SID: 1810980
Gold Cohort

Victor Xu
Professor Hwang
January 12th, 2019
Social Media Analytics Exploration

1. **<u>Exploring Social Network Graphs [in R]</u>**

```
#########################################################
# Network Analysis using Hypothetical Network  ##
#########################################################

install.packages("igraph")  # installs 'igraph' packages
library(igraph) # Activate igraph package

# creating toy network
edges <- rbind(c("Dave","Jenny"),c("Peter","Jenny"),c("John","Jenny"),
               c("Dave","Peter"),c("Dave","John"),c("Peter","Sam"),
               c("Sam","Albert"),c("Peter","John"))

#rbind is creating the lines

################################
###### undirected graph ########
################################

ug<-graph.edgelist(edges,directed=FALSE)      # undirected graph object
plot(ug,vertex.size=30,vertex.label.cex=0.6)  # plotting

E(ug) # view edge information, E for edge
V(ug) # view node information, V for vertex(node)

# degree centrality
indeg<-degree(ug,mode="in")      # calcuates indegree centrality
outdeg<-degree(ug,mode="out")    # calculates outdegree centrality
totaldeg<-degree(ug,mode="all")  # calculates totaldegree centrality

# Are they different?
x<-cbind(indeg,outdeg,totaldeg)
cor(x)

#correlation of 1, all the same because of undirected graph
```

SID: 1810980

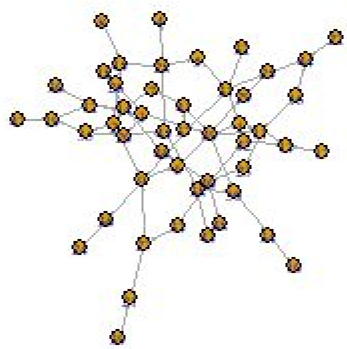
Gold Cohort

```
###################################
####### directed graph ############
###################################

dg<-graph.edgelist(edges,directed=TRUE)
plot(dg,vertex.size=20,vertex.label.cex=0.5
    ,edge.arrow.size=0.5)

# degree centrality
indeg<-degree(dg,mode="in")
outdeg<-degree(dg,mode="out")
totaldeg<-degree(dg,mode="all")

# Are they different?
x<-cbind(indeg,outdeg,totaldeg)
cor(x)

###################################
##### Using degree centrality to change node size ###
V(dg)$indeg<-indeg*10
plot(dg,vertex.size=V(dg)$indeg,vertex.label.cex=0.5,edge.arrow.size=0.05)

######network layouts ###############

plot(dg,vertex.size=V(dg)$indeg,vertex.label.cex=0.5,edge.arrow.size=0.05,
    layout=layout.random)

plot(dg,vertex.size=V(dg)$indeg,vertex.label.cex=0.5,edge.arrow.size=0.05,
    layout=layout.circle)

plot(dg,vertex.size=V(dg)$indeg,vertex.label.cex=0.5,edge.arrow.size=0.05,
    layout=layout.sphere)

plot(dg,vertex.size=V(dg)$indeg,vertex.label.cex=0.5,edge.arrow.size=0.05,
    layout=layout.fruchterman.reingold)


##########################################
######## Betweenness centrality ##########
##########################################

btw <- betweenness(dg,directed=TRUE)  #betweenness centrality
V(dg)$btw <- btw*10
```

SID: 1810980
Gold Cohort

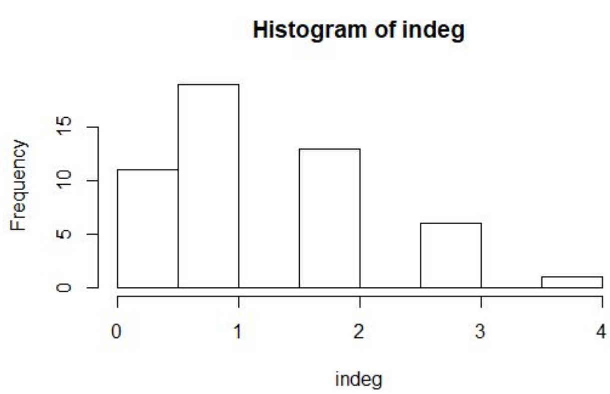
# Using betweenness centrality to change node size
plot(dg,vertex.size=V(dg)$btw,vertex.label.cex=0.5,edge.arrow.size=0.05)

##
Network Analysis using Real Network - CSV file of Facebook 'friend' circle
##



Q1.
Calculate indegree centrality of all the nodes. Create a histogram of the indegree centrality. Copy and paste your histogram below.

```
indeg<-degree(dg,mode="in")
> indeg
2 0 2 1 2 1 1 0 3 1 2 1 3 3 4 0 0 1 3 2 2 0 1 0 2 1 1 0 1 1 1 0 0 1 1 2 1 1 0 2 1 2 2 3
2 1 0 1 3 2
```

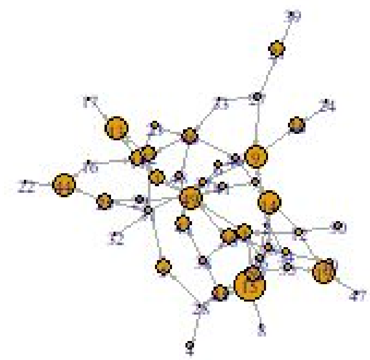


SID: 1810980

Gold Cohort

Q2

Create a plot with nodes sized by indegree centrality. You may choose any network layout that represents the network well. Copy and paste your plot below.



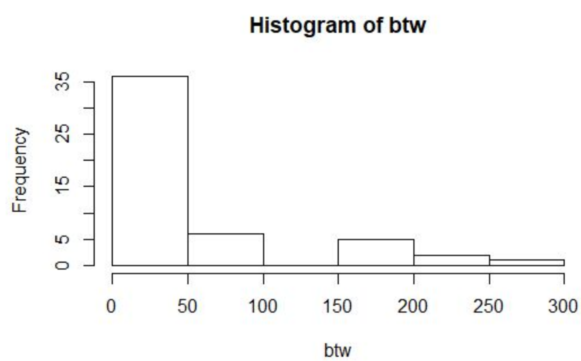
Q3

Which node is the most central one based on indegree centrality?

The node with the largest size on the plot is **#15**

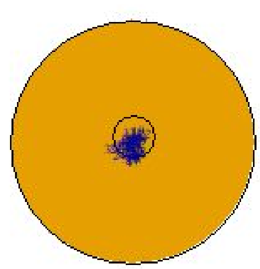
Q4.

Calculate betweenness centrality of all the nodes. Create a histogram of the betweenness centrality. Copy and paste your histogram below.



Q5.

Create a plot with nodes sized by betweenness centrality. You may choose any network layout that represents the network well. Copy and paste your plot below.



SID: 1810980

Gold Cohort

Q6.

Which node is the most central one based on betweenness centrality?

Node #49

Q7

What is the correlation of indegree centrality and betweenness centrality in this context?

```
> x<-cbind(indeg,btw)
> cor(x)
          indeg       btw
indeg 1.0000000 0.3106236
btw   0.3106236 1.0000000
```

Correlation is positive, 0.3106236

SID: 1810980
Gold Cohort

2. <u>Twitter Sentiment Analysis [Twitter API]</u>

```
#######################################################
#### Create connection to Twitter API ##################
#######################################################

#Setup Twitter auth
api_key <- "LYSaZFKsTEaMVXWEMLD3p42Sv"
api_secret <- "FCOuUfdUifrkbKiudWcEW7Vz0stY3YtBFvaXkVaEQVL5JvzaSk"
access_token <- "2590349732-A4jacrqM5FLGJ17fZvswcS4sMxIsuPsbwybVszB"
access_token_secret <- "CB48cDZniqduLHYqI3OtK8muMedWzbXWWa7oPUVbYkUoj"

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)


######################################################
##### Download Twitter Data ##########################
#######################################################
#get the n most recent tweets mentioning '?':
tweet <- searchTwitter("Donald Trump", lang = "en", n=1000)


#######################################################
########Sentiment Analysis##############################
#######################################################

### Data preprocessing ###
#1.Convert it to dataframe:
tweet_df = twListToDF(tweet)

# get the data ready for further processing
tweet_df2<-data_frame(line=1:1000,text=as.vector(tweet_df$text))
tweet_df3<-tweet_df2 %>% unnest_tokens(word,text)

#3. removing stop words
data("stop_words") #loads stop words

tweet_df4 <- tweet_df3 %>%
  anti_join(stop_words,by=c("word"="word"))
```

SID: 1810980
Gold Cohort

```
#4. tokenized tweets after removing stop words
tweet_df4 %>%
  count(word, sort = TRUE)
#size=dim(tweet_df4)
#total=size[1:1]

#5. sentiment analysis
bing<-get_sentiments("bing")  #calling sentiment library

bing_word_counts <- tweet_df4 %>%
  inner_join(bing,by="word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()

bing_word_counts  #word, sentiment, frequency


#6. calculating sentiment score
negatives=bing_word_counts %>%
  filter(sentiment=="negative")
positives=bing_word_counts %>%
  filter(sentiment=="positive")

negcount=sum(negatives[,3])
poscount=sum(positives[,3])

positive_to_negativeratio=poscount/negcount
positive_to_negativeratio


#7. wordcloud_negative words
bing_word_counts %>%
  filter(sentiment=="negative") %>%
  with(wordcloud(word, n, scale=c(3,0.6),max.words = 100))

#8. wordcloud_positive words
bing_word_counts %>%
  filter(sentiment=="positive") %>%
  with(wordcloud(word, n, scale=c(3,0.6),max.words = 100))
```

SID: 1810980

Gold Cohort

```
#########################################
#### Customizing Sentiment library#######
#########################################
# excluding from library
bing<-subset(bing,word!="trump")

# changing sentiment tag
which(bing$word == "crazy", arr.ind=TRUE) # find the row number for "crazy" word
bing[1122,2]<-"positive" # changing the emotion tag to positive
bing[1122,]  # verifying the change
```

SID: 1810980
Gold Cohort

3. Making Predictions with Social Media Nets: Bitcoin Value [Google Trends API]

```
####################################################################################
################## Predicting Bitcoin price ##################################
####################################################################################
library(gtrendsR)
#package to collect google trends data for us
library(stringr)
library(lubridate)
library(dplyr)
library(data.table)
library(reshape2)

################### Collecting google trends data ###########################################
# Below code collects google trends data for the search term "bitcoin". You can collect google
# trends data for different search term by chaning "bitcoin" to other term.
?gtrends # <- type this if you want to see a help file for gtrends function

google.trends = gtrends(c("bitcoin"), gprop = "web", time = "2017-01-08 2019-01-09")[[1]]


################### Loading bitcoin price data  ###########################################
# (bitcoin price data is obtained from https://www.blockchain.com/charts/market-price?timespan=2years)

bitcoin<-read.csv("C6_bitcoin.csv",header=FALSE)  #loading bitcoin data I downloaded for the class
names(bitcoin)[1]<-"time"                         # changing column names
names(bitcoin)[2]<-"priceinUSD"                   # changing column names
#our google data is in weekly, whereas the csv is in daily

bitcoin$time<-as.POSIXct(bitcoin$time)               # changing data type to use floor_date function
bitcoin$week<-floor_date(bitcoin$time,unit="week")    # changing date unit from day to week to match
with google trends data
#created a new column for week


# Aggregate bitcoin data by week to match with google trends data
bitcoin<-setDT(bitcoin)               #changing data type to 'data table' to use data.table package functions
bitcoin2<-bitcoin[,mean(priceinUSD),by="week"]  #aggregating bitcoin price data to weekly unit
names(bitcoin2)[2]<-"avgprice"                 # renaming the second colum
```

SID: 1810980
Gold Cohort

```
# combining bitcoin and google trends data
data<-cbind(bitcoin2,google.trends$hits)
names(data)[3]<-"gt_bitcoin"


################### Forecasting bitcoin price using google trends data 1 ##########

# DV: bitcoin price at t, IV: google trends data at t
ans1<-lm(avgprice~gt_bitcoin,data=data)
summary(ans1)

plot(data$gt_bitcoin,data$avgprice)          # scatter plot
abline(lm(avgprice~gt_bitcoin,data=data))     # adding regression line to the scatter plot



################### Forecasting bitcoin price using google trends data 2  ##########
### (Can we predict bitcoin price using one-week prior google trends data?)##############

### Creating one-week lagged google trends variable
data$gt_bitcoin_1 <- c(0,data$gt_bitcoin[1:nrow(data)-1])


### Creating one-week lagged bitcoin price variable (controlling for recent trend)
data$avgprice_1 <- c(0,data$avgprice[1:nrow(data)-1])


### Excluding the first data point for regression analysis because the first datapoints don't have lagged
variables
data_1<-data[-1,]


### Regression
ans2<-lm(avgprice~avgprice_1+gt_bitcoin_1,data=data_1)
summary(ans2)
```

SID: 1810980
Gold Cohort

```
################### Forecasting bitcoin price using google trends data 3 ###
#####(Let's try google trends data of different search terms.) ####################

google.trends = gtrends(c("blockchain"), gprop = "web", time = "2017-01-08 2019-01-09")[[1]]

data<-cbind(data,google.trends$hits)
names(data)[6]<-"gt_blockchain"

### Creating one-week lagged googld trends variable
data$gt_blockchain_1 <- c(0,data$gt_blockchain[1:nrow(data)-1])

### Excluding the first data point for regression analysis because the first datapoints don't have lagged variables
data_1<-data[-1,]

### Regression
ans3<-lm(avgprice~avgprice_1+gt_bitcoin_1+gt_blockchain_1,data=data_1)
summary(ans3)
```