [11] "S152" "S167" "S209" "S213" "S300" "S258" "S348" "S295" "S302" "S035"
```

```
##  [21] "S047" "S185" "S251" "S327" "S382" "S360" "S199" "S264" "S263" "S017"
##  [31] "S362" "S354" "S269" "S383" "S370" "S010" "S353" "S359" "S364" "S210"
##  [41] "S276" "S127" "S282" "S330" "S298" "S361" "S369" "S356" "S222" "S206"
##  [51] "S277" "S267" "S279" "S314" "S320" "S055" "S324" "S319" "S278" "S309"
##  [61] "S051" "S226" "S201" "S229" "S257" "S221" "S297" "S280" "S106" "S311"
##  [71] "S325" "S341" "S337" "S150" "S176" "S284" "S014" "S182" "S247" "S275"
##  [81] "S270" "S366" "S259" "S057" "S173" "S253" "S131" "S307" "S308" "S316"
##  [91] "S306" "S217" "S223" "S166" "S230" "S326" "S332" "S172" "S331" "S322"
## [101] "S313"
```

```
#find the possible republicans
possible_republicans = c()

index <- 1
for(senator in mcconnell_matrix[,"S174"]){
  if(senator > 0.2 & index != 1){ #index != to exclude Mitch McConnel
    #print(paste(senator, rownames(mcconnell_matrix)[index], index))
    possible_republicans <- c(possible_republicans, rownames(mcconnell_matrix)[index])
  }
  index = index + 1
}
```

```
## Error in if (senator > 0.2 & index != 1) {: missing value where TRUE/FALSE needed
```

```
#find the desired senators senators
index_neg_cor = 1

desired_senators <- c()

for(senator in negative_cor_with_mitch){
  index_poss_rep = 0
  correlation_values = c()
  for(sen in possible_republicans){
    #print(mcconnell_matrix[negative_cor_with_mitch[index_neg_cor],possible_republicans[index_poss_rep]
    correlation_values <- c(correlation_values, mcconnell_matrix[negative_cor_with_mitch[index_neg_cor]
    index_poss_rep = index_poss_rep + 1
  }
  #print(paste(mean(correlation_values, na.rm = TRUE), negative_cor_with_mitch[index_neg_cor]))

  if(mean(correlation_values, na.rm = TRUE) > 0.2){
    desired_senators <- c(desired_senators, negative_cor_with_mitch[index_neg_cor])
  }

  index_neg_cor = index_neg_cor + 1
}

desired_senators
```

```
## [1] "S375" "S374" "S118" "S300" "S382" "S383"
```

```
#import senator names
voters.json <- fromJSON("voters.json")
```

```
for(senator in desired_senators){
  print(voters.json[senator])
}
```

```
## $S375
## $S375$first_name
## [1] "Steve"
##
## $S375$last_name
## [1] "Daines"
##
## $S375$party
## [1] "R"
##
## $S375$state
## [1] "MT"
##
##
## $S374
## $S374$first_name
## [1] "Tom"
##
## $S374$last_name
## [1] "Cotton"
##
## $S374$party
## [1] "R"
##
## $S374$state
## [1] "AR"
##
##
## $S118
## $S118$first_name
## [1] "Orrin"
##
## $S118$last_name
## [1] "Hatch"
##
## $S118$party
## [1] "R"
##
## $S118$state
## [1] "UT"
##
##
## $S300
## $S300$first_name
## [1] "Richard"
##
## $S300$last_name
## [1] "Burr"
##
```

```
## $S300$party
## [1] "R"
##
## $S300$state
## [1] "NC"
##
##
## $S382
## $S382$first_name
## [1] "Ben"
##
## $S382$last_name
## [1] "Sasse"
##
## $S382$party
## [1] "R"
##
## $S382$state
## [1] "NE"
##
##
## $S383
## $S383$first_name
## [1] "Dan"
##
## $S383$last_name
## [1] "Sullivan"
##
## $S383$party
## [1] "R"
##
## $S383$state
## [1] "AK"
```

My code outputted the following names: Steve Daines, Tom Cotton, Orrin Hatch, Richard Burr, Ben Sasse, Dan Sullivan. Senator #375 (Steve Daines) actually has a slightly positive correlation with Mitch McConnel (0.005056894). S375 is being included in my list of negative correlation with McConnell, despite the conditional statement in my for loop telling R to exlude negative values. I have no idea why it is doing this and have tried many times to fix the bug, but I can't seem to figure it out. Daines was not in my list of possible republicans, so I know the error is attributable to this section of the code, but I was not able to fix that specific bug.

This group might be interesting because they vote against the leader of the party, but also have some tendency to vote with other members of the party. This could suggest that the party is divided between its leadership and some of it's members, and means that the 2 party system is not as enforced as previously thought.

Q4

Method 1:

```
OLS_list <- c()

for(index in 1:231){
  vector <- correlation_matrix[index,]
```

```
  sorted_vector <- order(vector, decreasing = TRUE)

  highest_correlation_with_current_senator = rownames(correlation_matrix)[sorted_vector[2]]
  current_senator <- rownames(correlation_matrix)[index]


  vector2 <- vector[!is.na(vector)]
  sorted2 <- order(vector2)


  most_neg_correlation <- rownames(correlation_matrix)[sorted2[length(sorted2)]]

  #print(paste(highest_correlation_with_current_senator, most_neg_correlation))

  OLS <- lm(voting_matrix[,current_senator]~
             voting_matrix[,highest_correlation_with_current_senator] + voting_matrix[,most_neg_correla

  OLS_list[index] <- OLS
}
```

```
## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
```

```
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
```

```
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
```

```
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
```

```
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
```

```
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
```

```
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Warning in OLS_list[index] <- OLS: number of items to replace is not a multiple
## of replacement length

## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...): 0 (non-NA) cases
```

Method 2 (did not complete):

```
#method 2

residuals_list <- c()

for(index in 1:231){
  vector <- correlation_matrix[index,]
  sorted_vector <- order(vector, decreasing = TRUE)
  highest_correlation_with_current_senator = rownames(correlation_matrix)[sorted_vector[2]]
  current_senator <- rownames(correlation_matrix)[index]


  OLS <- lm(voting_matrix[,current_senator]~
              voting_matrix[,highest_correlation_with_current_senator],
  )
  resid <- residuals(OLS)

}
```

```
## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...): 0 (non-NA) cases
```

Picking the senators with the highest positive correlations is a bad idea because it would violate the collinearity assumption of linear regression. To make unbiased models, you can't regress onto predictors that you know are correlated to each other, this would defeat the purpose of trying to discover an association between two variables since you know that the variables are associated already.

When a senator has no voting record for a bill, R will just ignore values of NA for the regression. It works for both dependent and independent variables.