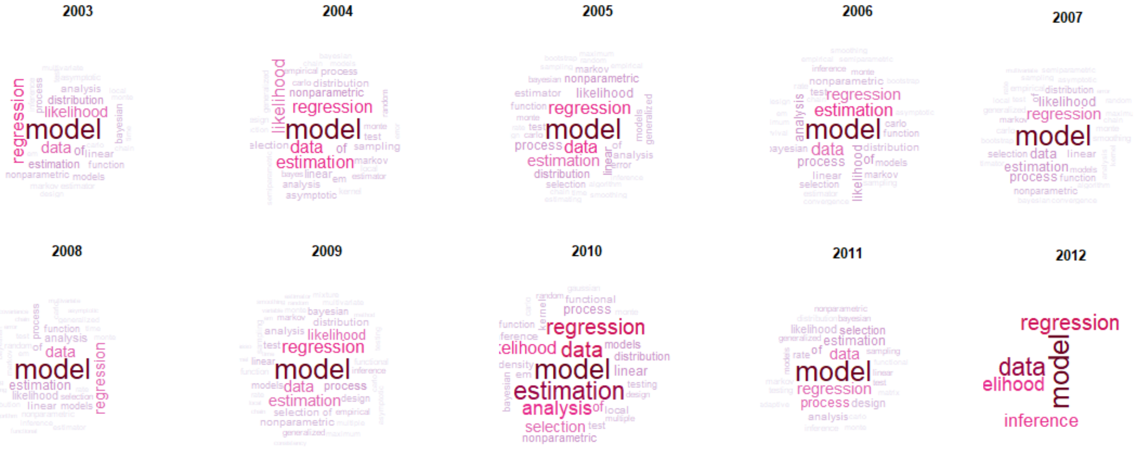
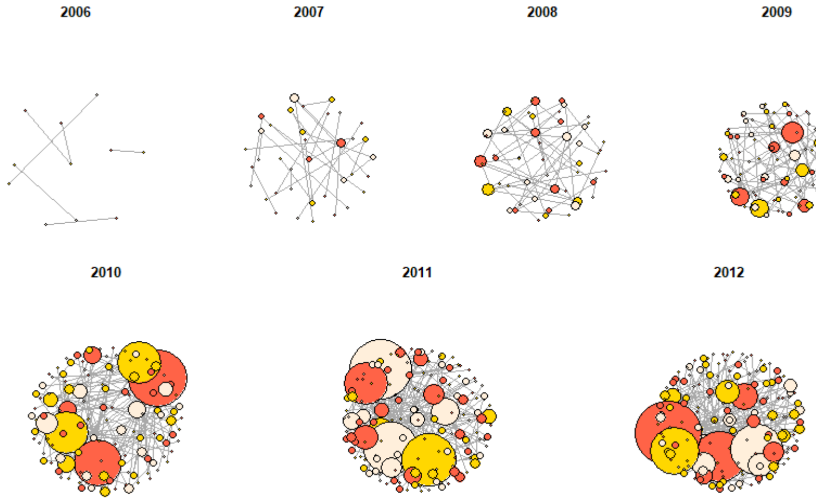
Problem 1

Problem 2

2.1) The word clouds for all of the years of abstract keywords only shows keywords that have a minimum frequency of 20 and excludes English stopwords. “model” is #1 word to include in abstract keywords, followed by the words “regression”, “data”, “estimation”, and “likelihood”.

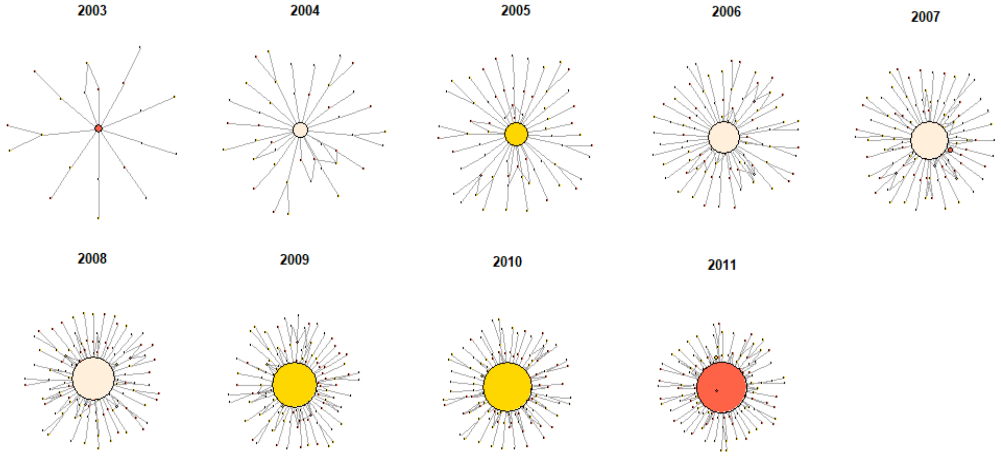


2.2) Part A: Statistician Chosen: Peter Hall



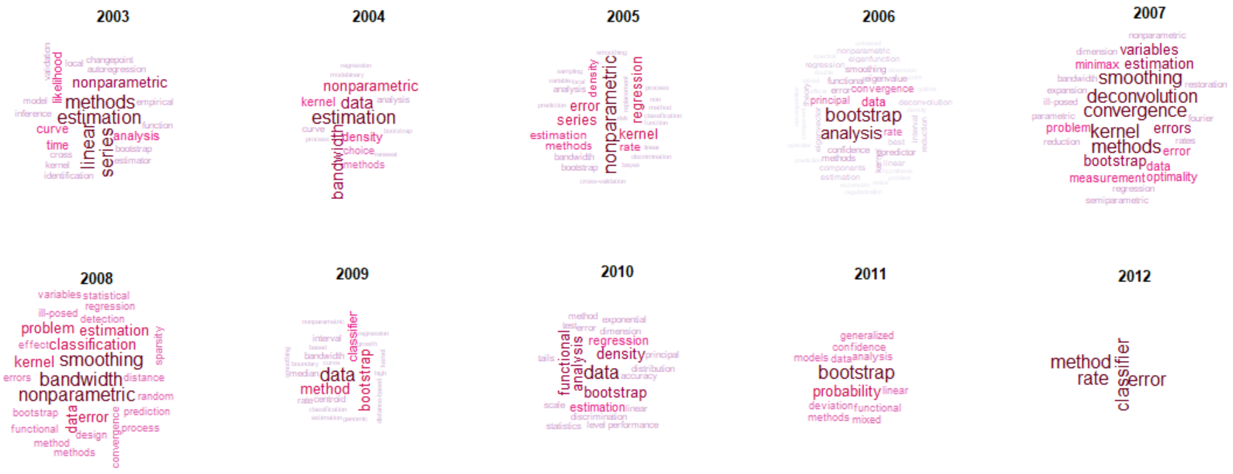
Other than the increase in network over time, the increasing sizes of some of the nodes indicate the amount of citations Hall had with those publications

2.2) Part B: Peter Hall has no new collaboration after 2011.



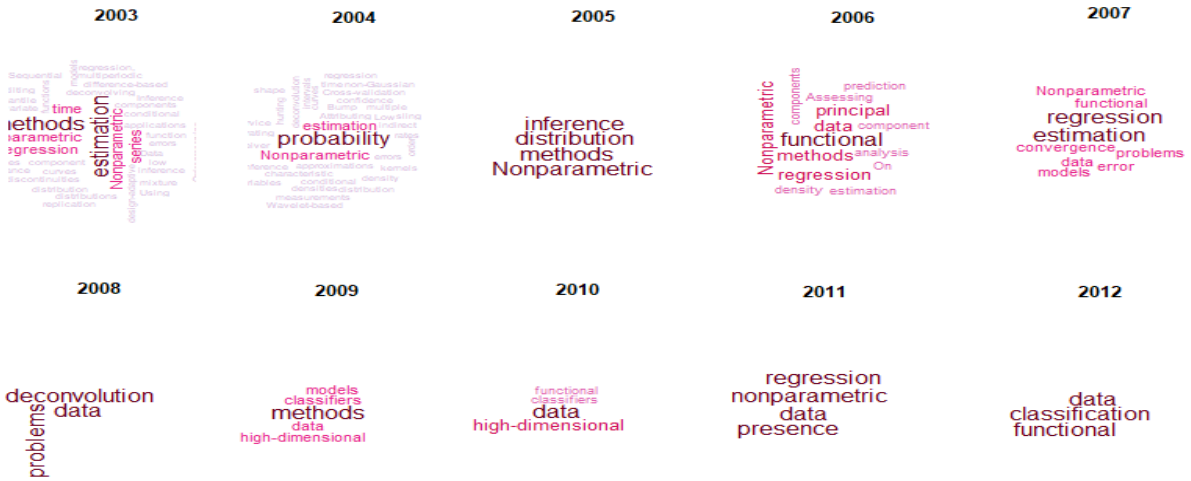
After 2005, his collaboration network expands rapidly and slows down around 2010. Unlike the citation network that has a few vertices with high weightings, this network has one single major vertex connected to other minor vertices, causing it to have the highest weight. This indicates that Hall was very active with publications.

2.2.) Part C:



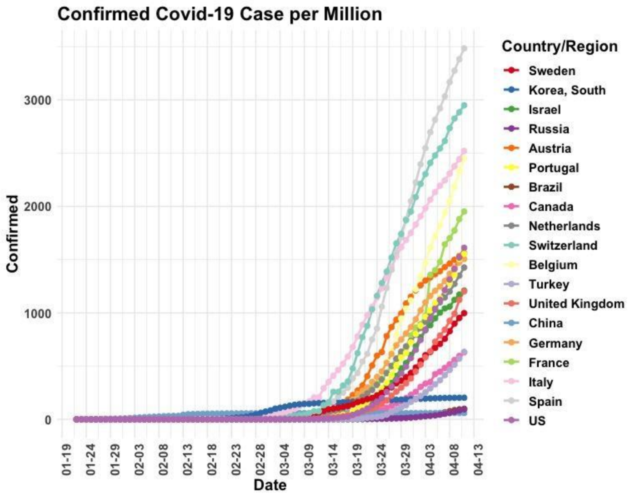
Peter Hall’s trend for keywords is not similar to the global trend. His keywords are focused on different topics over time. In order of the time period from early to late 2000s, the popular keywords seem to be: “estimation”, “nonparametric”, “methods”, “bandwidth”, “data”, “bootstrap”, and “rate”.

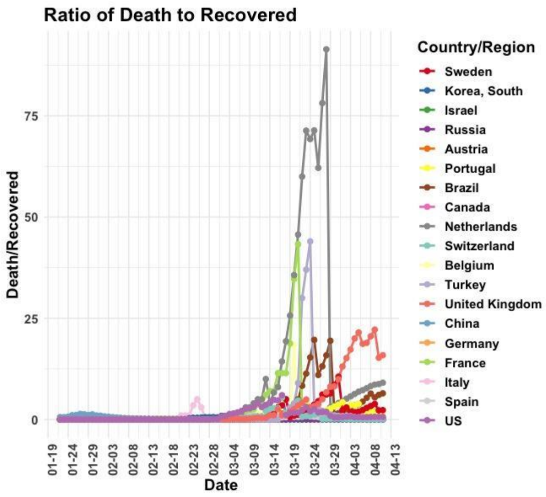
2.2.) Part D:



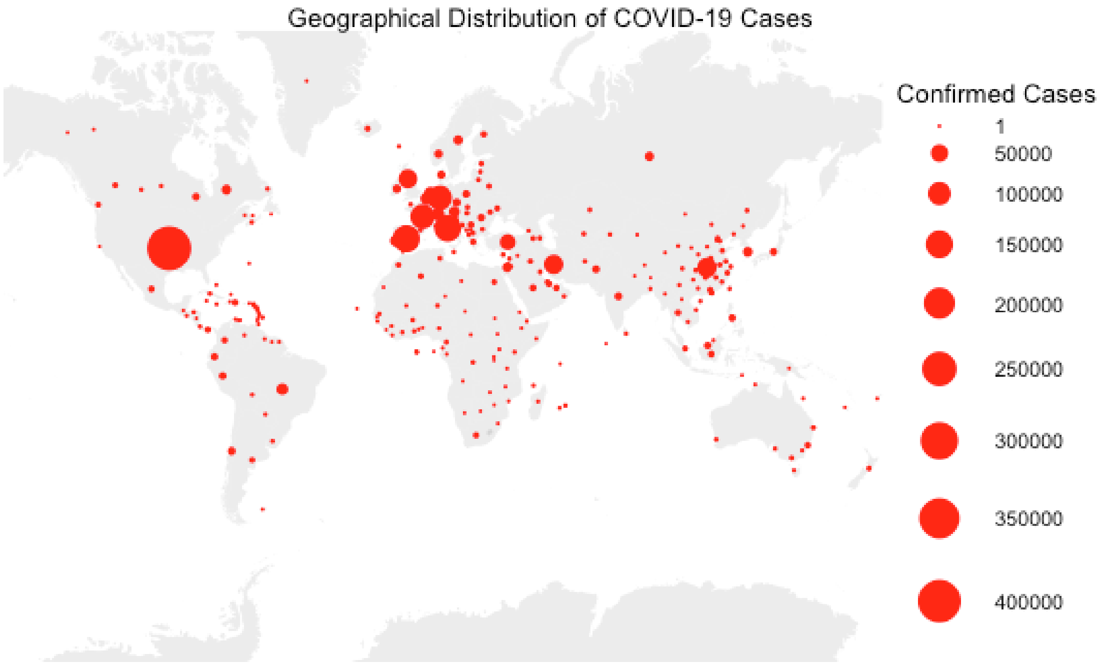
In comparison to C, Hall’s title clouds in D shows a less variety of keywords, sometimes limited to 3 or 4 (such as for 2005, 2008, 2010 - 2012). This is likely due to the limited data generated by Google Scholar that include Peter Hall as a coauthor. The search engine might not register him as a contributor, hence limiting the title search to only the publications that include him as a main author. In addition, keywords are used to highlight the important concepts of the publication while the title words are used to tell the work done, so they might not be perfectly lined up.

Problem 3


Confirmed Covid-19 Case per Million

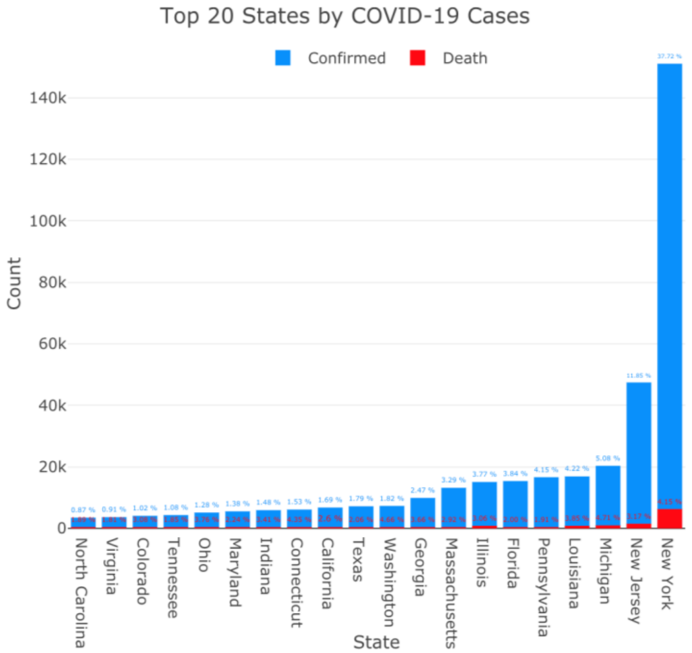

Ratio of Death to Recovered

**Added 1 to all number of recovered cases to mitigate the problem where recovered = 0 (got approval from TA)

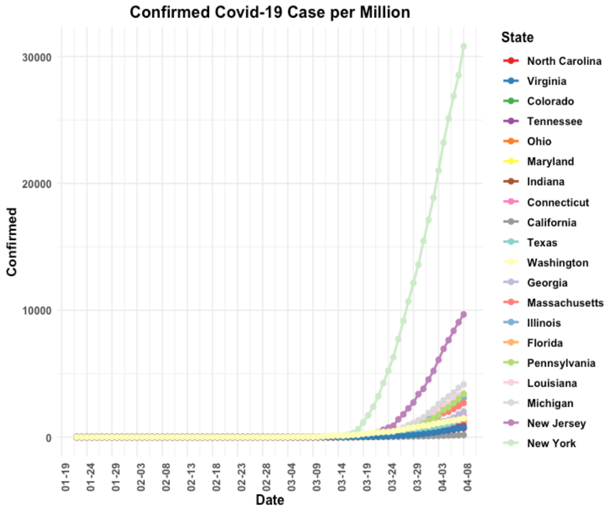

Geographical Distribution of COVID-19 Cases

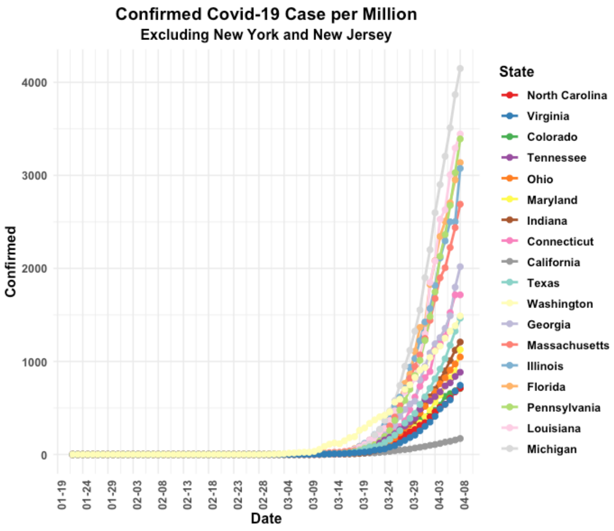
**US states data were removed by John Hopkins

The bubble map above demonstrates areas where confirmed COVID-19 cases were recorded as of April 8th, 2020. From the map, we could see that the United States and Europe have been hit particularly hard, followed by China, the Middle East, and South America.



In late March, the United States became the new epicenter of COVID-19 as its death count surpassed that of Europe and China. As its total COVID-19 cases made up 30% of all global cases on record, we investigated the top 20 U.S. states by COVID-19 Cases. From the graphs above, we can see that New York and New Jersey make up approximately 49% of total U.S. cases as of April 8th. This may be due to initial travel from Europe as well as the high density in New York City, where millions live and work–with some commuting between New York and New Jersey. It is also due to New York conducting the most tests per million people. However, despite having the most confirmed COVID-19 cases in the U.S., these two states are not necessarily the deadliest as Michigan, Washington, and Connecticut have higher death rates as of April 8th–potentially due to inadequate emergency response at the local level and/or healthcare systems.





For now, the coronavirus appears more concentrated in urban centers, primarily New York City and its surrounding suburbs, as well as Seattle, Boston, and San Francisco. Metro centers all over the US — Detroit, New Orleans, Miami (located in Michigan, Louisiana, and Florida respectively) have seen a spike in coronavirus cases. Their peak has not yet arrived.

Appendix

Problem 1

```
##airport data transformation
##generate global airline network data

##read in files
airport <- read.table("airports.txt", header = F, sep = ",")
routes <- read.table("routes.dat", header = F, sep=",")

library(maps)
library(igraph)

map("world", col="grey30", fill=TRUE, bg="black", lwd=.1)

# Add a point on the map for each airport:
points(x=airport$V8, y=airport$V7, pch=19, cex = .1, col="orange")

library(geosphere)

for(i in 1:nrow(routes))  {
  node1 <- airport[airport$V1 == routes[i,]$V4,]
  node2 <- airport[airport$V1 == routes[i,]$V6,]

  if( nrow(node2)==0 | nrow(node1)==0 )
    next

  arc <- gcIntermediate( c(node1[1,]$V8, node1[1,]$V7),
                         c(node2[1,]$V8, node2[1,]$V7),
                         n=1000, sp = TRUE, breakAtDateLine = TRUE )

 lines(arc, col="red", lwd=.03, cex=0.01)
}
```

Problem 2 - Statistician

Part 1:

```
text<-read.csv("paperList_Abstracts_Keyword.txt")
text$keywords<-unlist(lapply(text$keywords,function(x)gsub(';',' ',x)))
a<-text[text$paperYear==2003,]
b<-text[text$paperYear==2004,]
c<-text[text$paperYear==2005,]
d<-text[text$paperYear==2006,]
e<-text[text$paperYear==2007,]
f<-text[text$paperYear==2008,]
g<-text[text$paperYear==2009,]
h<-text[text$paperYear==2010,]
i<-text[text$paperYear==2011,]
j<-text[text$paperYear==2012,]

library(stylo)
library(RColorBrewer)
library(tm)
library(wordcloud)


pal <- brewer.pal(9, "PuRd")
a2words<-strsplit(a$keywords," ")
a2<-table(matrix(delete.stop.words(unlist(a2words), stop.words = stopwords('english'))))

par(mfrow=c(1,4))

wordcloud(
  words = names(a2),
  freq = as.numeric(a2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2003",line=-9)

b2words<-strsplit(b$keywords," ")
b2<-table(matrix(delete.stop.words(unlist(b2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(b2),
  freq = as.numeric(b2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2004",line=-9)

c2words<-strsplit(c$keywords," ")
c2<-table(matrix(delete.stop.words(unlist(c2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(c2),
  freq = as.numeric(c2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2005",line=-9)

d2words<-strsplit(d$keywords," ")
d2<-table(matrix(delete.stop.words(unlist(d2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(d2),
  freq = as.numeric(d2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2006",line=-9)

e2words<-strsplit(e$keywords," ")
e2<-table(matrix(delete.stop.words(unlist(e2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(e2),
  freq = as.numeric(e2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2007",line=-9)

f2words<-strsplit(f$keywords," ")
f2<-table(matrix(delete.stop.words(unlist(f2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(f2),
  freq = as.numeric(f2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2008",line=-9)

g2words<-strsplit(g$keywords," ")
g2<-table(matrix(delete.stop.words(unlist(g2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(g2),
  freq = as.numeric(g2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2009",line=-9)

h2words<-strsplit(h$keywords," ")
h2<-table(matrix(delete.stop.words(unlist(h2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(h2),
  freq = as.numeric(h2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2010",line=-9)

i2words<-strsplit(i$keywords," ")
i2<-table(matrix(delete.stop.words(unlist(i2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(i2),
  freq = as.numeric(i2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2011",line=-9)

j2words<-strsplit(j$keywords," ")
j2<-table(matrix(delete.stop.words(unlist(j2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(j2),
  freq = as.numeric(j2),
  random.order = F,
  colors=pal,
  scale = c(2.5, 0.25),
  min.freq = 20)
title("2012",line=-9)
```

Part B~D:

```
library("igraph")

citadj<-read.table("paperCitAdj.txt",header=F)
authors<-read.table("authorList.txt")
authorship<-read.table("authorPaperBiadj.txt",header=F)
authors[which(rowSums(authorship)>50),]
rownames(authorship)<-unlist(authors)
peter<-authorship[2489,]
colnumbPeter<-which(apply(peter, 2, function(x) any(grepl(1, x))))
names(colnumbPeter)<-NULL
PeterCitadj<-citadj[colnumbPeter,]

text<-read.csv("paperList_Abstracts_Keyword.txt")
work2003<-which(text$paperYear==2003)
work2004<-which(text$paperYear==2004)
work2005<-which(text$paperYear==2005)
work2006<-which(text$paperYear==2006)
work2007<-which(text$paperYear==2007)
work2008<-which(text$paperYear==2008)
work2009<-which(text$paperYear==2009)
work2010<-which(text$paperYear==2010)
work2011<-which(text$paperYear==2011)
work2012<-which(text$paperYear==2012)
test<-PeterCitadj
names(test)<-1:3248


test2k6<-as.matrix(subset(test,select=work2006))
test2k6bind<-matrix(0,272,354)
newtest2k6<-rbind(test2k6,test2k6bind)
net2k6 <- graph_from_adjacency_matrix(newtest2k6,mode="undirected")
isolated2k6=which(degree(net2k6)==0)
newnet2k6=delete.vertices(net2k6,isolated2k6)
V(newnet2k6)$size<-(degree(newnet2k6,mode="all"))*5
plot(newnet2k6,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)


test2k7<-as.matrix(subset(test,select=c(work2006,work2007)))
```

```
test2k7bind<-matrix(0,622,704)
newtest2k7<-rbind(test2k7,test2k7bind)
net2k7 <- graph_from_adjacency_matrix(newtest2k7,mode="undirected")
isolated2k7=which(degree(net2k7)==0)
newnet2k7=delete.vertices(net2k7,isolated2k7)
V(newnet2k7)$size<-(degree(newnet2k7,mode="all"))*5
plot(newnet2k7,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)


test2k8<-as.matrix(subset(test,select=c(work2006,work2007,work2008)))
test2k8bind<-matrix(0,992,1074)
newtest2k8<-rbind(test2k8,test2k8bind)
net2k8 <- graph_from_adjacency_matrix(newtest2k8,mode="undirected")
isolated2k8=which(degree(net2k8)==0)
newnet2k8=delete.vertices(net2k8,isolated2k8)
V(newnet2k8)$size<-(degree(newnet2k8,mode="all"))*5
plot(newnet2k8,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)


test2k9<-as.matrix(subset(test,select=c(work2006,work2007,work2008,work2009)))
test2k9bind<-matrix(0,1401,1483)
newtest2k9<-rbind(test2k9,test2k9bind)
net2k9 <- graph_from_adjacency_matrix(newtest2k9,mode="undirected")
isolated2k9=which(degree(net2k9)==0)
newnet2k9=delete.vertices(net2k9,isolated2k9)
V(newnet2k9)$size<-(degree(newnet2k9,mode="all"))*5
plot(newnet2k9,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)


test2k10<-as.matrix(subset(test,select=c(work2006,work2007,work2008,work2009,work2010)))
test2k10bind<-matrix(0,1756,1838)
newtest2k10<-rbind(test2k10,test2k10bind)
net2k10 <- graph_from_adjacency_matrix(newtest2k10,mode="undirected")
isolated2k10=which(degree(net2k10)==0)
newnet2k10=delete.vertices(net2k10,isolated2k10)
V(newnet2k10)$size<-(degree(newnet2k10,mode="all"))*5
plot(newnet2k10,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)
```

```
test2k11<-as.matrix(subset(test,select=c(work2006,work2007,work2008,work2009,work2010,work2011)))
test2k11bind<-matrix(0,2081,2163)
newtest2k11<-rbind(test2k11,test2k11bind)
net2k11 <- graph_from_adjacency_matrix(newtest2k11,mode="undirected")
isolated2k11=which(degree(net2k11)==0)
newnet2k11=delete.vertices(net2k11,isolated2k11)
V(newnet2k11)$size<-(degree(newnet2k11,mode="all"))*5
plot(newnet2k11,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)

test2k12<-as.matrix(subset(test,select=c(work2006,work2007,work2008,work2009,work2010,work2011,work2012)))
test2k12bind<-matrix(0,2222,2304)
newtest2k12<-rbind(test2k12,test2k12bind)
net2k12 <- graph_from_adjacency_matrix(newtest2k12,mode="undirected")
isolated2k12=which(degree(net2k12)==0)
newnet2k12=delete.vertices(net2k12,isolated2k12)
V(newnet2k12)$size<-(degree(newnet2k12,mode="all"))*5
plot(newnet2k12,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere)



############ comparison plots
par(mfrow=c(1,4))
plot(newnet2k6,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere,main=2006)
plot(newnet2k7,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere,main=2007)
plot(newnet2k8,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere, main=2008)
plot(newnet2k9,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere, main=2009)

par(mfrow=c(1,3))
plot(newnet2k10,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere, main=2010)
plot(newnet2k11,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere, main=2011)
plot(newnet2k12,edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.sphere, main=2012)
```

```
#####################Part B
library(tidyverse)

Collab<-authorship[,colnumbPeter]
testing<-strsplit(colnames(Collab),"V")
colnames(Collab)<-sapply(testing,"[[",2)
w2k3<-as.character(work2003)
w2k4<-as.character(work2004)
w2k5<-as.character(work2005)
w2k6<-as.character(work2006)
w2k7<-as.character(work2007)
w2k8<-as.character(work2008)
w2k9<-as.character(work2009)
w2k10<-as.character(work2010)
w2k11<-as.character(work2011)
w2k12<-as.character(work2012)



asdf2k3<-as.matrix(select(Collab,contains(w2k3)))
asdf2k3bind<-matrix(0,3607,3597)
newasdf2k3<-cbind(asdf2k3,asdf2k3bind)
netasdf2k3 <- graph_from_adjacency_matrix(newasdf2k3,mode="undirected")
isolatedasdf2k3=which(degree(netasdf2k3)==0)
newnetasdf2k3=delete.vertices(netasdf2k3,isolatedasdf2k3)
V(newnetasdf2k3)$size<-(degree(newnetasdf2k3,mode="all"))*1
plot(igraph::simplify(newnetasdf2k3),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k4<-as.matrix(select(Collab,contains(c(w2k3,w2k4))))
asdf2k4bind<-matrix(0,3607,3587)
newasdf2k4<-cbind(asdf2k4,asdf2k4bind)
netasdf2k4 <- graph_from_adjacency_matrix(newasdf2k4,mode="undirected")
isolatedasdf2k4=which(degree(netasdf2k4)==0)
newnetasdf2k4=delete.vertices(netasdf2k4,isolatedasdf2k4)
V(newnetasdf2k4)$size<-(degree(newnetasdf2k4,mode="all"))*1
plot(igraph::simplify(newnetasdf2k4),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k5<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5))))
asdf2k5bind<-matrix(0,3607,3576)
newasdf2k5<-cbind(asdf2k5,asdf2k5bind)
netasdf2k5 <- graph_from_adjacency_matrix(newasdf2k5,mode="undirected")
```

```
isolatedasdf2k5=which(degree(netasdf2k5)==0)
newnetasdf2k5=delete.vertices(netasdf2k5,isolatedasdf2k5)
V(newnetasdf2k5)$size<-(degree(newnetasdf2k5,mode="all"))*1
plot(igraph::simplify(newnetasdf2k5),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",v
ertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k6<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5,w2k6))))
asdf2k6bind<-matrix(0,3607,3607-44)
newasdf2k6<-cbind(asdf2k6,asdf2k6bind)
netasdf2k6 <- graph_from_adjacency_matrix(newasdf2k6,mode="undirected")
isolatedasdf2k6=which(degree(netasdf2k6)==0)
newnetasdf2k6=delete.vertices(netasdf2k6,isolatedasdf2k6)
V(newnetasdf2k6)$size<-(degree(newnetasdf2k6,mode="all"))*1
plot(igraph::simplify(newnetasdf2k6),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",v
ertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k7<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5,w2k6,w2k7))))
asdf2k7bind<-matrix(0,3607,3607-56)
newasdf2k7<-cbind(asdf2k7,asdf2k7bind)
netasdf2k7 <- graph_from_adjacency_matrix(newasdf2k7,mode="undirected")
isolatedasdf2k7=which(degree(netasdf2k7)==0)
newnetasdf2k7=delete.vertices(netasdf2k7,isolatedasdf2k7)
V(newnetasdf2k7)$size<-(degree(newnetasdf2k7,mode="all"))*1
plot(igraph::simplify(newnetasdf2k7),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",v
ertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k8<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5,w2k6,w2k7,w2k8))))
asdf2k8bind<-matrix(0,3607,3607-65)
newasdf2k8<-cbind(asdf2k8,asdf2k8bind)
netasdf2k8 <- graph_from_adjacency_matrix(newasdf2k8,mode="undirected")
isolatedasdf2k8=which(degree(netasdf2k8)==0)
newnetasdf2k8=delete.vertices(netasdf2k8,isolatedasdf2k8)
V(newnetasdf2k8)$size<-(degree(newnetasdf2k8,mode="all"))*1
plot(igraph::simplify(newnetasdf2k8),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",v
ertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k9<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5,w2k6,w2k7,w2k8,w2k9))))
asdf2k9bind<-matrix(0,3607,3607-71)
newasdf2k9<-cbind(asdf2k9,asdf2k9bind)
netasdf2k9 <- graph_from_adjacency_matrix(newasdf2k9,mode="undirected")
isolatedasdf2k9=which(degree(netasdf2k9)==0)
newnetasdf2k9=delete.vertices(netasdf2k9,isolatedasdf2k9)
V(newnetasdf2k9)$size<-(degree(newnetasdf2k9,mode="all"))*1
```

```
plot(igraph::simplify(newnetasdf2k9),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k10<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5,w2k6,w2k7,w2k8,w2k9,w2k10))))
asdf2k10bind<-matrix(0,3607,3607-78)
newasdf2k10<-cbind(asdf2k10,asdf2k10bind)
netasdf2k10 <- graph_from_adjacency_matrix(newasdf2k10,mode="undirected")
isolatedasdf2k10=which(degree(netasdf2k10)==0)
newnetasdf2k10=delete.vertices(netasdf2k10,isolatedasdf2k10)
V(newnetasdf2k10)$size<-(degree(newnetasdf2k10,mode="all"))*1
plot(igraph::simplify(newnetasdf2k10),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

asdf2k11<-as.matrix(select(Collab,contains(c(w2k3,w2k4,w2k5,w2k6,w2k7,w2k8,w2k9,w2k10,w2k11))))
asdf2k11bind<-matrix(0,3607,3607-82)
newasdf2k11<-cbind(asdf2k11,asdf2k11bind)
netasdf2k11 <- graph_from_adjacency_matrix(newasdf2k11,mode="undirected")
isolatedasdf2k11=which(degree(netasdf2k11)==0)
newnetasdf2k11=delete.vertices(netasdf2k11,isolatedasdf2k11)
V(newnetasdf2k11)$size<-(degree(newnetasdf2k11,mode="all"))*1
plot(igraph::simplify(newnetasdf2k11),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold)

################## PLOT PART B
par(mfrow=c(1,3))
plot(igraph::simplify(newnetasdf2k3),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2003)
plot(igraph::simplify(newnetasdf2k4),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2004)
plot(igraph::simplify(newnetasdf2k5),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2005)
plot(igraph::simplify(newnetasdf2k6),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2006)
plot(igraph::simplify(newnetasdf2k7),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2007)
plot(igraph::simplify(newnetasdf2k8),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2008)
plot(igraph::simplify(newnetasdf2k9),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2009)
plot(igraph::simplify(newnetasdf2k10),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle",vertex.color=c("antiquewhite1", "tomato", "gold"),layout=layout.fruchterman.reingold,main=2010)
plot(igraph::simplify(newnetasdf2k11),edge.arrow.size=.4,vertex.label=NA,vertex.shape="circle"
,vertex.color=c("antiquewhite1", "tomato",
"gold"),layout=layout.fruchterman.reingold,main=2011)

#################### PART C
text<-read.csv("paperList_Abstracts_Keyword.txt")
text$keywords<-unlist(lapply(text$keywords,function(x)gsub(';',' ',x)))
text<-text[colnumbPeter,]
a<-text[text$paperYear==2003,]
b<-text[text$paperYear==2004,]
c<-text[text$paperYear==2005,]
d<-text[text$paperYear==2006,]
e<-text[text$paperYear==2007,]
f<-text[text$paperYear==2008,]
g<-text[text$paperYear==2009,]
h<-text[text$paperYear==2010,]
i<-text[text$paperYear==2011,]
j<-text[text$paperYear==2012,]

library(stylo)
library(RColorBrewer)
library(tm)
library(wordcloud)


pal <- brewer.pal(9, "PuRd")
a2words<-strsplit(a$keywords," ")
a2<-table(matrix(delete.stop.words(unlist(a2words), stop.words = stopwords('english'))))

par(mfrow=c(1,4))

wordcloud(
  words = names(a2),
  freq = as.numeric(a2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),
  min.freq = 2)
title("2003",line=-9)

b2words<-strsplit(b$keywords," ")
b2<-table(matrix(delete.stop.words(unlist(b2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(b2),
  freq = as.numeric(b2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),
  min.freq = 2)
title("2004",line=-9)

c2words<-strsplit(c$keywords," ")
c2<-table(matrix(delete.stop.words(unlist(c2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(c2),
  freq = as.numeric(c2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2005",line=-9)

d2words<-strsplit(d$keywords," ")
d2<-table(matrix(delete.stop.words(unlist(d2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(d2),
  freq = as.numeric(d2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2006",line=-9)

e2words<-strsplit(e$keywords," ")
e2<-table(matrix(delete.stop.words(unlist(e2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(e2),
  freq = as.numeric(e2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2007",line=-9)

f2words<-strsplit(f$keywords," ")
f2<-table(matrix(delete.stop.words(unlist(f2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(f2),
  freq = as.numeric(f2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2008",line=-9)

g2words<-strsplit(g$keywords," ")
g2<-table(matrix(delete.stop.words(unlist(g2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(g2),
  freq = as.numeric(g2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2009",line=-9)

h2words<-strsplit(h$keywords," ")
h2<-table(matrix(delete.stop.words(unlist(h2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(h2),
  freq = as.numeric(h2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2010",line=-9)

i2words<-strsplit(i$keywords," ")
i2<-table(matrix(delete.stop.words(unlist(i2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(i2),
  freq = as.numeric(i2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2011",line=-9)

j2words<-strsplit(j$keywords," ")
j2<-table(matrix(delete.stop.words(unlist(j2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(j2),
  freq = as.numeric(j2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2012",line=-9)
```

```
############## Part D

dt <- scan("paperList.txt",sep="\n",what="")
dt <- dt[-1]
tmp1 <- lapply(dt,function(s)unlist(strsplit(s,"")))
years <- unlist(lapply(tmp1,function(x) return(x[5])))
years <- gsub(",","",years)
years <- as.numeric(years)
abss <- unlist(lapply(tmp1,function(x) return(x[6])))
abss <- unlist(lapply(abss,function(x)gsub('\\{','',x)))
abss <- unlist(lapply(abss,function(x)gsub("}","",x)))
abss <- unlist(lapply(abss,function(x)gsub("#","",x)))
abss <- unlist(lapply(abss,function(x)gsub("&","",x)))
abss <- unlist(lapply(abss,function(x)gsub("<","",x)))
abss <- unlist(lapply(abss,function(x)gsub(">","",x)))
abss <- unlist(lapply(abss,function(x)gsub("/","",x)))
abss <- unlist(lapply(abss,function(x)gsub(")","",x)))
abss <- unlist(lapply(abss,function(x)gsub("\\(","",x)))
abss <- unlist(lapply(abss,function(x)gsub("amp","",x)))
hoho<-as.data.frame(abss)
hoho$paperYear<-years
peteho<-hoho[colnumbPeter,]
peteho$abss<-as.character(peteho$abss)
peteho$abss<-unlist(peteho$abss)



text<-peteho
a<-text[text$paperYear==2003,]
b<-text[text$paperYear==2004,]
c<-text[text$paperYear==2005,]
d<-text[text$paperYear==2006,]
e<-text[text$paperYear==2007,]
f<-text[text$paperYear==2008,]
g<-text[text$paperYear==2009,]
h<-text[text$paperYear==2010,]
i<-text[text$paperYear==2011,]
j<-text[text$paperYear==2012,]

a2words<-strsplit(a$abss," ")
a2<-table(matrix(delete.stop.words(unlist(a2words), stop.words = stopwords('english'))))
library(stylo)
library(RColorBrewer)
library(tm)
library(wordcloud)
```

```
pal <- brewer.pal(9, "PuRd")
par(mfrow=c(1,4))

wordcloud(
  words = names(a2),
  freq = as.numeric(a2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),
  min.freq = 1)
title("2003",line=-9)

b2words<-strsplit(b$abss," ")
b2<-table(matrix(delete.stop.words(unlist(b2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(b2),
  freq = as.numeric(b2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),
  min.freq = 1)
title("2004",line=-9)

c2words<-strsplit(c$abss," ")
c2<-table(matrix(delete.stop.words(unlist(c2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(c2),
  freq = as.numeric(c2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2005",line=-9)

d2words<-strsplit(d$abss," ")
d2<-table(matrix(delete.stop.words(unlist(d2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(d2),
  freq = as.numeric(d2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2006",line=-9)
```

```
e2words<-strsplit(e$abss," ")
e2<-table(matrix(delete.stop.words(unlist(e2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(e2),
  freq = as.numeric(e2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2007",line=-9)

f2words<-strsplit(f$abss," ")
f2<-table(matrix(delete.stop.words(unlist(f2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(f2),
  freq = as.numeric(f2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2008",line=-9)

g2words<-strsplit(g$abss," ")
g2<-table(matrix(delete.stop.words(unlist(g2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(g2),
  freq = as.numeric(g2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2009",line=-9)

h2words<-strsplit(h$abss," ")
h2<-table(matrix(delete.stop.words(unlist(h2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(h2),
  freq = as.numeric(h2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2010",line=-9)

i2words<-strsplit(i$abss," ")
i2<-table(matrix(delete.stop.words(unlist(i2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(i2),
  freq = as.numeric(i2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2011",line=-9)

j2words<-strsplit(j$abss," ")
j2<-table(matrix(delete.stop.words(unlist(j2words), stop.words = stopwords('english'))))
wordcloud(
  words = names(j2),
  freq = as.numeric(j2),
  random.order = F,
  colors=pal,
  scale = c(1.5, 0.25),min.freq = 2)
title("2012",line=-9)
```

Problem 3

```
##Code for 3.1,2,3
confirmed_world <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_gl
obal.csv",
                        stringsAsFactors = FALSE,
                        check.names =  FALSE)



dim(confirmed_world)

names(confirmed_world)

head(confirmed_world[,1:6],10)

#install.packages("reshape2")

library(reshape2)

### again, we will use the package reshape2 to transform this data set to a more convenient
format

confirmed_world <- reshape2::melt(confirmed_world, id.vars = c("Province/State",
"Country/Region", "Lat", "Long"), variable.name = "Date", value.name = "Confirmed")
head(confirmed_world)


death_world <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_globa
l.csv",stringsAsFactors = FALSE,
                     check.names =  FALSE)

recovered_world <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_gl
obal.csv",stringsAsFactors = FALSE, check.names =  FALSE)

head(death_world[.1:6])

head(recovered_world[,1:6])


dim(death_world)
dim(recovered_world)

death_world <- reshape2::melt(death_world, id.vars = c("Province/State", "Country/Region",
"Lat", "Long"), variable.name = "Date", value.name = "Death")

recovered_world <- reshape2::melt(recovered_world, id.vars = c("Province/State",
"Country/Region", "Lat", "Long"), variable.name = "Date", value.name = "Recovered")


head(death_world)

head(recovered_world)


world_history_data <- dplyr::left_join(confirmed_world, death_world, by = c("Province/State",
"Country/Region", "Lat", "Long", "Date"))

head(world_history_data)

world_history_data <- dplyr::left_join(world_history_data, recovered_world, by =
c("Province/State", "Country/Region", "Lat", "Long", "Date"))


head(world_history_data,20)

world_history_data$Date <- as.Date(as.character(world_history_data$Date), format =
c("%m/%d/%y"))
head(world_history_data,20)


colnames(world_history_data) <- make.names(colnames(world_history_data))
colnames(world_history_data)

head(world_history_data,20)



library(RColorBrewer)

cols <- matrix(c(brewer.pal(9,"Set1"),brewer.pal(11,"Set3")),ncol=1)

length(unique(world_history_data$Country.Region))
```

```
library(plyr)

world.summary.data <- ddply(world_history_data,.(Country.Region, Date),function(x){
  colSums(x[,c("Confirmed","Death","Recovered")])
})


dim(world.summary.data)

#### too many of them

#### look at the countrires with the most confirmed cases

lastday <- max(world.summary.data$Date)
lastday <- "2020-04-11"

world.summary.data <- world.summary.data[world.summary.data$Date<=lastday,]

yesterday.data <- world.summary.data[world.summary.data$Date==lastday,]

sort.index <- sort(yesterday.data$Confirmed,decreasing=TRUE,index.return=TRUE)$ix

yesterday.data.major  <- yesterday.data[sort.index[1:20],]

yesterday.data.major  <-
data.frame(Country.Region=yesterday.data$Country.Region[sort.index[1:20]])

yesterday.data.major$Country.Region  <- as.character(yesterday.data.major$Country.Region )


major.summary.data <- dplyr::inner_join(world.summary.data,yesterday.data.major,by =
"Country.Region")

rownames(cols) <- unique(major.summary.data$Country.Region)

major.summary.data$Country.Region <- factor(major.summary.data$Country.Region,
                                            levels = rev(yesterday.data.major$Country.Region))


library(ggplot2)

library(wbstats)

pop_data = wb(indicator = "SP.POP.TOTL", startdate = 2018, enddate = 2019)
```

```
for (i in 1:length(major.summary.data$Country.Region)){
  if (major.summary.data$Country.Region[i] == "US" )
  {major.summary.data$Country_pop[i] = pop_data$value[pop_data$country == "United States"]
    next
  }
  else if (major.summary.data$Country.Region[i] == "Korea, South" )
  {
    major.summary.data$Country_pop[i] = pop_data$value[pop_data$country == "Korea, Rep."]
    next
  }
  else if (major.summary.data$Country.Region[i] == "Russia" )
  {
    major.summary.data$Country_pop[i] = pop_data$value[pop_data$country == "Russian
Federation"]
    next
  }
  else if (length(pop_data$value[pop_data$country == major.summary.data$Country.Region[i]])
== 0){
    next
  }
  major.summary.data$Country_pop[i] = pop_data$value[pop_data$country ==
major.summary.data$Country.Region[i]]
}


### Generate a line graph to visualize the confirmed cases since Jan 21 (except mainland
China)
major.summary.data = major.summary.data[major.summary.data$Country.Region != "Iran", ]

plot1 <- ggplot(major.summary.data,
       aes(x = Date, y = Confirmed/(Country_pop/1000000), col = Country.Region)) +
geom_line(lwd = 1) + geom_point(size = 2) +
  scale_color_manual(values = cols) + theme_minimal() + ylab("Confirmed") +labs(title =
"Confirmed Covid-19 Case per Million") +
  xlab("Date") + labs(color = "Country/Region") + scale_x_date(date_labels = "%m-%d",
                                                            date_breaks = "5 day") + theme(text = element_text(size =
14, face = "bold"),
                                                                                           axis.text.x = element_text(angle = 90),
legend.position = "right")
plot1

### generate a line graph to visualize the death cases since Jan 21 except china
plot2 <- ggplot(major.summary.data,
    aes(x = Date, y = Death/(Recovered+1), col = Country.Region)) + geom_line(lwd = 1) +
geom_point(size = 2) +
  scale_color_manual(values = cols) + theme_minimal() + ylab("Death/Recovered") + labs(title =
"Ratio of Death to Recovered") +
  xlab("Date") + labs(color = "Country/Region") + scale_x_date(date_labels = "%m-%d",
                                          date_breaks = "5 day") + theme(text = element_text(size =
14, face = "bold"),
                                                          axis.text.x = element_text(angle = 90),
legend.position = "right")
plot2

#animation
#install.packages("gganimate")
library(gganimate)

p1 <- plot1 + transition_reveal(along = Date) +shadow_wake(wake_length = 0.1, alpha =
FALSE)
animate(p1,nframes=50,renderer = gifski_renderer("test1.gif"),height=500,width=650)

p2 <- plot2 + transition_reveal(along = Date) +shadow_wake(wake_length = 0.1, alpha =
FALSE)
animate(p2,nframes=60,renderer = gifski_renderer("test2.gif"),height=500,width=650)


###################Top 20 States by COVID-19 Cases###############################
setwd('~/Desktop/STAT 3280')
confirmed_US <- read.csv("time_series_covid19_confirmed_US.csv",stringsAsFactors =
FALSE,
                  check.names =  FALSE)
deaths_US <- read.csv("time_series_covid19_deaths_US.csv",stringsAsFactors = FALSE,
                check.names =  FALSE) #same columns as confirmed_US but with a population
column
deaths_US<-deaths_US[,-c(1:6,12)]
confirmed_US<-confirmed_US[,-(1:6)]
#names(deaths_US)
#names(confirmed_US)


deaths_US <- reshape2::melt(deaths_US, id.vars = c("Province_State", "Country_Region",
"Lat", "Long_","Combined_Key"), variable.name = "Date", value.name = "Death")
confirmed_US <- reshape2::melt(confirmed_US, id.vars = c("Province_State",
"Country_Region", "Lat", "Long_","Combined_Key"), variable.name = "Date", value.name =
"Confirmed")
head(deaths_US)
head(confirmed_US)
lapply(deaths_US,class)
lapply(confirmed_US,class)

USdata <- dplyr::inner_join(confirmed_US, deaths_US, by = c("Province_State",
"Country_Region", "Lat", "Long_","Combined_Key","Date"))
USdata$Date <- as.Date(as.character(USdata$Date), format = c("%m/%d/%y"))
head(USdata)
lapply(USdata,class)


#deaths/confirmed cases
USdata$deaths_cases <- 100*(USdata$Death/USdata$Confirmed) #wasn't able to calculate
this?? Returns Nan
range(deaths_US$Death)
range(USdata$Death)
max(USdata$Date)
summary(USdata$deaths_cases)
colnames(USdata) <- make.names(colnames(USdata))



#april8=USdata[USdata$Date=="2020-04-08",]
#april8$dc=100*(april8$Death/april8$Confirmed)

library(RColorBrewer)
library(plyr)
cols <- matrix(c(brewer.pal(9,"Set1"),brewer.pal(11,"Set3")),ncol=1)
length(unique(USdata$Province_State))

US.summary.data <- ddply(USdata,.(Province_State, Date),function(x){
   colSums(x[,c("Confirmed","Death","deaths_cases")])})


dim(US.summary.data)

#### too many of them

#### look at the countrires with the most confirmed cases

lastday <- max(US.summary.data$Date)

US.summary.data <- US.summary.data[US.summary.data$Date<=lastday,]
state_pop=read.csv("state_2019_pop.csv",header=T)
```

```
head(state_pop)
for (i in 1:length(US.summary.data$Province_State)){
   US.summary.data$pop[i] = state_pop$POPESTIMATE2019[state_pop$NAME ==
US.summary.data$Province_State[i]]
}
head(US.summary.data)


US.summary.data <- US.summary.data[US.summary.data$Date<=lastday,]

yesterday.data <- US.summary.data[US.summary.data$Date==lastday,]

dim(yesterday.data)

head(yesterday.data)

sort.index <- sort(yesterday.data$Confirmed,decreasing=TRUE,index.return=TRUE)$ix

yesterday.data.major  <- yesterday.data[sort.index[1:20],]

yesterday.data.major

yesterday.data.major  <-
data.frame(Province_State=yesterday.data$Province_State[sort.index[1:20]])

yesterday.data.major

yesterday.data.major$Province_State  <- as.character(yesterday.data.major$Province_State )


major.summary.data <- dplyr::inner_join(US.summary.data,yesterday.data.major,by =
"Province_State")


dim(major.summary.data)

head(major.summary.data)



length(unique(major.summary.data$Province_State))


rownames(cols) <- unique(major.summary.data$Province_State)
```

```
major.summary.data$Province_State <- factor(major.summary.data$Province_State,
                                  levels = rev(yesterday.data.major$Province_State))
### Line Graph for US States####

p1=ggplot(major.summary.data,
     aes(x = Date, y = Confirmed/(pop/1000000), col = Province_State)) +
   geom_line(lwd = 1) + geom_point(size = 2) +
   scale_color_manual(values = cols) + theme_minimal() + ylab("Confirmed") +
   labs(title = "Confirmed Covid-19 Case per Million") +
   xlab("Date")  + scale_x_date(date_labels = "%m-%d",
date_breaks = "5 day") + theme(text = element_text(size = 12, face = "bold"),
axis.text.x = element_text(angle = 90), legend.position = "right") +
   theme(plot.title = element_text(hjust = 0.5))+
   guides(color=guide_legend(title="State"))



library(gifski)
#install.packages("gifski")
library(gganimate)

p11 <- p1 + transition_reveal(along = Date) +shadow_wake(wake_length = 0.1, alpha = FALSE)
animate(p1,nframes=50,renderer = gifski_renderer("test1.gif"),height=500,width=650)

#excluding new york and new jersey:
da=major.summary.data[major.summary.data$Province_State!="New York",]
da=da[da$Province_State!="New Jersey",]
ggplot(da,
     aes(x = Date, y = Confirmed/(pop/1000000), col = Province_State)) +
   geom_line(lwd = 1) + geom_point(size = 2) +
   scale_color_manual(values = cols) + theme_minimal() + ylab("Confirmed") +
   labs(title = "Confirmed Covid-19 Case per Million",
       subtitle="Excluding New York and New Jersey") +
   xlab("Date")  + scale_x_date(date_labels = "%m-%d",
                         date_breaks = "5 day") + theme(text = element_text(size = 12, face =
"bold"),
                                              axis.text.x = element_text(angle = 90), legend.position
= "right") +
   theme(plot.title = element_text(hjust = 0.5),plot.subtitle = element_text(hjust=.5))+
   guides(color=guide_legend(title="State"))

#######
```

```
data=major.summary.data[major.summary.data$Date==lastday,]
data=data[,-5]
ggplot(data, aes(x = Province_State, y = Confirmed, fill = Province_State)) +
  geom_col() + scale_fill_manual(values = cols) + theme_minimal() +
  ylab("Confirmed") + xlab("") + labs(color = "Province_State") + coord_flip() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 90),
     legend.position = "none")

#install.packages("plotly")
library(plotly)
plot_ly(data, x = ~Province_State,
    y = ~Confirmed, color=I('light blue'), type = 'bar', name = 'Confirmed') %>%
  add_trace(y = ~Death, color=I('red'),name = 'Death') %>%
  layout(title="Top 20 States by COVID-19 Cases",yaxis = list(title = 'Count'),
xaxis=list(title='State'), barmode = 'overlay')

#w/ hover text
text=paste(round(100*(data$Confirmed/sum(yesterday.data$Confirmed)),2),"% of national
cases")

text2=paste(round(100*(data$Death/data$Confirmed),2),"% of",data$Province_State,"cases")
plot_ly(data, x = ~Province_State, y = ~Confirmed, color=I('light blue'),
    type = 'bar', text=text, name = 'Confirmed') %>%
  add_trace(y = ~Death, text=text2,color=I('red'), name = 'Death') %>%
  layout(title="Top 20 States by COVID-19 Cases", yaxis = list(title = 'Count'),
      xaxis=list(title='State'),barmode = 'overlay',legend = list(orientation = "h",
    xanchor = "center",  x = 0.5,y=1))


#hover text and direct labels reference guide:
#https://plotly.com/chart-studio-help/documentation/r/bar-charts/
#https://plotly.com/r/horizontal-bar-charts/
#https://stackoverflow.com/questions/53285059/ordering-columns-in-plotly-horizontal-bar-chart
#https://rstudio-pubs-
static.s3.amazonaws.com/168008_080fe9650e14478f98487d525a76449d.html

text=paste(round(100*(data$Confirmed/sum(yesterday.data$Confirmed)),2),"%")
text2=paste(round(100*(data$Death/data$Confirmed),2),"%")
text2[4]<-"2.00 %"
plot_ly(data, x = ~Province_State, y = ~Confirmed, color=I('dodgerblue'),
    type = 'bar', text=text,textposition = 'outside', name = 'Confirmed') %>%
  add_trace(y = ~Death,color=I('red'), name = 'Death',text=text2,textposition = 'outside',t) %>%
   layout(title="Top 20 States by COVID-19 Cases", yaxis = list(title = 'Count'),
        xaxis=list(title='State'),barmode = 'overlay',legend = list(orientation = "h",
                                                          xanchor = "center",  x = 0.5,y=1))



#########Geographical Distribution of COVID-19 Cases##################
library(ggplot2)
library(dplyr)
library(maps)
library(igraph)
library(geosphere)
setwd('~/Desktop/STAT 3280')

confirmed_world <- read.csv("time_series_covid19_confirmed_global.csv",stringsAsFactors =
FALSE,
                        check.names =  FALSE)

#colnames(confirmed_world)[1] <- "Province/State"
#colnames(confirmed_world)[2] <- "Country/Region"
#confirmed_world$`Province/State`<-as.character(confirmed_world$`Province/State`)
#confirmed_world$`Country/Region`<-as.character(confirmed_world$`Country/Region`)
head(confirmed_world[,1:6],10)

library(reshape2)
### we will use the package reshape2 to transform this data set to a more convenient format
confirmed_world <- reshape2::melt(confirmed_world, id.vars = c("Province/State",
"Country/Region", "Lat", "Long"), variable.name = "Date", value.name = "Confirmed")

newdata=confirmed_world[confirmed_world$Date=="4/8/20",]

mybreaks <- c(1,50000,100000,150000,200000,250000,300000,350000,400000)
ggplot() +
   geom_polygon(data = map_data("world"), aes(x=long, y = lat, group = group), fill="grey",
alpha=0.3) +
   geom_point(data=newdata, aes(x=Long, y=Lat, size=Confirmed,
color=Confirmed),shape=20, stroke=FALSE,color="red") +
   scale_size_continuous(name="Confirmed Cases",range=c(1,12),breaks=mybreaks) +
   theme_void() +coord_map(xlim=c(-180,180))+
   theme(legend.key.size = unit(0, "cm"),legend.text = element_text(size = 8),
        legend.title = element_text(size = 10))+
   ggtitle("Geographical Distribution of COVID-19 Cases")+
   theme(plot.title = element_text(hjust = 0.83,size = 10))
```