

NYU APSTA GE 2014 Final Project

Claire Willeck

12/16/2019

Overview

In this project, I am interested in the network of campaign donors within a state for House elections. First, I look at descriptive statistics and visuals of the networks. Then, I simulate different networks based on different scenarios of the number of Democratic and Republican candidates running in an election and scenarios of the probability of a donor donating to either a Republican or Democratic candidate. The network data that I have is a one-mode network (described more below) with donor groups as the nodes. The original network was a bipartite network with candidates and donor groups. I don't have the bipartite network, but I do have data on the candidates that ran for the House elections in each district, for each state, for each election. I also have data on the percentage of donations given to Republican and Democratic candidates by each donor group. I use these percentages as the probability of a donor group to donate to a Republican or a Democratic candidate. For example, if a donor group donated 100% of their donations to a Democratic candidate then this donor group has a 100% probability of donating to Democratic candidates and 0% probability of donating to Republican candidates. This is clearly a rough approximation of the probability of a donor group to donate to a candidate. Donor groups likely do not donate to a candidate simply based on party, but also based on other strategy measures such as probability of winning, ideology, historical behavior, etc. Further, a donor group in reality can donate 100% of its donations to Democratic candidates, but that does not mean that the group donates evenly to all candidates or to some Democratic candidates at all; whereas, a donor group that donates 100% to Democratic candidates will donate to all Democratic candidates available in my simulations. In my first set of simulations, I vary the makeup of the candidates, and in my second set of simulations, I vary the probability of donor groups to donate to candidates.

First Set

The first set of simulations includes five scenarios. In the first scenario, I use the realized number of Republican and Democratic candidates running in each state for each election. Each donor group has a probability of donating to each candidate based on the percentage of donations given to Democratic and Republican candidates, and this probability remains fixed across all scenarios. The donor group then has a tie with a candidate if the donor group donates to the candidate and does not have a tie with a candidate if the donor group does not donate to that candidate. This produces an incidence matrix for a bipartite network with candidates and donor groups as the two modes. I then project this matrix to get a one-mode network with donor groups as the nodes, similar to the original network, to calculate measures of centrality. I simulate this scenario 100 times and retrieve a distribution of centrality measures. From this distribution, I take the mean centrality measures for the means of easy comparison with the other scenarios.

For the four remaining scenarios, I use the same methodology, except the number of candidates available to donate to changes. In the second scenario, there is one additional Republican candidate and the same number of Democratic candidates as there are in reality. In the third scenario, there is one additional Democratic candidate and the same number of Republican candidates as there are in reality. In the fourth scenario, there is one fewer Republican candidate and the same number of Democratic candidates as there are in reality. In the fifth scenario, there is one fewer Democratic candidate and the same number of Republican candidates as there are in reality. The mean centrality measures calculated in these scenarios will indicate whether the centrality of the donor network changes as the makeup of the candidates change. They will explore whether the donor group network centrality changes as an additional Republican or Democratic candidate either runs or drops out of the race.

Second Set

In the second set of scenarios, I keep the number of Republican and Democratic candidates running in each election fixed. I vary the probability of donor groups donating to Democratic and Republican candidates. In the first scenario, I change the probability of any donor group that donated 100% of donations to Republican candidates to 90% probability to Republican candidates and 10% to Democratic candidates. In the second scenario, I change the probability of any donor group that donated 100% of donations to Democratic candidates to 90% probability to Democratic candidates and 10% to Republican candidates. In the third scenario, I change any donor group that donated to both Republican and Democratic candidates to donate 100% to Republican candidates. In the fourth scenario, I change any donor group that donated to both Republican and Democratic candidates to donate 100% to Democratic candidates. As before, each scenario will be simulated 100 times and centrality measures will be calculated for each simulation. For ease of comparison, the mean centrality measure will be shown for each scenario.

The first two scenarios lessen the polarization in donor group donations to candidates. The latter two scenarios increase the polarization in the donor group donations to candidates. These scenarios explore how the centrality of the donor group network changes when donor groups become more or less polarized in their donations.

Data

For this project, I use two data sources. The first dataset that includes the network information is a dataset provided by Reuning (2019). The original data was in the form of a bipartite network where candidates and donating groups were the modes-CxG matrix. Reuning used a methodology proposed by Neal (2014). The backboning of a bipartite network creates edge weights to project the bipartite network CxG matrix to a GxG matrix network where the nodes are groups. The method involves finding a null distribution of edge weights by finding the propensity of a group to donate to a candidate, the propensity of a candidate to receive a donation, and the interaction of those two. Reuning had done this process and the resulting dataset includes the edgelist for the groups. Reuning used a 97.5% threshold for the observed edgeweights values in terms of the empirical null distribution of edgeweights. Reuning includes edges with a 90% threshold, but in the following analyses, I also exclude below the threshold of 97.5% to preserve computational capabilities. The data includes both House and Senate elections from 2000 to 2016. I only use data on House elections for this project. Further, for the sake of computational capacity, I only look at House elections for 2016 for each state. This dataset includes the percentage of a donor's donations given to a Democratic candidate and a Republican candidate. I will use these percentages as probabilities in the scenarios.

The second data set used in this project comes from the MIT Election Data and Science Lab. The dataset includes all of the candidates running for house districts from 1976 to 2018. Given the data availability of the donations, I only use data for house elections for 2016. To count the total number of Democratic and Republican candidates running in each state, I first have to classify candidates as either a Republican, Democrat or Other. Some states, such as Minnesota which has the Democratic Farmer Labor Party, have different names for the Democrat or Republican party. To account for this, I label any candidate running for a party that includes "republican" as Republican and any candidate running for a party that includes "democrat" as Democrat. Since I only have data on the percentage of donations given to a Democratic or Republican candidate, I exclude candidates running in the election that are not associated with either of those parties.

Importing Data and Cleaning Process

First, I load in the two datasets that will be required for the analyses.

```
setwd("/Users/clairewilleck/Documents/PhD Courses/NYU APSTA GE 2014 /Final Project/")
set.seed("573")
read_plus <- function(flnm) {
```

```

read_csv(flnm) %>%
  mutate(filename = flnm)
}

read_plusdta <- function(flnm) {
  read_dta(flnm) %>%
    mutate(filename = flnm)
}

##Load network data
md <-
  list.files(path = "./Election_Networks/Metadata/",
             pattern = "*.csv",
             full.names = T) %>%
  map_df(~read_plus(.))

el <-
  list.files(path = "./Election_Networks/Edge_Lists/",
             pattern = "*.csv",
             full.names = T) %>%
  map_df(~read_plus(.))

##load vote data
vote <- read_csv("./Election_Networks/1976-2018-house.csv")

```

Below I clean the edgelist data.

```

##Remove edges with a values less than 3 based on the threshold calculations done by the creator of the
el <- el[el$edge>=3,]

```

Below I clean the metadata for the network.

```

##Create year and state variables in metadata dataframe
md_house <- md[str_detect(md$filename, "House", negate=FALSE)==TRUE,]
md_house$year <- gsub('.{10}$', '', md_house$filename)
md_house$year <- substr(md_house$year, 39, 42)
md_house <- md_house[md_house$year=="2016",]
md_house$state <- substr(md_house$filename, 31, 32)
md_house$cat <- paste0(md_house$state, md_house$year)
x <- sub(".*//", "", unique(md_house$filename))

```

Now, I clean the voting data.

```

##Keep only observations from 2000 to 2016 elections.
vote_sub <- vote[vote$year>=2000,]

##Remove candidate NA rows-these are write-ins
vote_sub <- vote_sub[complete.cases(vote_sub$candidate),]

##Create a variable for merging
vote_sub$cat <- paste0(vote_sub$state_po, vote_sub$year)

uniquep <- length(unique(vote_sub$party))

```

The 213 unique parties that candidates ran under.

Clearly, there's a lot of parties that people are running under and it's difficult to ascertain whether these

parties fall on the ideological unidimensional liberal-conservative scale. For simplicity, I will code any party that includes “democrat” as Democrat and any party that includes “republican” as Republican and all other parties as Other. This simplification can also be justified by the fact that out of the candidates that won the election, only 4 were not Republican or Democrat and also that the US system is typically viewed as a two party system.

Below I recode the party to Democrat, Republican, and Other:

```
##Re-label candidate party. If party includes "republican"-label as republican. If party includes "democrat"
vote_sub$party <- tolower(vote_sub$party)
vote_sub$party_adj <- ifelse(str_detect(vote_sub$party, "democrat", negate=FALSE)==TRUE, "democrat", ifelse(str_detect(vote_sub$party, "republican", negate=FALSE)==TRUE, "republican", "other"))

##Find the number of candidates per party per district
vote_sub$cat <- paste0(vote_sub$state_po, vote_sub$year)
vote_sub <- vote_sub %>%
  group_by(cat) %>%
  mutate(num_rep=length(which(party_adj=="republican"))) %>%
  mutate(num_dem=length(which(party_adj=="democrat"))) %>%
  mutate(num_other=length(which(party_adj=="other")))

vote_cand <- vote_sub %>%
  group_by(cat) %>%
  distinct(num_rep, .keep_all=TRUE)
```

Below is the code to see how many districts had a winner that was not a Republican or Democrat

```
vote_sub$candidatevotes <- as.numeric(vote_sub$candidatevotes)
vote_win <- vote_sub %>%
  group_by(cat,district) %>%
  filter(candidatevotes==max(candidatevotes))

##See how many elections had a non republican or democrat winner
other <- vote_win[vote_win$party_adj=="other",]
other[complete.cases(other$year),c("year","state","party")]
```

```
# A tibble: 4 x 3
  year state   party
  <dbl> <chr>   <chr>
1 2000 Vermont independent
2 2000 Virginia independent
3 2002 Vermont independent
4 2004 Vermont independent
```

The number of Democrat, Republican, and Other candidates running in each election in each state will serve as a baseline. I will now re-create the bipartite network using different numbers of candidates that deviate from the observed numbers.

For this analysis, I will have four scenarios. One with an additional Republican candidate-all other candidates equal, one with an additional Democratic candidate-all other candidates equal, one with one less Republican candidate-all other candidates equal, and one with one less Democratic candidate-all other candidates equal.

For this analysis, I am ignoring the amount of money available to each donor to give. Instead, I am going to use the percentage of money donated to a Democratic candidate as the probability of donating to a Democratic candidate and the percentage of money donated to a Republican candidate as the probability of donating to a Republican candidate. The number of donors in each state and election will remain constant.

Descriptive Statistics and Visualization

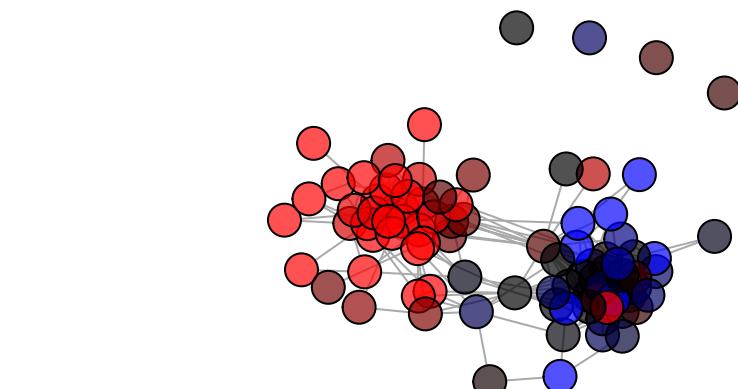
```
y <- x[str_detect(x,"House",negate=FALSE)==TRUE]
p <- length(y)
seq <- seq(1,by=1, len=p)
density <- list()
diameter <- list()
mean_dist <- list()
cd <- list()
cb <- list()
ce <- list()

##subset data
el_subdata <- lapply(seq, function(k) el[endsWith(el$filename,y[k]),] )
md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
##create network graph
g <- lapply(1:length(el_subdata), function(x) graph.data.frame(el_subdata[[x]],vertices=md_subdata[[x]])

for(i in 1:length(el_subdata)){
V(g[[i]])$color <- md_subdata[[i]]$DemCol
V(g[[i]])$attr <- md_subdata[[i]]$PerDem
}

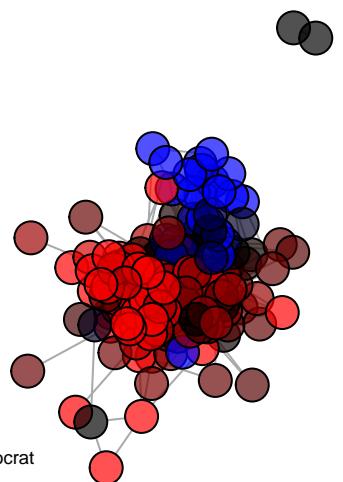
##calculate centrality measures
density <- lapply(1:length(el_subdata), function(x) edge_density(g[[x]], loops = FALSE))
diameter <- lapply(1:length(el_subdata), function(x) diameter(g[[x]]))
mean_dist <- lapply(1:length(el_subdata), function(x) mean_distance(g[[x]]))
cd <- lapply(1:length(el_subdata), function(x) centr_degree(g[[x]])$centralization)
cb <- lapply(1:length(el_subdata), function(x) centr_betw(g[[x]])$centralization)
ce <- lapply(1:length(el_subdata), function(x) centr_eigen(g[[x]])$centralization)
##create plot
colors = c("#0000FFAF", "#140000AF", "#FF0000AF")
labels = c("100% Democrat", "50% Republican; 50% Democrat", "100% Repbulican")
makeplots <- function(x){
  plot(g[[x]], vertex.label=NA,main=gsub('.{4}$', '', y[x]))
  legend("bottomleft", legend=labels, col=colors, pch=19, horiz=FALSE, cex=.6, yjust=0, x.intersp = .3, bty="r")
}
```

AK-2015-2016-House



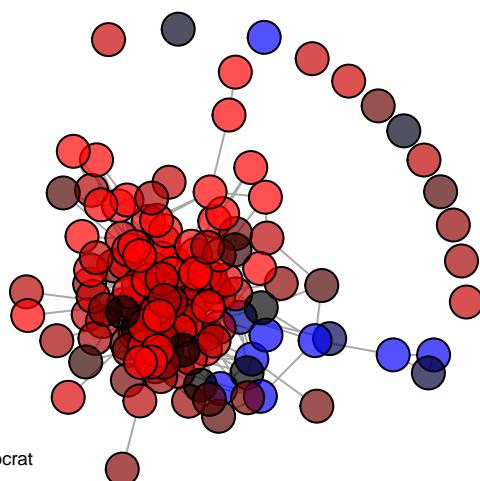
- 100% Democrat
- 50% Republican; 50% Democrat
- 100% Repbulican

AR-2015-2016-House



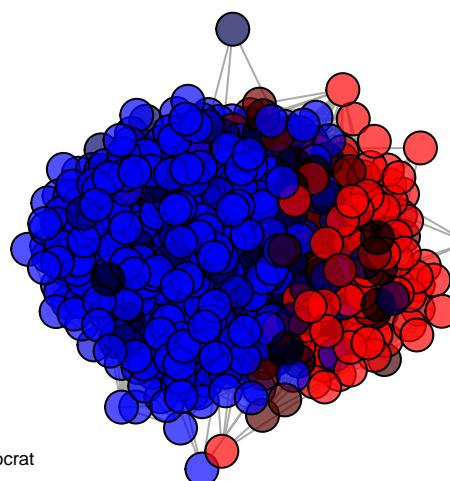
- 100% Democrat
- 50% Republican; 50% Democrat
- 100% Repbulican

AZ-2015-2016-House



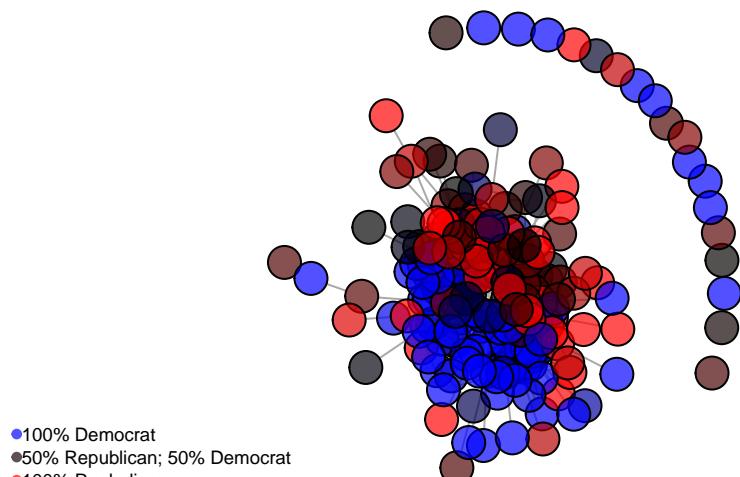
- 100% Democrat
- 50% Republican; 50% Democrat
- 100% Repbulican

CA-2015-2016-House

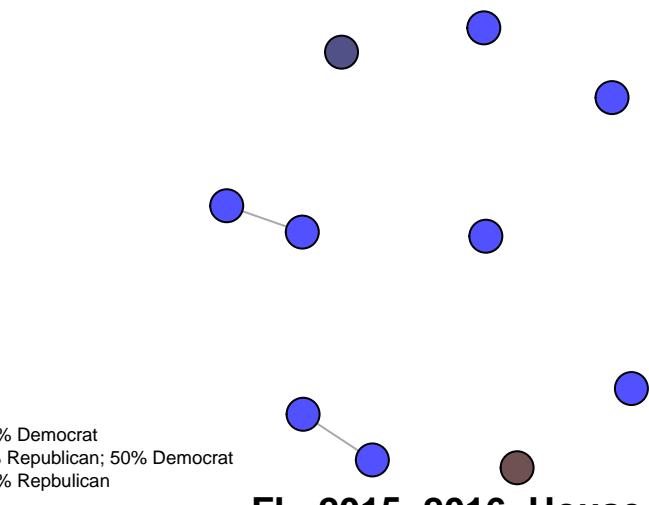


- 100% Democrat
- 50% Republican; 50% Democrat
- 100% Repbulican

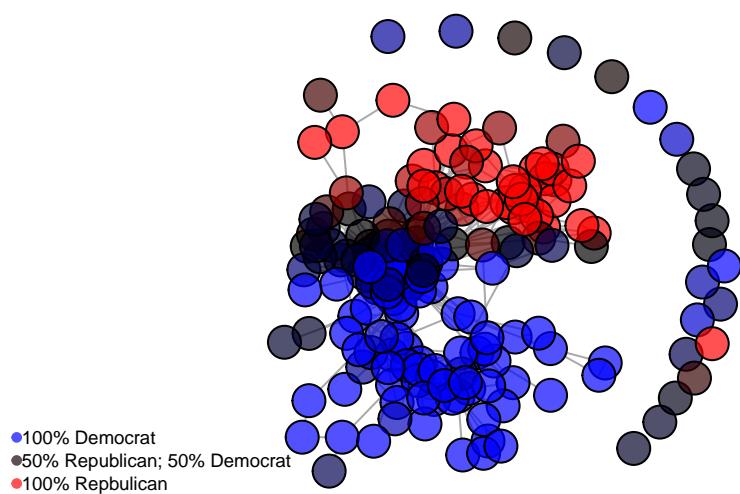
CO-2015-2016-House



CT-2015-2016-House

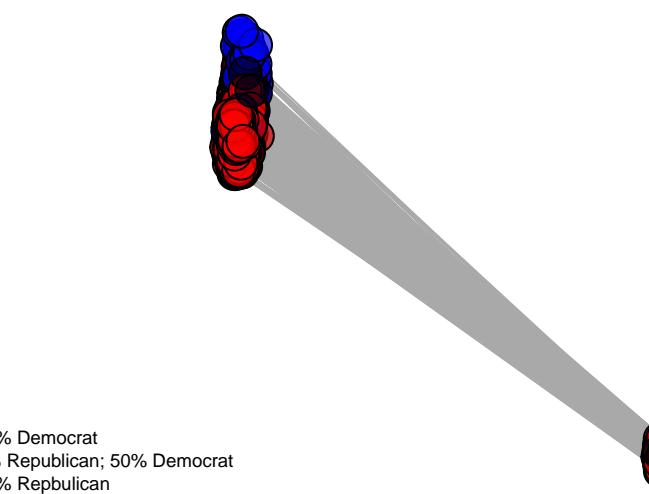


DE-2015-2016-House

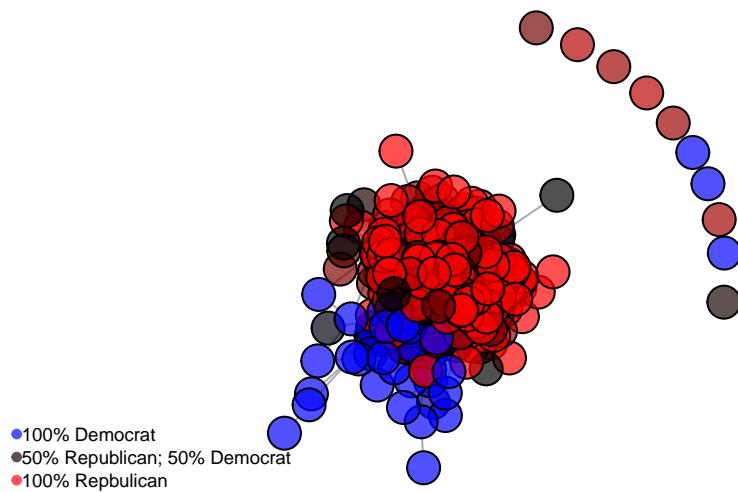


- 100% Democrat
- 50% Republican; 50% Democrat
- 100% Republican

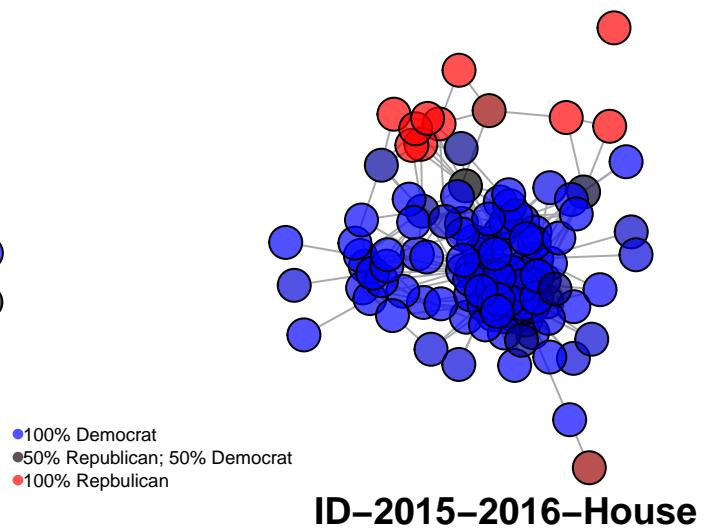
FL-2015-2016-House



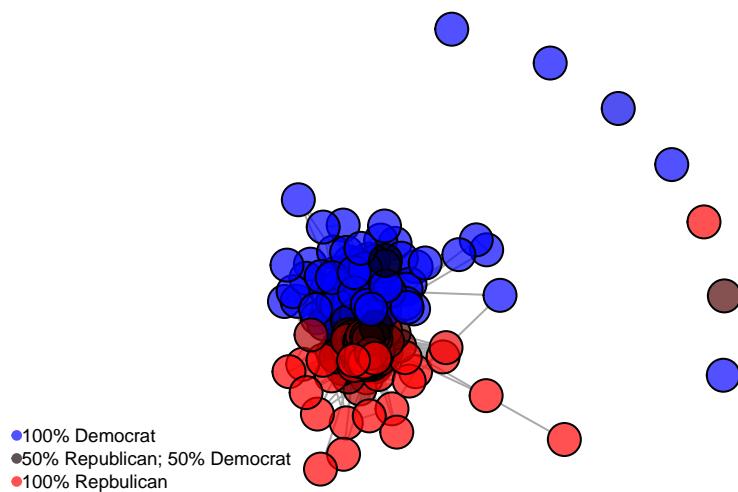
GA-2015-2016-House



HI-2015-2016-House

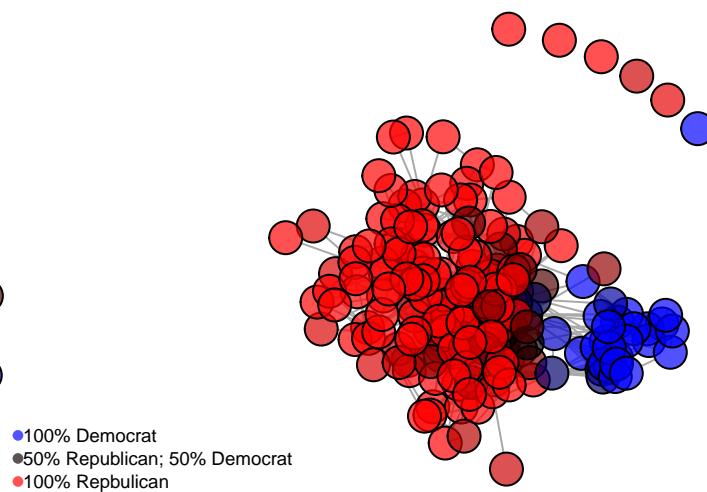


IA-2015-2016-House

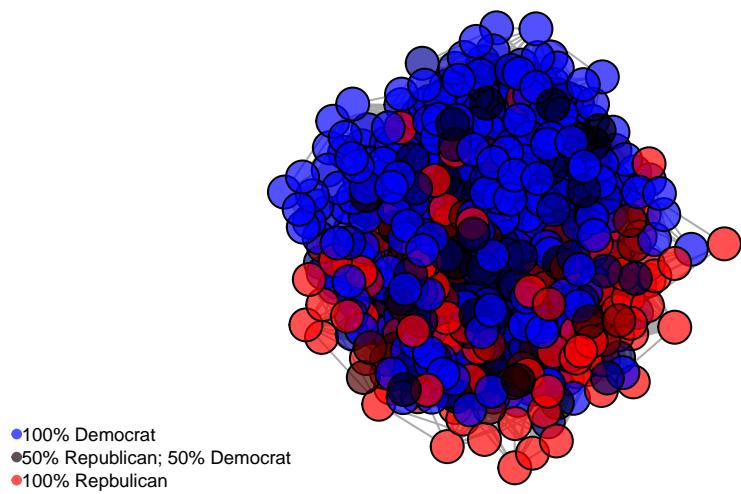


● 100% Democrat
● 50% Republican; 50% Democrat
● 100% Republican

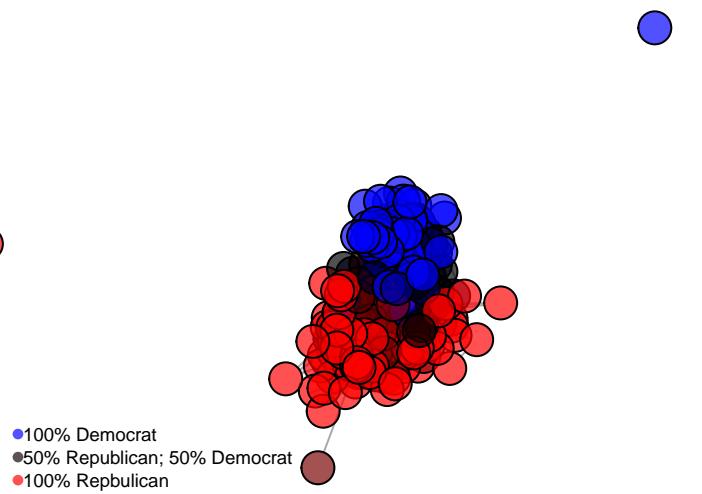
ID-2015-2016-House



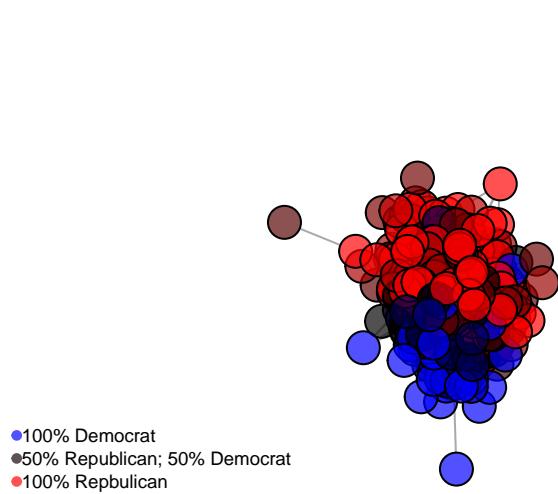
IL-2015-2016-House



IN-2015-2016-House

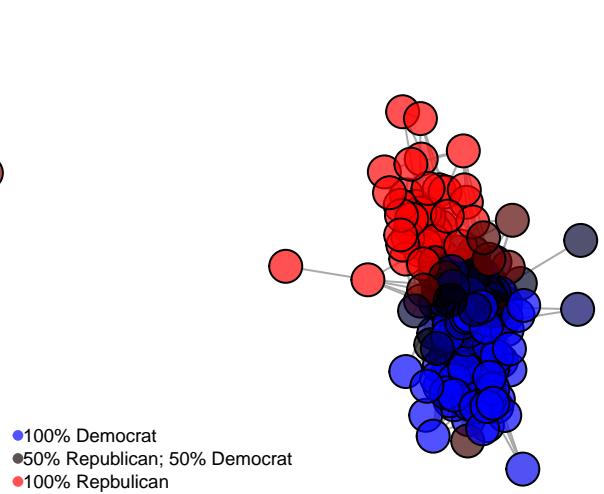


KS-2015-2016-House

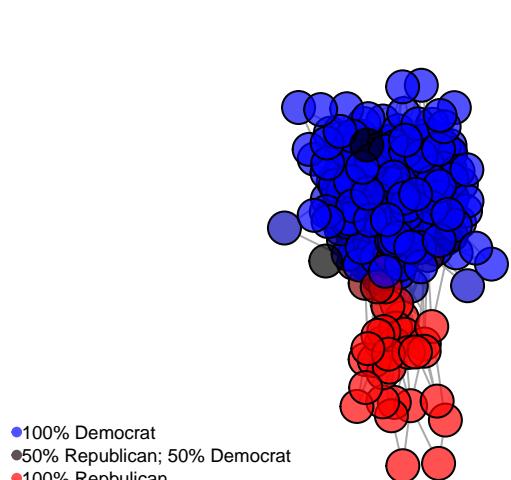


● 100% Democrat
● 50% Republican; 50% Democrat
● 100% Republican

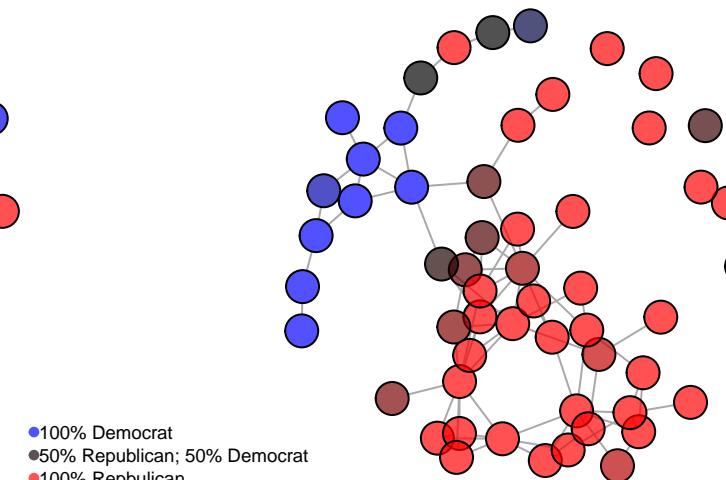
KY-2015-2016-House



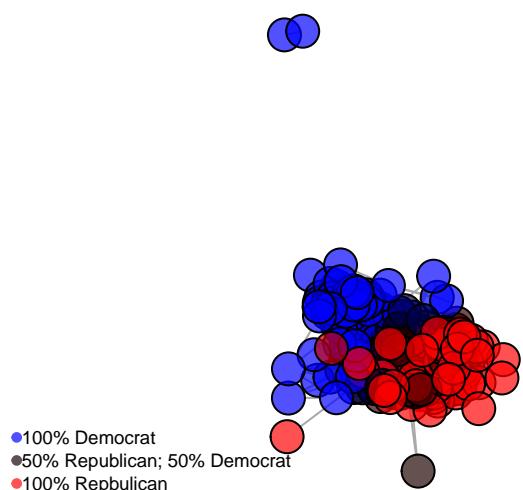
MA-2015-2016-House



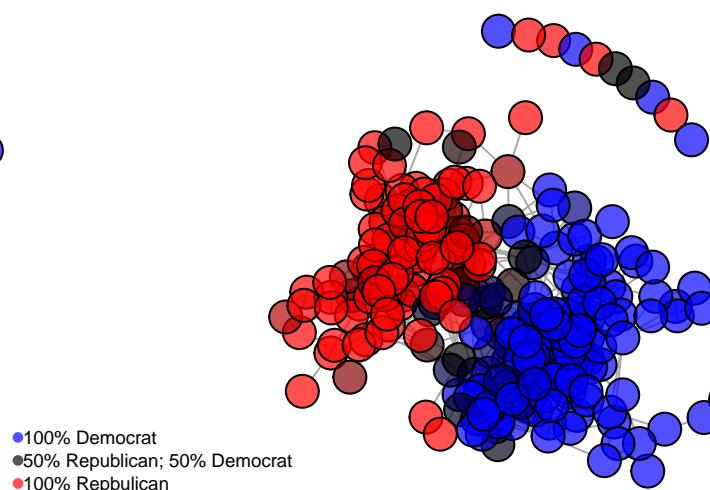
ME-2015-2016-House



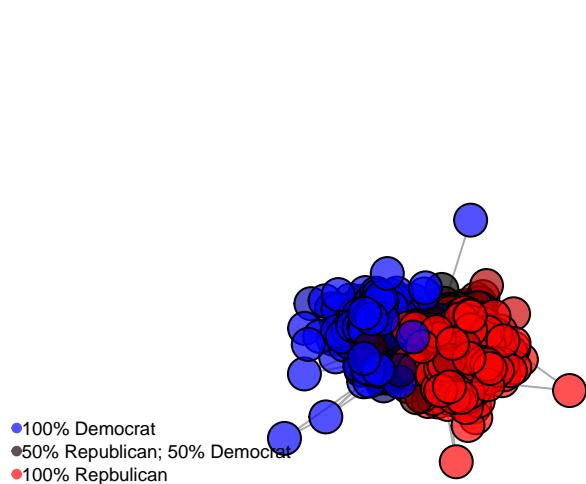
MI-2015-2016-House



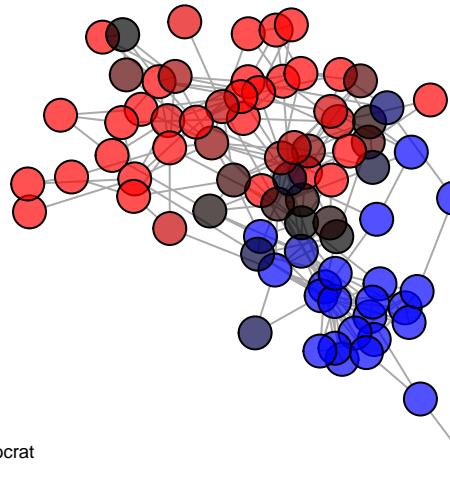
MN-2015-2016-House



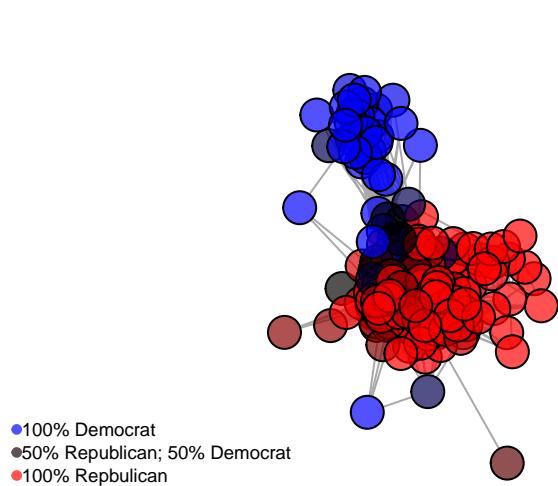
MO-2015-2016-House



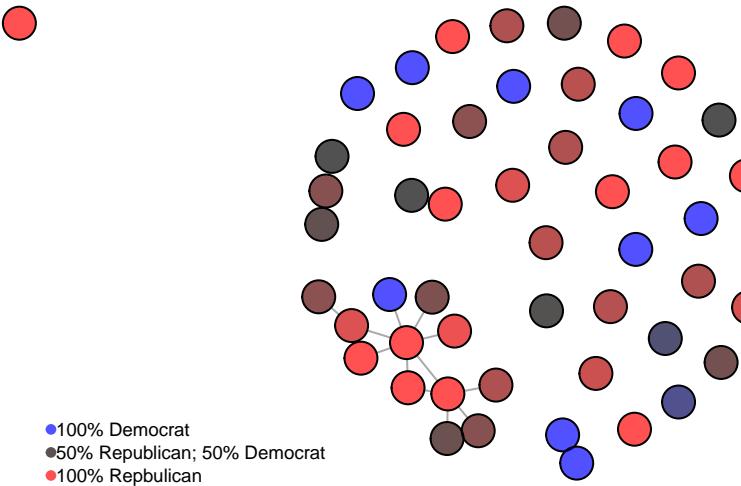
MT-2015-2016-House



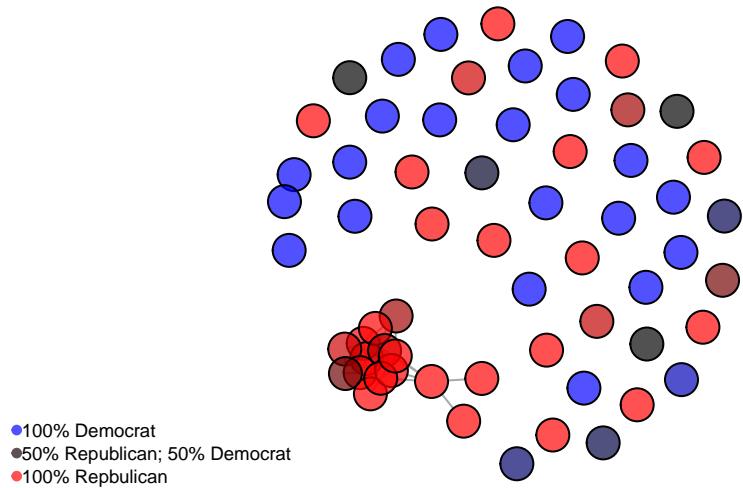
NC-2015-2016-House



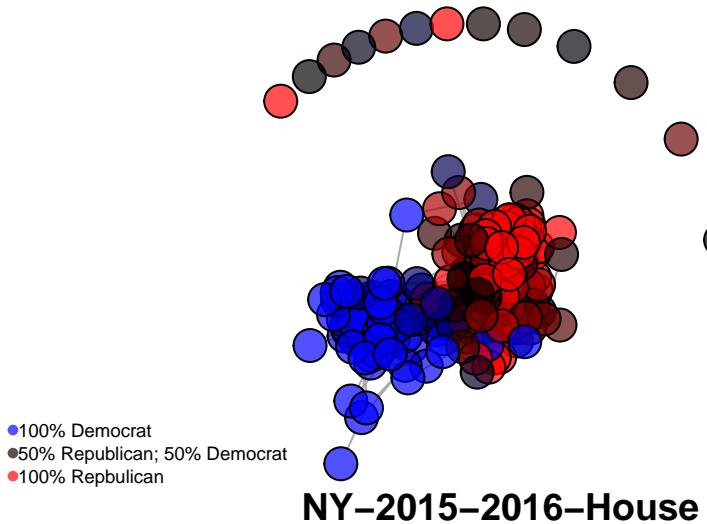
ND-2015-2016-House



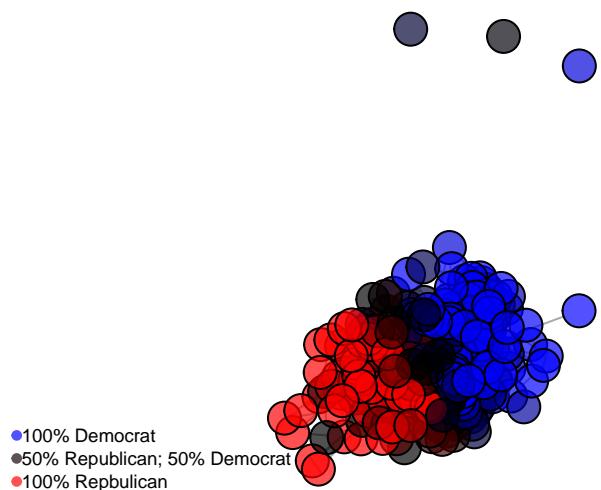
NH-2015-2016-House



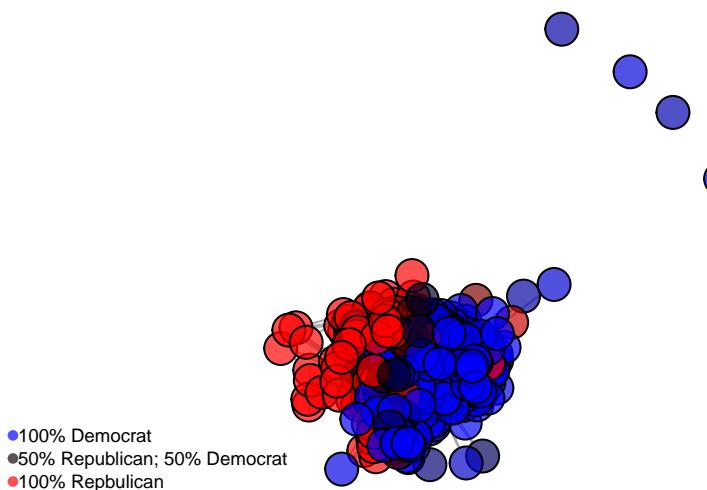
NM-2015-2016-House



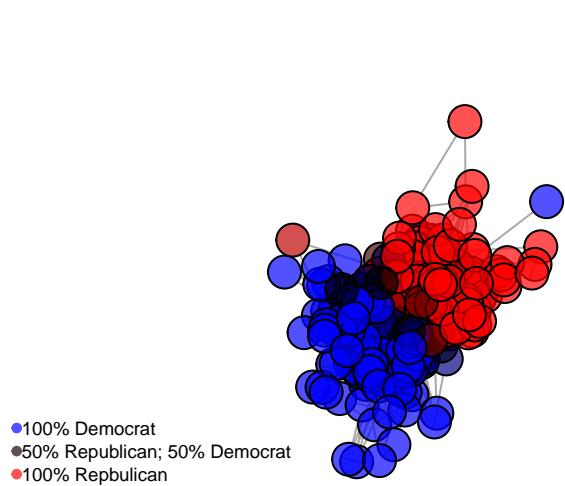
NV-2015-2016-House



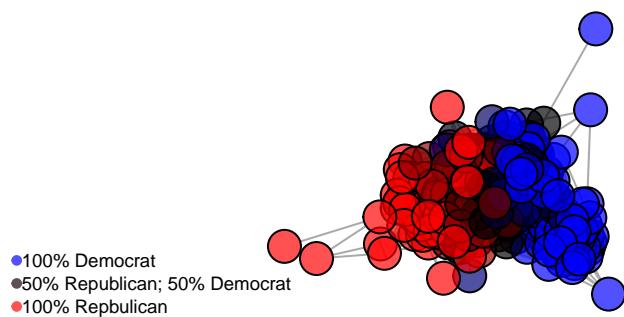
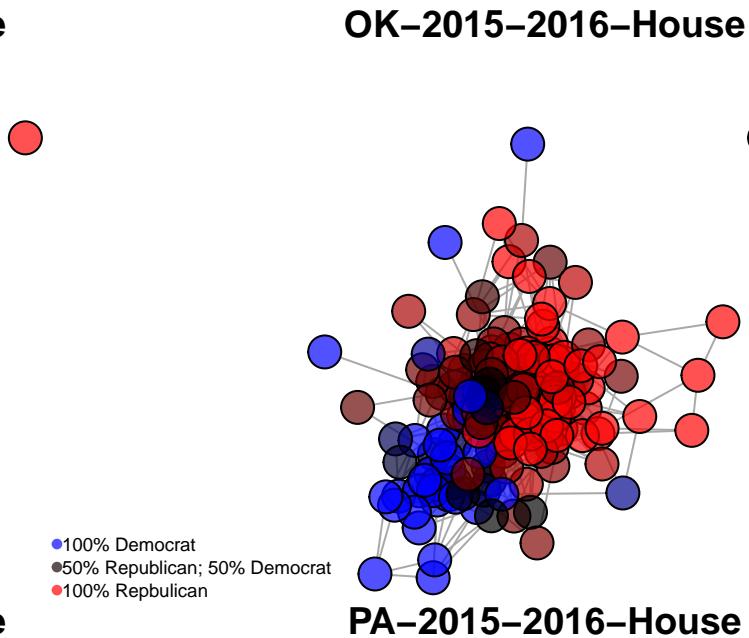
NY-2015-2016-House



OH-2015-2016-House

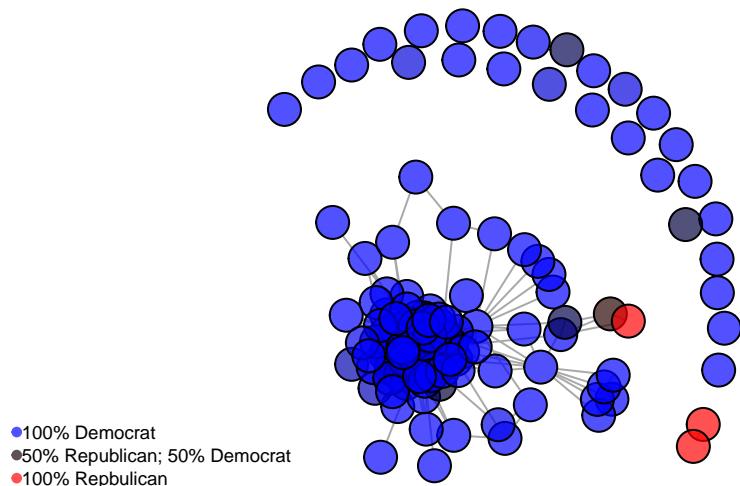


OR-2015-2016-House

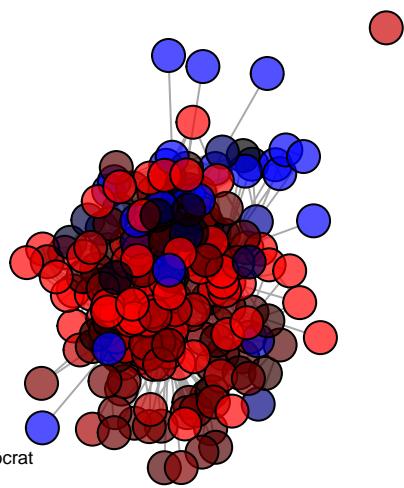


● 100% Democrat
● 50% Republican; 50% Democrat
● 100% Republican

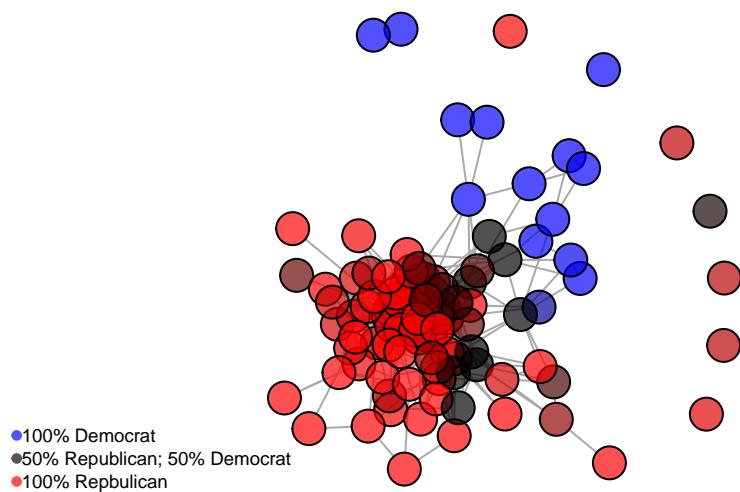
RI-2015-2016-House



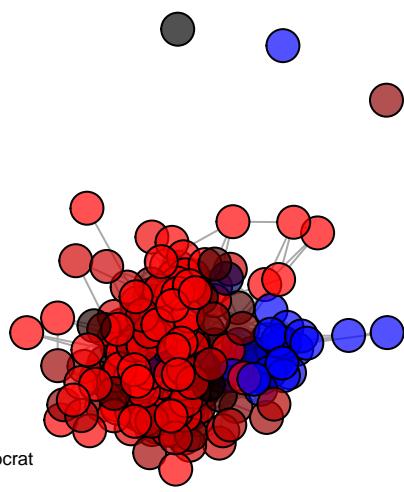
SC-2015-2016-House



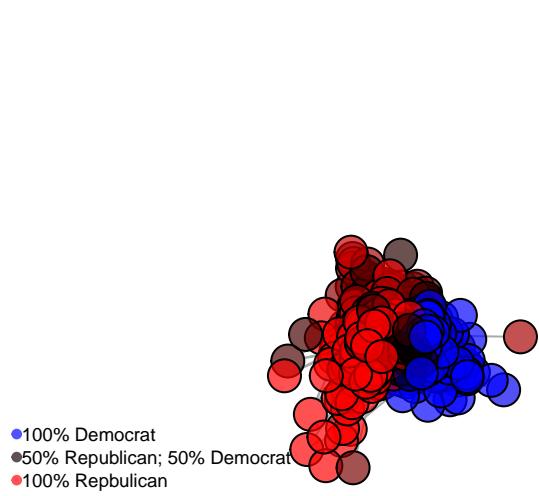
SD-2015-2016-House



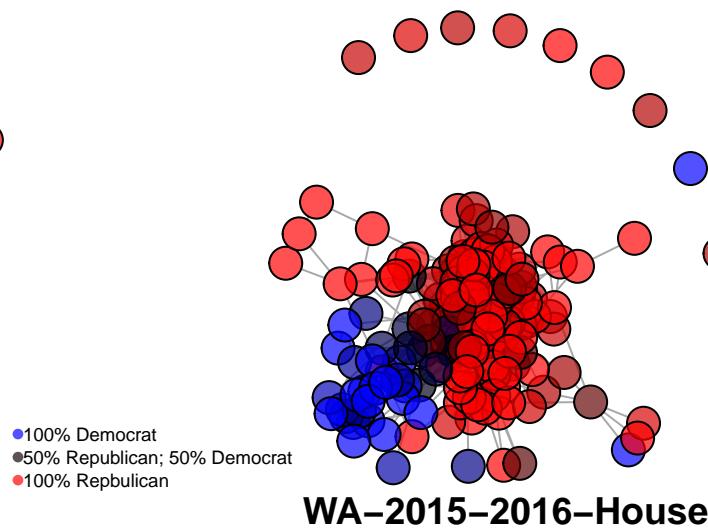
TN-2015-2016-House



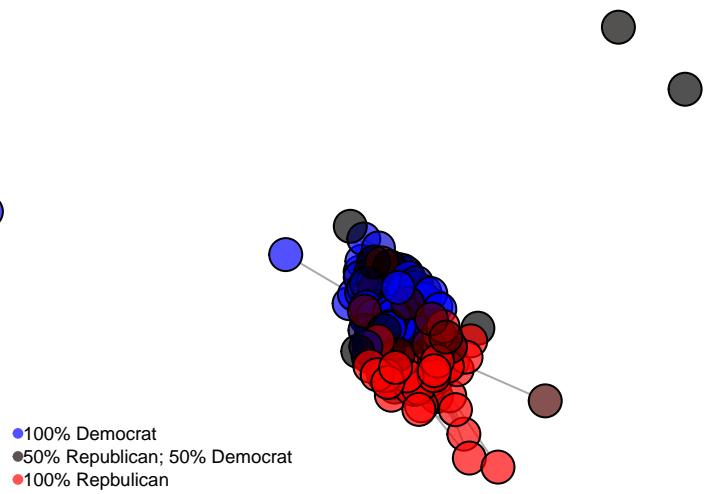
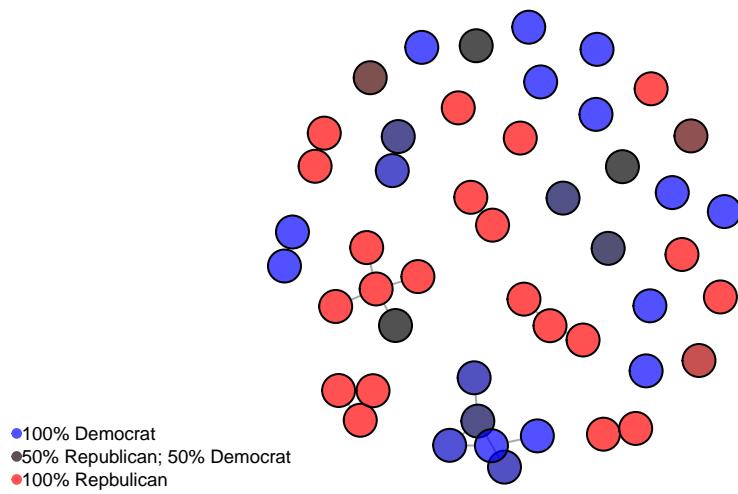
TX-2015-2016-House

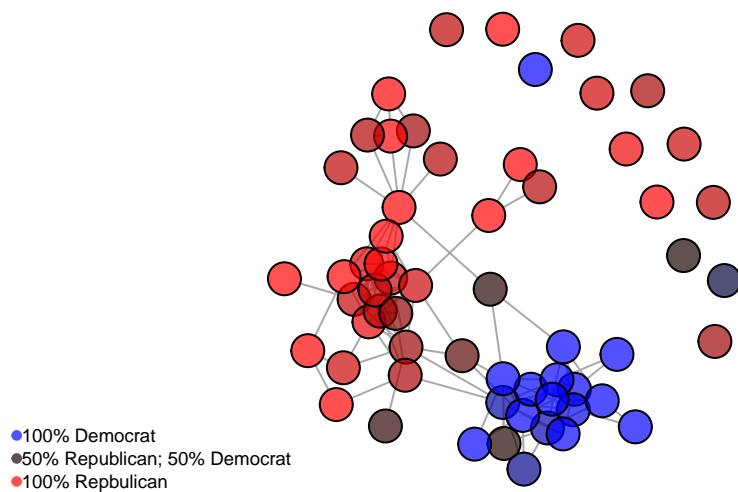
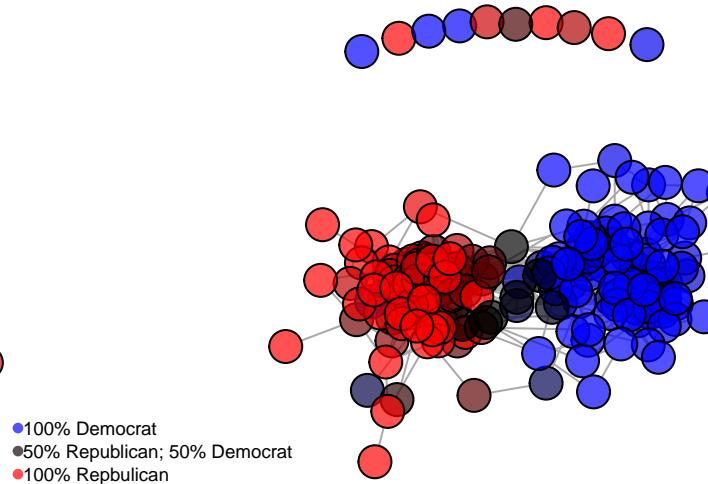
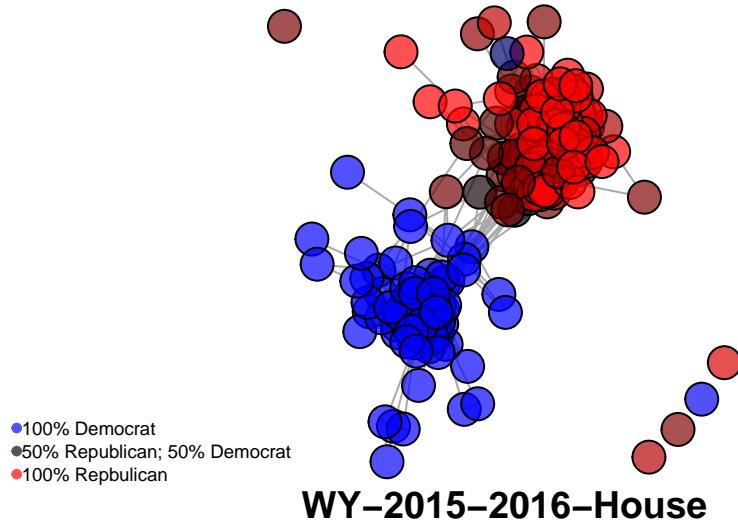
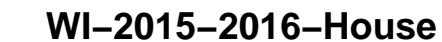


UT-2015-2016-House



VT-2015-2016-House





```

label <- substr(y,1,2)
table <- data.frame(label,density=round(unlist(density), 3), diameter=round(unlist(diameter),3), mean_d
col.names <- c("State","Density","Diameter","Mean\nDistance","Degree\nnCentrality","Betweenness\nCentrali

```

Below is a table showing the values for different centrality measures for each state for the 2016 election:

```
table %>%  
  mutate_all(linebreak) %>%  
  kable(format="latex", escape=F, row.names=FALSE, col.names=linebreak(col.names), caption="Network Level Su")
```

```
kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: Network Level Summary Statistics Election Year 2016

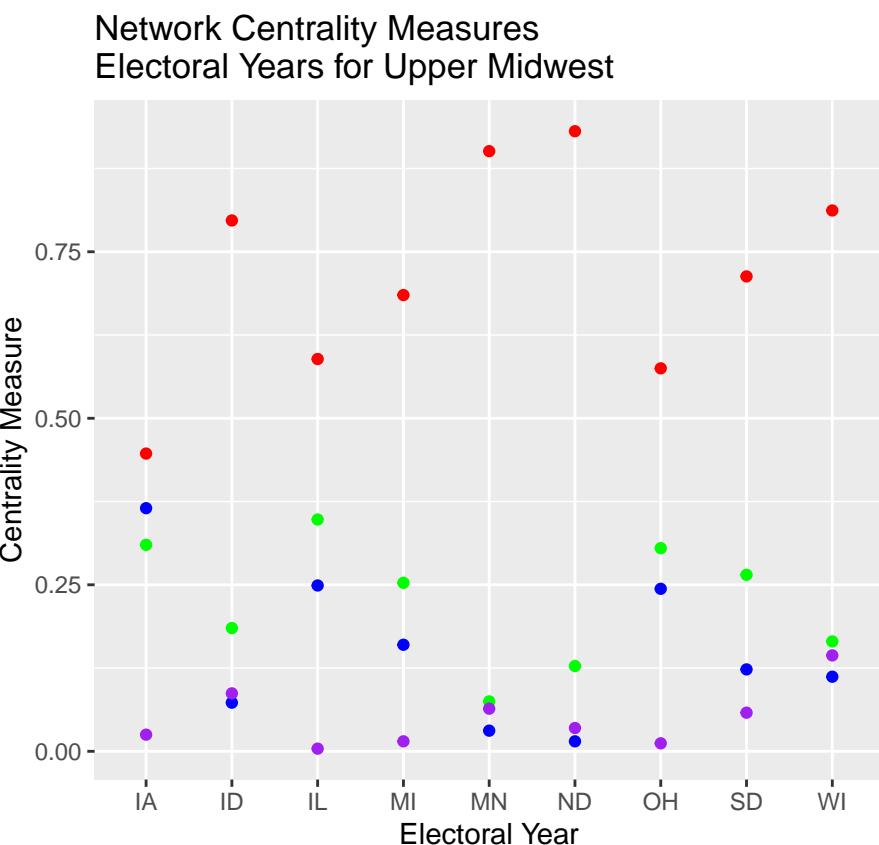
State	Density	Diameter	Mean Distance	Degree Centrality	Betweeness Centrality	Eigenvector Centrality
AK	0.118	6	2.563	0.145	0.182	0.740
AR	0.110	5	2.298	0.200	0.035	0.753
AZ	0.078	7	2.612	0.354	0.150	0.819
CA	0.229	4	1.782	0.285	0.002	0.584
CO	0.040	8	2.913	0.129	0.060	0.865
CT	0.036	1	1.000	0.064	0.000	1.000
DE	0.056	8	3.086	0.222	0.110	0.831
FL	0.173	4	1.860	0.377	0.006	0.659
GA	0.081	8	2.504	0.296	0.038	0.788
HI	0.117	7	2.518	0.272	0.105	0.743
IA	0.365	5	1.723	0.310	0.025	0.447
ID	0.073	6	2.769	0.185	0.087	0.797
IL	0.249	4	1.765	0.348	0.004	0.589
IN	0.187	4	1.946	0.310	0.031	0.651
KS	0.078	5	2.317	0.200	0.025	0.797
KY	0.171	5	2.059	0.218	0.034	0.654
MA	0.081	6	2.349	0.169	0.023	0.775
ME	0.063	11	4.021	0.140	0.262	0.794
MI	0.160	4	1.965	0.253	0.015	0.685
MN	0.031	10	3.591	0.075	0.064	0.901
MO	0.110	5	2.126	0.225	0.012	0.755
MT	0.082	7	2.927	0.184	0.212	0.788
NC	0.305	5	1.857	0.320	0.021	0.528
ND	0.015	4	2.113	0.128	0.035	0.931
NH	0.033	3	1.575	0.167	0.014	0.853
NM	0.112	6	2.500	0.260	0.062	0.759
NV	0.108	5	2.263	0.280	0.022	0.765
NY	0.109	5	2.166	0.248	0.014	0.744
OH	0.244	5	1.879	0.305	0.012	0.575
OK	0.139	6	2.222	0.253	0.031	0.708
OR	0.172	5	2.006	0.319	0.023	0.687
PA	0.100	5	2.157	0.241	0.011	0.761
RI	0.137	5	2.107	0.438	0.085	0.700
SC	0.068	6	2.583	0.218	0.116	0.817
SD	0.123	6	2.388	0.265	0.058	0.713
TN	0.112	6	2.292	0.260	0.040	0.739
TX	0.124	5	2.034	0.272	0.007	0.719
UT	0.100	6	2.536	0.302	0.039	0.755
VT	0.019	3	1.444	0.066	0.005	0.953
WA	0.212	4	1.854	0.274	0.006	0.620
WI	0.112	7	2.715	0.165	0.144	0.812
WV	0.099	7	2.770	0.237	0.051	0.772
WY	0.086	6	2.936	0.134	0.152	0.795

```
tablesub <- table[table$label=="MN" | table$label=="IA" | table$label=="WI" | table$label == "OH" | table$label == "MN"]
```

```

geom_point(aes(label, as.numeric(density), color="purple"))+
geom_point(aes(label, as.numeric(cd), color="red"))+
geom_point(aes(label, as.numeric(cb), color="blue"))+
geom_point(aes(label, as.numeric(ce), color="green"))+
labs(title="Network Centrality Measures\nElectoral Years for Upper Midwest", x="Electoral Year", y="Centrality Measure",
scale_color_manual(labels =c("Density","Degree Centrality","Degree Betweenness","Degree Eigenvector")),

```



Legend

- Density
- Degree Centrality
- Degree Betweenness
- Degree Eigenvector

Degree centrality is consistently higher than the other centrality measures in each state. Interestingly, there is no clear indication that a particular state has a higher network level centrality. For instance, North Dakota has the highest degree centrality but has the lowest degree betweenness. This means that there is one or a few donor groups that have many connections, but there aren't many donor groups that act as a bridge to other donor groups. Looking at the plot of North Dakota in 2016 on the previous page, this makes a lot of sense. There appears to be one or two donor groups that are connected to other donor groups and a lot of isolate donor groups. In contrast, South Dakota's centrality measures are closer together and this makes sense when looking at the plot for South Dakota. It is clear that there are two clusters of donor groups-one Republican and one Democrat-but there appear to be many donor groups that are connected to others and far fewer isolates than in North Dakota.

First Set

Scenario 1

Same number of Republican and Democratic candidates as reality

```
##I create a vector that captures the unique ID for state and year
y <- x[str_detect(x, "House", negate=FALSE)==TRUE]
seq <- seq(1,by=1, len=length(y))

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd1 <- vector()
mean_cb1 <- vector()
mean_ce1 <- vector()
mean_d1 <- vector()
l <- 10
```

I first create a list with all of the datasets broken up by state and election year.

```
md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))
```

Then I create vectors for the candidates that each donor had donated to for each election in each state using the number of Democratic and Republican candidates and the probability of a donor group donating to a Democratic or Republican candidate.

```
s <- replicate(l, lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_de
```

I create a function that will create a bipartite matrix and then an incident matrix of donor groups.

```
matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as rows and the states as columns
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate the network level measures
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}
```

Now I have the function to create the donor group networks and all of the simulated data, which is stored in s. I next create a list that includes all of the one-mode graphs for all states and all simulations.

```
all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))
```

Now I use those graphs to calculate network level centrality measures

```

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]] , 1))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]]))$cd
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]]))$cb
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]]))$ce

mean_cd1 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb1 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce1 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d1 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

Scenario 2

One more Republican candidate and same number of Democratic candidates as reality

I do not include the comments between each step to save space. The methodology is exactly the same as in scenario one except for the number of candidates changes.

```

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd2 <- vector()
mean_cb2 <- vector()
mean_ce2 <- vector()
mean_d2 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l, lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_de
matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates at
  vec <- unlist(j[[k,p]])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1]+1, ncol=n[[k]])

  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]] , 1))

```

```

cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]]))$cd)
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]]))$cb)
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]]))$ce)

mean_cd2 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb2 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce2 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d2 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

Scenario 3

One more Democratic candidate and same number of Republican candidates as reality

I do not include the comments between each step to save space. The methodology is exactly the same as in scenario one except for the number of candidates changes.

```

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd3 <- vector()
mean_cb3 <- vector()
mean_ce3 <- vector()
mean_d3 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l,lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_dem[i], 1, 0.5), rbinom(full[[x]]$num_rep[i], 1, 0.5)))))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as rows and the voters as columns
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+1+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate the centrality
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]]), 1))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]]))$cd)
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]]))$cb)
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]]))$ce)

```

```

# cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]]))$cb
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]]))$ce

mean_cd3 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb3 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce3 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d3 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

Scenario 4

One less Republican candidate and same number of Democratic candidates as reality

I do not include the comments between each step to save space. The methodology is exactly the same as in scenario one except for the number of candidates changes.

```

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd4 <- vector()
mean_cb4 <- vector()
mean_ce4 <- vector()
mean_d4 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l,lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_dem[i], 1, 0.5), rbinom(full[[x]]$num_rep[i], 1, 0.5))))))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as rows and the voters as columns
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1]-1, ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate the centrality
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]]), 1))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]])))
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]]))$cb)
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]]))$ce)

```

```

ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]]))$co

mean_cd4 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb4 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce4 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d4 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

Scenario 5

One less Democratic candidate and same number of Republican candidates as reality

I do not include the comments between each step to save space. The methodology is exactly the same as in scenario one except for the number of candidates changes.

```

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd5 <- vector()
mean_cb5 <- vector()
mean_ce5 <- vector()
mean_d5 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l, lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(ifelse(full[[x]]>0, full[[x]]-1, 0), l, 0.5), rbinom(ifelse(full[[x]]>0, full[[x]]-1, 0), r)))))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as rows and the voters as columns
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]-1+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate the centrality
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]], 1))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]])))
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]])))
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]])))$co

```

```

mean_cd5 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb5 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce5 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d5 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

I can now compare the centrality measures across all five scenarios.

Mean Density:

```

##Combine centrality measures
label <- paste0(substr(y,1,2))
table <- data.frame(label,round(as.numeric(unlist(mean_d1)),3),round(as.numeric(unlist(mean_d2)),3),round(as.numeric(unlist(mean_d3)),3),round(as.numeric(unlist(mean_d4)),3),round(as.numeric(unlist(mean_d5)),3))
col_names <- c("State","Scenario 1","Scenario 2","Scenario 3","Scenario 4","Scenario 5")

```

Below is a table showing the values for different centrality measures by state for election year 2016:

```

table %>%
mutate_all(linebreak) %>%
kable(format="latex",escape=F,row.names=FALSE,col.names=linebreak(col_names), caption="Network Level Degree Centrality by State Across Scenarios")
kable_styling(latex_options = c("striped", "hold_position"))

```

From the table above, it appears that the density is dependent on the number of candidates running in the election for some states. Scenarios 2 and 3 tend to have higher mean density than scenario 1 and scenarios 4 and 5 tend to have lower mean density than scenario 1. It seems like this trend is greater in states with fewer candidates running such as Alaska, North Dakota, Wyoming, Vermont, etc. In other states, the degree centrality is fairly constant across scenarios.

Degree Centrality:

```

##Combine centrality measures
label <- paste0(substr(y,1,2))
table <- data.frame(label,round(as.numeric(unlist(mean_cd1)),3),round(as.numeric(unlist(mean_cd2)),3),round(as.numeric(unlist(mean_cd3)),3),round(as.numeric(unlist(mean_cd4)),3),round(as.numeric(unlist(mean_cd5)),3))
col_names <- c("State","Scenario 1","Scenario 2","Scenario 3","Scenario 4","Scenario 5")

```

Below is a table showing the values for different centrality measures by state for election year 2016:

```

table %>%
mutate_all(linebreak) %>%
kable(format="latex",escape=F,row.names=FALSE,col.names=linebreak(col_names), caption="Network Level Degree Centrality by State Across Scenarios")
kable_styling(latex_options = c("striped", "hold_position"))

```

There doesn't appear to be a pattern in mean degree centrality over the five scenarios. In some states, adding an additional candidate-either a Republican or a Democrat-increases degree centrality and in other states, adding an additional candidate decreases degree centrality. The same applies for decreasing a candidate. It also doesn't appear to make much of a difference whether the candidate added or removed was a Republican or Democrat. Most states have the same-if not very similar-degree centrality measures for scenarios 2 and 3 and similarly for scenarios 4 and 5.

Eigenvector Centrality:

```

##Combine centrality measures
label <- paste0(substr(y,1,2))
table <- data.frame(label,round(as.numeric(unlist(mean_ce1)),3),round(as.numeric(unlist(mean_ce2)),3),round(as.numeric(unlist(mean_ce3)),3),round(as.numeric(unlist(mean_ce4)),3),round(as.numeric(unlist(mean_ce5)),3))
col_names <- c("State","Scenario 1","Scenario 2","Scenario 3","Scenario 4","Scenario 5")

```

Below is a table showing the values for different centrality measures by state for election year 2016:

```

table %>%
mutate_all(linebreak) %>%
kable(format="latex",escape=F,row.names=FALSE,col.names=linebreak(col_names), caption="Network Level Eigenvector Centrality by State Across Scenarios")
kable_styling(latex_options = c("striped", "hold_position"))

```

Table 2: Network Level Density for All Five Scenarios

State	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
AK	0.432	0.533	0.508	0.202	0.205
AR	0.775	0.787	0.801	0.780	0.769
AZ	0.946	0.945	0.946	0.975	0.977
CA	0.923	0.923	0.923	0.990	0.990
CO	0.837	0.845	0.842	0.908	0.910
CT	1.000	0.984	0.982	0.911	0.920
DE	0.533	0.586	0.629	0.234	0.259
FL	0.927	0.928	0.928	0.981	0.981
GA	0.898	0.899	0.901	0.983	0.983
HI	0.835	0.831	0.849	0.616	0.612
IA	0.769	0.788	0.807	0.814	0.814
ID	0.779	0.804	0.780	0.605	0.593
IL	0.902	0.903	0.903	0.966	0.967
IN	0.822	0.828	0.828	0.921	0.921
KS	0.814	0.840	0.838	0.829	0.828
KY	0.797	0.808	0.811	0.816	0.824
MA	0.848	0.852	0.852	0.949	0.949
ME	0.724	0.742	0.724	0.610	0.607
MI	0.858	0.860	0.863	0.962	0.962
MN	0.657	0.662	0.660	0.851	0.848
MO	0.792	0.795	0.797	0.912	0.912
MT	0.527	0.569	0.557	0.252	0.251
NC	0.884	0.886	0.887	0.959	0.958
ND	0.502	0.615	0.555	0.248	0.256
NH	0.578	0.599	0.607	0.564	0.562
NM	0.717	0.721	0.726	0.727	0.733
NV	0.820	0.836	0.837	0.830	0.829
NY	0.865	0.864	0.864	0.980	0.980
OH	0.838	0.838	0.838	0.943	0.944
OK	0.832	0.836	0.848	0.889	0.890
OR	0.791	0.808	0.807	0.824	0.826
PA	0.847	0.846	0.847	0.960	0.959
RI	0.943	0.942	0.945	0.612	0.613
SC	0.921	0.927	0.927	0.950	0.951
SD	0.612	0.723	0.667	0.243	0.247
TN	0.908	0.907	0.907	0.943	0.940
TX	0.934	0.935	0.935	0.989	0.989
UT	0.847	0.853	0.855	0.831	0.830
VT	0.194	0.467	0.246	0.000	0.000
WA	0.866	0.865	0.868	0.933	0.932
WI	0.671	0.673	0.677	0.847	0.847
WV	0.652	0.673	0.675	0.739	0.747
WY	0.471	0.586	0.541	0.258	0.248

```
kable(format="latex", escape=F, row.names=FALSE, col.names=linebreak(col_names), caption="Network Level Eigenvector Centrality for All Five Scenarios")
kable_styling(latex_options = c("striped", "hold_position"))
```

The same patterns found in degree centrality are found in eigenvector centrality.

Table 3: Network Level Degree Centrality for All Five Scenarios

State	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
AK	0.401	0.355	0.413	0.243	0.244
AR	0.215	0.203	0.194	0.150	0.155
AZ	0.054	0.055	0.054	0.025	0.023
CA	0.077	0.077	0.077	0.010	0.010
CO	0.163	0.155	0.158	0.080	0.079
CT	0.000	0.016	0.008	0.079	0.080
DE	0.383	0.365	0.326	0.248	0.248
FL	0.073	0.072	0.072	0.018	0.018
GA	0.102	0.101	0.099	0.017	0.017
HI	0.153	0.156	0.141	0.353	0.353
IA	0.230	0.212	0.192	0.160	0.161
ID	0.213	0.195	0.212	0.329	0.331
IL	0.098	0.097	0.097	0.030	0.030
IN	0.178	0.172	0.172	0.073	0.072
KS	0.181	0.159	0.160	0.143	0.143
KY	0.203	0.192	0.189	0.159	0.152
MA	0.149	0.145	0.146	0.037	0.037
ME	0.265	0.256	0.268	0.338	0.330
MI	0.142	0.140	0.137	0.037	0.037
MN	0.343	0.338	0.340	0.123	0.127
MO	0.208	0.205	0.203	0.081	0.082
MT	0.421	0.393	0.403	0.246	0.246
NC	0.116	0.114	0.113	0.038	0.039
ND	0.396	0.349	0.376	0.244	0.242
NH	0.404	0.389	0.386	0.328	0.331
NM	0.281	0.277	0.272	0.230	0.226
NV	0.179	0.163	0.162	0.148	0.148
NY	0.135	0.135	0.136	0.020	0.020
OH	0.162	0.162	0.162	0.053	0.052
OK	0.168	0.164	0.152	0.096	0.098
OR	0.207	0.190	0.193	0.147	0.146
PA	0.153	0.154	0.153	0.039	0.039
RI	0.044	0.051	0.045	0.366	0.364
SC	0.079	0.073	0.073	0.049	0.047
SD	0.304	0.249	0.289	0.247	0.246
TN	0.092	0.093	0.093	0.053	0.054
TX	0.066	0.065	0.065	0.010	0.010
UT	0.153	0.147	0.145	0.158	0.159
VT	0.240	0.458	0.250	0.000	0.000
WA	0.129	0.129	0.126	0.059	0.058
WI	0.329	0.327	0.323	0.131	0.131
WV	0.345	0.325	0.324	0.226	0.220
WY	0.396	0.369	0.377	0.245	0.245

Second Set

First, I need to create the probabilities for the scenarios.

Table 4: Network Level Eigenvector Centrality for All Five Scenarios

State	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
AK	0.433	0.351	0.413	0.561	0.557
AR	0.165	0.156	0.156	0.151	0.157
AZ	0.043	0.043	0.043	0.021	0.020
CA	0.062	0.062	0.062	0.009	0.009
CO	0.142	0.135	0.137	0.074	0.073
CT	0.000	0.011	0.011	0.083	0.080
DE	0.353	0.333	0.277	0.521	0.495
FL	0.055	0.055	0.055	0.017	0.017
GA	0.076	0.075	0.074	0.015	0.015
HI	0.095	0.100	0.088	0.336	0.338
IA	0.205	0.188	0.171	0.149	0.149
ID	0.147	0.129	0.150	0.323	0.329
IL	0.078	0.077	0.078	0.028	0.027
IN	0.152	0.146	0.147	0.067	0.067
KS	0.148	0.129	0.134	0.133	0.134
KY	0.173	0.164	0.161	0.148	0.142
MA	0.091	0.089	0.089	0.035	0.035
ME	0.196	0.179	0.198	0.328	0.326
MI	0.125	0.122	0.120	0.033	0.033
MN	0.305	0.301	0.302	0.116	0.118
MO	0.171	0.168	0.168	0.074	0.074
MT	0.397	0.357	0.393	0.508	0.509
NC	0.088	0.088	0.087	0.035	0.036
ND	0.397	0.303	0.372	0.519	0.513
NH	0.386	0.361	0.362	0.346	0.348
NM	0.250	0.244	0.242	0.219	0.214
NV	0.158	0.144	0.142	0.136	0.136
NY	0.104	0.104	0.105	0.018	0.018
OH	0.139	0.139	0.139	0.047	0.046
OK	0.139	0.136	0.128	0.089	0.089
OR	0.184	0.169	0.170	0.137	0.136
PA	0.134	0.134	0.134	0.034	0.035
RI	0.030	0.031	0.029	0.346	0.343
SC	0.065	0.061	0.061	0.045	0.044
SD	0.252	0.183	0.232	0.515	0.512
TN	0.074	0.075	0.075	0.048	0.049
TX	0.057	0.057	0.056	0.009	0.009
UT	0.107	0.103	0.104	0.147	0.148
VT	0.578	0.471	0.524	0.000	0.000
WA	0.114	0.115	0.111	0.054	0.054
WI	0.298	0.296	0.292	0.122	0.122
WV	0.317	0.296	0.294	0.215	0.209
WY	0.407	0.321	0.372	0.505	0.516

```

md_house$scen1r <- ifelse(md_house$PerRep==100, 90, md_house$PerRep)
md_house$scen1d <- ifelse(md_house$PerRep==100, 10, md_house$PerDem)

md_house$scen2r <- ifelse(md_house$PerDem==100, 10, md_house$PerRep)

```

```

md_house$scen2d <- ifelse(md_house$PerDem==100, 90, md_house$PerDem)

md_house$scen3r <- ifelse(md_house$PerRep < 100 & md_house$PerRep > 0, 100, md_house$PerRep)
md_house$scen3d <- ifelse(md_house$PerRep < 100 & md_house$PerRep > 0, 0, md_house$PerDem)

md_house$scen4r <- ifelse(md_house$PerDem < 100 & md_house$PerDem > 0, 0, md_house$PerRep)
md_house$scen4d <- ifelse(md_house$PerDem < 100 & md_house$PerDem > 0, 100, md_house$PerDem)

```

Scenario 1

```

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd1 <- vector()
mean_cb1 <- vector()
mean_ce1 <- vector()
mean_d1 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l,lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_de...))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as rows and the voters as columns
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]], 1))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]])$c...))
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]])$c...
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]])$c...

mean_cd1 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb1 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))

```

```

mean_ce1 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d1 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

Scenario 2

```

##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd2 <- vector()
mean_cb2 <- vector()
mean_ce2 <- vector()
mean_d2 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l,lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_dem[i], 1, full[[x]]$prob[i]])))))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as rows and the voters as columns
  vec <- unlist(j[[k,p]])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate the centrality of the network
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]], 1)))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]])))
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]])))
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]])))

mean_cd2 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb2 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce2 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d2 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

Scenario 3

```
##Create a variables
s <- list()
m <- list()
cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd3 <- vector()
mean_cb3 <- vector()
mean_ce3 <- vector()
mean_d3 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l,lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_dem[i], 1, full[[x]]$prob[i]])))))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as columns and voters as rows
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate the centrality
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]]), 1))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]])))
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]])))
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]])))

mean_cd3 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb3 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce3 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d3 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))
```

Scenario 4

```
##Create a variables
s <- list()
m <- list()
```

```

cd <- vector()
cb <- vector()
ce <- vector()
d <- vector()
n <- list()
mean_cd4 <- vector()
mean_cb4 <- vector()
mean_ce4 <- vector()
mean_d4 <- vector()
l <- 10

md_subdata <- lapply(seq, function(k) md_house[endsWith(md_house$filename,y[k]),] )
full <- lapply(1:length(md_subdata), function(x) merge(md_subdata[[x]], vote_cand, by="cat"))
n <- lapply(1:length(full), function(x) nrow(full[[x]]))

s <- replicate(l, lapply(1:length(n), function(x) lapply(1:n[[x]], function(i) c(rbinom(full[[x]]$num_dem[i], 1, full[[x]]$prob[i]])))))

matrix_graph_function <- function(j,k,p){
  ##Below, I combine these vectors to create a bipartite network incidence matrix with the candidates as columns and states as rows
  vec <- unlist(j[k,p])
  m <- matrix(vec,nrow=full[[k]]$num_dem[1]+full[[k]]$num_rep[1], ncol=n[[k]])

  ##Now that I have the incidence matrix, I can convert the incidence matrix into a graph object and then calculate centrality measures
  g <- graph_from_incidence_matrix(m)
  b <- bipartite.projection(g)$proj2
  return(b)
}

all_graphs <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) matrix_graph_function(s,t,x)))

##I now can calculate the centrality of the network.
d <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) edge_density(all_graphs[[t]][[x]], 1)))
cd <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_degree(all_graphs[[t]][[x]])))
#cb <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_betw(all_graphs[[t]][[x]])))
ce <- lapply(1:nrow(s), function(t) lapply(1:ncol(s), function(x) centr_eigen(all_graphs[[t]][[x]])))

mean_cd4 <- lapply(1:nrow(s), function(x) mean(unlist(cd[[x]])))
#mean_cb4 <- lapply(1:nrow(s), function(x) mean(unlist(cb[[x]])))
mean_ce4 <- lapply(1:nrow(s), function(x) mean(unlist(ce[[x]])))
mean_d4 <- lapply(1:nrow(s), function(x) mean(unlist(d[[x]])))

```

I can now compare the centrality measures across all four scenarios.

Mean Density:

```

##Combine centrality measures
label <- paste0(substr(y,1,2))
table <- data.frame(label,round(as.numeric(unlist(mean_d1)),3),round(as.numeric(unlist(mean_d2)),3),round(as.numeric(unlist(mean_d3)),3),round(as.numeric(unlist(mean_d4)),3))
col_names <- c("State","Scenario 1","Scenario 2","Scenario 3","Scenario 4")

```

Below is a table showing the values for different centrality measures by state for election year 2016:

```

table %>%
  mutate_all(linebreak) %>%
  kable(format="latex", escape=F, row.names=FALSE, col.names=linebreak(col_names), caption="Network Level Density Measures by State and Scenario")

```

```
kable_styling(latex_options = c("striped", "hold_position"))
```

Table 5: Network Level Density for All Four Scenarios

State	Scenario 1	Scenario 2	Scenario 3	Scenario 4
AK	0.428	0.425	0.731	0.564
AR	0.772	0.819	0.779	0.597
AZ	0.966	0.978	0.893	0.559
CA	1.000	0.998	0.512	0.837
CO	0.900	0.913	0.566	0.713
CT	0.998	0.996	0.673	1.000
DE	0.520	0.503	0.528	0.715
FL	0.994	0.993	0.869	0.500
GA	0.953	0.964	0.843	0.502
HI	0.855	0.840	0.513	0.853
IA	0.817	0.829	0.578	0.715
ID	0.781	0.805	0.799	0.498
IL	0.977	0.977	0.512	0.817
IN	0.908	0.911	0.682	0.554
KS	0.843	0.864	0.779	0.626
KY	0.861	0.862	0.538	0.692
MA	0.931	0.894	0.602	0.838
ME	0.744	0.749	0.747	0.511
MI	0.953	0.961	0.633	0.641
MN	0.844	0.844	0.504	0.556
MO	0.880	0.890	0.690	0.525
MT	0.482	0.531	0.585	0.539
NC	0.958	0.938	0.812	0.501
ND	0.506	0.511	0.673	0.607
NH	0.611	0.629	0.541	0.512
NM	0.747	0.750	0.616	0.601
NV	0.857	0.854	0.628	0.719
NY	0.986	0.977	0.499	0.782
OH	0.952	0.964	0.670	0.562
OK	0.857	0.883	0.733	0.637
OR	0.841	0.824	0.624	0.704
PA	0.965	0.964	0.649	0.585
RI	0.944	0.907	0.714	0.958
SC	0.949	0.953	0.848	0.578
SD	0.574	0.649	0.794	0.514
TN	0.942	0.955	0.824	0.580
TX	0.991	0.997	0.795	0.617
UT	0.878	0.883	0.846	0.505
VT	0.242	0.180	0.104	0.311
WA	0.918	0.899	0.595	0.717
WI	0.826	0.824	0.524	0.547
WV	0.709	0.717	0.559	0.642
WY	0.504	0.492	0.629	0.648

There really does not appear to be a trend with network level density at the probability of donor groups to donate to Republican or Democratic candidates. In some states, the network level density does change based on the polarization of donor groups. For example, it Missouri, lower levels of polarization have higher

mean densities and higher levels of polarization have lower mean densities. In contrast, Montana's mean density levels are fairly constant across donor group polarization. In some states, it depends on whether the polarization is based from Democrats or Republicans. For instance, in New York, density doesn't change much except when donor groups that were donating candidates from both parties start donating to only Republican candidates. A similar pattern exists in Illinois.

Degree Centrality:

```
##Combine centrality measures
label <- paste0(substr(y,1,2))
table <- data.frame(label,round(as.numeric(unlist(mean_cd1)),3),round(as.numeric(unlist(mean_cd2)),3),
col_names <- c("State","Scenario 1","Scenario 2","Scenario 3","Scenario 4"))
```

Below is a table showing the values for different centrality measures by state for election year 2016:

```
table %>%
mutate_all(linebreak) %>%
kable(format="latex",escape=F,row.names=FALSE,col.names=linebreak(col_names), caption="Network Level Degree Centrality")
kable_styling(latex_options = c("striped", "hold_position"))
```

Once again, there's not a systematic pattern with degree centrality across the four scenarios. However, it seems like it rarely matters whether the polarization is from donor groups donating to Republican or Democratic candidates. There are some states like South Carolina and Arizona where scenario 5 has higher degree centrality, but overall degree centrality is similar in scenarios 1 and 2 and similar in scenarios 3 and 4.

Eigenvector Centrality:

```
##Combine centrality measures
label <- paste0(substr(y,1,2))
table <- data.frame(label,round(as.numeric(unlist(mean_ce1)),3),round(as.numeric(unlist(mean_ce2)),3),
col_names <- c("State","Scenario 1","Scenario 2","Scenario 3","Scenario 4"))
```

Below is a table showing the values for different centrality measures by state for election year 2016:

```
table %>%
mutate_all(linebreak) %>%
kable(format="latex",escape=F,row.names=FALSE,col.names=linebreak(col_names), caption="Network Level Eigenvector Centrality")
kable_styling(latex_options = c("striped", "hold_position"))
```

Eigenvector centrality measures are different for most states between scenarios 3 and 4. It seems that more liberal states have higher eigenvector centrality in scenario 3 and more conservative states have higher eigenvector centrality in scenario 4. Scenario 3 is when donor groups donating to candidates from both parties change to donating to only candidates from the Republican party and scenario 4 is the opposite. This might indicate that central donor groups are connected to other central donor groups because the donor groups are becoming more polarized so there are two clusters-one Republican and one Democratic.

Conclusion

Overall, there are some patterns that arise with the centrality measures across the different scenarios where the number of candidates by party varies and the probability of donor groups donating to candidates from each party varies. However, these patterns are not systematic, which indicates that there are other varying aspects of the networks that would change the density of the network systematically. While varying the number of candidates and the probability of donor groups donating to certain candidates does change centrality measures, the direction of change in centrality and the heterogeneity of the change across scenarios indicates that there are aspects of the network that are interacting with the scenarios and centrality measures that are not measured in this project. For example, it could be the number of donor groups, the overall number of candidates, the sparsity of the network, the partisanship of the candidates, triads in the network, etc.

Table 6: Network Level Degree Centrality for All Four Scenarios

State	Scenario 1	Scenario 2	Scenario 3	Scenario 4
AK	0.371	0.390	0.115	0.129
AR	0.219	0.165	0.095	0.124
AZ	0.034	0.022	0.050	0.115
CA	0.000	0.002	0.067	0.074
CO	0.100	0.087	0.117	0.113
CT	0.002	0.004	0.127	0.000
DE	0.369	0.376	0.093	0.113
FL	0.006	0.007	0.060	0.001
GA	0.047	0.036	0.071	0.032
HI	0.127	0.144	0.090	0.068
IA	0.182	0.171	0.120	0.113
ID	0.205	0.184	0.088	0.017
IL	0.023	0.023	0.066	0.081
IN	0.092	0.089	0.120	0.111
KS	0.156	0.134	0.095	0.125
KY	0.139	0.137	0.103	0.118
MA	0.066	0.103	0.126	0.074
ME	0.247	0.237	0.105	0.081
MI	0.047	0.039	0.125	0.125
MN	0.156	0.156	0.047	0.112
MO	0.120	0.110	0.118	0.087
MT	0.418	0.393	0.124	0.106
NC	0.042	0.062	0.083	0.036
ND	0.367	0.352	0.122	0.127
NH	0.372	0.341	0.109	0.088
NM	0.249	0.246	0.125	0.124
NV	0.142	0.144	0.125	0.112
NY	0.014	0.023	0.005	0.094
OH	0.048	0.036	0.122	0.115
OK	0.143	0.117	0.109	0.125
OR	0.157	0.175	0.126	0.116
PA	0.035	0.036	0.124	0.121
RI	0.048	0.072	0.115	0.021
SC	0.051	0.047	0.069	0.120
SD	0.306	0.271	0.090	0.078
TN	0.058	0.045	0.079	0.121
TX	0.009	0.003	0.089	0.125
UT	0.122	0.116	0.070	0.053
VT	0.243	0.237	0.211	0.242
WA	0.075	0.095	0.127	0.113
WI	0.174	0.176	0.089	0.108
WV	0.288	0.280	0.114	0.125
WY	0.383	0.380	0.134	0.132

This project used campaign donor group networks to run scenarios varying both the composition of the candidates running for House election and the probability of a donor group donating to a Republican or Democratic candidate. The probability measures came from the percentage of donations a donor group gave to Republican and Democratic candidates. This is a rough approximation for the probability of a donor group donating to specific candidates. Currently, the composition of candidates running for election are of

Table 7: Network Level Eigenvector Centrality for All Four Scenarios

State	Scenario 1	Scenario 2	Scenario 3	Scenario 4
AK	0.435	0.432	0.156	0.311
AR	0.172	0.130	0.127	0.280
AZ	0.029	0.017	0.057	0.329
CA	0.000	0.002	0.421	0.090
CO	0.086	0.078	0.319	0.174
CT	0.002	0.004	0.222	0.000
DE	0.352	0.379	0.381	0.173
FL	0.006	0.005	0.070	0.500
GA	0.041	0.026	0.086	0.467
HI	0.083	0.106	0.400	0.080
IA	0.162	0.151	0.303	0.172
ID	0.158	0.128	0.114	0.486
IL	0.018	0.021	0.422	0.102
IN	0.082	0.073	0.198	0.335
KS	0.131	0.108	0.127	0.249
KY	0.115	0.122	0.361	0.191
MA	0.041	0.077	0.273	0.089
ME	0.197	0.173	0.151	0.415
MI	0.041	0.034	0.243	0.234
MN	0.127	0.135	0.451	0.333
MO	0.105	0.087	0.192	0.388
MT	0.436	0.378	0.295	0.359
NC	0.037	0.046	0.105	0.464
ND	0.380	0.360	0.208	0.271
NH	0.353	0.316	0.356	0.407
NM	0.223	0.213	0.259	0.275
NV	0.124	0.130	0.247	0.169
NY	0.010	0.022	0.496	0.125
OH	0.043	0.030	0.209	0.324
OK	0.123	0.096	0.160	0.239
OR	0.138	0.156	0.251	0.181
PA	0.031	0.031	0.227	0.294
RI	0.030	0.059	0.173	0.022
SC	0.045	0.039	0.083	0.303
SD	0.284	0.222	0.118	0.412
TN	0.051	0.036	0.098	0.301
TX	0.008	0.003	0.116	0.258
UT	0.095	0.081	0.084	0.444
VT	0.526	0.596	0.700	0.457
WA	0.067	0.085	0.278	0.169
WI	0.149	0.150	0.388	0.347
WV	0.258	0.251	0.329	0.234
WY	0.386	0.383	0.241	0.224

equal quality except for the difference between party affiliation. If this project were to be extended, it would benefit from having a better measure for the probability of donating to candidates that better approximate the actual decision making processes of donor groups. Further, this project does not account for the amount of donations given to candidates. This would likely impact the network structure of donor groups and would give greater weights to donor groups that donate more meaning than changing the probability of donating to

candidates would matter differently by donor group depending on the amount of the potential donation. This is a shortcoming of the analysis conducted above.

Citations:

HANSFORD, T., & GOMEZ, B. (2010). Estimating the Electoral Effects of Voter Turnout. *The American Political Science Review*, 104(2), 268-288. Retrieved from www.jstor.org/stable/40863720

MIT Election Data and Science Lab, 2017, “codebook-us-house-1976–2018.md”, U.S. House 1976–2018, <https://doi.org/10.7910/DVN/IGOUN2/XEGE3X>, Harvard Dataverse, V5

Neal, Zachary. 2014. “The Backbone of Bipartite Projections: Inferring Relationships from Co-Authorship, Co-Sponsorship, Co-Attendance and other Co-Behaviors.” *Social Networks* 39:84–97.

Reuning, Kevin, 2019, “Election Donation Networks”, <https://doi.org/10.7910/DVN/YMDFPW>, Harvard Dataverse, V1. The original data was in the form of a bipartite network where candidates and donating groups were the modes-CxG matrix.