```
library(igraph)
library(dplyr)


rm(list=ls())
getwd()
setwd("C:/Users/Jess/Dropbox/Grad School/Text Mining/Final_Project")

#Read in CSV files
edges<-read.csv("Edges and Weights.csv")
verts <- read.csv("Vertices (1).csv")

#Keep to 50 tweeters by follower count
verts_50<-head(arrange(verts,desc(followers)), n = 50)

#Create colors variable
verts_50$colors<-ifelse(verts_50$color=="Clinton","blue", ifelse(verts_50$color=="Trump","red", "white"))

#creating percent tweet direction variable
edges$perc <- with(edges, weight/total)

#subset of tweets less than 10% weight towards 1 candidate
edges_small <- subset(edges, perc <= 0.1 ) #23 edges less than 10%

#drop small edges
edges <- subset(edges, perc>0.1)

#keep edge variables of interest
links<-edges[,c(7,8,4,9)]
head(links)
links[1:2]<- lapply(links[1:2], as.numeric)

#Keep nodes variables of interest - 50 subset
nodes_50<-verts_50[,c(4,7,8,10,11,3)]
nodes_50[1]<- lapply(nodes_50[1], as.numeric)

#drop edges from vertices not in nodes_50
links_50 <- links[ links$vertex %in% nodes_50$vertex, ]

# check unique nodes & from-to combinations for 50 subset
nrow(nodes_50); length(unique(nodes_50$vertex))
nrow(links_50); nrow(unique(links_50[,c("vertex", "candidate_vertex")]))
head(nodes_50)


#Graph with top 50 tweeters subset
#Convert edges and nodes into graph data
g <- graph_from_data_frame(d=links_50, vertices=nodes_50, directed = FALSE)
g

plot(g)

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

colrs <- c("blue", "red", "white")
V(net)$color <- V(net)$colors
E(net)$width <- E(net)$weight/100
V(net)$size <- log(V(net)$sphere_of_infl)
V(net)$label <- V(net)$name
E(net)$edge.color <- "gray80"
l <- layout_with_dh(net)
l <- norm_coords(l, ymin=-1, ymax=1, xmin=-1, xmax=1)

#plot(net, vertex.label=NA, rescale=F, layout=l*1.5, edge.curved=0.1)

plot(net, vertex.label=ifelse(V(net)$color=="white", V(net)$label, NA), rescale=F, layout=l*1.5, edge.curved=0.1)

legend(x=-2, y=-0.5, c("Clinton","Trump", "Neutral"), pch=21,
       col="#777777", pt.bg=colrs, pt.cex=2, cex=.8, bty="n", ncol=1)


#Checkout different layouts
layouts <- grep("^layout_", ls("package:igraph"), value=TRUE)[-1]
# Remove layouts that do not apply to our graph.
layouts <- layouts[!grepl("bipartite|merge|norm|sugiyama|tree", layouts)]

par(mfrow=c(3,3), mar=c(1,1,1,1))
for (layout in layouts) {
  print(layout)
  l <- do.call(layout, list(net))
  plot(net, edge.arrow.mode=0, layout=l, main=layout) }
par(mfrow=c(1,1))
```