

```
setwd("~/Documents/R/Networks")
library(rtweet)
```

```
## Warning: package 'rtweet' was built under R version 3.5.2
```

```
library(rtweet)
library(readr)
library(tidyverse)
```

```
## — Attaching packages — tidyverse 1.2.1 —
```

```
## ✓ ggplot2 3.0.0      ✓ purrr 0.2.5
## ✓ tibble 1.4.2       ✓ dplyr 0.7.6
## ✓ tidyr 0.8.1        ✓ stringr 1.3.1
## ✓ ggplot2 3.0.0      ✓ forcats 0.3.0
```

```
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ purrr::flatten() masks rtweet::flatten()
## ✖ dplyr::lag() masks stats::lag()
```

```
load("~/Documents/R/Networks/sen_friends.RData")
load("~/Documents/R/Networks/rep_friends.RData")
legs <- read_csv("legislators-current.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   birthday = col_date(format = ""),
##   district = col_integer(),
##   senate_class = col_integer(),
##   cspan_id = col_integer(),
##   govtrack_id = col_integer(),
##   votesmart_id = col_integer(),
##   icpsr_id = col_integer()
## )
```

```
## See spec(...) for full column specifications.
```

```
legs <- legs %>% filter(type=='sen'|type=='rep') %>% select(twitter,party,govtrack_id,type)
legs <- na.omit(legs)
sens <- legs %>% filter(type == 'sen') %>% select(twitter,party,govtrack_id)
reps <- legs %>% filter(type=='rep') %>% select(twitter,party,govtrack_id)
```

Make dataframe of followers/followed Each row is a uni-directional follow from “follower” to “followed”

```
senrep <- rbind(sens, reps)
senrep_friends <- rbind(sen_friends, rep_friends)
senrep_friends
```

```
## # A tibble: 55,688 x 2
##   follower      followed
##   <chr>         <chr>
## 1 SenSherrodBrown RepAGonzalez
## 2 SenSherrodBrown RepBalderson
## 3 SenSherrodBrown WarrenDavidson
## 4 SenSherrodBrown SenBillCassidy
## 5 SenSherrodBrown SenRickScott
## 6 SenSherrodBrown SenatorBraun
## 7 SenSherrodBrown SenHawleyPress
## 8 SenSherrodBrown SenMcSallyAZ
## 9 SenSherrodBrown MarshaBlackburn
## 10 SenSherrodBrown SenKevinCramer
## # ... with 55,678 more rows
```

Break into Republicans and Democrats (slightly more of the former than the latter)

```
republicans <- senrep$twitter[sens$party=='Republican']
length(republicans)
```

```
## [1] 278
```

```
democrats <- senrep$twitter[sens$party=='Democrat']
length(democrats)
```

```
## [1] 243
```

Make two graphs: one of only Democrats, one of only Republicans

```
r_follows <- senrep_friends %>% filter(follower %in% republicans & followed %in% republicans)
d_follows <- senrep_friends %>% filter(follower %in% democrats & followed %in% democrats)

library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.5.2
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##   as_data_frame, groups, union
```

```
## The following objects are masked from 'package:purrr':  
##  
##   compose, simplify
```

```
## The following object is masked from 'package:tidyr':  
##  
##   crossing
```

```
## The following object is masked from 'package:tibble':  
##  
##   as_data_frame
```

```
## The following objects are masked from 'package:stats':  
##  
##   decompose, spectrum
```

```
## The following object is masked from 'package:base':  
##  
##   union
```

```
g <- graph_from_data_frame(senrep_friends, directed=TRUE)  
d <- graph_from_data_frame(d_follows, directed=TRUE, vertices=democrats)  
r <- graph_from_data_frame(r_follows, directed=TRUE, vertices=republicans)
```

Examine the edge density of Republican, Democrat, and full network

```
edge_density(g)
```

```
## [1] 0.1699069
```

```
edge_density(d)
```

```
## [1] 0.1862225
```

```
edge_density(r)
```

```
## [1] 0.1788562
```

The Democrat network is more dense than the Republican, which are both more dense than the overall network
Similarly, transitivity is lower for the overall network, highest for Republican network

```
transitivity(g)
```

```
## [1] 0.5383434
```

```
transitivity(d)
```

```
## [1] 0.5744101
```

```
transitivity(r)
```

```
## [1] 0.5917161
```

Analyze which users are followed by the highest number of other legislators

```
in_degree <- degree(g,mode='in')
sort(in_degree,decreasing=TRUE)[1:10 ] #who is followed by the most people
```

```
## SpeakerPelosi      GOPLeader      LeaderHoyer      SenatorSinema
##           228           220           206           197
## SteveScalise      cathymcmorris      JohnCornyn      SenSchumer
##           186           180           177           177
## SenatorTimScott    senrobportman
##           173           172
```

```
r_follows_d <- senrep_friends %>% filter(follower %in% republicans & followed %in% democrats)
d_follows_r <- senrep_friends %>% filter(follower %in% democrats & followed %in% republicans)
nrow(r_follows_d)/length(republicans) #average number of out-of-party follows for republicans
```

```
## [1] 42.08273
```

```
nrow(d_follows_r)/length(democrats) #average number of out-of-party follows for democrats
```

```
## [1] 47.36214
```

```
nrow(r_follows)/length(republicans) #average number of within-party follows for republicans
```

```
## [1] 49.54317
```

```
nrow(d_follows)/length(democrats) #average number of within-party follows for democrats
```

```
## [1] 45.06584
```

Democrats have more out-of-party follows than Republicans, and Republicans have more within-party follows than Democrats

Load tweets about Impeachment

```
load("~/Documents/R/Networks/total_tweets_3.RData")
impeach <- str_detect(tweets$text, regex('impeach', ignore_case=TRUE))
impeach_tweets_full <- tweets %>% filter(impeach)
impeach_tweets <- impeach_tweets_full %>% select(status_id, screen_name, text)
```

Make dataframe of legislators and their tweets on impeachment

```
load("~/Documents/R/Networks/legislators.RData")

names(impeach_tweets) <- c("status_id", "twitter", "text")
impeach_tweets$twitter <- tolower(impeach_tweets$twitter)
impeachment <- impeach_tweets %>% left_join(legislators, by='twitter')
impeachment$status_id_binary <- ifelse(is.na(impeachment$status_id), 0, 1)
impeachment <- impeachment %>% group_by(twitter) %>% mutate(numtweets = sum(status_id_binary)) %>% dplyr::select(twitter, party, numtweets)
impeachment <- distinct(impeachment)
num_tweets <- impeachment %>% dplyr::select(twitter, numtweets)
legislators <- legislators %>% left_join(num_tweets, by='twitter')
legislators$numtweets <- ifelse(is.na(legislators$numtweets), 0, legislators$numtweets)

tweet_count = impeach_tweets_full %>% dplyr::select(created_at, screen_name)
tweet_count$screen_name <- tolower(tweet_count$screen_name)
tweet_count$created_at <- as.Date(tweet_count$created_at)
names(tweet_count) <- c("created_at", "twitter")
tweet_dates <- aggregate(tweet_count$created_at, list(tweet_count$twitter), paste, collapse="/")
names(tweet_dates) <- c("twitter", "dates")

legislators <- legislators %>% left_join(tweet_dates, by='twitter')
legislators
```

```
## # A tibble: 532 x 10
##   twitter party govtrack_id type  terms followed_by follows gender
##   <chr>    <fct>      <int> <chr> <int>      <dbl>    <dbl> <fct>
## 1 senshe... Demo...   400050 sen     10        144     113 M
## 2 senato... Demo...   300018 sen      5        122     116 F
## 3 senato... Demo...   400064 sen     13        131     131 M
## 4 senato... Demo...   300019 sen      9        119      94 M
## 5 senbob... Demo...   412246 sen      3        134     104 M
## 6 senfei... Demo...   300043 sen      6        161      68 F
## 7 senamy... Demo...   412242 sen      3         52      61 F
## 8 senato... Demo...   400272 sen     11        132      88 M
## 9 sensan... Inde...   400357 sen     11        153      71 M
## 10 sensta... Demo...   300093 sen      6        115      62 F
## # ... with 522 more rows, and 2 more variables: numtweets <dbl>,
## #   dates <chr>
```

Make network about impeachment discussion, and interactions with other legislators on twitter

```
df <- impeach_tweets_full
# all mentions, as string
mention_list <- vector("list", nrow(impeach_tweets_full))
for(i in 1:nrow(impeach_tweets_full)){
  mentions <- impeach_tweets_full$mentions_screen_name[i][[1]]
  mentions <- paste(mentions,collapse=',')[[1]]
  mention_list[[i]]<-ifelse(mentions=="NA",NA,mentions)
}
df$mentions <- mention_list
df <- transform(df, mentions = as.character(mentions))
df <- df %>% dplyr::select(screen_name,reply_to_screen_name,quoted_screen_name,retweet_s
screen_name,mentions,text,status_id,created_at)
# all interactions as string
df <- unite(df,interaction_name,2:5,sep=" ",remove=FALSE)
# all interactions, separate rows, edges only when interacting with other legislators
df <- separate_rows(df,interaction_name) %>% filter(tolower(interaction_name) %in% legis
lators$twitter)
df <- unique(df)
# remove self edges
df <- filter(df,interaction_name!=screen_name)

# tweet dates, interaction types
df$created_at<-as.Date(df$created_at)
df$interaction_type <- ifelse(is.na(df$reply_to_screen_name),ifelse(is.na(df$quoted_scre
en_name),ifelse(is.na(df$retweet_screen_name),'mention','retweet'),'quote'),'reply')

df$screen_name<-tolower(df$screen_name)
df$interaction_name<-tolower(df$interaction_name)
df <- df %>% dplyr::select(screen_name,interaction_name,text,status_id,interaction_type,
created_at)
df$created_at<-as.character(df$created_at)

ntwk_graph <- graph_from_data_frame(df, directed = TRUE, vertices=legislators)
ntwk_graph
```

```
## IGRAPH cf72c66 DN-B 532 3149 --
## + attr: name (v/c), party (v/c), govtrack_id (v/n), type (v/c),
## | terms (v/n), followed_by (v/n), follows (v/n), gender (v/c),
## | numtweets (v/n), dates (v/c), text (e/c), status_id (e/c),
## | interaction_type (e/c), created_at (e/c)
## + edges from cf72c66 (vertex names):
## [1] johncornyn->speakerpelosi johncornyn->repmccaul
## [3] johncornyn->repdancrenshaw johncornyn->stevedaines
## [5] johncornyn->speakerpelosi johncornyn->repdancrenshaw
## [7] johncornyn->chuckgrassley johncornyn->repkevinbrady
## [9] johncornyn->mariodb johncornyn->repleezeldin
## + ... omitted several edges
```

Create (and optionally plot) a network of interactions restricted to a time window, and return legislators with highest eigenvector centrality

```

timed_graph_windowed <- function(j,plot=FALSE) {

upperdate = all_dates[j]
lower_week = all_dates[j-7]

#node size is proportional to the number of tweets they have on impeachment
V(ntwk_graph)$size <- 0
for(i in 1:length(V(ntwk_graph))){
  tweet_dates <- V(ntwk_graph)[i]$dates
  tweet_dates_list <- strsplit(tweet_dates,"/")
  tweet_dates_list <- as.Date(tweet_dates_list[[1]])
  bool_dates_list <- tweet_dates_list<upperdate
  V(ntwk_graph)[i]$size <- sqrt(sum(bool_dates_list))*5
}

#tweets from the past week
week_tweets <- tweet_count[tweet_count$created_at<upperdate & tweet_count$created_at>lower_week,]
#members who tweeted this week
week_filter <- (V(ntwk_graph)$name %in% week_tweets$twitter)

#graph of vertices from this week
g <- induced_subgraph(ntwk_graph,V(ntwk_graph)[week_filter])
#edges from this week
edge_filter <- E(g)$created_at>lower_week & E(g)$created_at<upperdate
g <- delete_edges(g,E(g)[!edge_filter])

#get the same layout
l2 <- l[week_filter,]

#label the legislators with the highest interactions
node_degree <- degree(g,mode='in')
V(g)$label<-NA
V(g)[V(g)$name %in% names(sort(node_degree,decreasing=TRUE)[1:N])]$label <- V(g)[V(g)$name %in% names(sort(node_degree,decreasing=TRUE)[1:N])]$name
V(g)$label.cex = .5

if(plot==TRUE){
title <- as.character(all_dates[j])
plot(g, layout=l2,edge.arrow.size=0.01,asp=0,rescale=F,xlim = c(0, 40),
ylim = c(0, 41),main=title)
}

eцент <- eigen_centrality(g,directed=TRUE)
return(names(sort(eцент$vector,decreasing=TRUE)[1:3]))
}

```

Run windowed network function for Sep-Dec of 2019


```

startDate = as.POSIXct("2019-09-01");
endDate = as.POSIXct("2019-12-09");
all_dates = seq(startDate, endDate, 86400)
N=3
set.seed(0)

l = layout.fruchterman.reingold(ntwk_graph)
V(ntwk_graph)$label <- 1:532

V(ntwk_graph)$size<-10
V(ntwk_graph)[party == 'Republican']$color <- "red"
V(ntwk_graph)[party == 'Republican']$frame.color <- "red"
V(ntwk_graph)[party == 'Democrat']$color <- "dodgerblue"
V(ntwk_graph)[party == 'Democrat']$frame.color <- "dodgerblue"

evs <- c()
for(j in seq(from=14, to=length(all_dates), by=1)){
  ev_names <- timed_graph_windowed(j,FALSE)
  evs <- c(evs,ev_names)
}

```

```

## Warning in eigen_centrality(g, directed = TRUE): At centrality.c:344 :graph
## is directed and acyclic; eigenvector centralities will be zeros

```

List legislators with highest eigenvector centralities, over time

```

top_evs <- sort(table(evs),decreasing=TRUE)[1:10]
top_evs

```

```

## evs
##      gopleader    stevescalise repadamschiff speakerpelosi repdougcollins
##           45           36           30           24           19
## repmarkmeadows jimpressooffice repfredkeller repstevechabot replizcheney
##           12           7           7           7           6

```

Measure the density of the interaction graph I, and subgraphs I_R, I_D

```

mention_graph_density <- function(j) {

upperdate = all_dates[j]
lower_week = all_dates[j-7]

edge_filter <- E(ntwk_graph)$created_at>lower_week & E(ntwk_graph)$created_at<upperdate
g<- subgraph.edges(ntwk_graph,which(edge_filter),delete.vertices=TRUE)
g_filter <-V(ntwk_graph) %in% V(g)

l2 <- l[g_filter,]

full_density <- edge_density(g)
reps <- V(g)$party=="Republican"
dems <- V(g)$party=="Democrat"
rep_density <- edge_density(induced_subgraph(g,V(g)[reps]))
dem_density <- edge_density(induced_subgraph(g,V(g)[dems]))
return(c(full_density,rep_density,dem_density))
}

```

Calculate and plot density of interaction graph I, and subgraphs I_D, I_R, over time

```

startDate = as.POSIXct("2019-09-06");
endDate = as.POSIXct("2019-12-09");
all_dates = seq(startDate, endDate, 86400)
N=3
set.seed(0)

l = layout.fruchterman.reingold(ntwk_graph)
V(ntwk_graph)$label <- 1:532

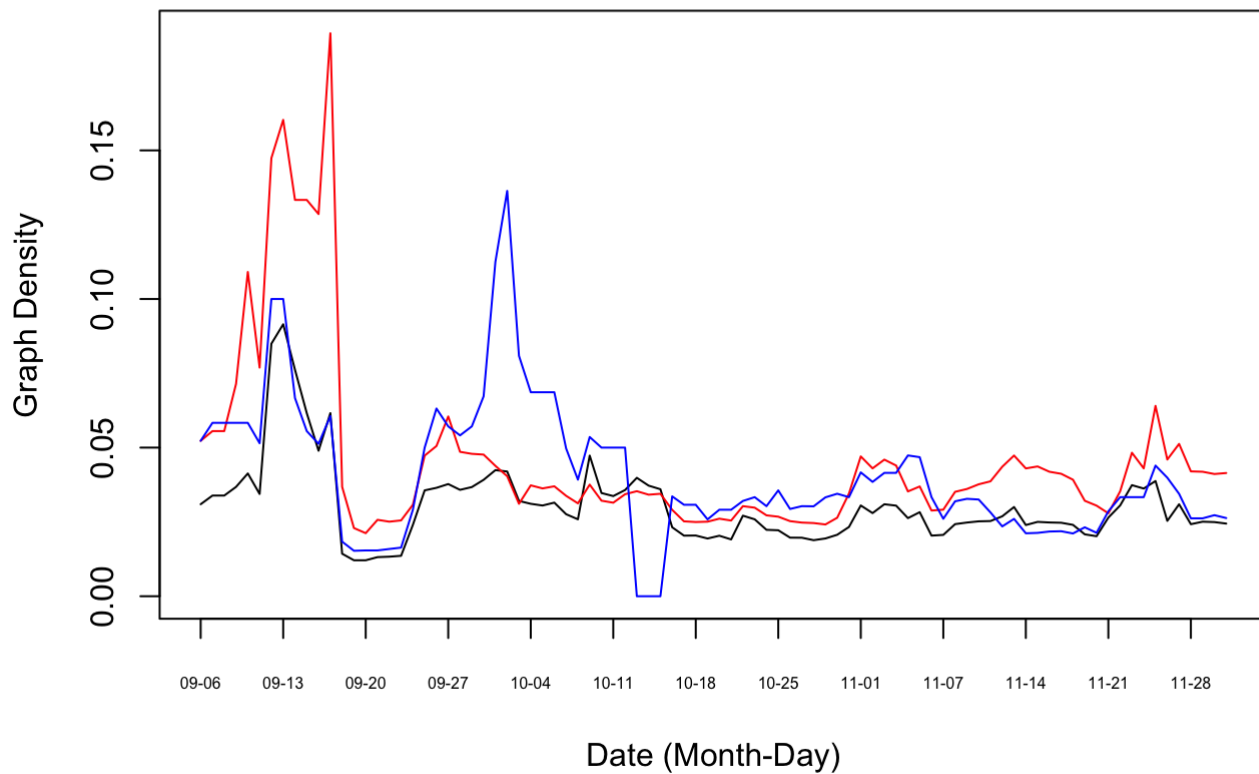
fd <- c()
rd <- c()
dd <- c()
for(j in seq(from=8, to=length(all_dates), by=1)){
  dens <- mention_graph_density(j)
  fd <- c(fd,dens[1])
  rd <- c(rd,dens[2])
  dd <- c(dd,dens[3])
}

xind = c()
xtick = c()
for(i in seq(from=1, to = length(fd),by=7)){
  xind = c(xind,i)
  xtick = c(xtick,substr(all_dates[i],6,10))
}

rd <- ifelse(is.nan(rd),0,rd)
fd <- ifelse(is.nan(fd),0,fd)
dd <- ifelse(is.nan(dd),0,dd)

y = fd
x = 1:length(y)
plot(x,y,type='l',col='black',ylim=c(0, max(rd)),xaxt='n',xlab='Date (Month-Day)',ylab=
'Graph Density')
axis(side=1, at=xind,labels = xtick,cex.axis=.5)
lines(x,rd,type='l',col='red')
lines(x,dd,type='l',col='blue')

```



Make follow dataframe and subsequent network F

```
load("~/Documents/R/Networks/sen_friends.RData")
load("~/Documents/R/Networks/rep_friends.RData")
legs <- read_csv("legislators-current.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   birthday = col_date(format = ""),
##   district = col_integer(),
##   senate_class = col_integer(),
##   cspan_id = col_integer(),
##   govtrack_id = col_integer(),
##   votesmart_id = col_integer(),
##   icpsr_id = col_integer()
## )
```

```
## See spec(...) for full column specifications.
```

```

legs <- legs %>% filter(type=='sen'|type=='rep') %>% dplyr::select(twitter,party,govtrack_id,type)
legs <- na.omit(legs)
sens <- legs %>%filter(type == 'sen') %>% dplyr::select(twitter,party,govtrack_id)
reps <- legs %>% filter(type=='rep') %>% dplyr::select(twitter,party,govtrack_id)

senrep <- rbind(sens,reps)
senrep_friends <-rbind(sen_friends,rep_friends)
senrep_friends$followed<-tolower(senrep_friends$followed)
senrep_friends$follower<-tolower(senrep_friends$follower)
senrep$twitter <- tolower(senrep$twitter)
senrep_friends

```

```

## # A tibble: 55,688 x 2
##   follower      followed
##   <chr>         <chr>
## 1 sensherrodbrown repagonzalez
## 2 sensherrodbrown repbalderson
## 3 sensherrodbrown warrendavidson
## 4 sensherrodbrown senbillcassidy
## 5 sensherrodbrown senrickscott
## 6 sensherrodbrown senatorbraun
## 7 sensherrodbrown senhawleypress
## 8 sensherrodbrown senmcsallyaz
## 9 sensherrodbrown marshablackburn
## 10 sensherrodbrown senkevincramer
## # ... with 55,678 more rows

```

```

senrep_friends2<-senrep_friends
senrep_friends2<-senrep_friends[senrep_friends$follower!='timkaine',]
senrep_friends2<-senrep_friends2[senrep_friends2$followed!='timkaine',]

follow_ntwk <- graph_from_data_frame(senrep_friends2, directed=TRUE,vertices=legislators)
follow_ntwk<-delete.vertices(follow_ntwk, degree(follow_ntwk)==0)
follow_ntwk

```

```

## IGRAPH f9459ad DN-B 529 55486 --
## + attr: name (v/c), party (v/c), govtrack_id (v/n), type (v/c),
## | terms (v/n), followed_by (v/n), follows (v/n), gender (v/c),
## | numtweets (v/n), dates (v/c)
## + edges from f9459ad (vertex names):
## [1] sensherrodbrown->repagonzalez      sensherrodbrown->repbalderson
## [3] sensherrodbrown->warrendavidson    sensherrodbrown->senbillcassidy
## [5] sensherrodbrown->senrickscott      sensherrodbrown->senatorbraun
## [7] sensherrodbrown->senhawleypress    sensherrodbrown->senmcsallyaz
## [9] sensherrodbrown->marshablackburn   sensherrodbrown->senkevincramer
## [11] sensherrodbrown->senatorromney     sensherrodbrown->sensinemas
## + ... omitted several edges

```

Make function to restrict follow network with time window (see paper), return legislators with highest in-degree

```

set.seed(0)
l_follow = layout.fruchterman.reingold(follow_ntwk)
V(follow_ntwk)[V(follow_ntwk)$party=='Democrat']$color<- "dodgerblue"
V(follow_ntwk)[V(follow_ntwk)$party=='Republican']$color<- "red"
V(follow_ntwk)$frame.color<-NA

N=3

timed_graph_follows <- function(j,plot=FALSE) {

upperdate = all_dates[j]
lower_week = all_dates[j-7]
lower_day = all_dates[j-1]

#tweets from the past week
week_tweets <- tweet_count[(tweet_count$created_at<=upperdate & tweet_count$created_at>lower_week),]
#all members who tweeted about impeachment in the past week
week_filter <- (V(follow_ntwk)$name %in% week_tweets$twitter)
#tweets from the past day
day_tweets <- tweet_count[(tweet_count$created_at<=upperdate & tweet_count$created_at>lower_day),]

#graph of legislators who tweeted about it in the past week
g <- induced_subgraph(follow_ntwk,V(follow_ntwk)[week_filter])

#members who tweeted about impeachment in the past day
day_filter <- (V(g)$name %in% day_tweets$twitter)
#delete any edges not from legislators who tweeted in the past day
g<- delete_edges(g,E(g)[!from(V(g)[day_filter])])

#get the same layout
l2 <- l_follow[week_filter,]

node_degree <- degree(g,mode='in')
V(g)$label<-NA

V(g)$size<-10
V(g)$size<-(strength(g,mode='in'))

title <- as.character(all_dates[j])

if(plot==TRUE){
plot(g, layout=l2,edge.arrow.size=0.01,asp=0,rescale=F,xlim = c(min(l_follow[,1]), max(l_follow[,1])),
ylim = c(min(l_follow[,2]), max(l_follow[,2])),main=title)
}
return(names(sort(node_degree,decreasing=TRUE)[1:N]))

}

```

Output users with highest in-degree over time

```

high_names <-c()
for(i in 8:length(all_dates)){
  val <- timed_graph_follows(i,FALSE)
  high_names <- c(high_names,val)
}

top_follows <- sort(table(high_names),decreasing=TRUE)[1:10]
top_follows

```

```

## high_names
##      gopleader      stevescalise      speakerpelosi      jim_jordan
##           57           46           19           17
##      repkevinbrady      leaderhoyer      johncornyn      repmarkmeadows
##           16           14           13           7
##      marshblackburn      repjeffduncan
##           6           6

```

Collect highest impact users

```

top_users <- c(top_evs,top_follows)
top_users <- unique(names(top_users))
top_users

```

```

## [1] "gopleader"      "stevescalise"    "repadamschiff"
## [4] "speakerpelosi"  "repdougcollins"  "repmarkmeadows"
## [7] "jimpressooffice" "repfredkeller"   "repstevechabot"
## [10] "replizcheney"   "jim_jordan"      "repkevinbrady"
## [13] "leaderhoyer"    "johncornyn"      "marshblackburn"
## [16] "repjeffduncan"

```

Plot follow network among highest impact users

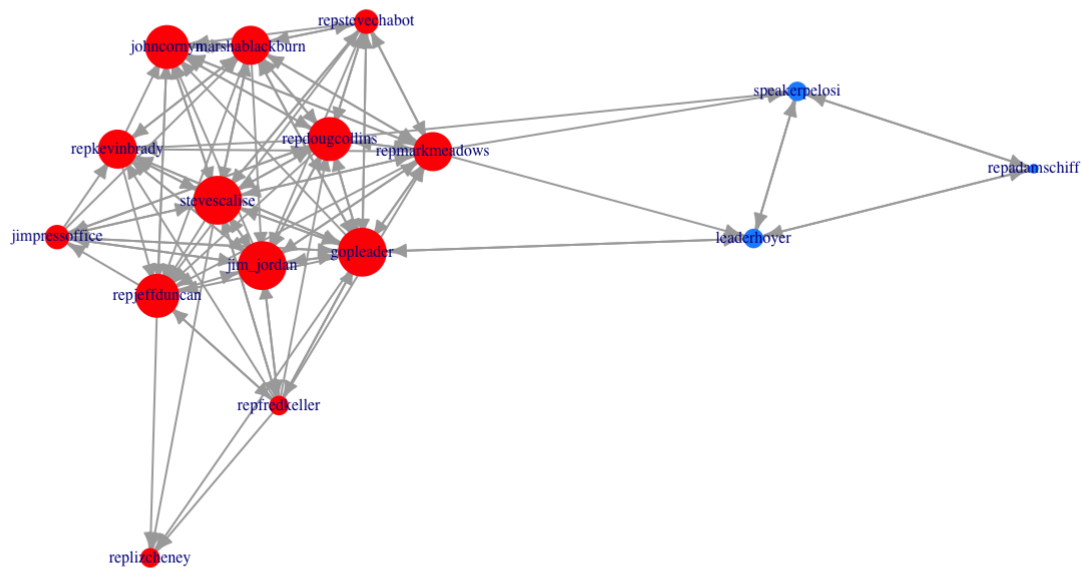
```

set.seed(0)
follow_ntwk <- graph_from_data_frame(senrep_friends2, directed=TRUE,vertices=legislators)
follow_ntwk<-delete.vertices(follow_ntwk, degree(follow_ntwk)==0)
V(follow_ntwk)[V(follow_ntwk)$party=='Democrat']$color<- "dodgerblue"
V(follow_ntwk)[V(follow_ntwk)$party=='Republican']$color<- "red"
V(follow_ntwk)$frame.color<-NA
g <- induced_subgraph(follow_ntwk,V(follow_ntwk)[V(follow_ntwk)$name %in% top_users])

V(g)$label<-V(g)$name
V(g)$label.cex = .5

V(g)$size<-(strength(g,mode='in'))
plot(g, layout=layout_fruchterman_reingold(g),edge.arrow.size=.5,asp=0)

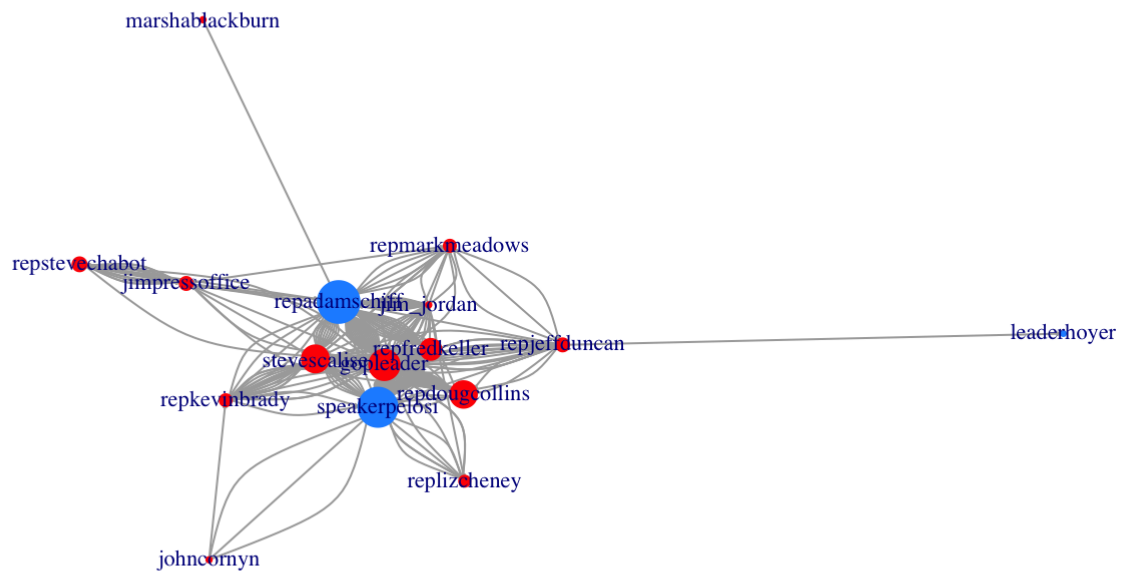
```



Plot interaction network among highest impact users

```
g <- induced_subgraph(ntwk_graph,V(ntwk_graph)[V(ntwk_graph)$name %in% top_users])
V(g)$label<-V(g)$name
V(g)$label.cex = .7
V(g)$size<-sqrt(strength(g,mode='in'))

plot(g, layout=layout.fruchterman.reingold(g),edge.arrow.size=.1,asp=0)
```

Collect tweets from top users

```
top_user_tweets <- impeach_tweets_full
top_user_tweets$screen_name <- tolower(top_user_tweets$screen_name)
top_user_tweets <- top_user_tweets %>% filter(screen_name %in% top_users)
top_user_tweets
```

```
## # A tibble: 1,595 x 90
##   user_id status_id created_at          screen_name text  source
##   <chr>    <chr>      <dtm>          <chr>      <chr> <chr>
## 1 132181... 12040470... 2019-12-09 14:36:04 johncornyn "\"F... Twitt...
## 2 132181... 12033357... 2019-12-07 15:29:40 johncornyn Than... Twitt...
## 3 132181... 12029863... 2019-12-06 16:21:20 johncornyn Pelo... Twitt...
## 4 132181... 12029348... 2019-12-06 12:56:39 johncornyn I do... Twitt...
## 5 132181... 12029348... 2019-12-06 12:56:39 johncornyn "\"J... Twitt...
## 6 132181... 12026620... 2019-12-05 18:52:22 johncornyn "Spe... Twitt...
## 7 132181... 12026561... 2019-12-05 18:29:03 johncornyn Unfo... Twitt...
## 8 132181... 12026045... 2019-12-05 15:04:00 johncornyn Sadl... Twitt...
## 9 132181... 12025546... 2019-12-05 11:45:52 johncornyn "Pin... Twitt...
## 10 132181... 12025478... 2019-12-05 11:18:41 johncornyn From... Twitt...
## # ... with 1,585 more rows, and 84 more variables:
## #   display_text_width <dbl>, reply_to_status_id <chr>,
## #   reply_to_user_id <chr>, reply_to_screen_name <chr>, is_quote <lgl>,
## #   is_retweet <lgl>, favorite_count <int>, retweet_count <int>,
## #   quote_count <int>, reply_count <int>, hashtags <list>, symbols <list>,
## #   urls_url <list>, urls_t.co <list>, urls_expanded_url <list>,
## #   media_url <list>, media_t.co <list>, media_expanded_url <list>,
## #   media_type <list>, ext_media_url <list>, ext_media_t.co <list>,
## #   ext_media_expanded_url <list>, ext_media_type <chr>,
## #   mentions_user_id <list>, mentions_screen_name <list>, lang <chr>,
## #   quoted_status_id <chr>, quoted_text <chr>, quoted_created_at <dtm>,
## #   quoted_source <chr>, quoted_favorite_count <int>,
## #   quoted_retweet_count <int>, quoted_user_id <chr>,
## #   quoted_screen_name <chr>, quoted_name <chr>,
## #   quoted_followers_count <int>, quoted_friends_count <int>,
## #   quoted_statuses_count <int>, quoted_location <chr>,
## #   quoted_description <chr>, quoted_verified <lgl>,
## #   retweet_status_id <chr>, retweet_text <chr>,
## #   retweet_created_at <dtm>, retweet_source <chr>,
## #   retweet_favorite_count <int>, retweet_retweet_count <int>,
## #   retweet_user_id <chr>, retweet_screen_name <chr>, retweet_name <chr>,
## #   retweet_followers_count <int>, retweet_friends_count <int>,
## #   retweet_statuses_count <int>, retweet_location <chr>,
## #   retweet_description <chr>, retweet_verified <lgl>, place_url <chr>,
## #   place_name <chr>, place_full_name <chr>, place_type <chr>,
## #   country <chr>, country_code <chr>, geo_coords <list>,
## #   coords_coords <list>, bbox_coords <list>, status_url <chr>,
## #   name <chr>, location <chr>, description <chr>, url <chr>,
## #   protected <lgl>, followers_count <int>, friends_count <int>,
## #   listed_count <int>, statuses_count <int>, favourites_count <int>,
## #   account_created_at <dtm>, verified <lgl>, profile_url <chr>,
## #   profile_expanded_url <chr>, account_lang <lgl>,
## #   profile_banner_url <chr>, profile_background_url <chr>,
## #   profile_image_url <chr>
```

Create df of top users' twitter activity over time

```

top_user_df <- data.frame(top_users)
names(top_user_df)<-"screen_name"
top_user_df$screen_name <- as.character(top_user_df$screen_name)

for(j in 2:length(all_dates)){
upperdate = all_dates[j]
lowerdate = all_dates[j-1]

#tweets from the past week
week_tweets <- top_user_tweets[top_user_tweets$created_at<upperdate & top_user_tweets$cr
eated_at>lowerdate,]

week_df <- week_tweets %>% group_by(screen_name) %>% count() %>% select(screen_name,n)
names(week_df) <- c("screen_name",as.character(upperdate))
top_user_df <- top_user_df %>% full_join(week_df,by='screen_name')

}
top_user_df[is.na(top_user_df)] <- 0

twitter_party <- legislators %>% select(twitter,party)
names(twitter_party)<-c("screen_name","party")
top_user_df <- top_user_df %>% left_join(twitter_party,by='screen_name')
top_user_df

```

##	screen_name	2019-09-07	2019-09-08	2019-09-09	2019-09-10	2019-09-11
## 1	gopleader	0	0	0	0	0
## 2	stevescalise	0	0	0	0	1
## 3	repadamschiff	0	0	0	0	0
## 4	speakerpelosi	0	0	0	0	0
## 5	repdougcollins	0	0	0	2	0
## 6	repmarkmeadows	0	0	0	2	3
## 7	jimpressooffice	0	0	0	0	0
## 8	repfredkeller	0	0	0	0	0
## 9	repstevechabot	0	0	0	0	0
## 10	replizcheney	0	0	0	0	1
## 11	jim_jordan	0	0	0	3	2
## 12	repkevinbrady	0	0	0	1	0
## 13	leaderhoyer	0	0	0	0	0
## 14	johncornyn	0	0	0	0	0
## 15	marshablackburn	0	0	0	0	0
## 16	repjeffduncan	0	0	0	0	0
##	2019-09-12	2019-09-13	2019-09-14	2019-09-15	2019-09-16	2019-09-17
## 1	0	4	2	0	0	0
## 2	0	4	1	1	0	0
## 3	0	0	0	0	1	0
## 4	0	0	0	0	0	0
## 5	0	3	0	0	0	0
## 6	1	2	0	0	0	1
## 7	0	0	0	0	0	0
## 8	0	0	0	0	0	0
## 9	0	1	0	0	0	0
## 10	0	0	0	0	0	0
## 11	0	2	0	0	0	1
## 12	0	0	0	0	0	0
## 13	1	0	0	0	0	0
## 14	0	0	0	0	0	0
## 15	0	0	0	0	0	0
## 16	0	0	0	0	0	0
##	2019-09-18	2019-09-19	2019-09-20	2019-09-21	2019-09-22	2019-09-23
## 1	2	5	1	1	1	0
## 2	1	2	0	1	0	0
## 3	0	0	0	0	0	0
## 4	0	0	0	0	0	0
## 5	1	0	0	0	0	0
## 6	0	0	0	0	0	0
## 7	0	0	0	0	0	0
## 8	0	0	0	0	0	0
## 9	1	0	0	0	0	0
## 10	0	1	0	0	0	0
## 11	0	0	0	0	0	0
## 12	1	0	0	0	0	0
## 13	0	0	0	0	0	0
## 14	0	0	0	0	0	0
## 15	0	0	0	0	0	0
## 16	1	0	0	0	0	0
##	2019-09-24	2019-09-25	2019-09-26	2019-09-27	2019-09-28	2019-09-29
## 1	0	5	5	6	8	1

## 2	0	3	3	2	1	1
## 3	0	1	0	0	0	0
## 4	0	1	0	0	0	0
## 5	0	2	2	2	1	0
## 6	1	2	3	2	0	0
## 7	0	0	0	1	0	0
## 8	0	1	0	0	1	0
## 9	0	0	0	0	0	0
## 10	0	1	0	0	0	0
## 11	0	2	3	2	0	0
## 12	0	0	1	0	1	0
## 13	0	1	1	0	0	0
## 14	0	0	0	0	0	0
## 15	0	0	2	0	0	0
## 16	0	2	5	1	0	0
##	2019-09-30	2019-10-01	2019-10-02	2019-10-03	2019-10-04	2019-10-05
## 1	2	2	3	5	5	3
## 2	3	1	0	2	9	4
## 3	2	0	0	0	0	0
## 4	1	0	0	1	0	0
## 5	0	0	1	0	1	1
## 6	0	0	1	0	1	5
## 7	0	0	0	1	0	0
## 8	0	0	1	0	2	1
## 9	0	0	0	0	0	0
## 10	0	0	0	1	0	0
## 11	0	0	1	3	2	0
## 12	0	1	2	1	0	0
## 13	0	0	0	0	1	2
## 14	3	2	2	1	1	2
## 15	1	1	2	0	1	0
## 16	0	0	0	2	3	0
##	2019-10-06	2019-10-07	2019-10-08	2019-10-09	2019-10-10	2019-10-11
## 1	1	3	6	4	4	3
## 2	0	1	1	1	2	3
## 3	0	0	0	1	0	0
## 4	0	0	0	0	0	0
## 5	0	0	1	2	1	1
## 6	0	0	0	1	0	1
## 7	0	0	0	0	0	0
## 8	0	0	0	1	1	0
## 9	0	0	0	0	0	0
## 10	0	0	1	0	0	0
## 11	1	1	2	1	2	1
## 12	0	0	2	0	1	1
## 13	3	0	0	0	0	0
## 14	1	0	1	2	1	0
## 15	0	0	0	1	0	0
## 16	0	0	1	0	0	2
##	2019-10-12	2019-10-13	2019-10-14	2019-10-15	2019-10-16	2019-10-17
## 1	2	2	2	0	2	2
## 2	1	0	0	2	1	5
## 3	0	0	0	0	0	0
## 4	0	0	0	0	0	0

## 5	0	1	1	0	1	0
## 6	2	0	1	1	0	1
## 7	0	0	0	0	0	0
## 8	0	0	0	0	0	0
## 9	0	0	0	0	0	0
## 10	0	0	0	0	0	1
## 11	1	2	0	2	1	1
## 12	0	0	0	0	1	0
## 13	0	0	0	0	0	1
## 14	1	0	1	1	3	3
## 15	0	0	1	1	0	0
## 16	0	0	0	0	0	3
##	2019-10-18	2019-10-19	2019-10-20	2019-10-21	2019-10-22	2019-10-23
## 1	4	7	3	1	17	10
## 2	4	4	1	2	3	17
## 3	0	0	0	0	0	1
## 4	0	0	0	0	0	0
## 5	0	1	0	0	4	0
## 6	0	1	0	0	2	1
## 7	0	0	0	0	0	0
## 8	0	1	1	0	3	0
## 9	0	0	0	0	0	0
## 10	0	0	0	0	0	1
## 11	0	1	3	0	1	1
## 12	0	2	0	0	0	1
## 13	0	0	0	0	0	1
## 14	2	4	0	0	0	4
## 15	0	0	0	0	0	0
## 16	0	2	0	0	1	1
##	2019-10-24	2019-10-25	2019-10-26	2019-10-27	2019-10-28	2019-10-29
## 1	9	0	0	3	0	2
## 2	12	4	1	2	0	2
## 3	0	0	0	0	0	1
## 4	0	0	0	0	0	1
## 5	2	1	0	1	0	3
## 6	2	1	0	1	0	5
## 7	0	0	0	0	0	0
## 8	2	4	2	0	0	0
## 9	0	0	0	0	0	0
## 10	0	0	0	0	0	0
## 11	1	1	0	1	0	5
## 12	3	0	1	0	0	2
## 13	0	0	0	0	0	0
## 14	2	3	3	1	0	1
## 15	0	0	0	0	0	0
## 16	0	2	0	0	0	0
##	2019-10-30	2019-10-31	2019-11-01	2019-11-02	2019-11-03	
## 1	10	7	12	2	1	
## 2	5	6	9	4	2	
## 3	0	1	2	0	0	
## 4	0	0	0	1	0	
## 5	3	1	4	0	0	
## 6	2	2	9	2	0	
## 7	0	0	0	0	0	

## 8	1	1	2	1	0
## 9	0	0	1	0	0
## 10	0	0	1	0	1
## 11	3	1	3	0	0
## 12	2	1	4	0	0
## 13	0	2	1	0	0
## 14	3	1	7	4	0
## 15	0	0	0	0	0
## 16	1	0	7	1	0
##	2019-11-03 23:00:00	2019-11-04 23:00:00	2019-11-05 23:00:00	2019-11-05 23:00:00	
## 1		4		7	1
## 2		3		1	1
## 3		1		0	2
## 4		0		0	0
## 5		0		0	0
## 6		0		1	1
## 7		0		7	4
## 8		0		1	0
## 9		0		1	6
## 10		0		0	0
## 11		1		1	0
## 12		0		0	1
## 13		2		0	0
## 14		1		0	0
## 15		0		0	0
## 16		0		1	0
##	2019-11-06 23:00:00	2019-11-07 23:00:00	2019-11-08 23:00:00	2019-11-08 23:00:00	
## 1		4		9	7
## 2		3		2	1
## 3		1		0	0
## 4		0		0	0
## 5		1		0	2
## 6		0		1	1
## 7		0		0	0
## 8		0		1	0
## 9		0		1	0
## 10		0		0	0
## 11		1		1	0
## 12		0		0	0
## 13		1		0	0
## 14		0		2	0
## 15		1		0	0
## 16		0		0	1
##	2019-11-09 23:00:00	2019-11-10 23:00:00	2019-11-11 23:00:00	2019-11-11 23:00:00	
## 1		2		0	0
## 2		0		1	0
## 3		0		0	0
## 4		0		0	0
## 5		0		0	0
## 6		1		0	0
## 7		0		0	0
## 8		0		0	0
## 9		0		0	0
## 10		0		0	0

## 11	1	0	0
## 12	0	0	0
## 13	0	0	0
## 14	1	0	4
## 15	0	0	0
## 16	0	0	0
##	2019-11-12 23:00:00	2019-11-13 23:00:00	2019-11-14 23:00:00
## 1	5	20	4
## 2	2	16	6
## 3	1	1	0
## 4	1	1	1
## 5	0	2	2
## 6	3	20	9
## 7	1	1	0
## 8	2	2	3
## 9	0	2	0
## 10	1	1	0
## 11	1	1	1
## 12	0	5	2
## 13	0	1	1
## 14	0	2	0
## 15	0	2	1
## 16	2	5	3
##	2019-11-15 23:00:00	2019-11-16 23:00:00	2019-11-17 23:00:00
## 1	24	3	3
## 2	5	2	0
## 3	1	0	0
## 4	0	0	0
## 5	0	0	0
## 6	8	1	2
## 7	0	0	0
## 8	1	1	0
## 9	1	0	0
## 10	1	0	0
## 11	2	0	2
## 12	1	0	0
## 13	0	0	0
## 14	0	2	1
## 15	1	1	0
## 16	8	0	0
##	2019-11-18 23:00:00	2019-11-19 23:00:00	2019-11-20 23:00:00
## 1	4	8	7
## 2	1	10	8
## 3	0	1	0
## 4	0	1	0
## 5	0	1	0
## 6	4	13	6
## 7	0	0	0
## 8	2	2	1
## 9	0	0	0
## 10	0	1	0
## 11	2	3	2
## 12	1	1	4
## 13	0	1	2

## 14	1	4	2
## 15	0	2	1
## 16	3	3	7
##	2019-11-21 23:00:00	2019-11-22 23:00:00	2019-11-23 23:00:00
## 1	17	7	2
## 2	10	3	0
## 3	0	0	0
## 4	0	0	0
## 5	0	0	1
## 6	2	2	0
## 7	0	0	0
## 8	3	0	0
## 9	1	0	0
## 10	0	0	0
## 11	1	1	0
## 12	5	2	0
## 13	0	0	0
## 14	5	0	3
## 15	1	1	0
## 16	2	2	0
##	2019-11-24 23:00:00	2019-11-25 23:00:00	2019-11-26 23:00:00
## 1	2	2	3
## 2	1	1	1
## 3	0	0	0
## 4	0	0	0
## 5	0	0	0
## 6	1	3	4
## 7	0	0	0
## 8	0	1	1
## 9	0	0	0
## 10	0	0	0
## 11	0	1	1
## 12	0	0	1
## 13	0	0	0
## 14	2	0	2
## 15	0	1	0
## 16	0	0	1
##	2019-11-27 23:00:00	2019-11-28 23:00:00	2019-11-29 23:00:00
## 1	1	0	2
## 2	1	0	0
## 3	0	0	0
## 4	0	0	0
## 5	1	0	0
## 6	1	0	1
## 7	0	0	0
## 8	1	0	0
## 9	0	0	0
## 10	0	0	0
## 11	0	0	0
## 12	0	0	0
## 13	0	0	0
## 14	0	0	0
## 15	0	0	0
## 16	0	0	0

##	2019-11-30 23:00:00	2019-12-01 23:00:00	2019-12-02 23:00:00	
## 1	1	3	4	
## 2	1	0	1	
## 3	0	0	1	
## 4	0	0	0	
## 5	1	0	2	
## 6	0	0	2	
## 7	0	0	0	
## 8	0	0	2	
## 9	0	0	0	
## 10	0	0	0	
## 11	0	0	1	
## 12	0	0	0	
## 13	0	0	0	
## 14	0	1	2	
## 15	0	0	1	
## 16	0	0	0	
##	2019-12-03 23:00:00	2019-12-04 23:00:00	2019-12-05 23:00:00	
## 1	12	27	13	
## 2	6	11	7	
## 3	2	1	0	
## 4	0	1	3	
## 5	1	15	1	
## 6	9	18	6	
## 7	1	12	1	
## 8	2	3	2	
## 9	0	5	0	
## 10	1	0	0	
## 11	0	4	1	
## 12	2	3	3	
## 13	0	4	2	
## 14	1	6	5	
## 15	2	6	0	
## 16	4	14	3	
##	2019-12-06 23:00:00	2019-12-07 23:00:00	2019-12-08 23:00:00	party
## 1	9	2	2	Republican
## 2	3	1	1	Republican
## 3	0	1	1	Democrat
## 4	1	1	1	Democrat
## 5	4	0	1	Republican
## 6	4	0	2	Republican
## 7	5	0	0	Republican
## 8	2	0	1	Republican
## 9	2	0	0	Republican
## 10	0	0	0	Republican
## 11	1	0	0	Republican
## 12	0	0	0	Republican
## 13	0	0	0	Democrat
## 14	3	1	0	Republican
## 15	0	1	1	Republican
## 16	1	0	0	Republican

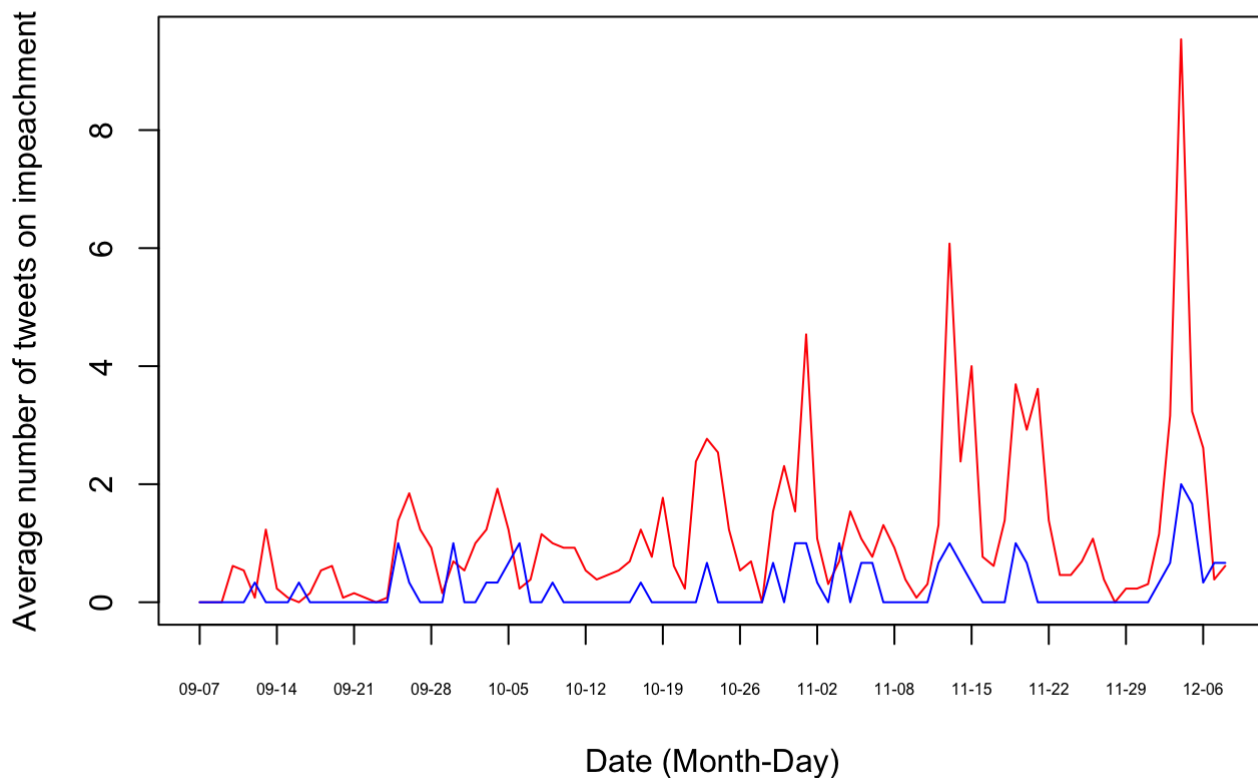
Smooth and plot this data

```

xind = c()
xtick = c()
for(i in seq(from=1, to = length(top_user_df),by=7)){
  xind = c(xind,i)
  xtick = c(xtick,substr(all_dates[i+1],6,10))
}

rep_df <- top_user_df %>% filter(party=="Republican")
dem_df <- top_user_df %>% filter(party=="Democrat")
rep_mean = c()
dem_mean = c()
for(i in 2:95){
  rep_mean <- c(rep_mean,mean(rep_df[,i]))
  dem_mean <- c(dem_mean,mean(dem_df[,i]))
}
plot(1:length(rep_mean),rep_mean,type='l',col='red',xaxt = 'n',xlab="Date (Month-Day)",
  ylab="Average number of tweets on impeachment")
axis(side=1,at=xind,labels=xtick,cex.axis=.5)
lines(1:length(dem_mean),dem_mean,type='l',col='blue')

```



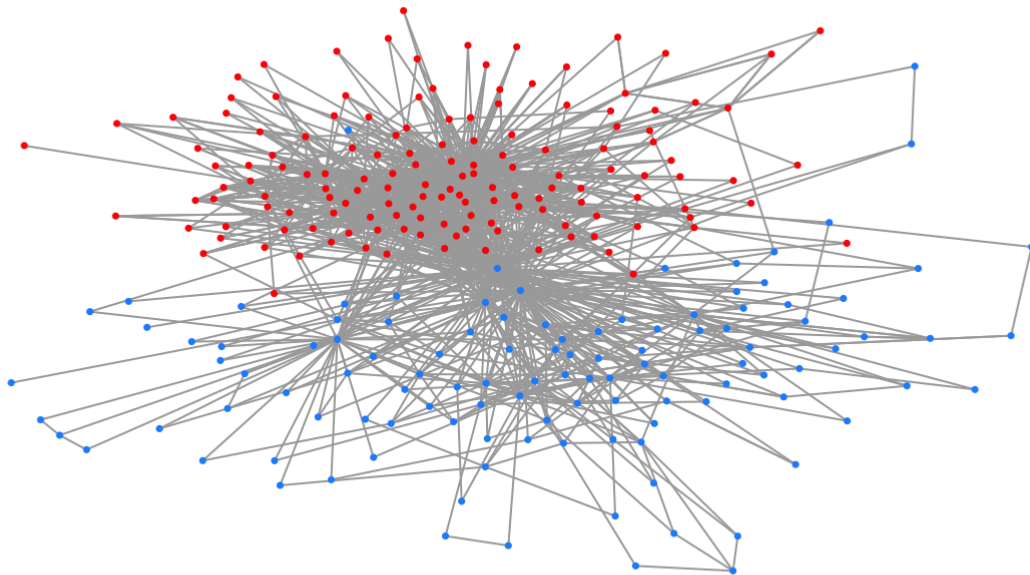
Plot interaction graph for a topic, colored for party and for cluster

```

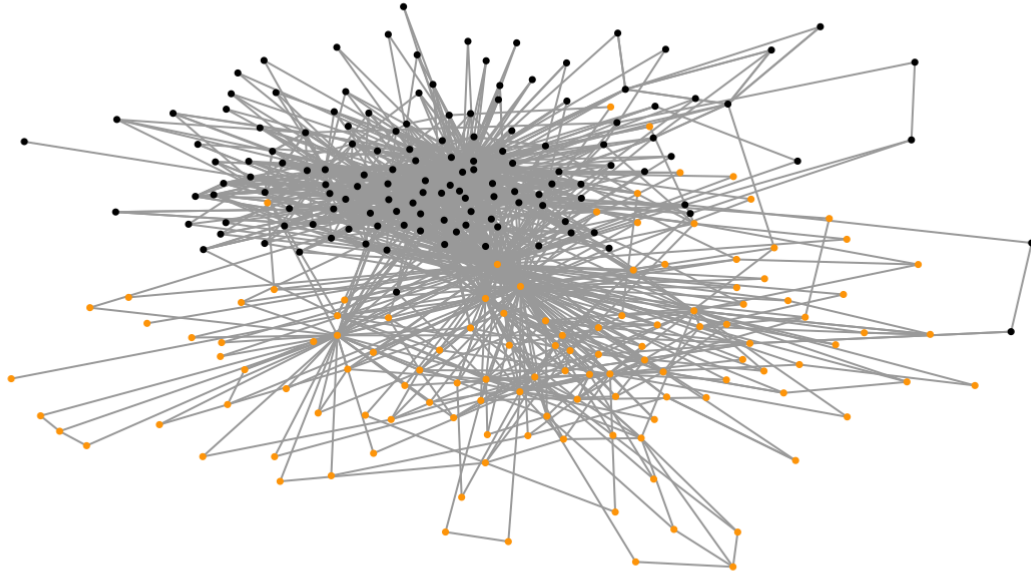
tg <-ntwk_graph

E(tg)$weights<-1
g <- delete_edge_attr(tg,"created_at")
g <- delete_edge_attr(g,"text")
g <- delete_edge_attr(g,"status_id")
g <- delete_edge_attr(g,"interaction_type")
g<- simplify(g, edge.attr.comb="sum")
g<- delete.vertices(g, degree(g)<=1)
g<- decompose.graph(g,max.comps=1)[[1]]
comms <- cluster_fast_greedy(as.undirected(g))
V(g)$group <- cutat(comms,2)
V(g)$label<- NA
V(g)$size<- 1
V(g)[length(V(g))-$party = "Republican"
V(g)[party == 'Republican']$color <- "red"
V(g)[party == 'Republican']$frame.color <- "red"
V(g)[party == 'Democrat']$color <- "dodgerblue"
V(g)[party == 'Democrat']$frame.color <- "dodgerblue"
l = layout.fruchterman.reingold(g)
plot(g,layout = l,edge.arrow.size=0.01,asp=0)

```



```
comms <- cluster_fast_greedy(as.undirected(g))
V(g)$group <- cutat(comms,2)
V(g)[group == 1]$color <- "black"
V(g)[group == 2]$color <- "orange"
V(g)$frame.color<-"black"
V(g)[group == 2]$frame.color <- "orange"
plot(g,layout = 1,edge.arrow.size=0.01,asp=0)
```



Define the partisan number for a chosen word

```

partisan_number <- function(chosen_word){

  impeach <- str_detect(tweets$text, regex(chosen_word,ignore_case=TRUE))
  impeach_tweets <- tweets %>% filter(impeach)

  df <- impeach_tweets
  # all mentions, as string
  mention_list <- vector("list", nrow(impeach_tweets))
  for(i in 1:nrow(impeach_tweets)){
    mentions <- impeach_tweets$mentions_screen_name[i][[1]]
    mentions <- paste(mentions,collapse=',')[[1]]
    mention_list[[i]]<-ifelse(mentions=="NA",NA,mentions)
  }
  df$mentions <- mention_list
  df <- transform(df, mentions = as.character(mentions))
  df <- df %>% dplyr::select(screen_name,reply_to_screen_name,quoted_screen_name,retweet
_screen_name,mentions,text,status_id,created_at)
  # all interactions as string
  df <- unite(df,interaction_name,2:5,sep="," ,remove=FALSE)
  # all interactions, separate rows, edges only when interacting with other legislators
  df <- separate_rows(df,interaction_name) %>% filter(tolower(interaction_name) %in% leg
islators$twitter)
  df <- unique(df)
  # remove self edges
  df <- filter(df,interaction_name!=screen_name)

  # tweet dates, interaction types
  df$created_at<-as.Date(df$created_at)
  df$interaction_type <- ifelse(is.na(df$reply_to_screen_name),ifelse(is.na(df$quoted_sc
reen_name),ifelse(is.na(df$retweet_screen_name),'mention','retweet'),'quote'),'reply')

  df$screen_name<-tolower(df$screen_name)
  df$interaction_name<-tolower(df$interaction_name)
  df <- df %>% dplyr::select(screen_name,interaction_name,text,status_id,interaction_typ
e,created_at)
  #df
  df$created_at<-as.character(df$created_at)

  ntwk_graph <- graph_from_data_frame(df, directed = TRUE, vertices=legislators)

  tg <-ntwk_graph
  E(tg)$weights<-1
  g <- delete_edge_attr(tg,"created_at")
  g <- delete_edge_attr(g,"text")
  g <- delete_edge_attr(g,"status_id")
  g <- delete_edge_attr(g,"interaction_type")
  g<- simplify(g, edge.attr.comb="sum")
  g<- delete.vertices(g, degree(g)<=1)
  g<- decompose.graph(g,max.comps=1)[[1]]
  comms <- cluster_fast_greedy(as.undirected(g))
  V(g)$group <- cutat(comms,2)

  group_df <- data.frame(V(g)$name,V(g)$group)

```

```
names(group_df)<-c("twitter","group")
group_df$twitter <- as.character(group_df$twitter)
leg_group <- legislators %>% inner_join(group_df,by='twitter')

df1 <- leg_group %>% select(party,group) %>% filter(group==1) %>% group_by(party) %>%
count() %>% arrange(desc(n))
df2 <- leg_group %>% select(party,group) %>% filter(group==2) %>% group_by(party) %>%
count() %>% arrange(desc(n))
return((max(df1$n)/sum(df1$n)+max(df2$n)/sum(df2$n))/2)
}
```