Assignment2

Pradnya Nimbekar

Problem 1

Problem 1 (45%). First, briefly describe the fourth real-world network you chose to work with for this problem and provide a URL to where it is located. For each of the four real-world networks you study, identify the node(s) in the network with the highest scores in terms of the following centrality measures: (i) Degree, (ii) Eccentricity, (iii) Closeness, (iv) Betweenness, (v) Katz index, (vi) PageRank, (vii) Kleinberg's Authority score, and (viii) Kleinberg's Hub score. Try to find out in what way the nodes identified by these measures as the most important might be important in the network. Are there cases in which the different centrality measures identify the same node(s) as the most important? Discuss your results.

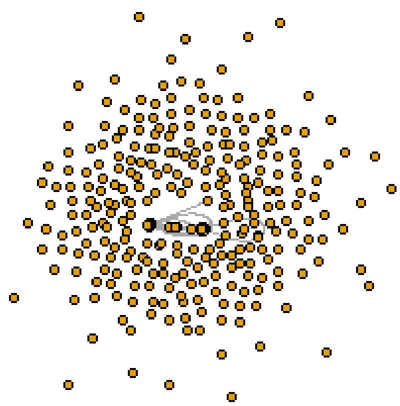
Answer:

I'm selecting fourth real world network as 'American College Football' and th URL is http://www-personal.umich.edu/~mejn/netdata/. This data set contains the network of American football games for Fall 2000. The nodes have values that indicate to which conferences they belong. 0 = Atlantic Coast, 1 = Big East, 2 = Big Ten, 3 = Big Twelve, 4 = Conference USA, 5 = Independents, 6 = Mid-American, 7 = Mountain West, 8 = Pacific Ten, 9 = Southeastern, 10 = Sun Belt, 11 = Western Athletic

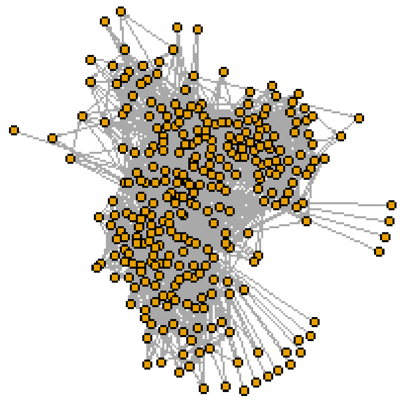
```
library(igraph)

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.4

library(centiserve)

## Warning: package 'centiserve' was built under R version 4.0.4
```

```
## Loading required package: Matrix
```

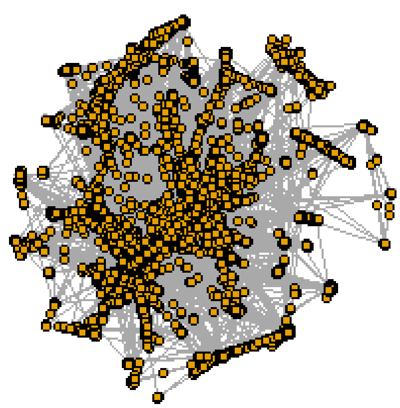
```
pb <- read.graph("polblogs.gml","gml")
nn <- read.graph("celegansneural.gml","gml")
inn <- read.graph("as-22july06.gml","gml")
fb <- read.graph("football.gml","gml")

pb_nodes <- vcount(pb)
nn_nodes <- vcount(nn)
inn_nodes <- vcount(inn)
fb_nodes <- vcount(fb)

pb_edges <- gsize(pb)
nn_edges <- gsize(nn)
inn_edges <- gsize(inn)
fb_edges <- gsize(fb)

plot(pb, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5)
```



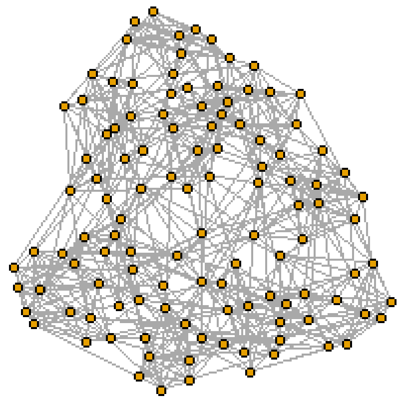
```
plot(nn, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5)
```



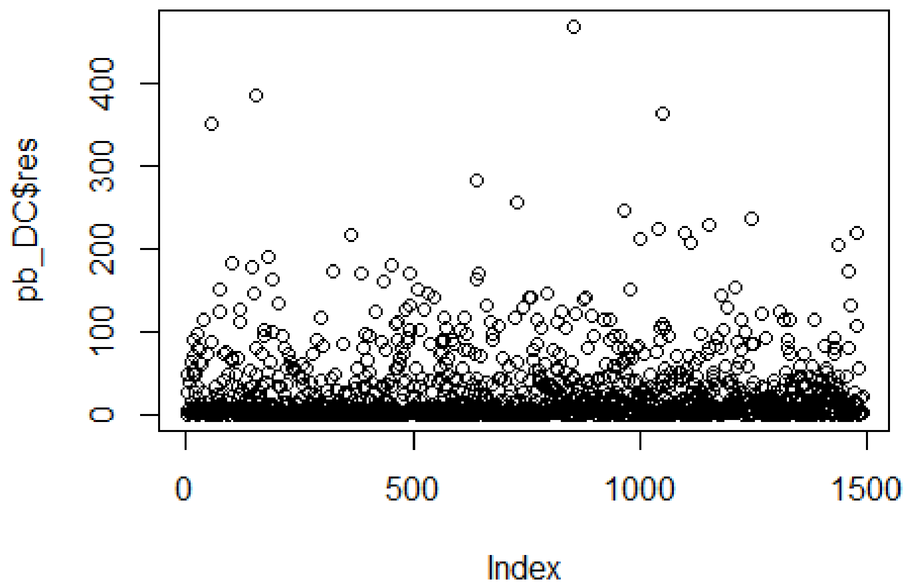
```
plot(inn, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5)
```



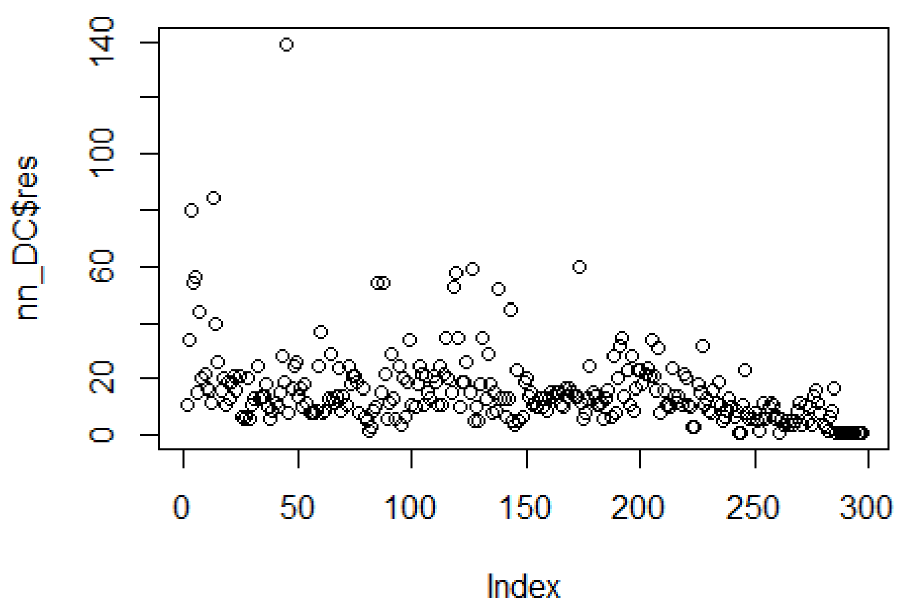
```
plot(fb, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5)
```



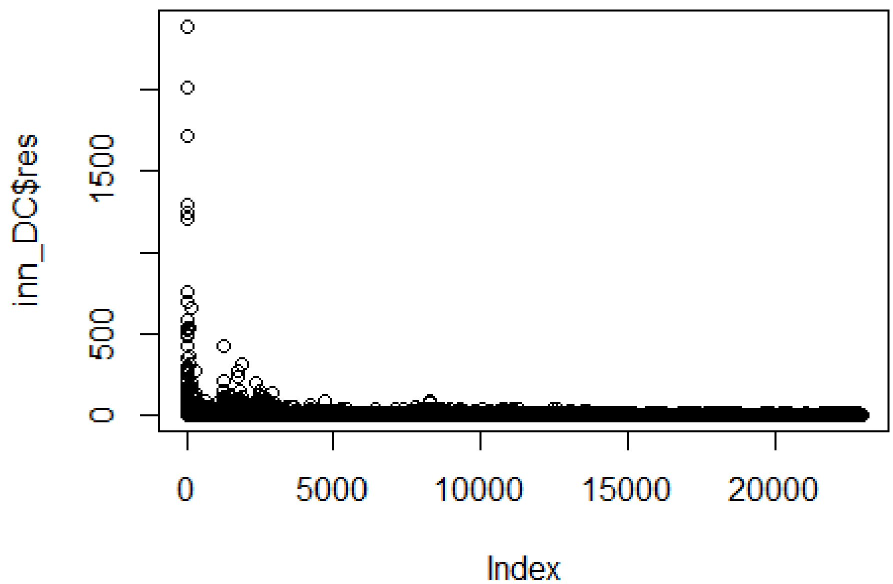
```
#Degree Centrality

pb_DC <- centr_degree(graph = pb , mode = "all" )
nn_DC <- centr_degree(graph =nn , mode = "all" )
inn_DC <- centr_degree(graph =inn , mode = "all" )
fb_DC <- centr_degree(graph =fb , mode = "all" )

plot(pb_DC$res)
```



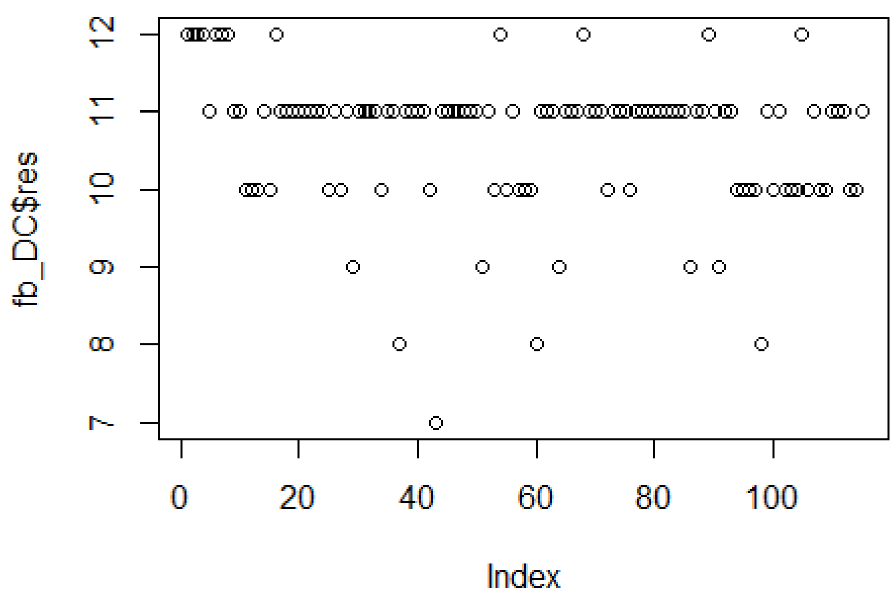

```
plot(nn_DC$res)
```




```
plot(inn_DC$res)
```




```
plot(fb_DC$res)
```



```r
#Eccentricity

pb_ec <- eccentricity(pb, vids = V(pb), mode = c("all", "out", "in", "total")
)
nn_ec <- eccentricity(nn, vids = V(nn), mode = c("all", "out", "in", "total")
)
inn_ec <- eccentricity(inn, vids = V(inn), mode = c("all", "out", "in", "tota
l"))
fb_ec <- eccentricity(fb, vids = V(fb))

#Closeness

pb_close <- closeness(pb, vids = V(pb))

## Warning in closeness(pb, vids = V(pb)): At centrality.c:2784 :closeness
## centrality is not well-defined for disconnected graphs

nn_close <- closeness(nn, vids = V(nn))

## Warning in closeness(nn, vids = V(nn)): At centrality.c:2784 :closeness
## centrality is not well-defined for disconnected graphs

inn_close <- closeness(inn, vids = V(inn))
fb_close <- closeness(fb)

#Betweenness

pb_bet <- betweenness(pb, directed = TRUE)
nn_bet <- betweenness(nn, directed = TRUE)
inn_bet <- betweenness(inn, directed = TRUE)
fb_bet <- betweenness(fb, directed = TRUE)

# PageRank

pb_pr <- page_rank(pb,  directed = TRUE)$vector
nn_pr <- page_rank(nn,  directed = TRUE)$vector
inn_pr <- page_rank(inn,  directed = TRUE)$vector
fb_pr <- page_rank(fb,  directed = TRUE)$vector

# Kleinberg’s Authority score
pb_ka <- authority.score(pb,scale = TRUE)$vector
nn_ka <- authority.score(nn,scale = TRUE)$vector
inn_ka <- authority.score(inn,  scale = TRUE)$vector
fb_ka <- authority.score(fb,  scale = TRUE)$vector

# Kleinberg’s Hub score
pb_kh <- hub_score(pb,  scale = TRUE)$vector
nn_kh <- hub_score(nn,  scale = TRUE)$vector
inn_kh <- hub_score(inn,  scale = TRUE)$vector
fb_kh <- hub_score(fb,  scale = TRUE)$vector

#res <- katzcent(fb,vids= V(fb), alpa=0.1)
```

```r
#which.max(res)

#political Blog
which.max(pb_DC$res)

## [1] 855

which.max(pb_ec)

## [1] 794

which.max(pb_close)

## [1] 293

which.max(pb_bet)

## [1] 855

which.max(pb_pr)

## [1] 155

which.max(pb_ka)

## [1] 155

which.max(pb_kh)

## [1] 512

#-------------------------------------------------
#Neural Network
which.max(nn_DC$res)

## [1] 45

which.max(nn_ec)

## [1] 82

which.max(nn_close)

## [1] 260

which.max(nn_bet)

## [1] 178

which.max(nn_pr)

## [1] 45

which.max(nn_ka)

## [1] 45
```

```r
which.max(nn_kh)

## [1] 126

#------------------------------------------------
#Internet
which.max(inn_DC$res)

## [1] 4

which.max(inn_ec)

## [1] 9200

which.max(inn_close)

## [1] 23

which.max(inn_bet)

## [1] 4

which.max(inn_pr)

## [1] 4

which.max(inn_ka)

## [1] 4

which.max(inn_kh)

## [1] 4

#------------------------------------------------
#Football Network
which.max(fb_DC$res)

## [1] 1

which.max(fb_ec)

## [1] 2

which.max(fb_close)

## [1] 59

which.max(fb_bet)

## [1] 83

which.max(fb_pr)

## [1] 6
```

```
which.max(fb_ka)

## [1] 68

which.max(fb_kh)

## [1] 68
```

Graph	Degree	Eccentricity	Closeness	Betweenness	Page Rank	Kleinberg's Authority score	Kleinberg's Hub Score	Katz Index
Political Blogs	855	794	293	855	155	155	512	-
Neural Network	45	82	260	178	45	45	126	-
Internet	4	9200	23	4	4	4	4	-
Football	1	2	59	83	6	68	68	-

Observations:

We know that centrality denotes the order of importance on the vertices or edges of a network by assigning real values to them. From above values, we can observe that for political blogs, some values are repeated, for instance, 855 and 155 which means they are overpowering other values. So, node 855 has the highest importance as per degree centrality and betweenness centrality for Political blogs. For PageRank and Kleinberg's Hub score 155 is highest among all. The most commonly used node in the neural network is node 45, which has the highest degrees of centrality, page rank, and Klienberg Authority ranking. Klienberg Authority and Hub score (68) are same for last network, that is, football graph and is dominant one with highest scores.

Problem 2:

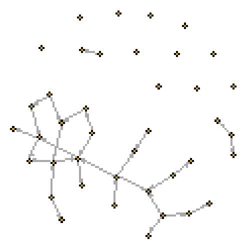
Generate two types of random graphs using igraph's methods for the Erdos-Renyi and Barabasi-Albert (preferential attachment) models. (The Barabasi-Albert model is a simple stochastic algorithm that generates a scale-free graph, a graph with a Power-Law degree distribution.) Let the number of nodes in each case be 20. Choose the parameters in the two models such that the two graphs you generate have roughly the same number of edges. In the same fashion, generate another set of two graphs, this time with the number of nodes in each case being 40, instead of 20.

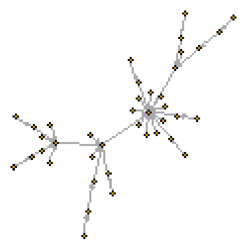
Answer:

```
# Two types of random graphs using Erdos-Renyi and Barabasi-Albert with of no
des 20 and 40

rg1 <- erdos.renyi.game(n = 20, p = 0.04, type = c("gnp"), directed = TRUE)
rg2 <- barabasi.game(n = 20, power = 1, m = NULL, directed = TRUE)
rg3 <- erdos.renyi.game(n = 40, p = 0.02, type = c("gnp"), directed = TRUE)
rg4 <- barabasi.game(n = 40, power = 1, m = NULL, directed = TRUE)

gsize(rg1)

## [1] 16

gsize(rg2)

## [1] 19

gsize(rg3)

## [1] 29

gsize(rg4)

## [1] 39

vcount(rg1)

## [1] 20

vcount(rg2)

## [1] 20

vcount(rg3)

## [1] 40

vcount(rg4)

## [1] 40

par(mfrow=c(1,2))
plot(rg1, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5, xlab = "Erd
os Renyi Graph: 20 nodes")
plot(rg2, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5, xlab = "Bar
abasi Graph: 20 nodes")
```

Erdos Renyi Graph: 20 nodes   Barabasi Graph: 20 nodes

```
par(mfrow=c(1,2))
plot(rg3, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5, xlab = "Erd
os Renyi Graph: 40 nodes")
plot(rg4, vertex.label= NA, edge.arrow.size=0.1, vertex.size = 5, xlab = "Bar
abasi Graph : 40 nodes")
```





Erdos Renyi Graph: 40 nodes   Barabasi Graph : 40 nodes

Part A: For each of the four graphs you generated, compute all eigenvalues and corresponding

eigenvectors of the Laplacian of the graph (e.g. using MatLab's (or Numpy's) eig function). Do not include the results of this computation in your submission, but you will need the result for the subsequent subproblems.

Answer:

```
# eigenvalues and respective eigenvectors of the Laplacian of the graph

l_m1 <- laplacian_matrix(rg1, normalized = FALSE)
e1 <- eigen(l_m1)
e_val1 <- e1$values
e_vect1 <- e1$vectors

l_m2 <- laplacian_matrix(rg2, normalized = FALSE)
e2 <- eigen(l_m2)
e_val2 <- e2$values
e_vect1 <- e2$vectors

l_m3 <- laplacian_matrix(rg3, normalized = FALSE)
e3 <- eigen(l_m3)
e_val3 <- e3$values
e_vect1 <- e3$vectors

l_m4 <- laplacian_matrix(rg4, normalized = FALSE)
```

```r
e4 <- eigen(l_m4)
e_val4 <- e4$values
e_vect1 <- e4$vectors
```

Part B:

Complete the following table for each graph. (In case the graph you generated consists of more than one component, in populating the table, consider the largest connected component.) Graph n m dmin dmax l D ccg λ2 λn

where n is the number of nodes; m is the number of edges; dmin is the minimum degree in the graph; dmax is the maximum degree in the graph; l is the average path length; D is the diameter; ccg is the global clustering coefficient; λ2 is the second smallest eigenvalue (the algebraic connectivity); and λn is the largest eigenvalue. Discuss the observations you make out of these data.

Answer:

```r
#Minimum Degree in the graph

min(degree(rg1))

## [1] 0

min(degree(rg2))

## [1] 1

min(degree(rg3))

## [1] 0

min(degree(rg4))

## [1] 1

# #Minimum Degree in the graph

max(degree(rg1))

## [1] 4

max(degree(rg2))

## [1] 8

max(degree(rg3))

## [1] 5
```

```r
max(degree(rg4))

## [1] 14

#Average path length

average.path.length(rg1)

## [1] 1.416667

average.path.length(rg2)

## [1] 1.515152

average.path.length(rg3)

## [1] 2.065789

average.path.length(rg4)

## [1] 1.792683

#Diameter

diameter(rg1)

## [1] 3

diameter(rg2)

## [1] 3

diameter(rg3)

## [1] 5

diameter(rg4)

## [1] 4

# Clustering coefficient

transitivity(graph = rg1, type = c("global"))

## [1] 0

transitivity(graph = rg2, type = c("global"))

## [1] 0

transitivity(graph = rg3, type = c("global"))

## [1] 0

transitivity(graph = rg4, type = c("global"))
```

```
## [1] 0

#largest eigenvalue
max(abs(e_val1))

## [1] 2

#second smallest eigenvalue
abs(sort(e_val1, decreasing=F)[2])

## [1] 0

#largest eigenvalue
max(abs(e_val2))

## [1] 1

#second smallest eigenvalue
abs(sort(e_val2, decreasing=F)[2])

## [1] 1

#largest eigenvalue
max(abs(e_val3))

## [1] 3

#second smallest eigenvalue
abs(sort(e_val3, decreasing=F)[2])

## [1] 0

#largest eigenvalue
max(abs(e_val4))

## [1] 1

#second smallest eigenvalue
abs(sort(e_val4, decreasing=F)[2])

## [1] 1
```

Graph	n	m	dmin	dmax	l	D	ccg	λ2	λn
Erdos Renyi Graph 1	20	16	0	4	1.42	3	0	0	2
Barabasi Random Graph 1	20	19	1	8	1.52	3	0	1	1
Erdos Renyi Graph 2	40	29	0	5	2.07	5	0	0	3
Barabasi Random Graph 2	40	39	0	14	1.79	4	0	1	1

Observations:

Calculations for four randomly generated graphs using the Erdos Renyi and Barabasi-Albert methods are shown in the table below. For both approaches, networks has been created with 20 and 40 nodes, respectively, keeping the number of edges close. Despite the fact that the networks have the same number of nodes and vertices, I can say from the table that the Barabasi Random graph with nodes of 20 and 40 is the most prominent, with the lowest and highest degrees. The average path length has closer values for all four networks

---

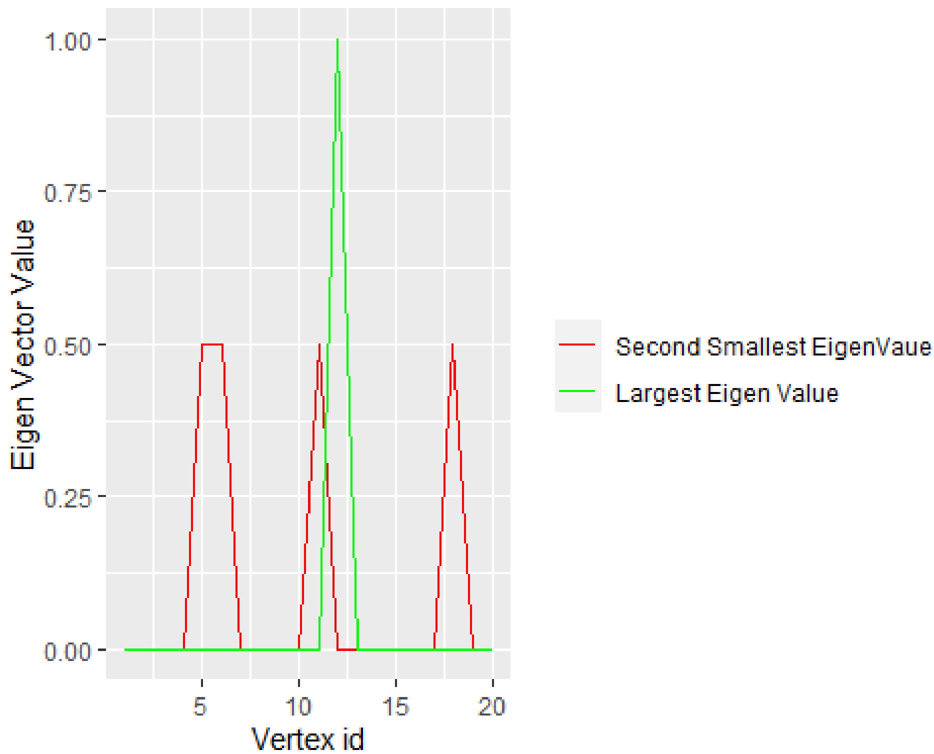
Part3:

For this subproblem you will focus on the 20-nodes graphs in each of the two models. You will generate two figures, one for each model. In each figure, plot two quantities together: the eigenvector corresponding to the second-smallest eigenvalue (λ2) and the eigenvector corresponding to the largest eigenvalue (λn). In the plots, the x-axis would show vertex ids and the y-axis shows value of the eigenvector. Compare and contrast the plots of the two graphs, and discuss any observations you make.

Answer:

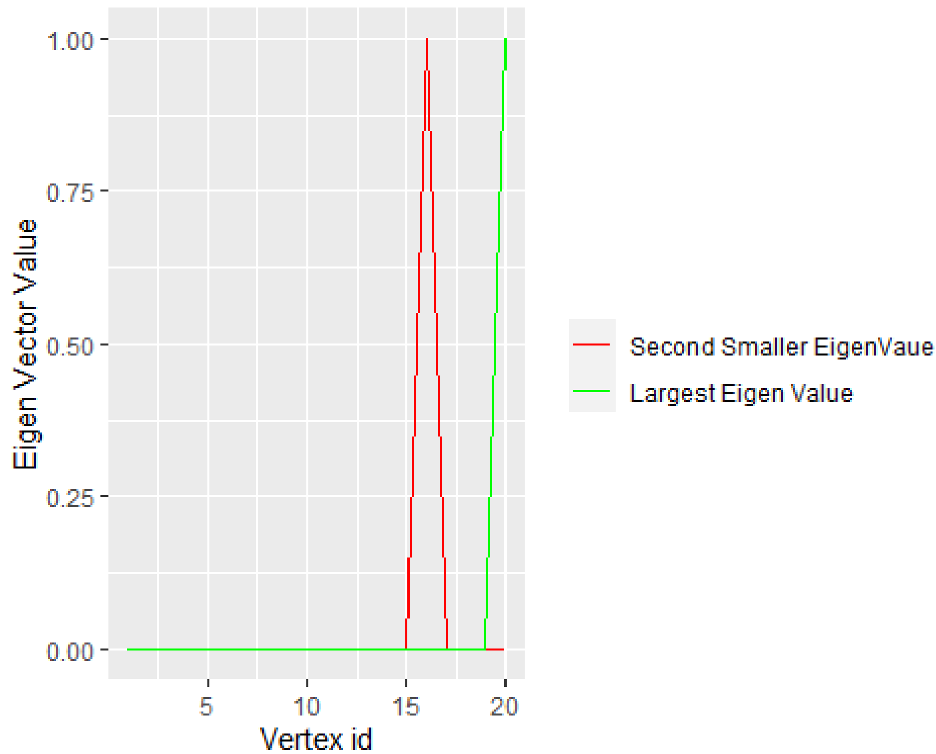
```
#For Erdos Renyi network with 20 nodes
data1 <-  data.frame(x = c(1:20),
                     secondSmallestEigenVaue = eigen(graph.laplacian(rg1))$vec
tors[,19],
                     largestEigenValue = eigen(graph.laplacian(rg1))$vectors[,
```

```
which.max(eigen(graph.laplacian(rg1))$values)] )

ggplot(data1, aes(x = x)) +
  geom_line(aes(y = secondSmallestEigenVaue, colour = "Second Smallest EigenV
aue")) +
  geom_line(aes(y = largestEigenValue, colour="Largest Eigen Value")) +
  xlab("Vertex id") +
  ylab("Eigen Vector Value")+
  scale_colour_manual("",
                      breaks = c("Second Smallest EigenVaue", "Largest Eigen
Value"),
                      values = c("red", "green"))
```




```
#For Barabasi network with 20 nodes
data2 = data.frame(x = c(1:20),
                      secondSmallestEigenVaue = eigen(graph.laplacian(rg2))$v
ectors[,19],
                      largestEigenValue = eigen(graph.laplacian(rg2))$vectors
[,which.max(eigen(graph.laplacian(rg2))$values)])


#Plot for Barabasi network with 20 nodes
ggplot(data2, aes(x = x)) +
  geom_line(aes(y = secondSmallestEigenVaue, colour = "Second Smaller EigenVa
ue")) +
  geom_line(aes(y = largestEigenValue, colour="Largest Eigen Value")) +
```

```
  xlab("Vertex id") +
  ylab("Eigen Vector Value")+
  scale_colour_manual("",
                      breaks = c("Second Smaller EigenVaue", "Largest Eigen V
alue"),
                      values = c("red", "green"))
```



Observations:

Above plots shows that largest eigen values mostly is equal to zero except for some nodes.

Poblem3:

Read the first four sections of the SIAM Review article "PageRank Beyond the Web" by David Gleich (note a copy of the article is posted on the course's page on Canvas under the Module PageRank). From the very wide range of applications of PageRank discussed in section 4 of the article, pick three that you personally found most fascinating (or interesting) and tell me why. For each application you picked, a couple of sentences stating your reasons is adequate, but you are welcome to write a few more sentences, if you wish.

Answer:

First fascinating application I found is the pagerank in recommender system. Amazon and Netflix are using popular recommendation systems for their products and movies respectively. The most popular one is item based collaborative filtering. According to the article, many research studies on recommender systems use localized page rank to rate possible predictions. On a database reformulationgraph that explains how users rewrite queries, Boldi et al. [2008] run localized PageRank. Link prediction attempts to predict which edges will appear in the future based on the current state of a network. Interesting thing about page rank in recommendation system is that it had a lot of success with several tasks related to query recommendation, and it is often rated as on of the best methods. So, applying pagerank to this recommendation system would provide accurate results.

Second application is PageRank in literature, where page rank as a centrality measure useful in finding most important books. For a single book, hypertextual literature includes several potential story paths. The random surfer model for PageRank, according to Kontopoulou et al. [2012], perfectly maps to how users read these books which would eventually help in getting an understanding of properties of the stories and tells which story paths in hypertextual literature are most suitable. To provide the solution to traditional content search problem, BookRank which is localized PageRank gives accurate suggestions for what to read next which would not only save the time but also cost.

Third common but most interesting application is the PageRank in social network. PageRank addresses many problems in social networking, for example, link prediction problem, evaluating the centrality of the people and the most importatn one is finding influential individuals.PageRank has been used to rate individuals in the Twitter network based on their relevance and to characterize properties of the Twitter social network based on their users' PageRank values.