stat992HW1

SongWang

10/08/2015

read the referral/ physisian feature/payment data sets

and select a subset from it conditional on the state = CA, City = “San Francisco”, entity_type = “individual”

```
rm(list=ls())
library(data.table)  # so fast!
# install.packages('igraph')
library(igraph)  # all the basic graph operations.
##############
setwd("~/Stat/Courses/Physisian_Referral_Network")
DataPath <- "./Data/"
ResultsPath <- "./Results/"
PlotsPath <- "./Plots/"
RScriptsPath <- "./RScripts/"


#### payment data

# Payment = fread(paste0(DataPath,
#        "Medicare_Provider_Util_Payment_PUF_CY2013/Medicare_Provider_Util_Payment_PUF_CY2013.txt"),
#                   sep = "\t")
# Payment <- Payment[-1]
# setkey(Payment, NPI)
# head(Payment)
#
# Payment_NPI_ca <- Payment[NPPES_PROVIDER_STATE=="CA"&NPPES_ENTITY_CODE=="I"]
# Payment_NPI_total_ca= Payment_NPI_ca[,.(NPI,totalPay=AVERAGE_MEDICARE_ALLOWED_AMT * LINE_SRVC_CNT)]
# Payment_NPI_total_ca <- Payment_NPI_total_ca[,.(totalPay=sum(totalPay)),by=NPI]
#
# save(Payment_NPI_total_ca,file = paste0(DataPath, "Payment_NPI_total_ca.RData"))


system.time(load(paste0(DataPath, "EtDT.RData")))
```

```
##    user  system elapsed
##  60.358   0.514  61.016
```

```
system.time(load(paste0(DataPath,"Payment_NPI_ca.RData"))) ## payment data constrained to individual phy
```

```
##    user  system elapsed
##   4.050   0.054   4.106
```

```
system.time(load(paste0(DataPath,"Payment_NPI_total_ca.RData")))
```

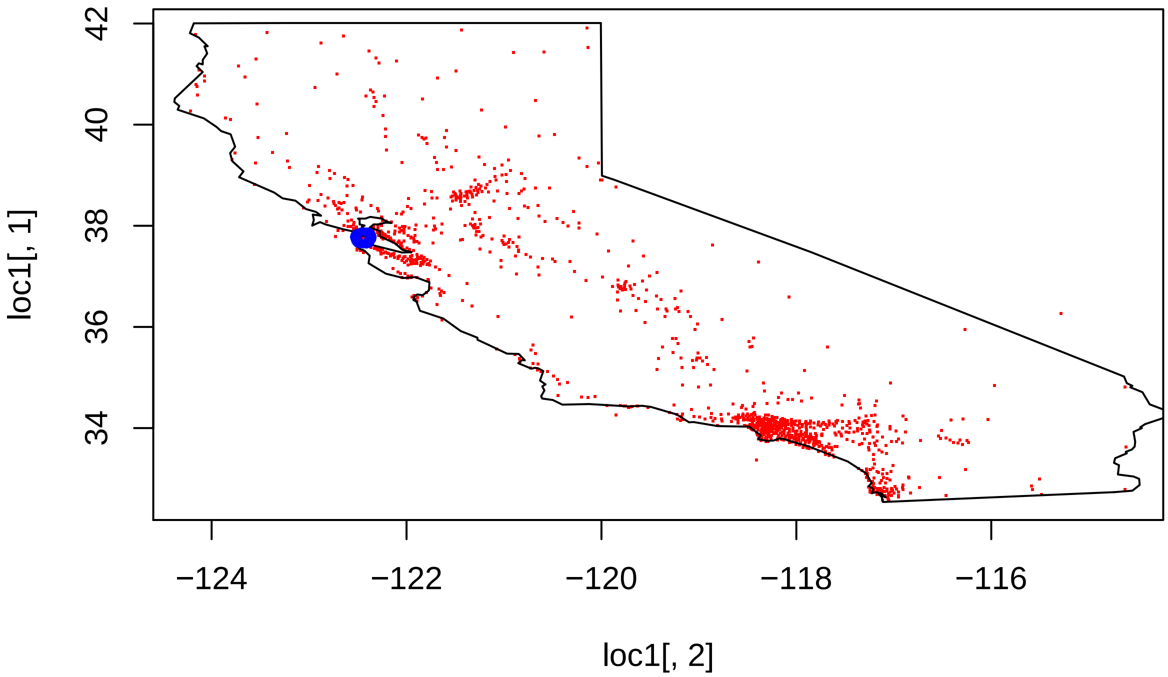
```
##    user  system elapsed
##   0.006   0.000   0.009
## Payment_NPI_total_ca
## physisian --individual  & in ca
NPI_SF <- DT[City=="SAN FRANCISCO" & NPI%in%Payment_NPI_total_ca$NPI ]
setkey(NPI_SF,NPI)
#NPI_SF = NPI_SF[unique(NPI_SF$NPI), mult="first"]
Edge_SF <- Et[V1 %in% unique(NPI_SF$NPI)]
setkey(Edge_SF, V1)

setkey(Payment_NPI_total_ca,NPI)
Payment_SF <- Payment_NPI_total_ca[NPI%in%NPI_SF$NPI]
Payment_SF <- Payment_SF[,.(NPI,totalPay,logPay = log(totalPay+1))]
```

Part 1.1 Look at the positions of Physician in San Francisco.

```
library(zipcode)
library(data.table)
data(zipcode)   # this contains the locations of zip codes
setkey(NPI_SF,NPI)
zip = NPI_SF[as.character(Payment_SF$NPI)]$"Zip Code"
zip = substr(zip, start = 1, stop = 5)

zipcode = as.data.table(zipcode); setkey(zipcode, zip)
loc =  zipcode[zip, c("latitude", "longitude"), with = F]
loc = loc[complete.cases(loc)]
loc = data.frame(loc)

### show the geographic positions
library(maps); library(ggplot2)
```

```
##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990. #
##  # Type '?world' or 'news(package="maps")'. See README_v3. #
```

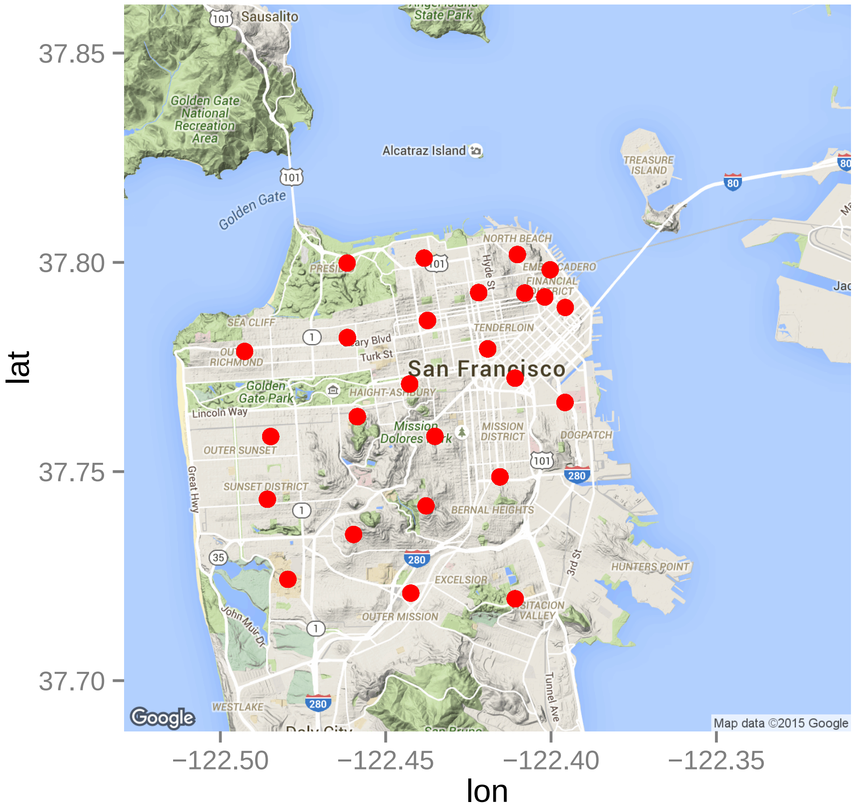
```
library(ggmap)
ca <- DT[State=="CA"]
zip = ca$"Zip Code"
zip = substr(zip, start = 1, stop = 5)

data(zipcode)   # this contains the locations of zip codes
zipcode = as.data.table(zipcode); setkey(zipcode, zip)
loc1 =  zipcode[zip, c("latitude", "longitude"), with = F]
loc1 = loc1[complete.cases(loc1)]
loc1 = data.frame(loc1)
plot(loc1[,2],loc1[,1], pch=".",col="red")
map(database = 'state', region = c('california'),fill=F, add = T)
points(loc[,2],loc[,1],col="blue")
```



```
sfMap = get_map(location = 'San Francisco', zoom = 12)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=San+Francisco&zoom=12&size=640x64
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=San%20Francisco&sense
```

```
ggmap(sfMap) +  geom_point(data=loc,aes(x = longitude, y = latitude,
                                        position="jitter"),color="red", size=3)
```

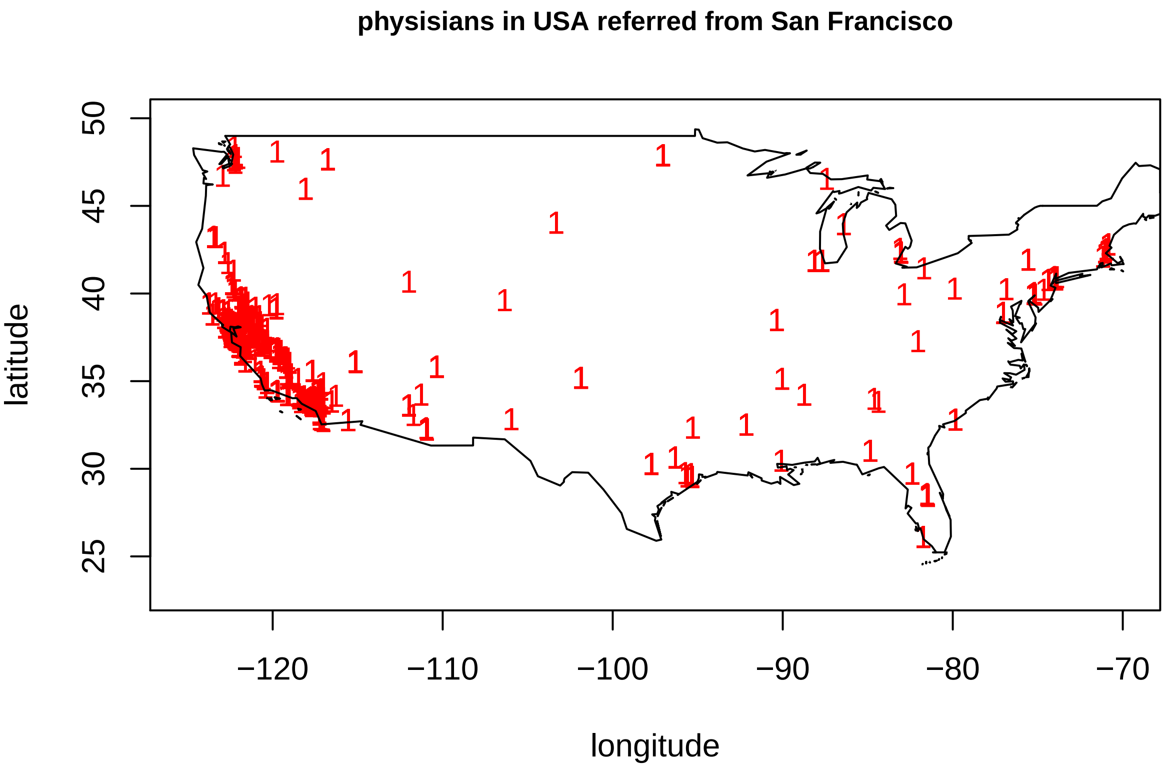


Part 1.2, take a look at how many physicians are outside the San Francisco. They are located all over the country.

```
#Edge_SF <- Edge_SF[V2 %in% V1]
length(unique(Edge_SF$V1))
```

```
## [1] 1743
```

```
outNode <- Edge_SF[,.(V2)]
zip <- DT[outNode]$"Zip Code"
zip = substr(zip, start = 1, stop = 5)

data(zipcode)   # this contains the locations of zip codes
zipcode = as.data.table(zipcode); setkey(zipcode, zip)
loc1 =  zipcode[zip, c("latitude", "longitude"), with = F]
loc1 = loc1[complete.cases(loc1)]
loc1 = data.frame(loc1)
plot(loc1[,2],loc1[,1], pch="1",col="red", xlim= c(-125, -70), ylim= c(23,50),
     xlab = "longitude", ylab ="latitude")
title(main="physicians in USA referred from San Francisco",cex.main=0.8)
map(database = 'world', region = c('usa'),fill=F, add = T)
```



Finding, There are a lot long-distance referrals going on. They are difficult to explain. Even after I already restricted the both nodes in the referral network to be in San Francisco. some doctors may have two or multiple billing address. Also maybe one year 365 time window is too big. reduced the time window may help.

Part 2, show the referral network confined to network among physicians in SF, Trying to show the relationship between network and total payment from Medicare

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

```
Edge_SF1 <- Edge_SF[V2 %in% V1]
paylevel <- function(x){
    high <- quantile(x,probs = 0.90)
    high_medium <- quantile(x,probs = 0.70)
    low_medium <- quantile(x,probs = 0.30)
    low <- quantile(x,probs = 0.10)
    y <- as.character(x)
    y[which(x>=high)]="high"
    y[which(x<high &x>= high_medium)] ="high_medium"
    y[which(x<high_medium &x>= low_medium)] ="medium"
    y[which(x<low_medium &x>= low)] ="low_medium"
    y[which(x<low)] ="low"
    y[is.na(x)]="NA"
    return(y)
}
Payment_SF <-Payment_SF[,.(NPI,totalPay,logPay,payLevel=paylevel(totalPay))]

el=as.matrix(Edge_SF1)[,1:2] #igraph needs the edgelist to be in matrix format
g=graph.edgelist(el,directed = F) # this creates a graph.
g= simplify(g)  # removes any self loops and multiple edges
vcount(g)
```

```
## [1] 1000
```

```
ecount(g)
```

```
## [1] 4742
```

```
ids <- unique(Edge_SF1[,.(V1)])
cities <- DT[ids, mult="first"]$City  ## cannot just simply pick one, having multiple address.
sort(table(cities), decreasing=TRUE)[1:30]
```

```
## cities
##       SAN FRANCISCO          SANTA ROSA           DALY CITY
##                 743                  24                  10
##          BURLINGAME              FRESNO           SAN MATEO
##                   9                   8                   8
##          SACRAMENTO           GREENBRAE        REDWOOD CITY
##                   5                   4                   4
##            PARADISE            SAN JOSE               SELMA
##                   3                   3                   3
##             ALAMEDA             ANTIOCH             HANFORD
##                   2                   2                   2
##           KENTFIELD              MERCED             OAKLAND
##                   2                   2                   2
##             REDDING            ROSEBURG SOUTH SAN FRANCISCO
##                   2                   2                   2
##        WALNUT CREEK       CASTRO VALLEY       COEUR D ALENE
##                   2                   1                   1
##        CORTE MADERA             FREMONT           HOLLISTER
##                   1                   1                   1
##                KATY                LODI          LOMA LINDA
##                   1                   1                   1
```

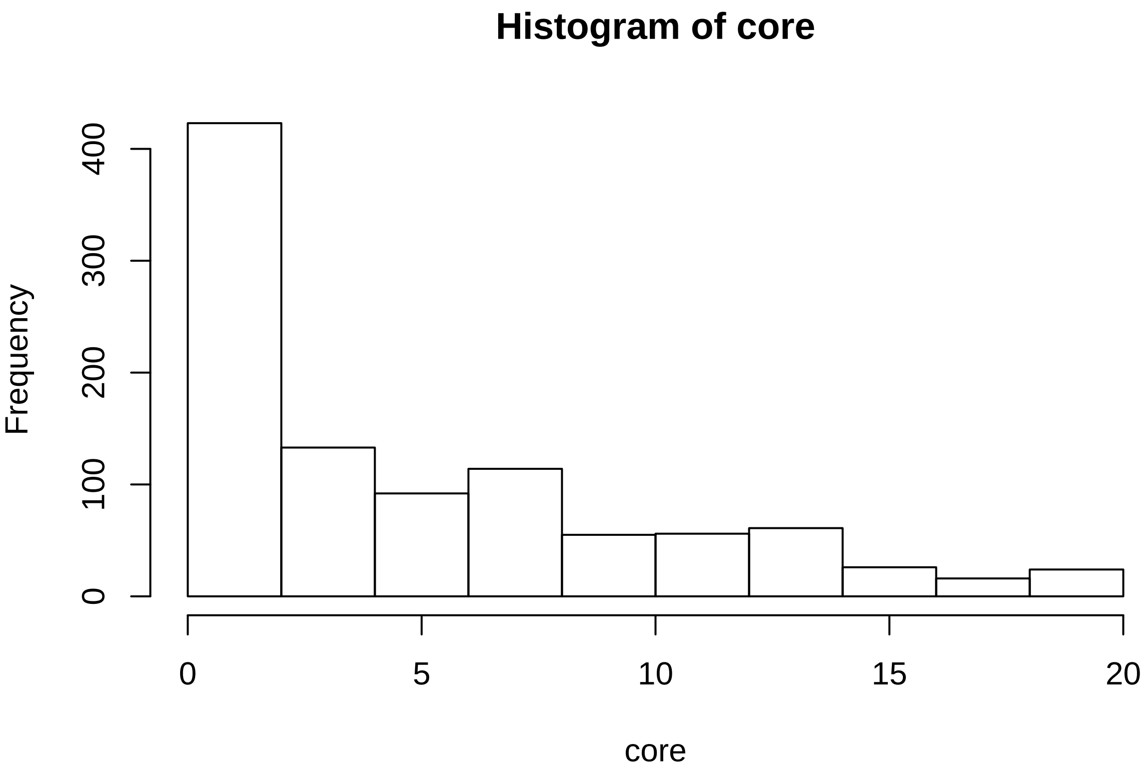
```
states <- DT[ids]$State## cannot just simply pick one, having multiple address.
sort(table(states), decreasing=TRUE)  # most are in CA, many are out sides of SF
```

```
## states
##   CA   TX   AZ   OR   DC   ID   IL   NJ   NV   WA
## 3348    4    3    2    1    1    1    1    1    1
```

```r
clust <- clusters(g)
clust$csize
```

```
##  [1]   4 868  20   3  24   5   2   2   2  18   2   2   2   3   2   7   4
## [18]   3   2   2   4   2   2   2   2   3   2   2   2   2
```

```r
core = graph.coreness(g)  # talk about core.
hist(core)
```



```r
sum(core>3)
```

```
## [1] 485
```

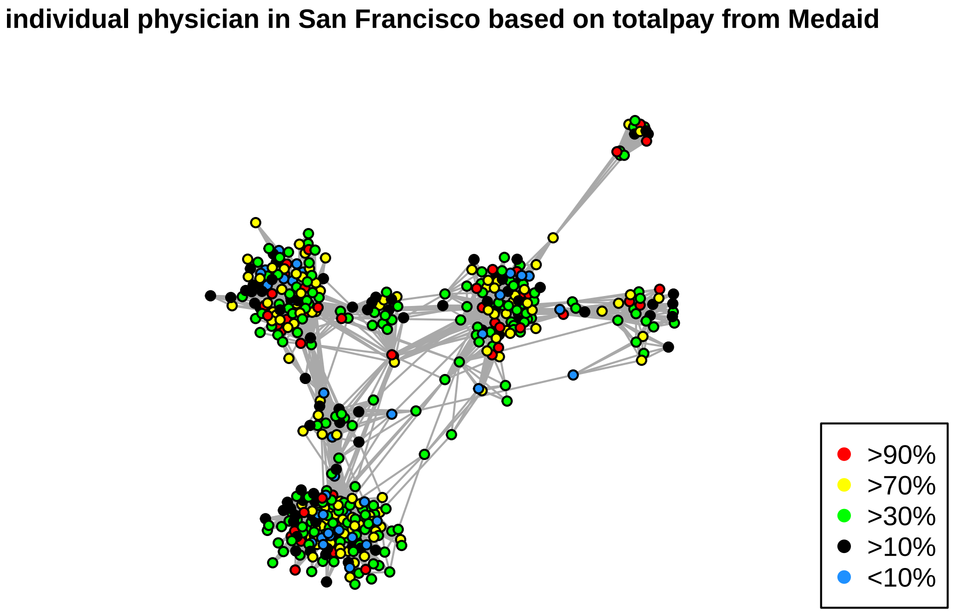
```r
g1 = induced.subgraph(graph = g,vids = V(g)[core>3])  # talk about induced subgraphs.
clust1 <- clusters(g1)
clust1$csize
```

```
## [1] 455  17  13
```

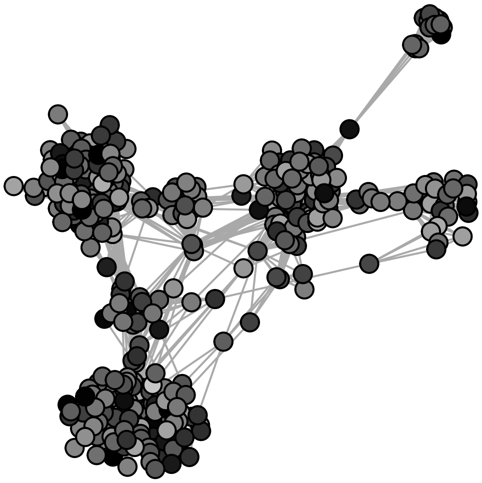
```r
## look at the biggest connected component
g2 <- induced_subgraph(g1,vids = names(which(clust1$membership==1)))
clusters(g2)$csize
```

```
## [1] 455
```

```
layout(1)
v.colors <- as.character(Payment_SF[V(g1)]$payLevel)
v.colors[v.colors=="high"]="red"
v.colors[v.colors=="low"] ="dodgerblue"
v.colors[v.colors=="high_medium"]="yellow"
v.colors[v.colors=="medium"]="green"
v.colors[v.colors=="low_medium"]="black"

set.seed(42)
plot(g2,layout = layout.fruchterman.reingold, vertex.label = NA,
     edge.arrow.size=0.05,  vertex.size=4,
     vertex.color=v.colors)
title(main="individual physician in San Francisco based on totalpay from Medaid",cex.main=0.8)
legend("bottomright",legend=c(">90%",">70%",">30%",">10%","<10%"),
                    col=c("red","yellow","green","black","dodgerblue"), pch=19,
       border = "white",cex =0.8)
```



```
Payment_NPI_SF <- Payment_NPI_ca[NPPES_PROVIDER_CITY=="SAN FRANCISCO"]
NPI_servicecount <- Payment_NPI_SF[,.(countService = sum(LINE_SRVC_CNT)),by=NPI]
NPI_servicecount$logCount <- log(NPI_servicecount$countService)

set.seed(42)
logCount <- NPI_servicecount[V(g2)]$logCount
plot(g2,layout = layout.fruchterman.reingold, vertex.label = NA,
     edge.arrow.size=0.05,  vertex.size=8,
     vertex.color = grey((logCount - min(logCount))/(max(logCount) - min(logCount))) )
title(main="physician network in San Francisco colored on countService",cex.main=0.8)
```



physician network in San Francisco colored on countService

Finding: results show that the clusters in physician referral network are not consistent with the total pay or total number of services. Need to further Explore those high-paid/high service giver may be hubs of the network?

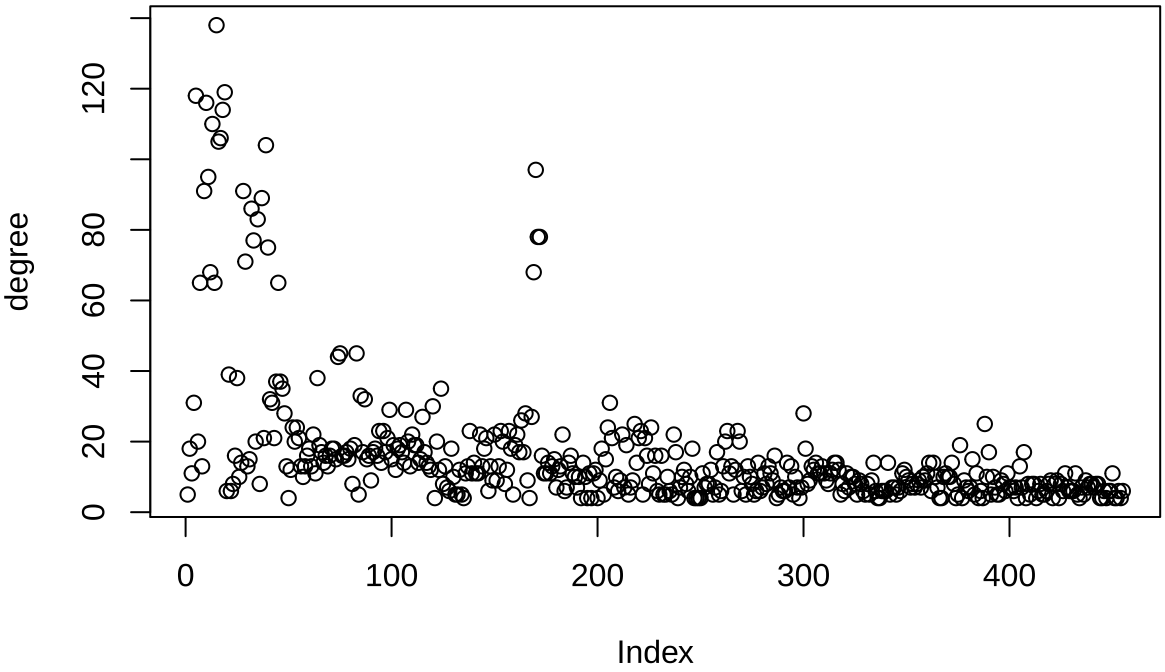
Part 3, Results based on Spectral clustering.

– using spectral clustering to partition the network; – Looking the nodes features, and inteprete the results

```
#get.adjacency(graph, type=c("both", "upper", "lower"),  attr=NULL, names=TRUE, binary=FALSE, sparse=FA
library(Matrix)
Adj2 <- get.adjacency(g2)  ## This is 'dgCMatrix' -- i, p
Matrix::isSymmetric(Adj2)
```

```
## [1] TRUE
```

```
degree <- Matrix::rowSums(Adj2)
plot(degree)
```



```
source("~/Stat/Courses/Physisian_Referral_Network/RScripts/regularSpec/specClust.R")
specClust <- specClust(Adj2,nBlocks = 10, verbose = T)
```

```
## Loading required package: irlba
```

```
V(g2)$label.dist <- 0
set.seed(42)
plot(g2,layout = layout.fruchterman.reingold, vertex.label = NA,
     edge.arrow.size=0.05,  vertex.size=10,
     vertex.color=specClust$cluster)
specClust$eigenVals
```

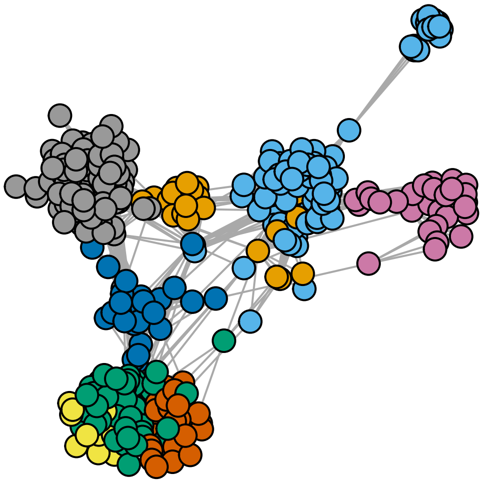
```
##  [1] 0.6216877 0.5385939 0.5060015 0.4043161 0.3781330 0.3656144 0.3388099
##  [8] 0.3237747 0.3116689 0.3038802 0.2981098
```

```
table(specClust$cluster)
```

```
## 
##   1   2   3   4   5   6   7   8   9  10 
##  19  81  72  35  28  54  32 104  16  14
```

```
title(main="physician network in San Francisco colored on SpecCluster",cex.main=0.8)
```

physician network in San Francisco colored on SpecCluster

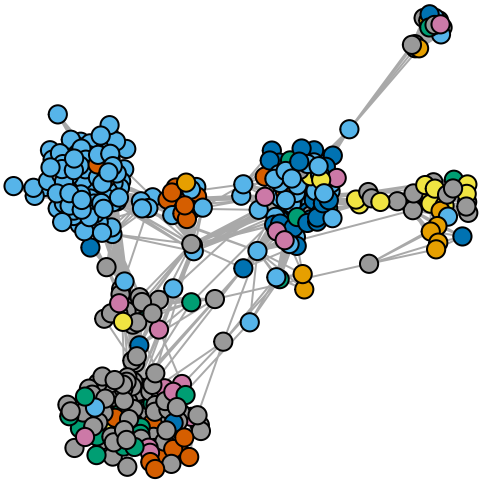


```
##  Trying to interpret the clusters
#1 zip code seems strongly correlated with the network clusterss
zip_SF <- substr(DT[names(V(g2)),mult="first"]$`Zip Code`,1,5)
#table(specClust$cluster,DT[names(V(g2)),mult="first"]$`Primary specialty`)
tab <- table(specClust$cluster,substr(DT[names(V(g2)),mult="first"]$`Zip Code`,1,5))
for( i in 1:10){
    print(colnames(tab)[order(tab[i,],decreasing = T)[1:4]])
}
```

```
## [1] "94110" "94143" "94116" "94010"
## [1] "94117" "94109" "94132" "94133"
## [1] "94115" "94118" "94110" "95405"
## [1] "94115" "95405" "94118" "95816"
## [1] "94115" "94114" "94118" "94109"
## [1] "94115" "94110" "94114" "94118"
## [1] "94108" "94133" "94134" "94115"
## [1] "94143" "94110" "94115" "94117"
## [1] "94117" "94062" "94015" "94109"
## [1] "94904" "94118" "94143" "94010"
```

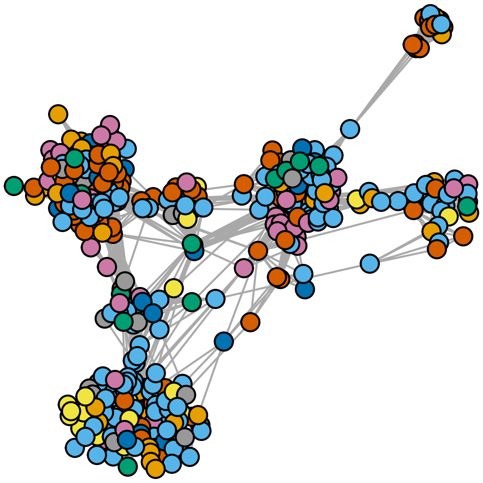
```
set.seed(42)
plot(g2,layout = layout.fruchterman.reingold, vertex.label = NA,
      edge.arrow.size=0.05,  vertex.size=8,
      vertex.color=as.factor(zip_SF))
title(main="physician network in San Francisco Colored based on zip code",cex.main=0.8)
```

physician network in San Francisco Colored based on zip code



```
set.seed(42)
plot(g2,layout = layout.fruchterman.reingold, vertex.label = NA,
     edge.arrow.size=0.05,  vertex.size=8,
     vertex.color=as.factor(DT[names(V(g2)),mult="first"]$`Primary specialty`))
title(main="physician network in San Francisco Colored based on specialty", cex.main=0.8)
```

physician network in San Francisco Colored based on specialty



Findings, Specialties don't correspond to clusters in the network of physisians, they are scattly distributed in the network. It seems that most correlated feature is zip code.

Part 5

Potential direction to try: – construct specialty network, individual physician may not be very informative. specialty is a concentration version of the network. – constrain the data further to a zip code, to exclude the location effect on network. – Look at referral network of 60-day or 30-day to exclude the long range referral.