####Refer to code at the following URL: http://www.nyu.edu/projects/politicsdatalab/localdata/workshops/twitter.pdf

#libraries to extract twitter data

library(twitteR)

library(ROAuth)

#libraries to parse data and plot network graph

library(RCurl)

library(RJSONIO)

library(igraph)

#set required keys to access Twitter data

consumer\_key <- '9gSseVvJXuEfSsDm3HuuizmMn'

consumer\_secret <- 'JYkSgJeHjMlPxIU8k5A2BChNGb7x75R6t6Fm1VeNjSuZiYtTem'

access\_token <- '2560297818-X7lIinJeC3EUlkSNPmCuDvXgNqgqny5xGs5etlg'

access\_secret <- '3YDvFeFHisRxQMMO1jydUkzCIsix7LDYtWPf35PZcFSn9'

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)

#function to create edge matrix given a seed user. Edge list is a list of matrices where each matrix has two columns: friends, whom the user is following and followers who are following the user. This function will create edge matrix for a given user

edge.mat <- function(username){

user <- getUser(username)

#get screen names of all the twitter friends user has and convert it into character vector. This would be the source column of our edge list matrix

friends <- user$getFriends()

friends.screen.names <- as.character(lapply(friends, function(f) f$getScreenName()))

#get screen names of all the twitter followers user has and convert it into character vector. This would be the target column of our edge list matrix

followers <- user$getFollowers()

followers.screen.names <- as.character(lapply(followers, function(f) f$getScreenName()))

#before combining friends and followers into a matrix, check if number of friends or followers is 0. If it is, replace friends or followers vectors by vector of all 0s

if(length(friends.screen.names) == 0){

friends.screen.names <- integer(length(followers.screen.names))

}

if(length(followers.screen.names) == 0){

followers.screen.names <- integer(length(friends.screen.names))

}

if(length(followers.screen.names) == 0 & length(friends.screen.names) == 0){

followers.screen.names <- integer(0)

friends.screen.names <- integer(0)

}

#create edge matrix

edge.mat <- cbind(friends.screen.names, followers.screen.names)

colnames(edge.mat) <- c('friends', 'followers')

return(edge.mat)

}

#The above function gives us edge matrix which has friends and followers of a given user. From these columns, we want to find other potential seed users to implement snowball sampling.

get.seed <- function(edge.matrix, seed){

#there is possibility to have a user as your friend and follower on twitter. In that case, we only want to extract one username and avoid dups

new.seed <- unique(c(edge.mat[, 1], edge.mat[, 2]))

#since we can potentially have 0s in our friends and / or followers columns, make sure to remove them

new.seed <- new.seed[new.seed != '0']

#finally, make sure that the original seed user is not in the new.seed vector

new.seed <- new.seed[new.seed != seed]

return(new.seed)

}

#Implement snowball sampling for a given seed user.

twitter.snowball <- function(seed, k = 2){

#get edge matrix for given seed user

seed.em <- edge.mat(seed)

#get new seeds from the original seed users friends and followers to be used in next round of snowball

new.seeds <- get.seed(seed.em, seed)

#keep a count of rounds completed

rounds <- 1

#keep a record of all the users / nodes that are already hit

all.nodes <- seed

#begin the snowball search

while(rounds < k){

next.seeds <- c()

for (user in new.seeds){

#get user information only if we have already not extracted their information in earlier round(s)

if (!user %in% all.nodes) {

#get user's edge matrix

user.em <- edge.mat(user)

#make sure that followers exist

if (dim(user.em)[2] > 0){

#combine user.em and seed.em

seed.em <- rbind(seed.em, user.em)

#compute newer seeds and add them to orinal vector of new seeds

next.seeds <- c(next.seeds, get.seed(user.em, user))

#add user to vector of all nodes

all.nodes <- c(all.nodes, user)

}

}

}

#remove dups from next seeds vector

new.seeds <- unique(next.seeds)

#remove any entries that already exist in all.nodes

new.seeds <- new.seeds[!which(new.seeds %in% all.nodes)]

#increment round counter

rounds <- rounds + 1

}

#remove any possible dups from seed.em

seed.em <- seed.em[!duplicated(seed.em), ]

return(graph.edgelist(seed.em))

}

graph <- twitter.snowball('datascientist17')

write.graph(graph, 'data/datascientist17/datascientist17\_net', format = c('graphml'))