Elements of Network Science : Assignment2

Sharath Kumar Karnati

2024-02-29

```r
knitr::opts_chunk$set(echo = TRUE)
```

Question 1: The fourth real network is taken from http://www-personal.umich.edu/~mejn/netdata/. The name of the network chosen is: Zachary's karate club It is a social network of 34 friends in a karate club at a US university that is described by Wayne Zachary in 1977.

Degree: The node with the greatest central connectivity in the graph or network has the highest degree. Centrality gauges the strength of the network. Thus, the most centrally connected nodes of the aforementioned networks are the Internet (node 4), Neural Network (node 45), Political Blogs (node 885), and Karate (node 34). networks, correspondingly. Eccentricity: Eccentricity is the shortest path a node can take to go to any other node that can be reached.
The metric known as "closeness" indicates a node's proximity to other nodes inside the network. Node 23 is the nearest node to every other node in the internet network, for instance. Betweenness: The number of pathways a node uses to connect to other nodes in the network.In an internet network, for instance, node 4 has the greatest number of paths in the graph.
Page rank: This indicates how significant a node is inside the graph.
Hub Score and Authority of Kleinberg - The total of the hub values that are used to determine the authority value cite the page that is in use. In a similar vein, the scaled authority values of the pages a hub links to establish its value.

```r
# importing the igraph package
library(igraph)
```

```
##
## Attaching package: 'igraph'

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

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

```r
setwd("C:/Users/shara/OneDrive/Desktop/elemts")
getwd()
```

```
## [1] "C:/Users/shara/OneDrive/Desktop/elemts"
```

```r
# reading the graphs for Political Blogs, Neural Network and Internet
politicalBlogs <- read.graph("as-22july06/as-22july06.gml","gml")
```

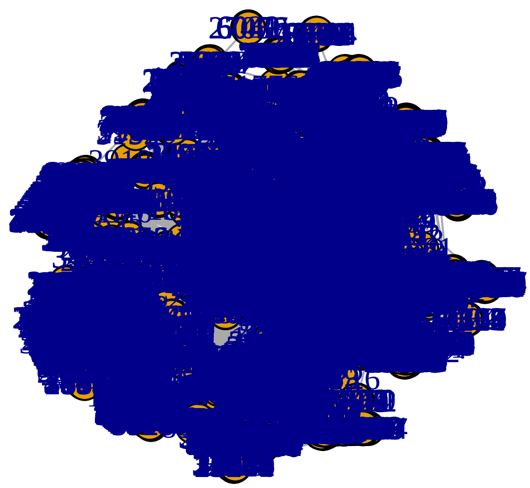
```
## Warning: `read.graph()` was deprecated in igraph 2.0.0.
## i Please use `read_graph()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## generated.
```

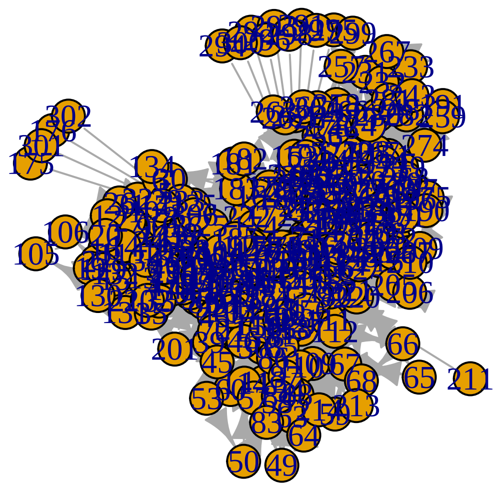
```r
neuralNetwork <- read.graph("celegansneural/celegansneural.gml","gml")
internet <- read.graph("polblogs/polblogs.gml","gml")
```

```
## Warning in read.graph.gml(file, ...): At vendor/cigraph/src/io/gml.c:149 : One
## or more unknown entities will be returned verbatim (&#38;).
```

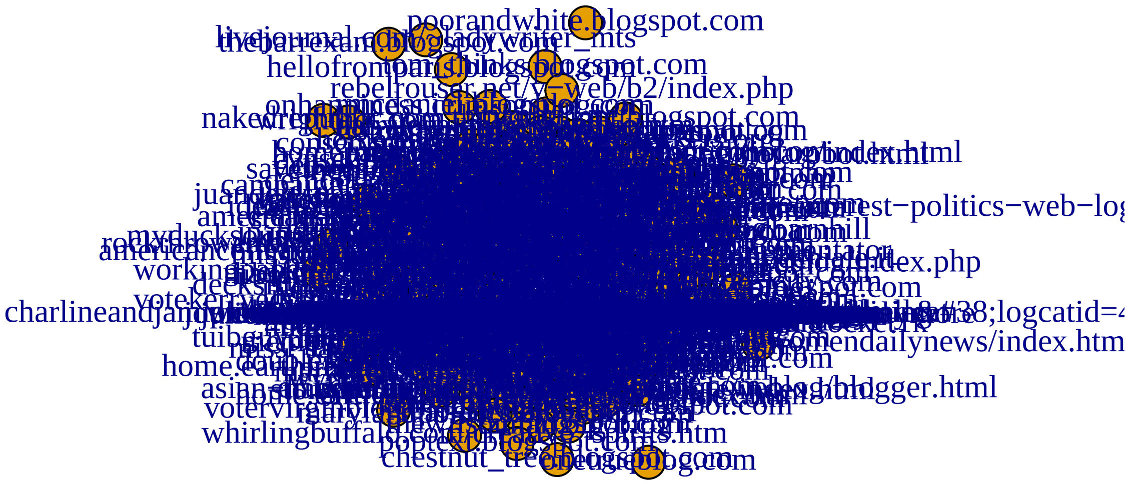
```r
# Plotting the three graphs
 plot(politicalBlogs)
```



```r
 plot(neuralNetwork)
```

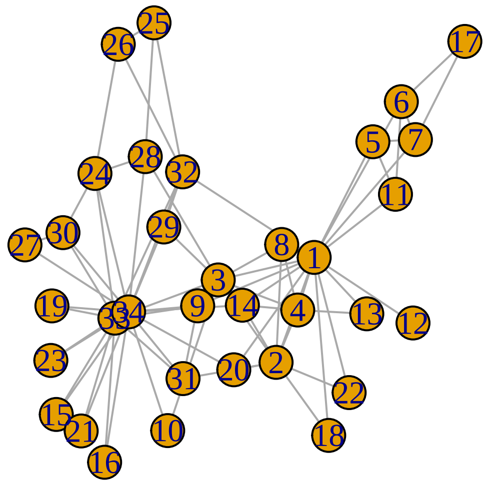


```r
plot(internet)
```



```r
#reading the fourth graph I chose "Zachary Karate Club"
KarateClub <- read.graph("karate/karate.gml","gml")

# Plotting the Karate club network
plot(KarateClub)
```




```r
# List of network and their corresponding names

network <- list(politicalBlogs, neuralNetwork, internet, KarateClub)
networkNames <- list("Political Blogs", "Neural Network", "Internet", "KarateClub")
 i <- 1

for(net in network){
# Centrality measures to be calculated based on Problem 1
   deg <- degree(net)
   ecc <- eccentricity(net)
   cl <- closeness(net)
   btw <- betweenness(net)
   pr <- page_rank(net)$vector
   kAuth <- authority_score(net)$vector
   kHub <- hub_score(net)$vector

# Finding highest scores for each centrality measure
   degHighest <- which(deg == max(deg))
   eccHighest <- which(ecc == max(ecc))
   clHighest <- which.max(cl)
   btwHighest <- which(btw == max(btw))
   prHighest <- which(pr == max(pr))
   kAuthHighest <- which(kAuth == max(kAuth))
   kHubHighest <- which(kHub == max(kHub))

   # Printing values
```

```r
    cat("------------------------------", toString(networkNames[i]), "------------------------------", "
    cat("Node(s) with highest degree centrality:", degHighest, "\n")
    cat("Node(s) with highest eccentricity centrality:", eccHighest, "\n")
    cat("Node(s) with highest closeness centrality:", clHighest, "\n")
    cat("Node(s) with highest betweenness centrality:", btwHighest, "\n")
    cat("Node(s) with highest PageRank:", prHighest, "\n")
    cat("Node(s) with highest Kleinberg's Authority score:", kAuthHighest, "\n")
    cat("Node(s) with highest Kleinberg's Hub score:", kHubHighest, "\n")
i=i+1
}
```

```
## ------------------------------ Political Blogs ------------------------------
## Node(s) with highest degree centrality: 4
## Node(s) with highest eccentricity centrality: 9200 16852
## Node(s) with highest closeness centrality: 23
## Node(s) with highest betweenness centrality: 4
## Node(s) with highest PageRank: 4
## Node(s) with highest Kleinberg's Authority score: 4
## Node(s) with highest Kleinberg's Hub score: 4
## ------------------------------ Neural Network ------------------------------
## Node(s) with highest degree centrality: 45
## Node(s) with highest eccentricity centrality: 82 127 129 243 244 296 297
## Node(s) with highest closeness centrality: 26
## Node(s) with highest betweenness centrality: 178
## Node(s) with highest PageRank: 45
## Node(s) with highest Kleinberg's Authority score: 45
## Node(s) with highest Kleinberg's Hub score: 126
## ------------------------------ Internet ------------------------------
## Node(s) with highest degree centrality: 855
## Node(s) with highest eccentricity centrality: 794 1259
## Node(s) with highest closeness centrality: 116
## Node(s) with highest betweenness centrality: 855
## Node(s) with highest PageRank: 155
## Node(s) with highest Kleinberg's Authority score: 155
## Node(s) with highest Kleinberg's Hub score: 512
## ------------------------------ KarateClub ------------------------------
## Node(s) with highest degree centrality: 34
## Node(s) with highest eccentricity centrality: 15 16 17 19 21 23 24 27 30
## Node(s) with highest closeness centrality: 1
## Node(s) with highest betweenness centrality: 1
## Node(s) with highest PageRank: 34
## Node(s) with highest Kleinberg's Authority score: 34
## Node(s) with highest Kleinberg's Hub score: 34
```

Question 2:

```r
set.seed(123)
erGraph1 <- erdos.renyi.game(20,19,type = "gnm")
barabasiGraph1 <- barabasi.game(20, directed = FALSE)
```

```
## Warning: `barabasi.game()` was deprecated in igraph 2.0.0.
## i Please use `sample_pa()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
E(erGraph1)
```

```
## + 19/19 edges from 732a8eb:
##  [1]  2-- 3  2-- 6  4-- 6  5-- 6  2--11  9--11  1--12  8--12  6--13  7--13
## [11]  5--14 10--14  5--15  5--17  7--17 10--18  4--20 11--20 19--20
```

```r
E(barabasiGraph1)
```

```
## + 19/19 edges from 7331b65:
##  [1]  1-- 2  2-- 3  2-- 4  3-- 5  5-- 6  4-- 7  5-- 8  4-- 9  5--10  3--11
## [11]  2--12 12--13 12--14  9--15 12--16  1--17  7--18 12--19  3--20
```

```r
erGraph2 <- erdos.renyi.game(40,39, type = "gnm")
barabasiGraph2 <- barabasi.game(40, directed = FALSE)
E(erGraph2)
```

```
## + 39/39 edges from 7333c47:
##  [1]  2-- 8  6--11  9--14 11--15 13--16 16--17 18--19 16--21 21--22 12--23
## [11] 14--24 15--24 23--26 10--27 13--27 24--28  5--29  4--30  9--31 12--31
## [21] 13--31  1--32  8--32 18--33  1--34 23--34 25--34  7--35 10--35 14--35
## [31] 19--35  5--36 11--36 24--36 34--36 16--40 25--40 31--40 36--40
```

```r
E(barabasiGraph2)
```

```
## + 39/39 edges from 73347e6:
##  [1]  1-- 2  1-- 3  1-- 4  1-- 5  3-- 6  2-- 7  5-- 8  1-- 9  3--10 10--11
## [11] 10--12 11--13  2--14  1--15 10--16  4--17 10--18  4--19  2--20 14--21
## [21]  1--22  8--23 10--24 11--25  6--26 10--27 26--28 10--29 25--30 27--31
## [31] 15--32 10--33  3--34 31--35  8--36  1--37 35--38 25--39  3--40
```

```r
# # Print the number of edges in each graph
cat("Erdos-Renyi graph 1 with", ecount(erGraph1), "edges\n")
```

```
## Erdos-Renyi graph 1 with 19 edges
```

```r
cat("Barabasi-Albert graph 1 with", ecount(barabasiGraph1), "edges\n")
```

```
## Barabasi-Albert graph 1 with 19 edges
```

```r
cat("Erdos-Renyi graph 2 with", ecount(erGraph2), "edges\n")
```

```
## Erdos-Renyi graph 2 with 39 edges
```

```r
cat("Barabasi-Albert graph 2 with", ecount(barabasiGraph2), "edges\n")
```

```
## Barabasi-Albert graph 2 with 39 edges
```

```r
#### Answer P2.1:


# List of randomly generated networks and their corresponding names
graphList <- list(erGraph1, barabasiGraph1, erGraph2, barabasiGraph2)
graphNames <- list("Erdos-Renyi Graph 1 (20 nodes)", "Barabasi-Albert Graph 1 (20 nodes)", "Erdos-Renyi

i <- 1
for(g in graphList){
  # Compute Laplacian Matrix for the randomly generated ER and BA graphs in previous section
  laplacian <- laplacian_matrix(g)
```

```r
  # Eigenvalues and Eigenvectors computation
  eig <- eigen(laplacian)
}
```

```r
i <- 1
for(g in graphList){
  n <- vcount(g) # No. of nodes
  m <- ecount(g) # No. of edges
  dmin <- min(degree(g)) # Minimum Degree
  dmax <- max(degree(g)) # Maximum Degree
  l <- average.path.length(g) #average path length
  D <- diameter(g) # Diameter
  ccg <- transitivity(g) # Global clustering coefficient

  laplacian <- laplacian_matrix(g)
  eig <- eigen(laplacian)
  lambdan <-max(eig$values) # Largest eigenvalue
  lambda2 <- sort(eig$values)[2] # 2nd smallest eigenvalue



  # Printing final values
  cat("------------------------------", toString(graphNames[i]), "------------------------------", "\n")
  cat("Number of nodes:", n, "\n")
  cat("Number of edges:", m, "\n")
  cat("Minimum degree:", dmin, "\n")
  cat("Maximum degree:", dmax, "\n")
  cat("Average path length:", l, "\n")
  cat("Diameter:", D, "\n")
  cat("Global clustering coefficient:", ccg, "\n")
  cat("Second smallest eigenvalue (algebraic connectivity):", lambda2, "\n")
  cat("Largest eigenvalue:", lambdan, "\n")
i=i+1
}
```

Problem 2.2:

```
## Warning: `average.path.length()` was deprecated in igraph 2.0.0.
## i Please use `mean_distance()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## ------------------------------ Erdos-Renyi Graph 1 (20 nodes) ------------------------------
## Number of nodes: 20
## Number of edges: 19
## Minimum degree: 0
## Maximum degree: 4
## Average path length: 3.089431
## Diameter: 7
## Global clustering coefficient: 0
## Second smallest eigenvalue (algebraic connectivity): 2.220734e-16
## Largest eigenvalue: 5.85901
## ------------------------------ Barabasi-Albert Graph 1 (20 nodes) ------------------------------
```

```
## Number of nodes: 20
## Number of edges: 19
## Minimum degree: 1
## Maximum degree: 5
## Average path length: 3.373684
## Diameter: 6
## Global clustering coefficient: 0
## Second smallest eigenvalue (algebraic connectivity): 0.1100541
## Largest eigenvalue: 6.491826
## ------------------------------ Erdos-Renyi Graph 2 (40 nodes) ------------------------------
## Number of nodes: 40
## Number of edges: 39
## Minimum degree: 0
## Maximum degree: 5
## Average path length: 4.228733
## Diameter: 11
## Global clustering coefficient: 0
## Second smallest eigenvalue (algebraic connectivity): -1.130594e-15
## Largest eigenvalue: 7.031744
## ------------------------------ Barabasi-Albert Graph 2 (40 nodes) ------------------------------
## Number of nodes: 40
## Number of edges: 39
## Minimum degree: 1
## Maximum degree: 9
## Average path length: 4.119231
## Diameter: 9
## Global clustering coefficient: 0
## Second smallest eigenvalue (algebraic connectivity): 0.04553323
## Largest eigenvalue: 10.17029
```

Observations: The Erdos-Renyi Graph 1, consisting of 20 nodes and 19 edges, reflects a random network structure. Nodes in this graph have degrees ranging from 0 to 4, resulting in a relatively short average path length of 3.09 but a longer diameter of 7. The absence of clustering (Global clustering coefficient: 0) suggests a lack of local connectivity. Additionaly, the second smallest eigenvalue being close to zero indicates minimal connectivity within the graph, highlighting its stochastic and less organized nature.

Contrastingly, Barabasi-Albert Graph 1, also with 20 nodes and 19 edges, demonstrates a scale-free network characterized by preferential attachment. Nodes in this graph have degrees ranging from 1 to 5, leading to a slightly longer average path length of 3.37 and a shorter diameter of 6. Despite the lack of clustering, the positive second smallest eigenvalue suggests improved connectivity compared to the Erdos-Renyi counterpart, emphasizing the presence of hubs and a more structured network.

Moving to Erdos-Renyi Graph 2, featuring 40 nodes and 39 edges, we observe a larger random network. The nodes' degrees range from 0 to 5, resulting in a longer average path length (4.23) and diameter (11) compared to the smaller Erdos-Renyi graph. Similar to Graph 1, the absence of clustering (Global clustering coefficient: 0) and an almost zero second smallest eigenvalue indicate low local connectivity and overall randomness in the network structure.

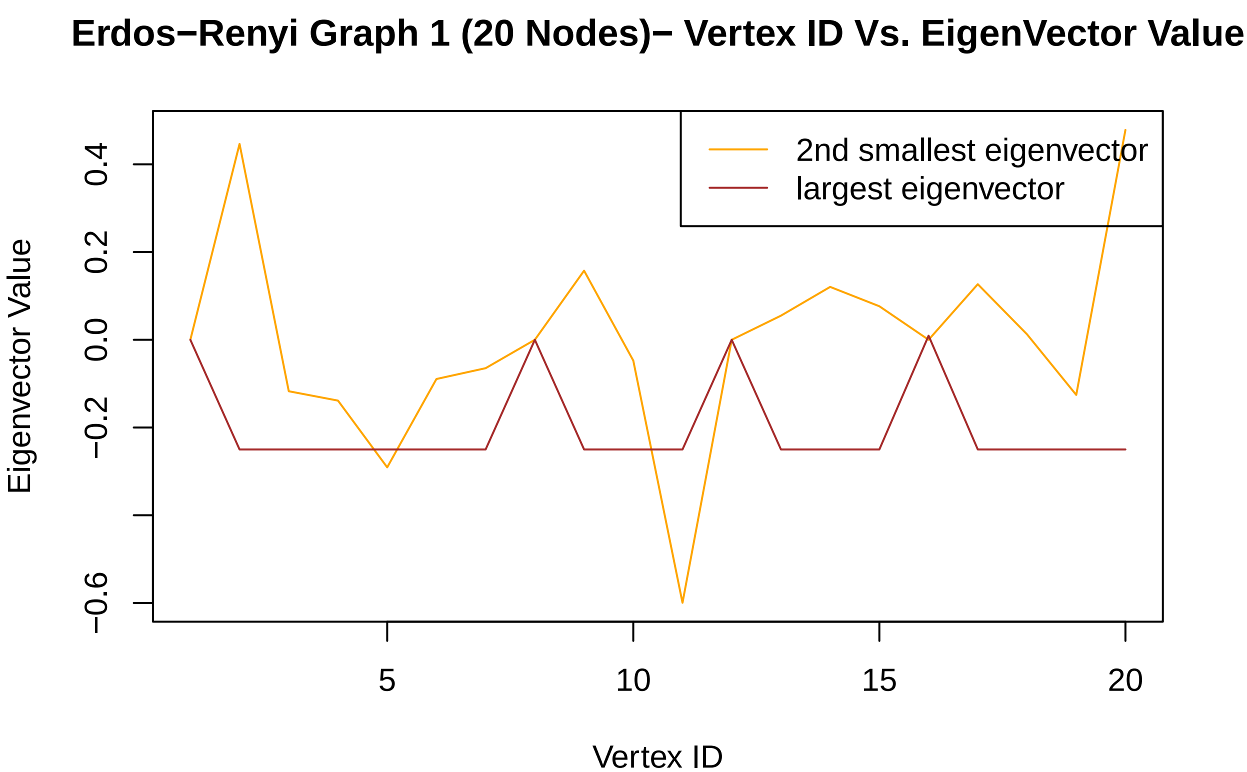
In contrast, Barabasi-Albert Graph 2, with 40 nodes and 39 edges, exhibits a scale-free network with nodes having degrees ranging from 1 to 9. Despite the absence of clustering (Global clustering coefficient: 0), this graph demonstrates a slightly shorter average path length (4.12) and diameter (9) compared to Erdos-Renyi Graph 2. The positive second smallest eigenvalue suggests enhanced connectivity, aligning with the preferential attachment mechanism and the formation of hubs in the scale-free network.

In summary, the Erdos-Renyi graphs display random characteristics with lower connectivity, while the Barabasi-Albert graphs showcase scale-free properties with improved overall connectivity and a more structured network topology. The choice between these models depends on the desired network characteristics for specific applications.

2.3 #Graphs #Code:

```
laplacian <- laplacian_matrix(erGraph1)
  eig <- eigen(laplacian)
  eigVector2 <- eig$vectors[, 2] # Second smallest eigenvector
  eigVectorn <- eig$vectors[, length(eig$values)] # Largest eigenvector
  # Let eigVector2 and eigVectorn are the eigen vectors corresponding to the

#second-smallest and largest eigenvalues respectively
  plot(eigVector2, type="l", col="orange",
       xlab="Vertex ID", ylab="Eigenvector Value",
       main="Erdos-Renyi Graph 1 (20 Nodes)- Vertex ID Vs. EigenVector Value")
  lines(eigVectorn, col="brown")
  legend("topright", legend=c("2nd smallest eigenvector",
                              "largest eigenvector"),
         col=c("orange", "brown"),lty=1)
```



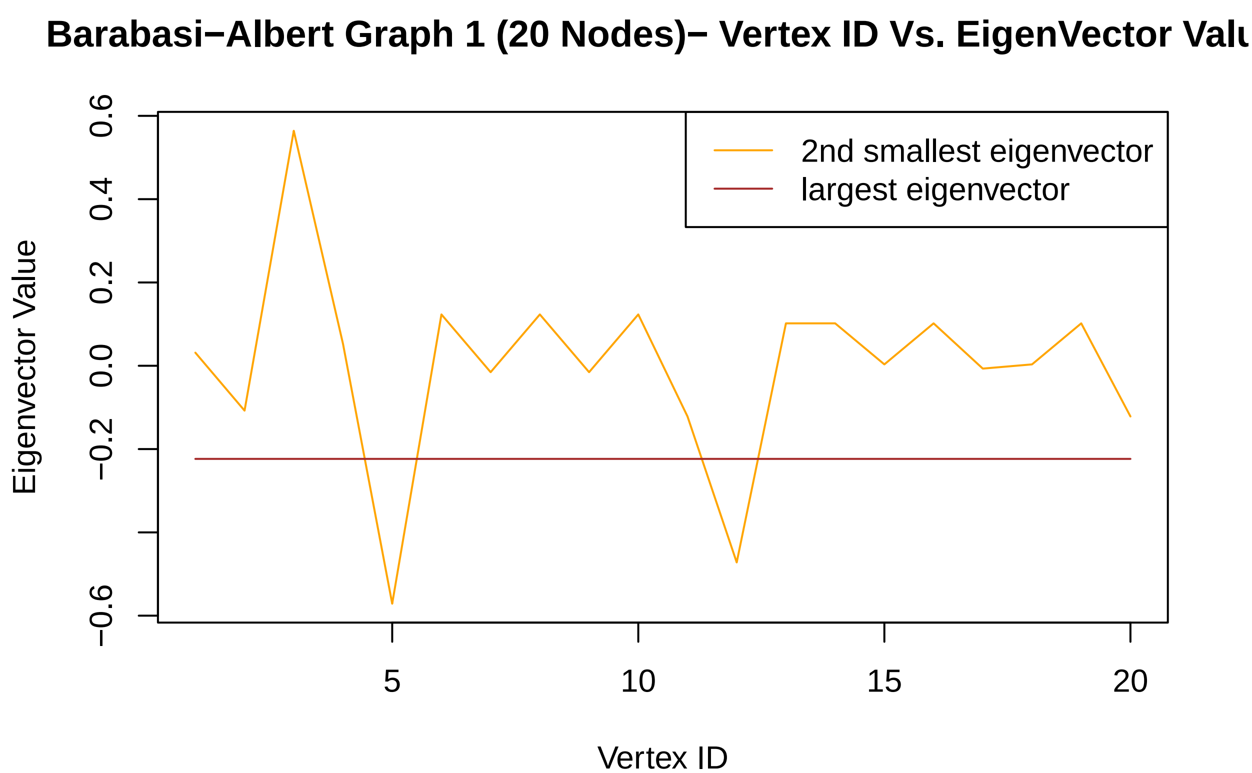
For Erdos-Renyi graph, The highest value of eigen vector is around the 5th and 15th node, and it decreases after that. It starts off at a high value and drastically decreases around node 10 and drastically increases at node 15. For the second minimum graph, it is comparatively varying less. Therefore, the largest eigen value is smaller than the second smallest eigen values.

```
  laplacian <- laplacian_matrix(barabasiGraph1)
  eig <- eigen(laplacian)
  eigVector2 <- eig$vectors[, 2] # Second smallest eigenvector
```

```r
eigVectorn <- eig$vectors[, length(eig$values)] # Largest eigenvector
# Let eigVector2 and eigVectorn are the eigen vectors corresponding to the
#second-smallest and largest eigenvalues respectively
plot(eigVector2, type="l", col="orange", xlab="Vertex ID", ylab="Eigenvector Value",

    main="Barabasi-Albert Graph 1 (20 Nodes)- Vertex ID Vs. EigenVector Value")
lines(eigVectorn, col="brown")
legend("topright", legend=c("2nd smallest eigenvector", "largest eigenvector"),

       col=c("orange", "brown"),lty=1)
```



For Barabasi albert graph, the highest value of eigen vector is linear through out. For the second minimum graph, it is comparatively varying less and does not reach the positive eigen vector values. Therefore, the largest eigen value is smaller than the second smallest eigen values.

Question 3 :

The intriguing applications of PageRank in chemistry, neuroscience, and literature are discussed in the article "Page Rank beyond the Web." 1)PageRank in Chemistry : It's great to see network science being applied in Chemistry. It's fascinating to create a graph of water molecules and hydrogen bonds, then use page rank to determine the degree to which a node is highly corelated. It's also interesting to retrieve the solute's position and confirm whether or not it has a hydrogen bond.

2) PageRank in Neuroscience: It's intriguing to use page rank to research connectome-related network features and brain areas. One effective technique to bridge two distinct topics is to relate network science to the study of neurons and brain networks.

3) PageRank in literature: Relevance is determined by applying the distance metric in books that employ

topic modeling concepts. In this manner, it assists readers in identifying recommended reading material and important works.