Name-Fenil Tailor Netid-fst216

Assignment-4

Q-1 A)

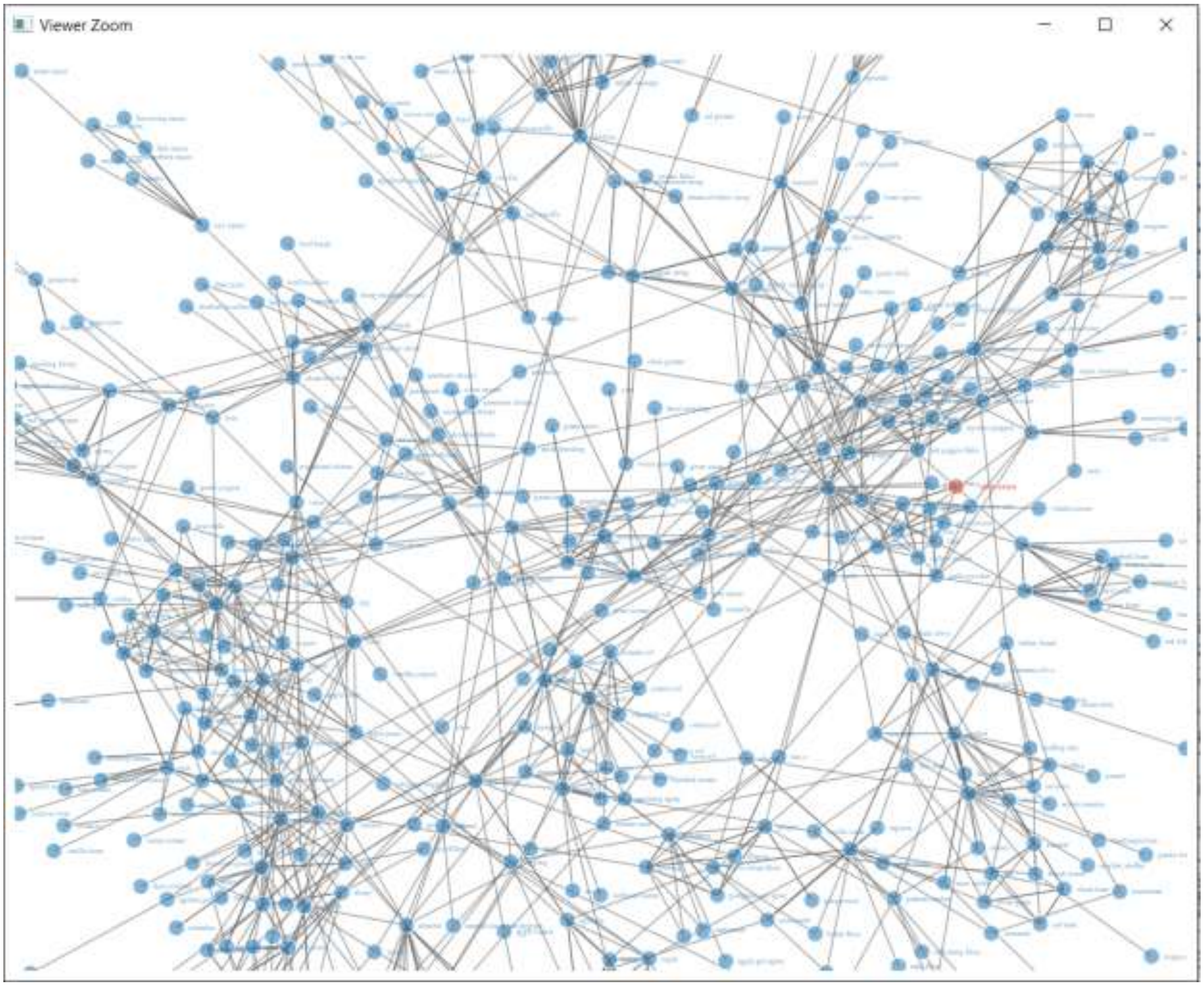
a) Read in the files and visualize the network (try using ggplot2 or networkD3 libraries).

Ans:

Code:

```
#Read the subs file over here
edges<-read.table(file.choose(),sep="\t")
colnames(edges)<-c("idsrc","iddest")

#Read the keys file over here
nodes<-read.table(file.choose(),sep="\t")
colnames(nodes)<-c("id","item")

require(sqldf)

#for changing the name of ids(Source ID,Destination id) to the Item Name
sourcenodenames<-sqldf("select item from edges,nodes where id=idsrc")
destinationnodenames<-sqldf("select item from edges,nodes where id=iddest")

#Created a frame called total which has item names for source and destination
total <- cbind(sourcenodenames,destinationnodenames)
colnames(total)<-c("itemsrc","itemdest")

#ploted graph using networkD3
networkdata<-data.frame(total$itemsrc,total$itemdest)
library(networkD3)
simpleNetwork(networkdata)
```

Output:



Conclusion:

For visualization I’ve used networkD3 library

Q-1 B)

Calculate the degree centrality of each node.

Ans.

Code:

```
install.packages("igraph")
```

Name-Fenil Tailor Netid-fst216

```
library(igraph)

net <- graph.data.frame(edges, nodes, directed=T)

#plot(net, edge.arrow.size=.4)

#net <- simplify(net, remove.multiple = F, remove.loops = T)


#calculating centrality for each node in graph

degreecentrality<-centralization.degree(net)


#Calculating count of indegree+outdegree

degcentdf<-data.frame(degreecentrality$res)

View(degcentdf)
```

Output:

	degreecentrality.res
1	3
2	33
3	4
4	8
5	1
6	19
7	1
8	5
9	5
10	5
11	5
12	29
13	5
14	19
15	22
16	7
17	13
18	9

Showing 1 to 18 of 562 entries

Conclusion:

Created the degree centrality(indegree+outdegree) for each node using centralization.degree() function.

Node 1’s total(indegree+outdegree) is 3

Node 2’s total(indegree+outdegree) is 33 e.t.c

Q-1D) Which are the most “connected” node(s).

Ans:

Code:

```
net <- graph.data.frame(edges, nodes, directed=T)

#calculating centrality for each node in graph

degreecentrality<-centralization.degree(net)


#Calculating count of indegree+outdegree

degcentdf<-data.frame(degreecentrality$res)

View(degcentdf)

#As we can see from data frame that node no with ID 2(degree-33),12(degree-29),79(degree-29) are the most three connected nodes
```

Conclusion:

We can see from data frame(degcentdf) that node no with ID 2(degree-33),12(degree-29),79(degree-29) are the most three connected nodes(Here I have taken only top 3 most connected nodes)

Name-Fenil Tailor Netid-fst216

Q-1C) Visually determine what are the furthest ingredients from cocoa powder?

Ans.

Code:

```
#tmp2 = get.shortest.paths(net, from='408', to='142')

#fromcoca<-shortest_paths(net, from=408)


#Created distMatrix that showa the distance to all other nodes from node cocoa powder(408)

distMatrix <- shortest.paths(net, v=408)

max<-0

coll<-c()

cnt<-0

#for(i in 1:562){

#  if(distMatrix[i][1]==1 && !is.na(distMatrix[i][1]))

#  {

#  cnt<-cnt+1

#  }

#}

#Pls empty the coll first and then run the for loop to get required output


#Visually we can see that the nodes which are furthest from node cocoa powder are that nodes which are not connected to it

#Created a simple for loop to verify the results which I got from visualization for the query

for(i in 1:562){

  if(distMatrix[i][1]==Inf && !is.na(distMatrix[i][1]))

  {

  cnt<-cnt+1
```

```
  coll<-union(coll,c(i))
  }
}
coll
require(sqldf)
collnames<-sqldf("select item from nodes where id in
(93,94,107,108,109,110,111,113,114,256,353,431,432,459,460,463,464,480,481)")
```

Output:

```
#These no of nodes are furthest from node cocoa powder
#[1]  93  94 107 108 109 110 111 113 114 256 353 431 432 459 460 463 464 480 481

#                item
#1         marshmallow
#2    marshmallow creme
#3       yellow mustard
#4        honey mustard
#5              mustard
#6        dijon mustard
#7  spicy brown mustard
#8               mussel
#9                 clam
#10      mustard powder
#11        mustard seed
#12          pound cake
#13     angel food cake
#14           toothpick
#15              skewer
#16          baking mix
```

#17 pancake mix

#18 avocado

#19 guacamole

Conclusion:

The nodes which are not connected to the node no 408(cocoa powder) are the furthest node from node no-408.We are getting 19 such nodes in our graph.

Q-2A)

Download 100 users ids that have tweeted about this, and their friends/followers. Note that due to rate limits you may need to include a pause in order to be able to download data on this many users

Code:

setup credentials: note you need to change to your own credentials

```{r}

require(twitteR)

require(RCurl)

consumer_key <-'x3DkrJTjJ1PjMAJx3HfCgJQya'

consumer_secret <-'XD3tQ5eODm7lCW9bhn2Ptg4oJEBLWrCW6ShDzrVIRde5urbbXw'

access_token <-'1154193151-Fq8xxFjr90ODVEj2La9kTshvpUUd5OGLbl5Fmhp'

access_secret <-'DDgYlEuEUvTBd576VFCFe4RnNKOstpFv39rapLHlKhiwS'

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

```

search using keywords:

I use 10 here for simplicity, large number of n will cause the program running slow or reaching the twitter request limit, result in throwing a following error: In twInterfaceObj$doAPICall("account/verify_credentials", ...) : Rate limit encountered & retry limit reached - returning partial results

Name-Fenil Tailor Netid-fst216

Change 10 to at least 200, so we would most likely get at least 100 unique users.

searchTwitter returns List

```{r, echo=FALSE}
myTweets <- searchTwitter("#SXSW2016", n=200, lang="en")
myTweet
```

Extract the usernames:

Change k = 100, if you want to process 100 users

```{r}
k = 100
tweetsDF <- twListToDF(myTweets)
nameDF <- tweetsDF[, c("screenName")]
uniqueNameDF <- unique(nameDF)
hundredNamesDF <- head(uniqueNameDF, k)
hundredNamesDF
```

use networkD3 to plot the graph, igraph to assess the degree distribution of the graph

```
{r}
# Load package
require(networkD3)
require(igraph)

network_tw <- data.frame(src=character(), target=character(), stringsAsFactors=FALSE)
```

Name-Fenil Tailor Netid-fst216

```
for(i in 1:100)

{

start <- getUser(uniqueNameDF[i],retryOnRateLimit=900)
```

#I used retryOnRateLimit=900 when it reached to it's maximum allowed rate limit to get User,Friends and followers.

```
  friends.object<-lookupUsers(start$getFriendIDs(retryOnRateLimit=900))

  follower.object<-lookupUsers(start$getFollowerIDs(retryOnRateLimit=900))


  n<- length(friends.object)

  m<- length(follower.object)


  friends <- sapply(friends.object[1:n],screenName)

  followers <- sapply(follower.object[1:m],screenName)


  networkData <- data.frame(src=uniqueNameDF[i], target=friends)

  network_tw <- merge(network_tw, networkData,  all=T)

  networkData <- data.frame(src=followers, target=uniqueNameDF[i])

  network_tw <- merge(network_tw, networkData,  all=T)

}
```

Output:

Note---Here I have taken **retryOnRateLimit=900** argument for getting User,Friends and their Followers.If rate limit crosses the threshold it would repeat the same function(900 times—for ensuring that we get enough data).

Then after I build a network of friends and followers and feed them into a dataframe called network_tw

Name-Fenil Tailor Netid-fst216

	src	target
491	GDiddy210	WillyO_LA
492	GDiddy210	WolfeNamedCrizz
493	GDiddy210	wooaahnelly
494	GDiddy210	WxLLxxM
495	GDiddy210	x_Missjassy_
496	GDiddy210	XXL
497	GDiddy210	YeahReQ
498	GDiddy210	Yesjulz
499	GDiddy210	yng_kara
500	GDiddy210	YoShowtime
501	GDiddy210	YoungRicans
502	GDiddy210	YoungZillaGTM
503	GDiddy210	YouOttoKnow
504	GDiddy210	YSF_Mel
505	GDiddy210	YSFSean
506	GDiddy210	YungBarnes
507	GDiddy210	yungdiddy_
508	GDiddy210	YungHandsome19

Showing 491 to 509 of 27,623 entries

I got **27,623** for 100 users that I have taken from tweetsDF.

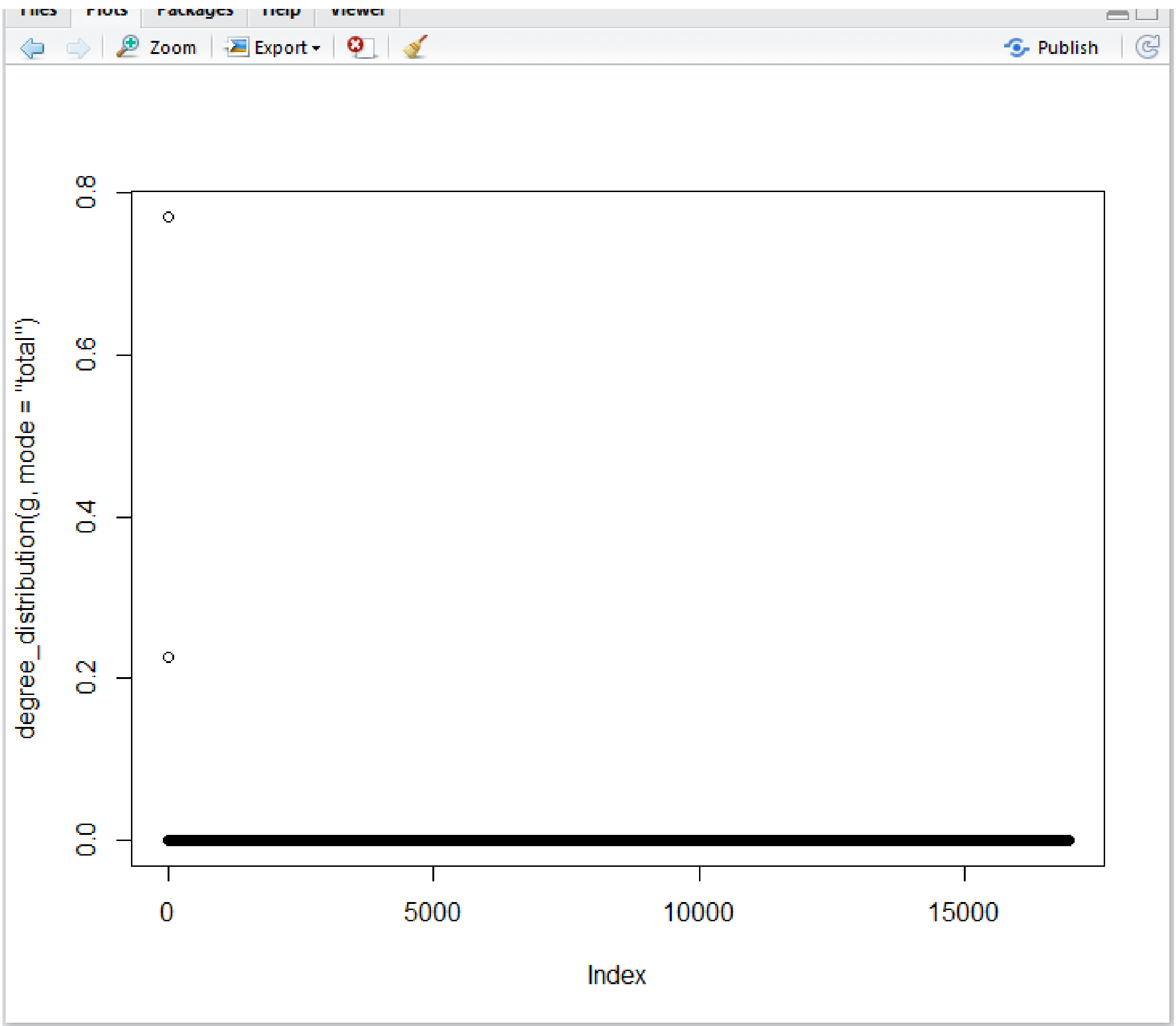
Q-2B) Assess and plot the degree distribution of your network (choose either in-degree or out-degree and motivate why you chose the metric).

Code:

```
# degree assess

g <- graph.data.frame(network_tw, directed = F)


degree(g, mode = "total")


degree_distribution(g)

plot(degree_distribution(g,mode="total"))
```

Output:

Name-Fenil Tailor Netid-fst216



Conclusion:

I have taken mode=total(indegree+outdegree).The reason for choosing total degree is that we can get Complete visualization of each user's friends and followers.If I only choose indegree than I can only be a le to get distribution details of each users followers and If I only choose outdegree than I can only be abl e to get details of each users friends.

Q-2C) Visualize the network (try using ggplot2 or networkD3 libraries).

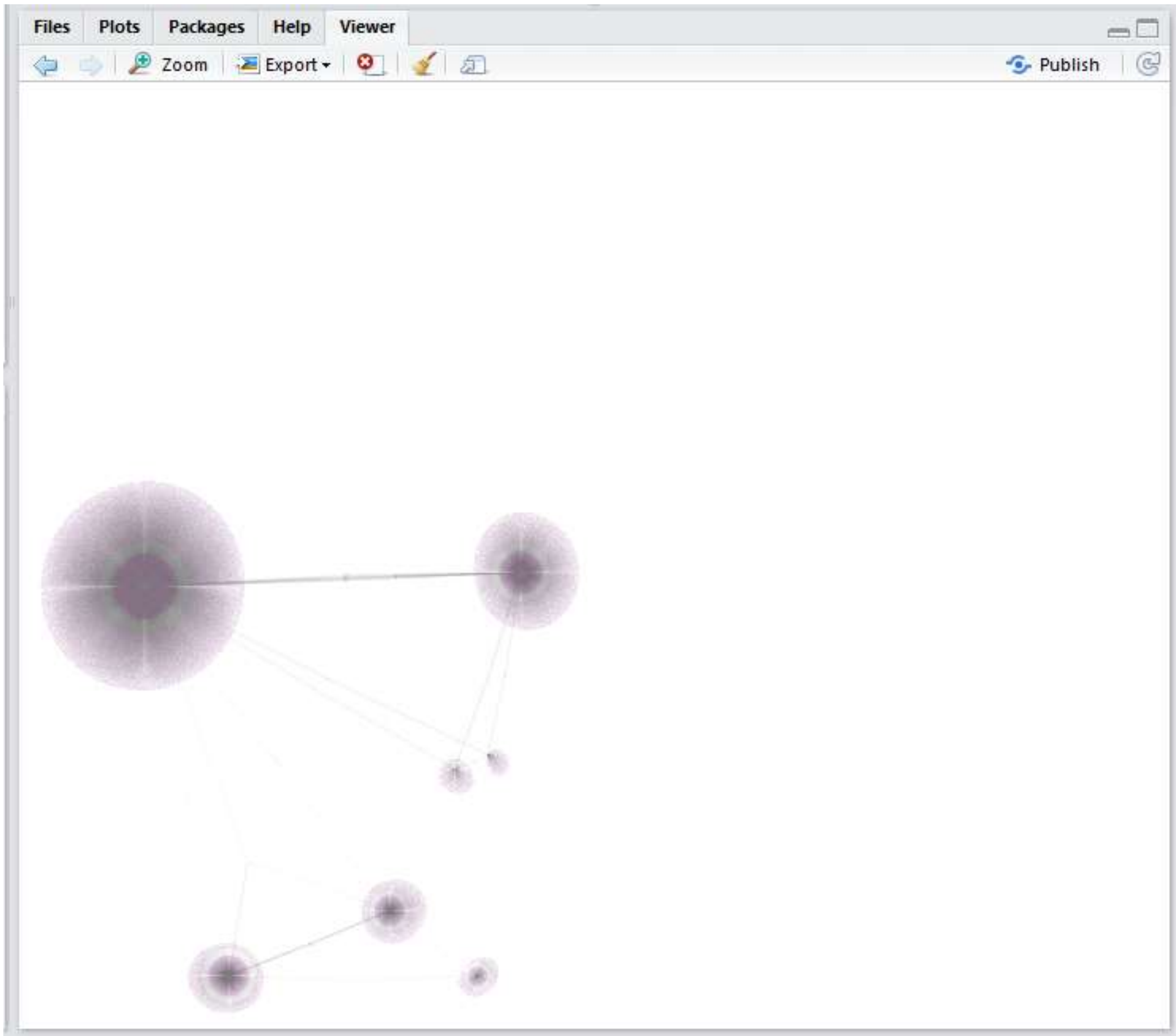
Plot

simpleNetwork(network_tw, zoom = T, linkDistance = 60, opacity = 0.5, linkColour = "grey", nodeColour = "purple",

Name-Fenil Tailor Netid-fst216

```
nodeClickColour = "red", textColour = "blue")
```

Output:



Conclusion

Note---Request to wait for approximately 10-15 minutes to load complete graph after running the script.

I have pasted the zoom out version of the graph so that we can able to see the overall picture of user,friends and their followers(it is for 100 users).

Name-Fenil Tailor   Netid-fst216

Q-3A) Download Tweets from each user above that mention the hashtag you selected (over an appropriate time period)?

Code:

```
From <- "from:"

sxsw <- "+#SXSW2016"


tweets_list <- data.frame(src=character(), target=character(), stringsAsFactors=FALSE)


#looping to get the tweets from specific user by getting the hundredNamedDF[j](unique usernames) and searching tweets that they've put in recent times

for(j in 1:100)

{

  if(j %% 5 == 0){Sys.sleep(600)}

#Pausing the R execution if the rate limit reached to the specific threshold

    tweets_object<-do.call("rbind",lapply(searchTwitter(paste(From,hundredNamesDF[j],sxsw,sep = ""),resultType = "recent",lang="en"), as.data.frame))

    tweets_list <- merge(tweets_list, tweets_object,  all=T)

}

tweetsDF<-tweets_list
```

Conclusion:

For this problem I’ve used following function

```
tweets_object<-do.call("rbind",lapply(searchTwitter(paste(From,hundredNamesDF[j],sxsw,sep = ""),resultType = "recent",lang="en"), as.data.frame))
```

This line of code will get the tweets from specific user that he/she has posted in recent times and then after I merge all user’s tweets in the recent times into the data frame called tweets_object

Name-Fenil Tailor Netid-fst216

Output:

	text	favorited	favoriteCount	replyToSN	created	truncated	replyToSID	id	replyToUID
1	#Austin, TX #Engineering #Job: Power Electronics En...	FALSE	0	NA	2016-04-11 22:08:48	FALSE	NA	719648421003419648	NA
2	#Austin, TX #Engineering #Job: Power Electronics En...	FALSE	1	NA	2016-04-11 22:08:48	FALSE	NA	719648421003419648	NA
3	#Clerical alert: Legal Intern | SunPower | #Austin, TX ...	FALSE	0	NA	2016-04-11 20:43:36	FALSE	NA	719626980509949953	NA
4	#Clerical in #Milpitas, CA: Mechantronics Intern at Su...	FALSE	0	NA	2016-04-12 04:23:30	FALSE	NA	719742717991919616	NA
5	#Finance #Job alert: General Ledger Accountant | Su...	FALSE	1	NA	2016-04-11 08:20:39	FALSE	NA	719440010303307776	NA
6	#IT #Job in #BinanROHQSPSS: O2C Business Proces...	FALSE	1	NA	2016-04-11 23:31:50	FALSE	NA	719669314521923584	NA
7	#Sales in #BinanROHQSPSS: Partner Support Repres...	FALSE	0	NA	2016-04-12 04:47:46	FALSE	NA	719748825431183360	NA
8	#SanJose, CA #Construction : Sales Instructional Des...	FALSE	1	NA	2016-04-11 15:13:50	FALSE	NA	719543988621959168	NA
9	#SanJose, CA #HR #Job: Senior Director, Talent Mana...	FALSE	0	NA	2016-04-11 16:27:27	FALSE	NA	719582515634458624	NA
10	Austin was LITTT with @khofmoscrill Channel K: #SX...	FALSE	7	NA	2016-04-11 21:11:03	FALSE	NA	719633885362065408	NA
11	Austin was LITTT with @khofmoscrill Channel K: #SX...	FALSE	3	NA	2016-04-11 21:52:15	FALSE	NA	719644254432141312	NA
12	Can you recommend anyone for this #Construction ...	FALSE	1	NA	2016-04-12 01:33:20	FALSE	NA	719699891656597504	NA
13	Can you recommend anyone for this #Construction ...	FALSE	1	NA	2016-04-11 15:38:21	FALSE	NA	719550162016731136	NA
14	Can you recommend anyone for this #Construction ...	FALSE	2	NA	2016-04-11 12:47:51	FALSE	NA	719507252348977152	NA
15	Can you recommend anyone for this #Engineering #j...	FALSE	0	NA	2016-04-11 20:57:12	FALSE	NA	719630403754004481	NA
16	Can you recommend anyone for this #Engineering #j...	FALSE	1	NA	2016-04-12 03:52:11	FALSE	NA	719734835774787584	NA
17	Can you recommend anyone for this #Finance #Job? ...	FALSE	1	NA	2016-04-12 04:11:20	FALSE	NA	719739655436939264	NA

	replyToUID	statusSource	screenName	retweetCount	isRetweet	retweeted	longitude	latitude	src	target
21003419648	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-97.7036561	30.3870686	NA	NA
21003419648	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-97.7036561	30.3870686	NA	NA
80509949953	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-97.7036561	30.3870686	NA	NA
17991919616	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-121.8995741	37.4323341	NA	NA
10303307776	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	102.1821816	2.4472386	NA	NA
14521923584	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	121.774017	12.879721	NA	NA
25431183360	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	121.774017	12.879721	NA	NA
88621959168	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-121.8854337	37.3904943	NA	NA
15634458624	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-121.9529992	37.4308503	NA	NA
85362065408	NA	<a href="http://twitter.com/download/iphone" rel="...	TaylorGang	1	FALSE	FALSE	NA	NA	NA	NA
54432141312	NA	<a href="http://twitter.com/download/iphone" rel="...	TaylorGang	1	FALSE	FALSE	NA	NA	NA	NA
91656597504	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-97.7036561	30.3870686	NA	NA
62016731136	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-97.7036561	30.3870686	NA	NA
52348977152	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	1	FALSE	FALSE	NA	NA	NA	NA
03754004481	NA	<a href="http://www.tweetmyjobs.com" rel="nofollo...	HydeCareers	0	FALSE	FALSE	78.486671	17.385044	NA	NA
35774787584	NA	<a href="http://www.tweetmyjobs.com" rel="nofollo...	HydeCareers	0	FALSE	FALSE	-8.6304975	52.6680204	NA	NA
55436939264	NA	<a href="http://tweetmyjobs.com" rel="nofollow">Sa...	SunPowerTalent	0	FALSE	FALSE	-121.9529992	37.4308503	NA	NA

Q-3B) For n = 1, n = 2 and n = 3, submit the list of the 10 most frequent sequences

Ans:

Code:

```
myTweets <- searchTwitter("#SXSW2016", n=200, lang="en")

tweetsDF <- twListToDF(myTweets)


install.packages("tm")

library(tm)

tweets_source <- VectorSource(tweetsDF$text)
```

```
corpus <- Corpus(tweets_source)


corpus <- tm_map(corpus, removePunctuation)


corpus <- tm_map(corpus, stripWhitespace)


corpus <- tm_map(corpus, removeWords, stopwords("english"))


dtm <- DocumentTermMatrix(corpus)
dtm2 <- as.matrix(dtm)
#For n=1(unigram)
frequency <- colSums(dtm2)
frequency <- sort(frequency, decreasing=TRUE)
#Output
frequency[1:10]


#For n=2(Bigram)
BigramTokenizer <-
  function(x)
    unlist(lapply(ngrams(words(x), 2), paste, collapse = " "), use.names = FALSE)


tdm <- TermDocumentMatrix(corpus, control = list(tokenize = BigramTokenizer))
tdm2 <- as.matrix(tdm)


frequency_2 <- rowSums(tdm2)


frequency_2 <- sort(frequency_2, decreasing=TRUE)
#Output
frequency_2[1:10]
```

```
#For Trigram
trigramTokenizer <-
  function(x)
    unlist(lapply(ngrams(words(x), 3), paste, collapse = " "), use.names = FALSE)

tdm_3 <- TermDocumentMatrix(corpus, control = list(tokenize = trigramTokenizer))
tdm2_3 <- as.matrix(tdm_3)

frequency_3 <- rowSums(tdm2_3)
frequency_3 <- sort(frequency_3, decreasing=TRUE)
#Output
frequency_3[1:10]
```

Output:

For n=1

```
sxsw2016       austin      sxsweco          job         sxsw
192           49           41               36            26

channel khofmoscrill        littt     sunpower         work
23           23           23             20                20
```

```
For n=2

sxsweco sxsw2016    austin littt    channel k              k sxsw2016
41                          23            23                     23

khofmoscrill channel littt khofmoscrill austin rt    httpstcozvp3uu09aw tgod
23                     23                 15          15

rt ganggang sxsw2016 httpstcozvp3uu09aw
15                     15


For n=3

austin littt khofmoscrill  channel k sxsw2016      khofmoscrill channel k
23                              23                     23


littt khofmoscrill channel
23
```

```
austin rt ganggang httpstcozvp3uu09aw tgod Austin k sxsw2016 httpstcozvp3uu09
15                     15                             15


sxsw2016 httpstcozvp3uu09aw tgod
15

tgod austin rt      realtaylorgang austin littt
15                     14
```

Q-3C)

c) For n = 1, n = 2 and n = 3, submit the sum of all frequencies of all seque
nces for   that n.

Code:

```
sum(frequency_3)
sum(frequency_2)
sum(frequency)
```

Output:

```
sum(frequency_3)
[1] 1967
> sum(frequency_2)
[1] 2167
> sum(frequency)
[1] 2057
```

Conclusion:
As we can see that for n=1 sum of all frequencies is 2057(as we have removed stopwords(“English) and punctuation we got less count for n=1)

For n=2 we got 2167 frequency sum count

And For n=3 we got 1967 frequency sum count

Q-3D)

Using these frequencies, generate a distance measure for individuals (e.g. th
ey   share the X most common frequency 3-gram terms, or 2-gram terms, or 1-g
ram term). How does this network look compared to the network generated in q
uestion 2?

Ans:
Code:

```
myTweets <- searchTwitter("#SXSW2016", n=200, lang="en")
k = 100
tweetsDF <- twListToDF(myTweets)
nameDF <- tweetsDF[, c("screenName")]
```

```
uniqueNameDF <- unique(nameDF)
hundredNamesDF <- head(uniqueNameDF, k)
networkdata1 <- data.frame(src=character(), target=character(),distance=numer
ic(),stringsAsFactors=FALSE)

for(i in 1:100)
{
  for(j in 1:100)
  {
  networkdata2<-data.frame(src=uniqueNameDF[i],target=uniqueNameDF[j],strings
AsFactors=FALSE)
  networkdata1<-merge(networkdata1, networkdata2,  all=T)
  }
}


tweets_frame1<-data.frame(screenname=tweetsDF$screenName,tweets=tweetsDF$text
,stringsAsFactors=FALSE)

tweets_frame1$tweets<-iconv(tweets_frame1$tweets,"UTF-8")

for (k in 1:length(networkdata1$src))
{
  networkdata1$distance[k]<-stringdist(as.character(tweets_frame1$tweets[twee
ts_frame1$screenname==networkdata1$src[k]]),as.character(tweets_frame1$tweets
[tweets_frame1$screenname == networkdata1$target[k]]), method = "qgram", q =
1)
}

View(networkdata1)
```

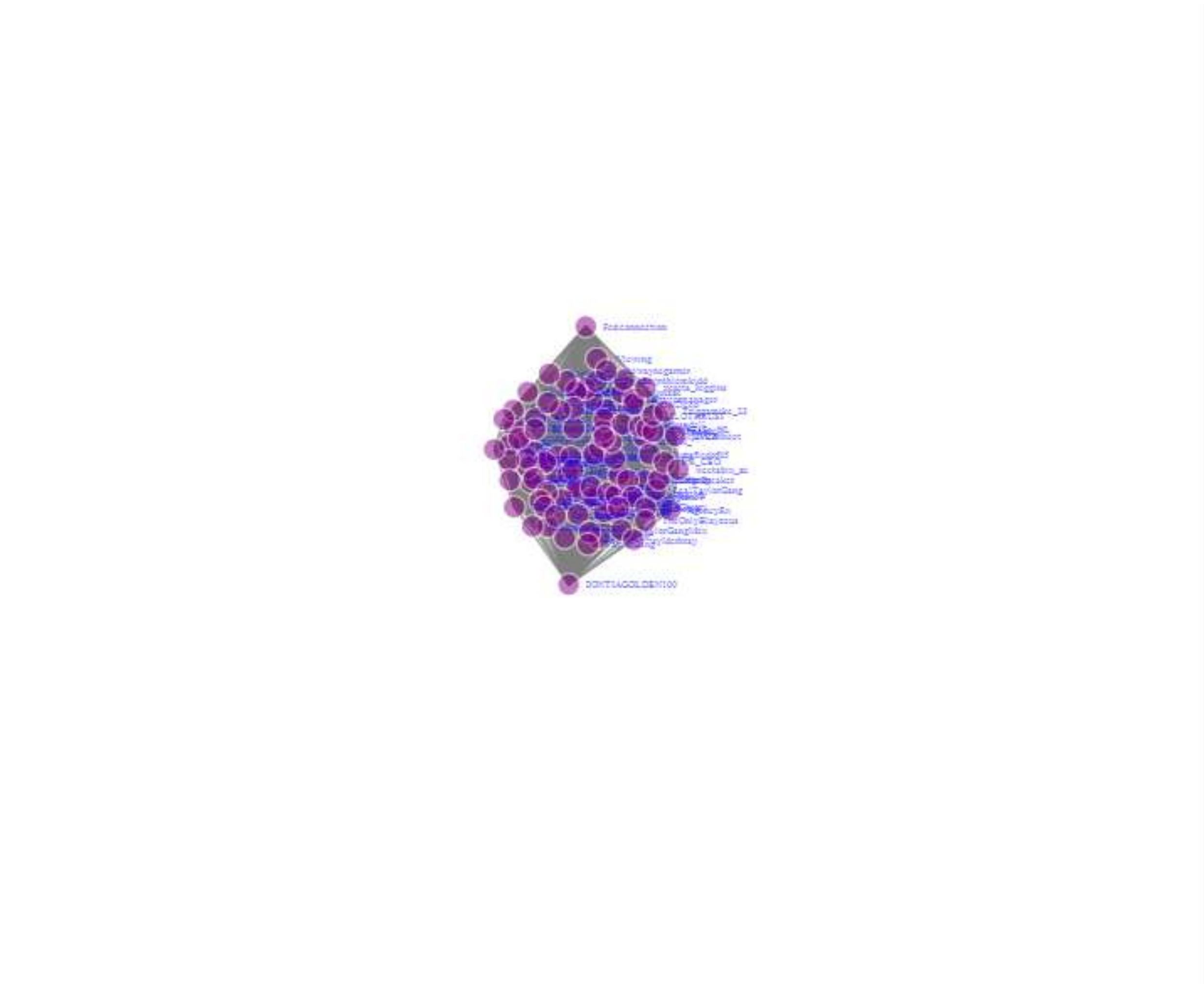
Output:

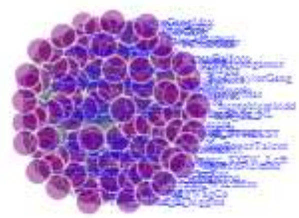
Name-Fenil Tailor Netid-fst216

Filter

	src	target	distance
1	414work	414work	0
2	414work	Adams_tgod	84
3	414work	AgencyRx	79
4	414work	ajzhang	90
5	414work	alittlesoundatx	87
6	414work	AWLoves	77
7	414work	BDUBMuzikGroup	84
8	414work	BellasNattie4vr	78
9	414work	Cardboard_VR	105
10	414work	CBS_1987	77
11	414work	Chance100k	NA
12	414work	Chi_daSHOOTA	92
13	414work	chrisreedfilm	102
14	414work	cjrutter	82
15	414work	CliveWoodger	73
16	414work	cmnewland	90
17	414work	darcmarc1736	95
18	414work	davidmartino23	96

Showing 1 to 18 of 10,000 entries





Conclusion:

We have choose n=1 for this analysis and I’ve generated 100 graphs to compute distance measure of each user. Here,I have shown just two graphs

From these Picture we can say that
1)The distance measure between tweets from user’s screen name=”414work” to user’s screen name=”Adams_tgod” is 84 which means that 84 characters are different in their tweets