Assignment 4: U.S. Senators on Twitter

Xuejing Li

2021-04-08

Network Analysis of U.S. Senate Tweets

Overview

Twitter is a great tool to analyze the public interactions of political actors. For this assignment, I want you to use the information about who follows whom on Twitter as well as past tweets of the current U.S. Senate members to analyze how they interact and what they tweet about.

Data

Twitter Handles of Senators

Twitter does not allow us to search for past tweets (beyond about a week back) based on keywords, location, or topics (hashtags). However, we are able to obtain the past tweets of users if we specify their Twitter handle. The file `senators_twitter.csv` contains the Twitter handles of the current U.S. Senate members (obtained from UCSD library). We will focus on the Senators' *official Twitter accounts* (as opposed to campaign or staff members). The data also contains information on the party affiliation of the Senators.

Followers

The file `senators_follow.csv` contains an edge list of connections between each pair of senators who are connected through a follower relationship (this information was obtained using the function `rtweet::lookup_friendships`). The file is encoded such that the `source` is a follower of the `target`. You will need to use the subset of `following = TRUE` to identify the connections for which the `source` follows the `target`.

Tweets by Senators

To make your life a bit easier, I have also already downloaded all available tweets for these Twitter accounts using the following code. You **do not need to repeat this step**. Simply rely on the file `senator_tweets.RDS` in the exercise folder.

```
## Run / Install before executing slides

# Load packages.
packages <- c("devtools","knitr","widgetframe","readr","igraph","svglite",
              "ggnetwork","GGally","network","sna","ggplot2",
              "svglite","rsvg","tidyverse",
              "ggraph","igraph","tidygraph",
              "gganimate","randomNames","threejs","visNetwork",
              "ergm","tweenr","rtweet","twitteR", "kableExtra",
              "ggthemes","DT")

packages <- lapply(packages, FUN = function(x) {
  if(!require(x, character.only = TRUE)) {
    install.packages(x)
  library(x, character.only = TRUE)
  }
}
)
library(RColorBrewer)
library(wesanderson)
```

```
library(tidyverse)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:igraph':
##
##     %--%, union
```

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

```
#install.packages("rtweet", dependencies=TRUE)
library(rtweet)
library(ggrepel)

# Read in the Senator Data
senate <- read_csv("senators_twitter.csv")
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   senator = col_character(),
##   twitter_handle = col_character(),
##   state = col_character(),
##   party = col_character()
## )
```

```
# Read in the Tweets
senator_tweets <- readRDS("senator_tweets.RDS")
```

The data contains about 280k tweets and about 90 variables. Please note, that the API limit of 3,200 tweets per twitter handle actually cuts down the time period we can observe the most prolific Twitter users in the Senate down to only about one year into the past.

Tasks for the Assignment

1. Who follows whom?

a) Network of Followers

Read in the edgelist of follower relationships from the file `senators_follow.csv`. Create a directed network graph. Identify the three senators who are followed by the most of their colleagues (i.e. the highest “in-degree”) and the three senators who follow the most of their colleagues (i.e. the highest “out-degree”). [Hint: You can get this information simply from the data frame or use `igraph` to calculate the number of in and out connections: `indegree = igraph::degree(g, mode = "in")`.] Visualize the network of senators. In the visualization, highlight the party ID of the senator nodes with an appropriate color (blue = Democrat, red = Republican) and size the nodes by the centrality of the nodes to the network. Briefly comment.

```
# Read in the Senator Data
follow <- read_csv("senators_follow.csv") %>%
  filter(following=="TRUE") %>%
  select(source, target)
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   source = col_character(),
##   target = col_character(),
##   following = col_logical(),
##   followed_by = col_logical()
## )
```

```
senate_attrs <- senate %>%
  select(twitter_handle,state,party) %>%
  filter(twitter_handle != "senatemajldr")

f <- graph_from_data_frame(d = follow,
                           vertices = senate_attrs, #attach attributes to vertices
                           directed = TRUE)
V(f)$Followed = igraph::degree(f, mode="in")
V(f)$Following = igraph::degree(f, mode="out")
V(f)$Betweenness = igraph::betweenness(f, directed=TRUE)
V(f)$Color = ifelse(f$party=="D","steelblue", "darkred") #color political party into the graph by assiçgning them to a vertex attribute
```

```
## Warning in length(vattrs[[name]]) <- vc: length of NULL cannot be changed
```

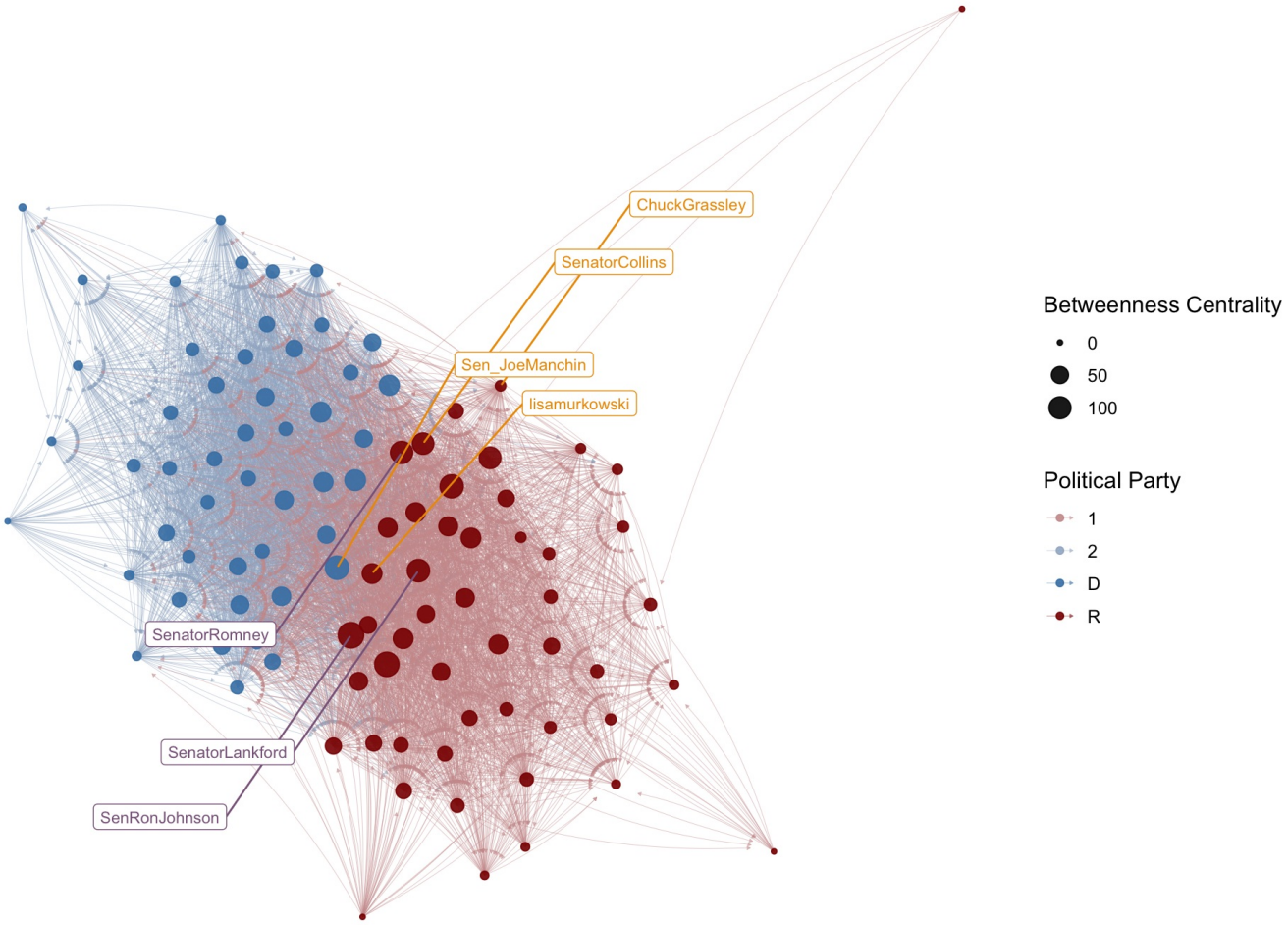
```
#vertex_attr(f)
```

```
f <- igraph::simplify(f, edge.attr.comb = "sum") #simplify igraph

#Convert igraph object to dataframe
dat <- ggnetwork(f,
          layout=igraph::with_fr()) %>%
  filter(party != "I") %>%
  mutate(party_edge = factor(ifelse(party == "R", "1", "2")))

#filter for top followed/following senators
top_followed <- dat %>%
  arrange(desc(Followed)) %>%
  distinct(name, .keep_all = TRUE) %>%
  filter(Followed >= 91)

top_following <- dat %>%
  arrange(desc(Following)) %>%
  distinct(name, .keep_all = TRUE) %>%
  filter(Following >= 76)

#plot dataframe into network graph using ggplot
gg1 <- ggplot() +
  geom_edges(data=dat,
             aes(x=x, y=y, xend=xend, yend=yend, color=party_edge),
             arrow = arrow(length = unit(2, "pt"), type = "closed"),
             curvature=0.1, size=0.15, alpha=0.5) +
  geom_nodes(data=dat,
             aes(x=x, y=y, xend=xend, yend=yend, color=party, size=Betweenness),alpha=0.9) +
  geom_label_repel(data=top_followed, max.overlaps = 100,
                   aes(x=x, y=y, label=name),
                   nudge_x=-0.2,
                   nudge_y=-0.2,
                   size=3, color="plum4") +
  geom_label_repel(data=top_following, max.overlaps = 100,
                   aes(x=x, y=y, label=name),
                   nudge_x=0.2,
                   nudge_y=0.2,
                   size=3, color="orange2")+
  scale_color_manual(values=c("rosybrown3","lightsteelblue3","steelblue","darkred"))+
  theme_blank()+
  theme(
  plot.title = element_text(color="#4D4D4D", size=14, face="bold"),
  plot.caption = element_text(color="#898989", face="italic", size=8))+
  labs(title="Network Followers",
       caption = "Source: twitter",
       size='Betweenness Centrality',
       color='Political Party')
```

```
## Warning: Ignoring unknown aesthetics: xend, yend
```

```
gg1
```

Network Followers



Source: twitter

Names in purple are the top three senators who are followed the most by other senators. Names in orange are the four senators who are following the most of other senators. The size of the nodes showcases betweenness centrality which measures centrality based on the shortest paths. It represents how many times the node stands between the shortest path from any two nodes in the network. Higher betweenness centrality means the senator potentially have more power on twitter, because more colleagues will pass through that senator to get to others.

187 vertices

b) Communities

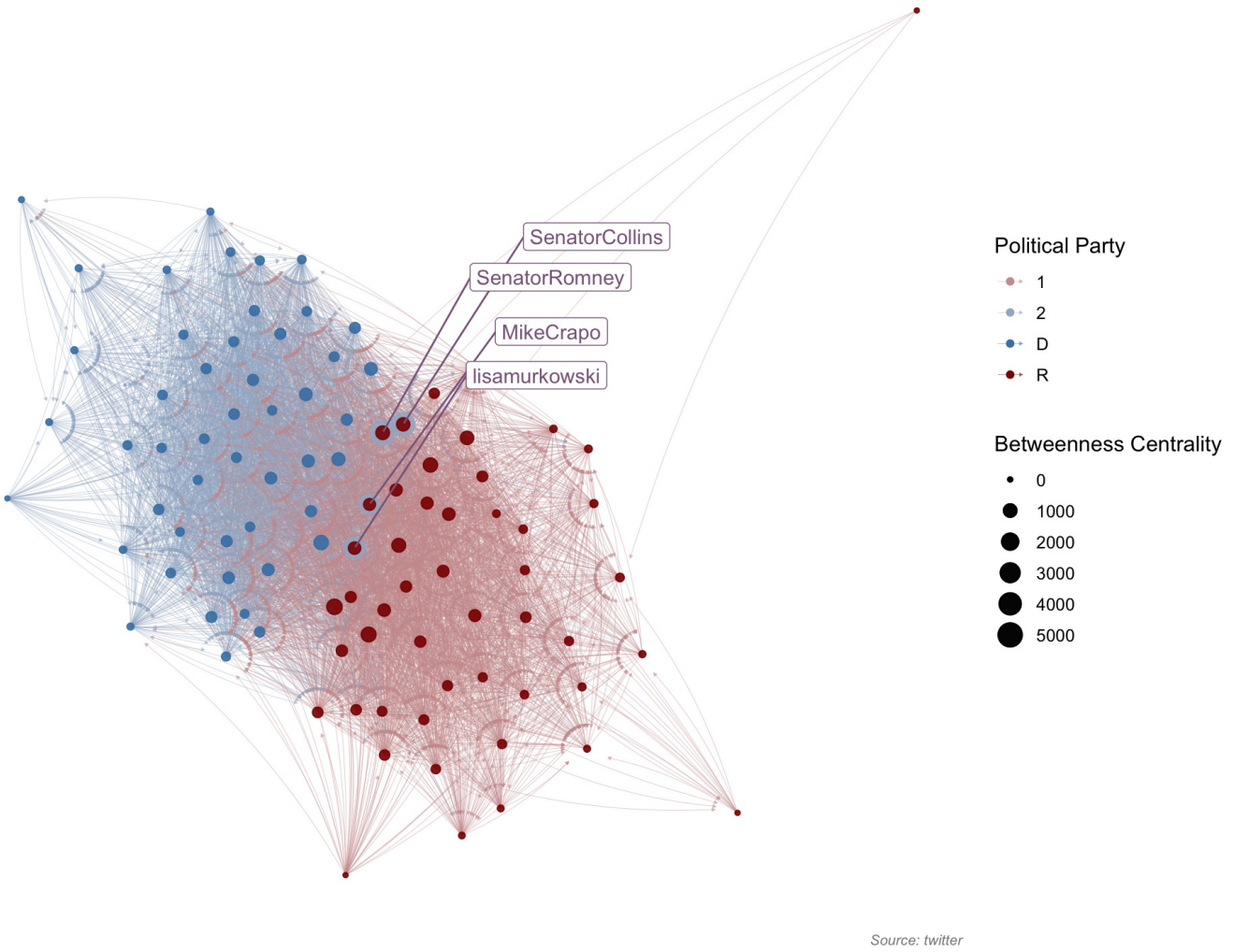
Now let's see whether party identification is also recovered by an automated mechanism of cluster identification. Use the `cluster_walktrap` command in the `igraph` package to find densely connected subgraphs. Based on the results, visualize how well this automated community detection mechanism recovers the party affiliation of senators. This visualization need not be a network graph. Comment briefly.

```
wc <- cluster_walktrap(f)  # find "communities"
com <- cbind(V(f)$name,membership(wc))

#convert list to dataframe
com_df <- data.frame(matrix(unlist(com), nrow=99),stringsAsFactors=FALSE) %>%
  rename(twitter_handle=X1, group=X2)

member_automate <- dat %>%
  left_join(com_df, by=c("name"="twitter_handle"))

#find senators grouped wrong by the automated mechanism of cluster identification
wrong <- member_automate %>%
  distinct(name, .keep_all = TRUE) %>%
  filter(party=="R" & group==2)

gg2 <- ggplot() +
  geom_edges(data=member_automate,
             aes(x=x, y=y, xend=xend, yend=yend, color=group),
             arrow = arrow(length = unit(2, "pt"), type = "closed"),
             curvature=0.1, size=0.15, alpha=1/2) +
  geom_nodes(data=wrong,
             aes(x=x, y=y, xend=xend, yend=yend, color=group, size=Betweenness*50),alpha=0.8)+
  geom_nodes(data=member_automate,
             aes(x=x, y=y, xend=xend, yend=yend, color=party, size=Betweenness*10),alpha=0.9)+
  geom_label_repel(data=wrong, max.overlaps = 100,
             aes(x=x, y=y, label=name),
             nudge_x=0.2,
             nudge_y=0.2,
             size=4, color="plum4")+
  scale_color_manual(values=c("rosybrown3","lightsteelblue3","steelblue","darkred"))+
  theme_blank()+
  theme(
  plot.title = element_text(color="#4D4D4D", size=14, face="bold"),
  plot.caption = element_text(color="#898989", face="italic", size=8))+
  labs(title="Automated Community Detection",
       caption = "Source: twitter",
       size='Betweenness Centrality',
       color='Political Party')
```

```
## Warning: Ignoring unknown aesthetics: xend, yend

## Warning: Ignoring unknown aesthetics: xend, yend
```

```
gg2
```

Automated Community Detection



I plotted a similar network graph where different colors of the nodes and the edges represents different political parties. Moreover, I highlighted the four senators that are misclassified by the automated mechanism of cluster identification. For these four nodes, the larger circle represents the party they are being misclassified into, and the smaller circle represents the party they are in.

Comparing the two network graphs side by side tells us the automated cluster identification mechanism does a good job of recovery the senators'party affliations. Out of the four republican senators being misclassified, three of them either have the most number of senator followers or is following the most of other senators. Therefore, the mistakes the mechanism made isn't completely unexpected. It's more likely to misclassify nodes with a high degree of centrality.

2. What are they tweeting about?

From now on, rely on the information from the tweets stored in `senator_tweets.RDS`.

a) Most Common Topics over Time

Remove all tweets that are re-tweets (`is_retweet`) and identify which topics the senators tweet about. Rather than a full text analysis, just use the variable `hashtags` and identify the most common hashtags over time. Provide a visual summary.

```
#TOPICS
hashtag <- unnest(senator_tweets, hashtags) %>%
  filter(is_retweet==FALSE) %>%
  mutate(hashtags=tolower(hashtags)) %>%
  select(created_at, hashtags) %>%
  filter(!is.na(hashtags)) %>%
  mutate(Date=as.Date(created_at, format = "%Y-%m-%d")) %>%
  mutate(year= format(Date, format="%Y")) %>% #2009-2021
  select(hashtags, year) %>%
  group_by(hashtags, year) %>%
  summarise(Frequency= n()) %>%
  arrange(desc(year), desc(Frequency)) %>%
  filter(Frequency > 10 & year %in% c("2016","2017","2018","2019","2020","2021") & hashtags %in% c("covid19
","coronavirus")==FALSE)
```

```
## `summarise()` has grouped output by 'hashtags'. You can override using the `.groups` argument.
```

```
# #install.packages("ggwordcloud", dependencies=TRUE)
library(ggwordcloud)
library(viridis)
```

```
## Loading required package: viridisLite
```

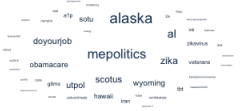
```
set.seed(42)
hashtag_year <- ggplot(hashtag, aes(label = hashtags, size = Frequency, color=Frequency)) +
  geom_text_wordcloud_area(rm_outside = TRUE) +
  scale_size_area(max_size = 8) +
  scale_fill_viridis(discrete=FALSE, direction = -1, option="inferno")+
  facet_wrap(~year, ncol=2) +
  theme_minimal()+
  theme(
  plot.title = element_text(color="#4D4D4D", size=20, face="bold"),
  plot.caption = element_text(color="#898989", face="italic", size=12),
  strip.text = element_text(size=20))+
  labs(title="Most Common Tags Over Time",
       caption = "Source: twitter")

hashtag_year
```

```
## Some words could not fit on page. They have been removed.
```

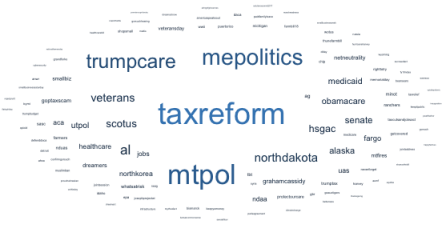
```
## Some words could not fit on page. They have been removed.
```

Most Common Tags Over Time

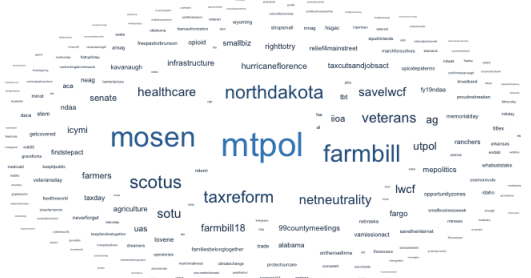
2016



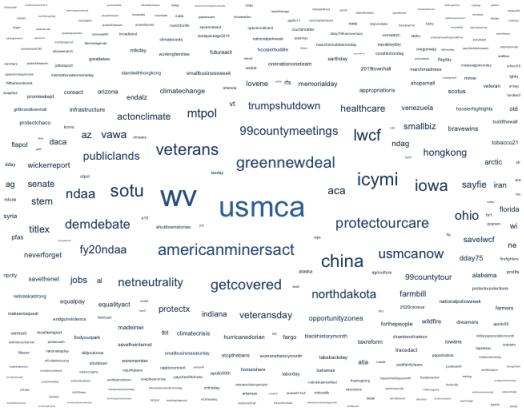
2017



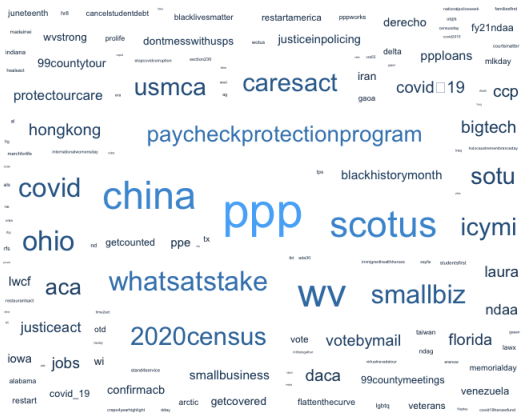
2018



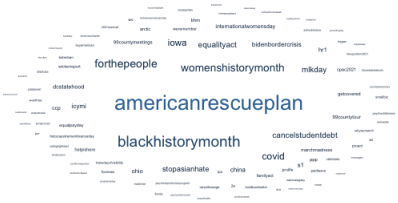
2019



2020



2021



Source: twitter

I calculated the most frequent hashtags each year and created a facet word cloud where the side of the words is the frequency the hashtags have been used. To create a more balanced visualization, I excluded the tag “covid19” and “coronavirous”, because the frequency is simply too high.

b) BONUS ONLY: Election Fraud 2020 - Dems vs. Reps

One topic that did receive substantial attention in the recent past the issue whether the [2020 presidential election involved fraud] and should be overturned. The resulting far-right and conservative campaign to *Stop the Steal* promoted the conspiracy theory that falsely posited that widespread electoral fraud occurred during the 2020 presidential election to deny incumbent President Donald Trump victory over former vice president Joe Biden.

Try to identify a set of 5-10 hashtags that signal support for the movement (e.g. #voterfraud, #stopthesteal, #holdtheline, #trumpwon, #voterid) while other expressed a critical sentiment towards the protest (e.g. #trumplost).

Sites like hashtagify.me or ritetag.com can help with that task. Using the subset of senator tweets that included these hashtags you identified, show whether and how senators from different parties talk differently about the issue of the 2020 election outcome.

3. Are you talking to me?

Often tweets are simply public statements without addressing a specific audience. However, it is possible to interact with a specific person by adding them as a friend, becoming their follower, re-tweeting their messages, and/or mentioning them in a tweet using the `@` symbol.

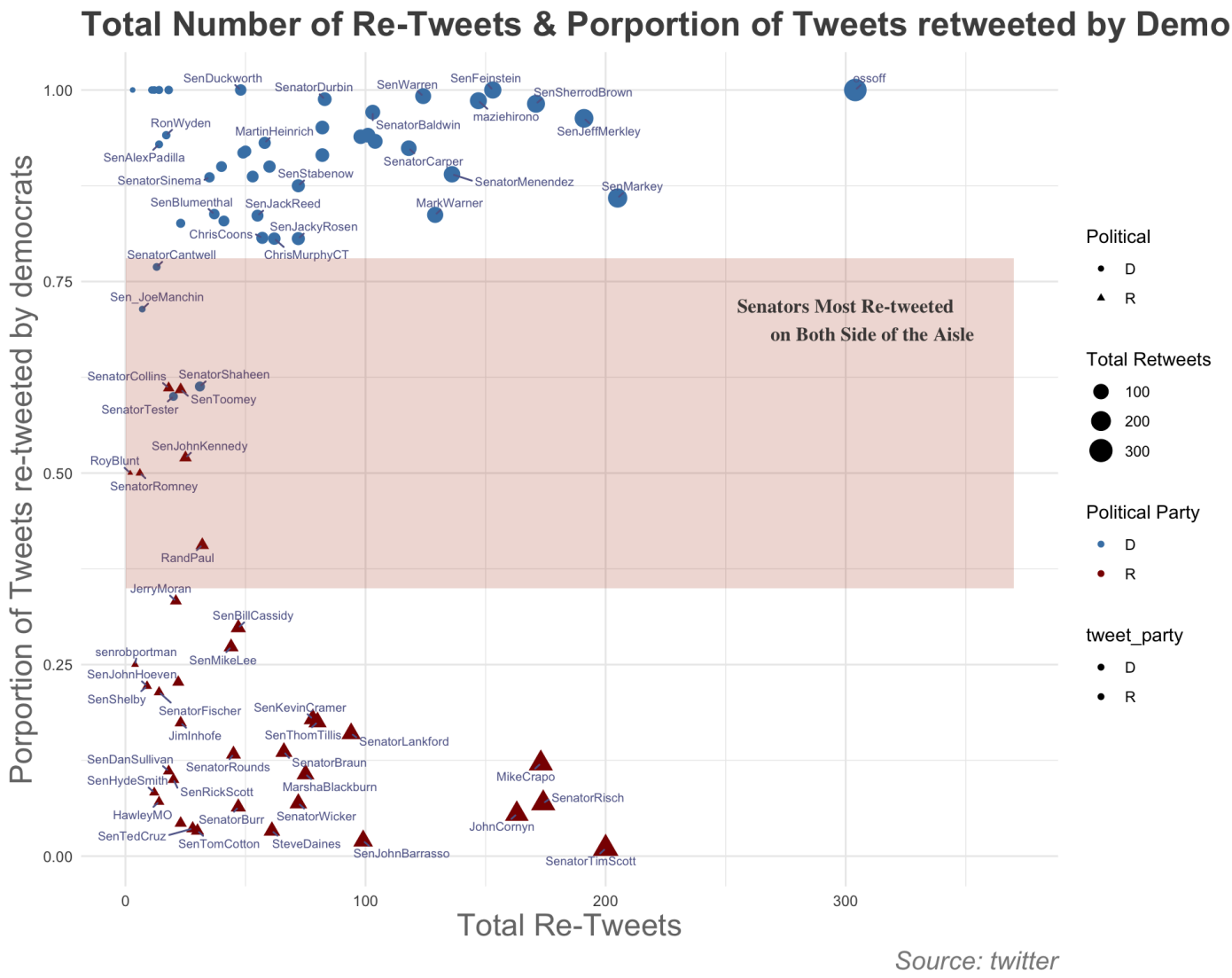
a) Identifying Re-Tweets

Select the set of re-tweeted messages from other senators and identify the source of the originating message. Calculate by senator the amount of re-tweets they received and from which party these re-tweets came. Essentially, I would like to visualize whether senators largely re-tweet their own party colleagues' messages or whether there are some senators that get re-tweeted on both sides of the aisle. Visualize the result and comment briefly.

```
#Identify Retweets
retweets <- senator_tweets %>%
  filter(is_retweet==TRUE) %>%
  select(screen_name, retweet_screen_name) %>%
  left_join(senate, by=c("screen_name"="twitter_handle")) %>%
  left_join(senate, by=c("retweet_screen_name"="twitter_handle")) %>%
  select(screen_name, retweet_screen_name, party.x, party.y) %>%
  rename(tweet_party = party.x, retweet_party =party.y) %>%
  filter(!is.na(retweet_party)) %>%
  group_by(screen_name, tweet_party, retweet_party) %>%  #groupby original senator and the party of thethe s
enator that retweeted their tweets
  summarize(total=n()) %>%
  ungroup(tweet_party, retweet_party) %>%
  mutate(D.prop= round(total/sum(total),3), retweet.total=sum(total)) %>%
  filter(retweet_party=="D" & tweet_party != "I")
```

```
## `summarise()` has grouped output by 'screen_name', 'tweet_party'. You can override using the `.groups` ar
gument.
```

```
#Visualize the amound of retweets every senator received & percentage of the retweets by democrats
Retweets_party <-ggplot(retweets, aes(x=retweet.total, y=D.prop, size=retweet.total,
                      fill=tweet_party)) +
  geom_point(aes(shape=tweet_party, color=tweet_party), stroke=0.3)+
  scale_color_manual(values=c("steelblue", "darkred"))+
  geom_text_repel(label=retweets$screen_name, color="#5F6697", size=2.5,
                  max.overlaps=7,
                  min.segment.length=0,
                  boc.padding=0.5,
                  segment.inflect=TRUE)+
  annotate("rect", xmin=0, xmax=370,
           ymin=0.35, ymax=0.78,#add shaded rect area to highlight senators who get re-tweeted by both parti
es
           alpha=0.2, fill="#B23D1A")+
  annotate(geom="text", x=300, y=0.70, family="serif", size=4,
           fontface="bold",
           color="#494645",
           label="Senators Most Re-tweeted
           on Both Side of the Aisle")+
  theme_minimal()+
  theme(
  plot.title = element_text(color="#4D4D4D", size=20, face="bold"),
  plot.caption = element_text(color="#898989", face="italic", size=15),
  axis.title.x = element_text(color="#767676", size=18),
  axis.title.y = element_text(color="#767676", size=18)
  )+
  labs(title="Total Number of Re-Tweets & Porportion of Tweets retweeted by Democrats", y="Porportion of Twe
ets re-tweeted by democrats",
       x="Total Re-Tweets",caption = "Source: twitter",
       size="Total Retweets",
       color="Political Party",
       shape="Political")

Retweets_party
```



I created a scatter plot where the colors and the shapes represent the political parties. The size of each point represent how many times their tweets are re-tweeted. This graph not only showcases the total number of re-tweets each senator received, but it also presents the proportion of tweets re-tweeted by senators of either party. In addition, by highlighting the area in the middle, the reader is able to locate the senators that get re-tweeted on both side of the aisle.

b) Identifying Mentions

Identify the tweets in which one senator mentions another senator directly (the variable is `mentions_screen_name`). For this example, please remove simple re-tweets (`is_retweet == FALSE`). Calculate who re-tweets whom among the senate members. Convert the information to an undirected graph object in which the number of mentions is the strength of the relationship between senators. Visualize the network graph using the party identification of the senators as a group variable (use blue for Democrats and red for Republicans) and some graph centrality measure to size the nodes. Comment on what you can see from the visualization.

```
#create mentions edge list
mentions <-unnest(senator_tweets, mentions_screen_name) %>%
  filter(is_retweet==FALSE) %>%
  select(screen_name, mentions_screen_name) %>%
  inner_join(senate, by=c("mentions_screen_name"="twitter_handle")) %>%
  group_by(screen_name, mentions_screen_name, party) %>%
  summarize(mentions=n()) %>%
  left_join(senate, by=c("screen_name"="twitter_handle")) %>%
  select(screen_name, mentions_screen_name, mentions) %>%
  rename(weight=mentions)
```

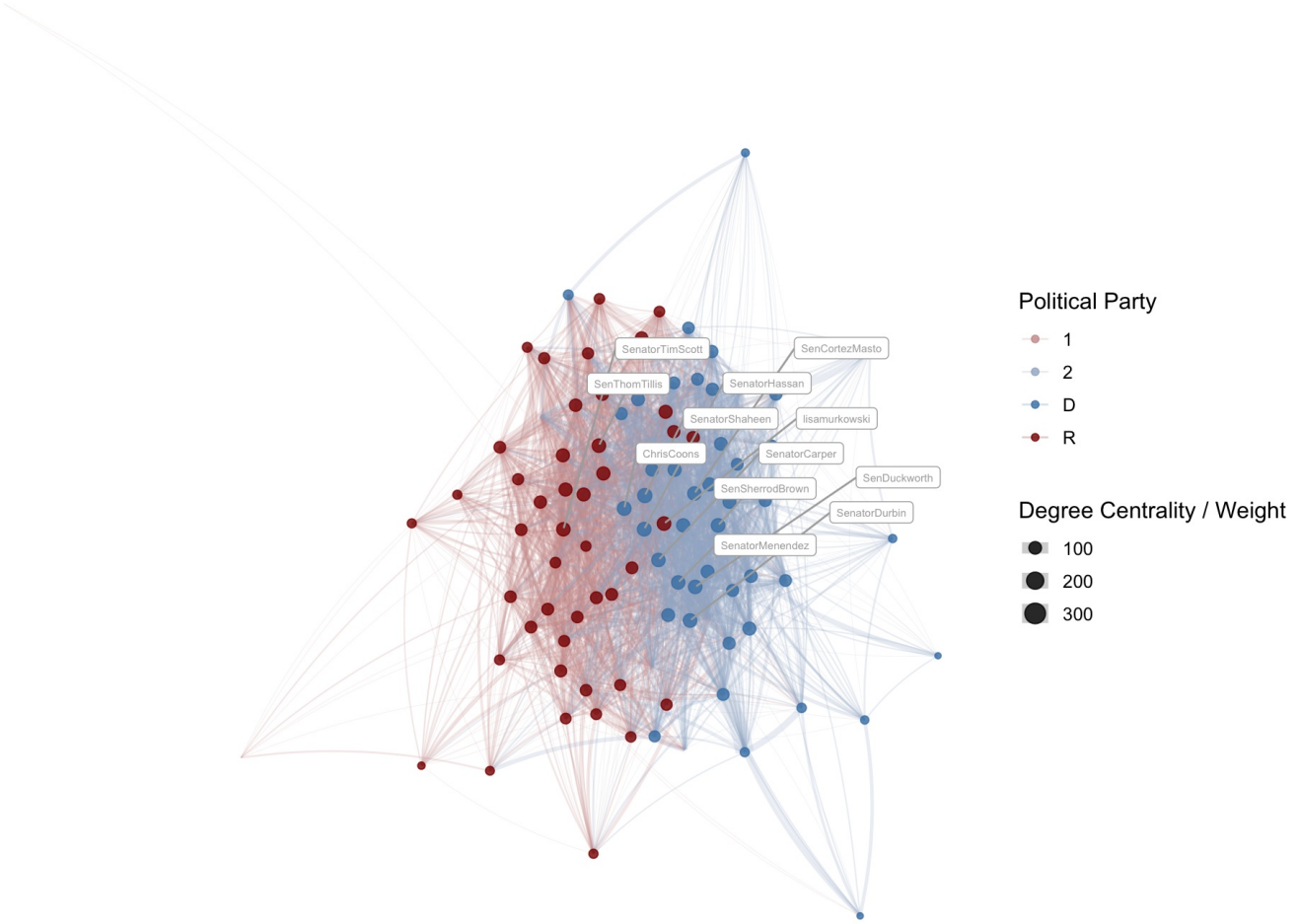
```
## `summarise()` has grouped output by 'screen_name', 'mentions_screen_name'. You can override using the `.g
roups` argument.
```

```
#Convert dataframe to graph object
mentions_graph <- graph_from_data_frame(mentions, directed=FALSE, vertices=senate_attrs)

#Calculate the degree ventrality of each vertex
V(mentions_graph)$degree = igraph::degree(mentions_graph, mode="all")

#Check if network graph is weighted
#is_weighted(mentions_graph)

mentions_simp <- igraph::simplify(mentions_graph, edge.attr.comb="sum")

#graph to dataframe
simp <- ggnetwork(mentions_simp,
                  layout=igraph::with_fr()) %>%
  filter(party != "I" & !is.na(weight)) %>%  #Clean the dataframe
  mutate(party_edge = factor(ifelse(party == "R", "1", "2")))

#filter for top followed/following senators
power <- simp %>%
  arrange(desc(degree)) %>%
  distinct(name, .keep_all = TRUE) %>%
  filter(degree >= 110)

gg3 <- ggplot() +
  geom_edges(data=simp,
             aes(x=x, y=y, xend=xend, yend=yend, color=party_edge, size=weight),
             curvature=0.1, alpha=0.2) +
  geom_nodes(data=simp,
             aes(x=x, y=y, xend=xend, yend=yend, color=party, size=degree),alpha=0.8)+
  scale_color_manual(values=c("rosybrown3","lightsteelblue3","steelblue","darkred"))+
  scale_size_continuous(range=c(0.1,5))+
  geom_label_repel(data=power, max.overlaps =50,
             aes(x=x, y=y, label=name),
             nudge_x=0.1,
             nudge_y=0.1,
             size=2, color="#ACACAC")+
  theme_blank()+
  theme(
  plot.title = element_text(color="#4D4D4D", size=14, face="bold"),
  plot.caption = element_text(color="#898989", face="italic", size=8))+
  labs(title="Mentions as Strength of Relationship",
       caption = "Source: twitter",
       size='Degree Centrality / Weight',
       color='Political Party',
       width='Density')
```

```
## Warning: Ignoring unknown aesthetics: xend, yend
```

```
gg3
```

Mentions as Strength of Relationship



Source: twitter

Similar to the graphs in the first question, the size of the node also showcases the degree centrality of each senator. The bigger the nodes, the more frequent that senator is mentioned or mentions other. Alternatively, one could interpret the degree centrality for the number of mentions as the how active or engaged the senators are on twitter. The listed names on the graph are senators with highest degree centrality, and therefore the most active senators on twitter.

Moreover, the width of the edge represents the number of times the two senators mentioned each other on the tweets. Depends on the scenarios, a more accentuated edge could either indicate the strength of political liasion or hostility. #### c) BONUS ONLY: Who is popular on Twitter?

Using the twitter handles, access the user information of the senators to identify the number of followers they have (obviously, this will require to actually connect to the Twitter server). Re-do the previous graph object but now use the number of followers (or some transformation of that info) to size the nodes. Comment how graph degree centrality (via mentions) and the number of followers are related.

Submission

Please follow the instructions to submit your homework. The homework is due on Thursday, April 8.

Please stay honest!

If you do come across something online that provides part of the analysis / code etc., please no wholesale copying of other ideas. We are trying to evaluate your abilities to visualized data not the ability to do internet searches. Also, this is an individually assigned exercise – please keep your solution to yourself.