Web and Network Science Assignment 3

Sai Krishna Lakshminarayanan

1 March 2019

1. Introduction

The presence of hubs that are the nodes of high degree is considered to be a vital one in shortening average path length in the network. The assignment is done here to explore this hypothesis. For this purpose, d target and k target are considered. Here, d refers to the average distance and k reference to the average degree.

From the Koblenz network repository, two network data sets are considered satisfying the given condition of present between 1000 to 30000 nodes. The data sets are Blog Dataset and Open flight dataset.

2. Blog Dataset

The blog dataset is a directed network that contains front-page hyperlinks between blogs in the context of the 2004 United States elections. Here, the vertices represent the blog and the edges represent the hyperlinks between the blogs. There are total of 1,224 vertices and 19,025 edges present in the network.

```
#Loading all the required packages.
library(ggplot2)
library(gglorenz)
```

```
## Warning: package 'gglorenz' was built under R version 3.5.2
```

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.5.2
```

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

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

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

```
library(igraphdata)
```

```
## Warning: package 'igraphdata' was built under R version 3.5.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

2.1 Original World Network for Blog

The given data set is loaded into the R studio. There are some irrelevant columns present in the end and so the columns 3 and 4 are removed. Some junk values are present in the first row and it is also removed . Then, the columns are named are from and to denoting the vertices.

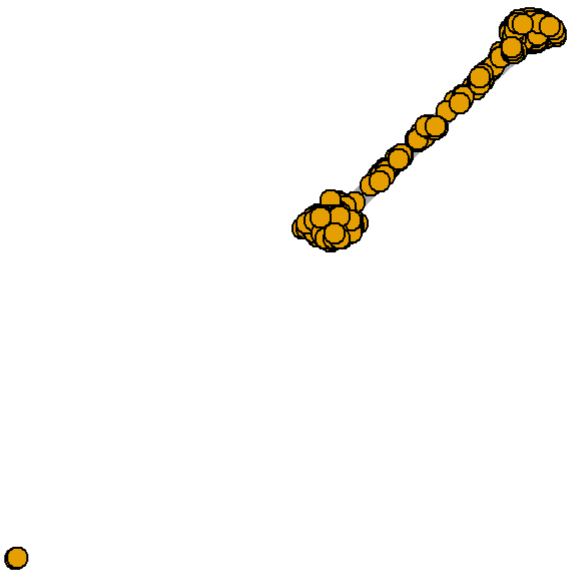
```
Blogs<- read.table("out.moreno_blogs_blogs.txt",sep='',fill = TRUE,row.names = NULL)#loading the
file
Blogs<-Blogs[,-4]#removing the irrelevant contents
Blogs<-Blogs[,-3]
Blogs<-Blogs[-1,]
colnames(Blogs)<-c('from','to')#providing the column name
```

Now, the Original world network model is considered. For this purpose, the data frame is converted into an undirected graph as said in the assignment question and this will be followed for all the coming models.

The graph is then simplified and the output is given out.

```
Bloggraph<-graph.data.frame(Blogs, directed=F)#generating the original networkgraph
Bloggraph<-simplify(Bloggraph, remove.multiple = TRUE, remove.loops = TRUE)#simplifying the grap
h
plot(Bloggraph,vertex.label= NA, edge.arrow.size=0.3,vertex.size = 8,main="Figure 1-Original Wor
ld Network for Blogs")#output
```

Figure 1-Original World Network for Blogs



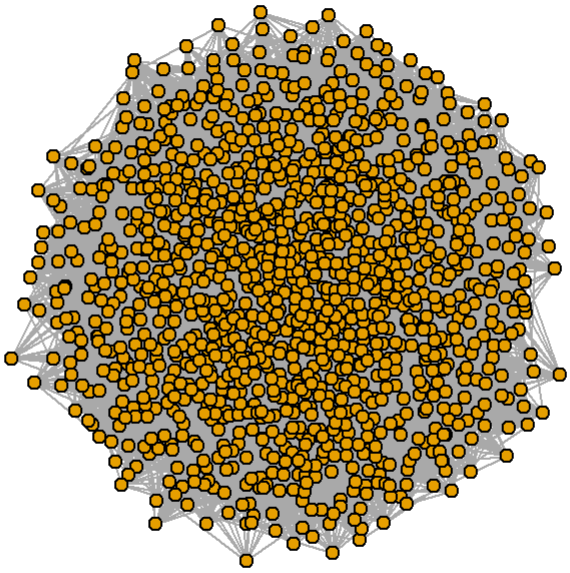
2.2 GNP Equivalent Model for Blog

In order to perform the GNP model, n and p values are needed. n is calculated as the total number of vertices present in the Blog network. k is calculated as the rounded value of the average degree of the Blog network. p is taken as the value of k/(n-1) which are obtained previously.

The GNP model is done by using the sample_gnp() function and the graph is simplified and the output is given.

```
k1 <- round(mean(degree(Bloggraph,loops = FALSE)))#calculating k as the rounded value of the mea
n degree
n1 <- vcount(Bloggraph)#count of the vertices present
p1 <- k1/((n1-1))#calculating p
gnp1 <- erdos.renyi.game(n1, p1, type = "gnp",directed = F)#generating gnp model
gnp1<-simplify(gnp1, remove.multiple = TRUE, remove.loops = TRUE)#simplifying the graph
plot(gnp1, vertex.label= NA, edge.arrow.size=0.05,vertex.size = 5,main = "Figure 2- GNP Equivale
nt Model for Blog")
```

Figure 2- GNP Equivalent Model for Blog



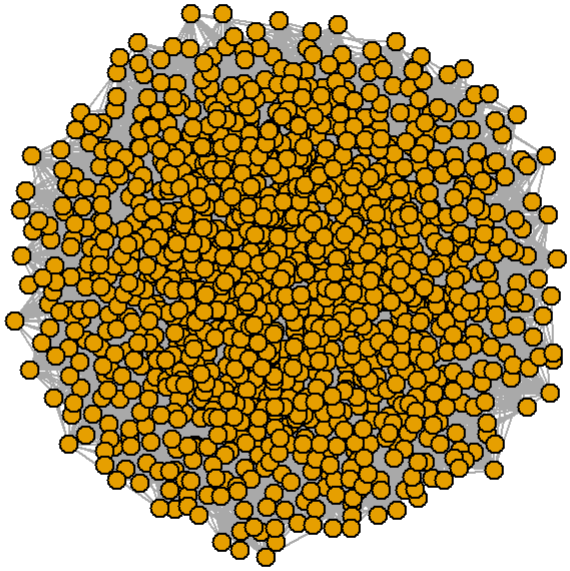
```
#output
```

2.3 Scale Free Equivalent Model for Blog

The scale free equivalent can be done by using the preferential attachment model.The preferential attachment model is obtained using the sample_pa() function. The power is given as 1 and is done based on the k and n values as calculated above. The model is simplified after generation and the result is given out.

```
sf1<- sample_pa(n1, power=1, k1, directed=F)#generating the preferential attachment model
sf1<-simplify(sf1, remove.multiple = TRUE, remove.loops = TRUE)#simplifiying the graph
plot(sf1, vertex.label= NA, edge.arrow.size=0.1,vertex.size =7,main = "Figure 3- Scale Free Equi
valent Model for Blog")
```

Figure 3- Scale Free Equivalent Model for Blog



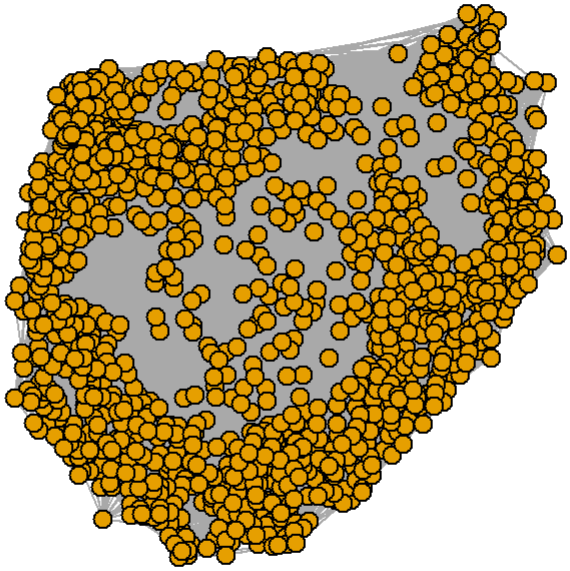
2.4 Small Network Equivalent for Blog

In the case of the small world equivalent model, beta value is required in calculation. Here, beta is calculated through means of clustering coefficient average and its c0 value with the corresponding formula of 3(k-2)/4(k-1). This is obtained from the formula of clustering coefficient =(1-beta)^3.

Now based on the values obtained for clustering coefficient and c0, the beta is then found as 1-(clustering coefficient/c0)^0.33. The sample_smallworld() function is used to obtain the small world model and it is simiplified before giving the result out.

```
clus.coeff1<-transitivity(Bloggraph,type="global")#obtaining the clustering coefficient of blog
 using transitivity and type global
c01<-(3*(k1-2))/(4*(k1-1))# getting c0 through the given formula in the lectures
beta1<-1-((clus.coeff1)/(c01))^0.33 #rearranging the formula given in the lectures to obtain the
beta
sw1<-sample_smallworld(dim=1,size=n1,nei = k1/2,p=beta1)#generating pa model
sw1<- simplify(sw1, remove.multiple = TRUE, remove.loops = TRUE)#simiplifying the graph
plot(sw1, vertex.label= NA, edge.arrow.size=0.1,vertex.size =7,main="Figure 4- Small Network Equ
ivalent Model for Blog")#output
```

Figure 4- Small Network Equivalent Model for Blog



2.5 k target vs d target for Blogs all 4 model

The k target and d target values are to be found for the original network,random equivalent, scale free equivalent and small world equivalent for both the datasets. Hence inorder to support reusability, a function can be built inorder to generate and store the values of k and d targets for each model respectively.

Now , inorder to perform this, the variables are created for storing k target and d target by their respective names. The inital value is assigned adn the range is providied to check the degree along its values.

Inorder to perform these, the from and to vertices are needed to be found . So, the from vertices is collected as all the vertices degree and mean degree are the same. Now, as given in the lecture, k value is found with the gap of 5 namely 5,10,15 and so on. So for these values, the to vertices are found are the ones which are greater than and lesser than the sum and difference of the ranges with the present k value.

Based on this from and to vertices, the d target values are found as the average distance present between the given from and to vertice. The values are then stored in a data frame with the column names as k target and d target respectively so that they can be called and used in the ggplot later.

```
k_d_targets <- function(x) {#initialising the funciton
  k_target <- vector(mode="numeric", length=0)
  #assiging k and d target variables
  d_target <- vector(mode="numeric", length=0)
  a <- 0#initialising the values of k and range for calculation
  r <- 5
  from <- which(degree(x,loops = FALSE) == round(mean(degree(x,loops = FALSE))))
  #obtaining the from vertices values as the total number of vertices where degree is equal to t
he mean degree.
  for(i in 1:25){# to perform the calculation for k values successively
    k_target <- c(k_target,a)#assigning the k values as theyre generated in each loop
    to = which(degree(x,loops = FALSE) > (a-r)&degree(x,loops = FALSE) < (a+r))
    #obtaining the to vertices as all the vertices with the degrees present between the k and ra
nge values
    d_target <- c(d_target,mean(distances(x, v=from, to=to)))
    #obtaining the d target in every loop as the average mean distance between the from and to v
ertices for the given k value in the loop
    a <- a + 5 # increasing the value of k in multiples of 5
  }
  k_d_values <- data.frame(k_target,d_target)#storing the values
  colnames(k_d_values) <- c("k_target","d_target")# assigning the column names
  return(k_d_values)#returning the dataset as the result of the k_d_targets function
}
```

Now, the ggplot is used to building a plot with all four lines with each 1 for 1 model values of the k target vs d target for the Blog dataset.

In this, the data is given as the result of the function of k_d_targets(). In this, x axis is the k target and y axis is the d target as given in the question. Therefore, giving x as k target and y as d target and each colour as one of the 4 models. In this way, the plot is obtained.

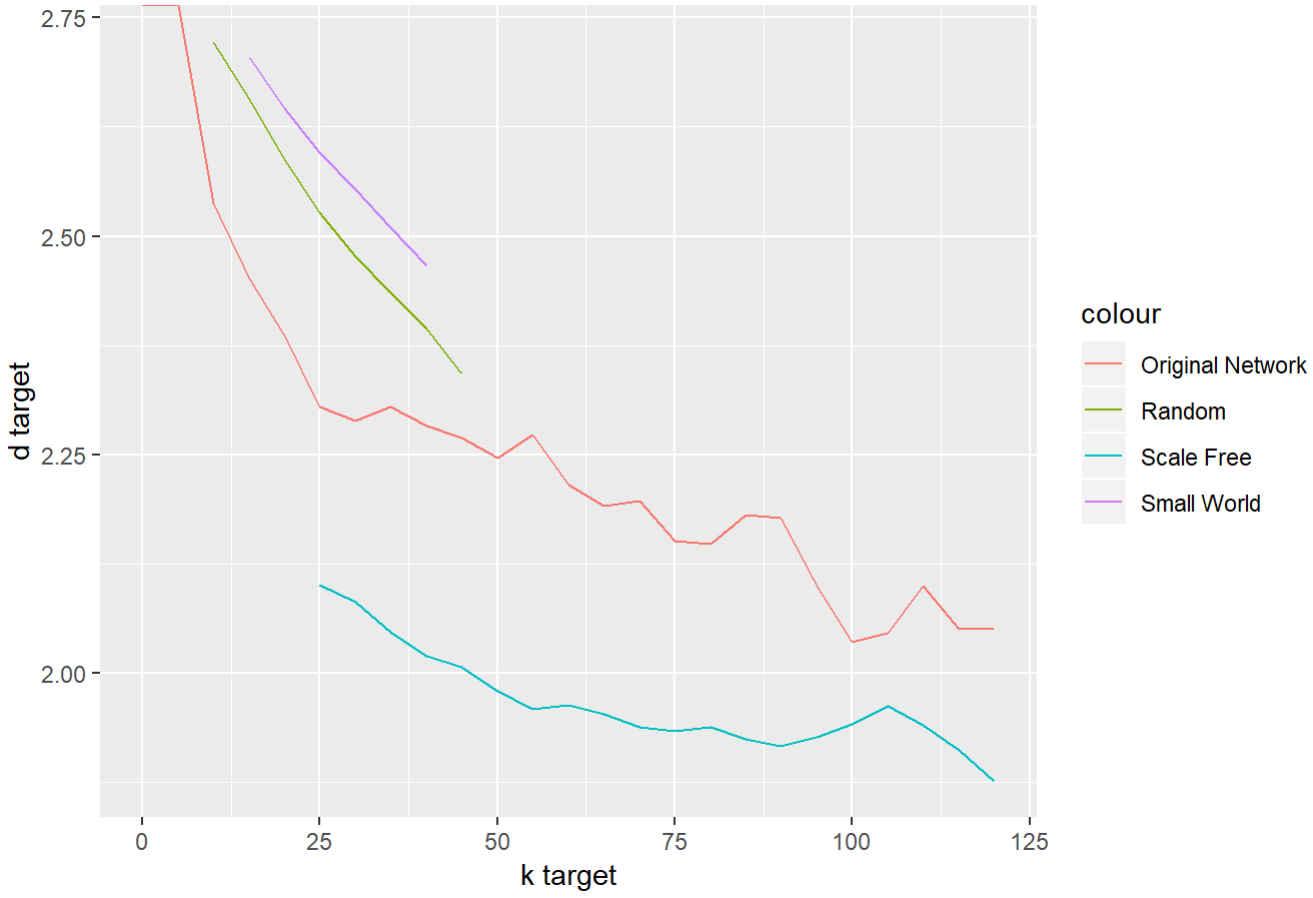
```
kd_blog<-k_d_targets(Bloggraph)#storing the k and d targets for all 4 models
kd_gnp1<-k_d_targets(gnp1)
kd_sf1<- k_d_targets(sf1)
kd_sw1<-k_d_targets(sw1)
Blogsplot<- ggplot() +#generating the plot
  geom_line(data =kd_blog , aes(x =k_target, y =d_target,colour = "Original Network"))+
  #plotting each line with different colour
  geom_line(data =kd_gnp1, aes(x = k_target, y = d_target, colour = "Random"))+
  geom_line(data =kd_sf1, aes(x = k_target, y = d_target, colour = "Scale Free"))+
  geom_line(data =kd_sw1 , aes(x = k_target, y = d_target,colour = "Small World"))+
  xlab("k target") +ylab("d target") +
  ggtitle(" Figure 5-Blogs k target vs d target for all 4")
Blogsplot
```

```
## Warning: Removed 17 rows containing missing values (geom_path).
```

```
## Warning: Removed 5 rows containing missing values (geom_path).
```

```
## Warning: Removed 19 rows containing missing values (geom_path).
```

Figure 5-Blogs k target vs d target for all 4



3. Open Flight Dataset

The open flight dataset is the datset set collected as part of OpenFlights.org project. In this, each node is an airport with the edges denoting the flight from one airport to another by one airline. In this, there are 3,425 vertices present and 67,663 edges present denoting the total number of airports and flights respectively.

There are some irrelevant columns present in the end and so the columns 3 is removed. Some junk values are present in the first row and it is also removed . Then, the columns are named are from and to denoting the vertices.

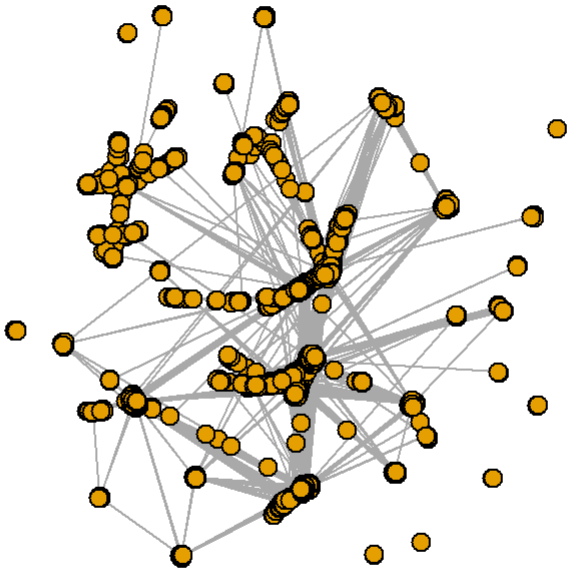
```
flights<- read.table("out.openflights.txt",sep='',fill = TRUE)#obtaining the file
flights<-flights[,-3]#cleaning the file
flights<-flights[-1,]
colnames(flights)<-c('from','to')#providing column names
flightgraph<-graph.data.frame(flights, directed=F)# generating graph
flightgraph<-simplify(flightgraph)
```

3.1 Original World Network for Open Flight

The original network model is obtained by using the graph.data.frame () function and as an undirected graph. It is then simplified and plotted with the edge and vertex sizes specified.

```
plot(flightgraph,vertex.label= NA, edge.arrow.size=0.1,vertex.size = 7,main="Figure 6-Original W
orld Network for Open Flight")#generating original network model
```

Figure 6-Original World Network for Open Flight



3.2 GNP Equivalent Model for Open Flight

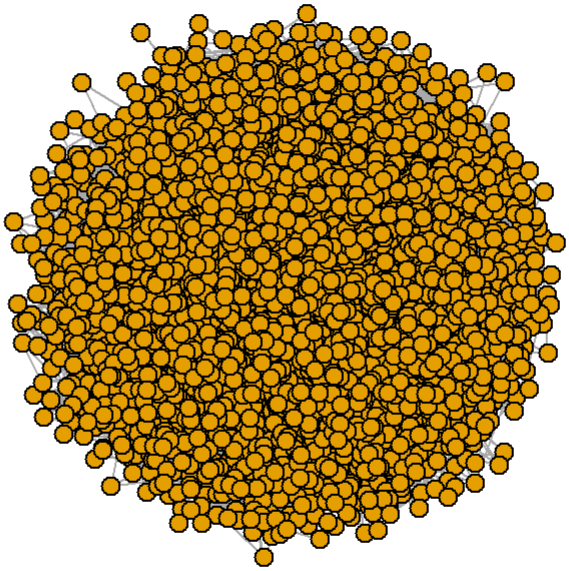
The same strategy is followed in open flight like in blog for getting the GNP equivalent model.

In order to perform the GNP model, n and p values are needed. n is calculated as the total number of vertices present in the Open Flight network. k is calculated as the rounded value of the average degree of the Open Flight network. p is taken as the value of k/(n-1) which are obtained previously.

The GNP model is done by using the sample_gnp() function and the graph is simplified and the output is given.

```
k <- round(mean(degree(flightgraph,loops = FALSE)))#calculating the average degree
n <- vcount(flightgraph)#total count of the vertices
p <- k/((n-1))#calculating p
gnp <- erdos.renyi.game(n, p, type = "gnp",directed = F)#generating the model
gnp<-simplify(gnp, remove.multiple = TRUE, remove.loops = TRUE)
plot(gnp, vertex.label= NA, edge.arrow.size=0.1,vertex.size = 7,main = "Figure 7- GNP Equivalent
Model for Open Flight")
```

Figure 7- GNP Equivalent Model for Open Flight



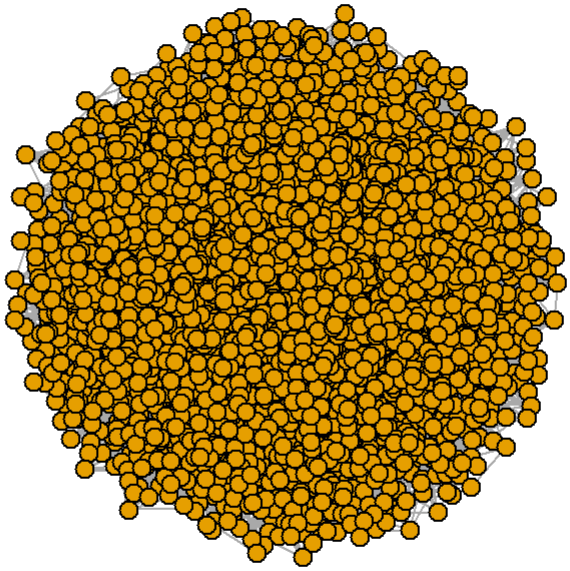
3.3 Scale Free Equivalent Model for Open Flight

The same approach is followed here in generating the scale free equivalent model like in blog.

The scale free equivalent can be done by using the preferential attachment model.The preferential attachment model is obtained using the sample_pa() function. The power is given as 1 and is done based on the k and n values as calculated above. The model is simplified after generation and the result is given out.

```
sf<- sample_pa(n, power=1, k, directed=F)#putting the pa model
sf<-simplify(sf, remove.multiple = TRUE, remove.loops = TRUE)
plot(sf, vertex.label= NA, edge.arrow.size=0.1,vertex.size =7,main = "Figure 8- Scale Free Equiv
alent Model for Open Flight")#output
```

Figure 8- Scale Free Equivalent Model for Open Flight



3.4 Small Network Equivalent for Open Flight

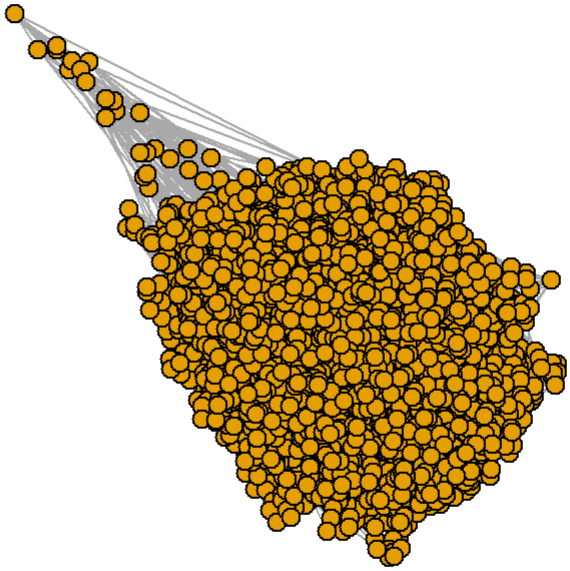
Similar to small network model generation in blog, the same process is done in open flights also for obtaining the model.

In the case of the small network equivalent model, beta value is required in calculation. Here, beta is calculated through means of clustering coefficient average and its c0 value with the corresponding formula of 3(k-2)/4(k-1). This is obtained from the formula of clustering coefficient =(1-beta)^3.

Now based on the values obtained for clustering coefficient and c0, the beta is then found as 1-(clustering coefficient/c0)^0.33. The sample_smallworld() function is used to obtain the small world model and it is simiplified before giving the result out.

```
clus.coeff<-transitivity(flightgraph,type="global")#obtaining the clustering coefficient
c0<-(3*(k-2))/(4*(k-1)) #getting c0
beta<-1-((clus.coeff)/(c0))^0.33 #calculating beta
sw<-sample_smallworld(dim=1,size=n,nei = k/2,p=beta) #generating small world model
sw<- simplify(sw, remove.multiple = TRUE, remove.loops = TRUE)
plot(sw, vertex.label= NA, edge.arrow.size=0.1,vertex.size =7,main="Figure 9- Small Network Equi
valent Model for Open Flight") #output
```

Figure 9- Small Network Equivalent Model for Open Flight



3.5 k target vs d target for open flight

Now, the ggplot is used to building a plot with all four lines with each 1 for 1 model values of the k target vs d target for the Blog dataset.

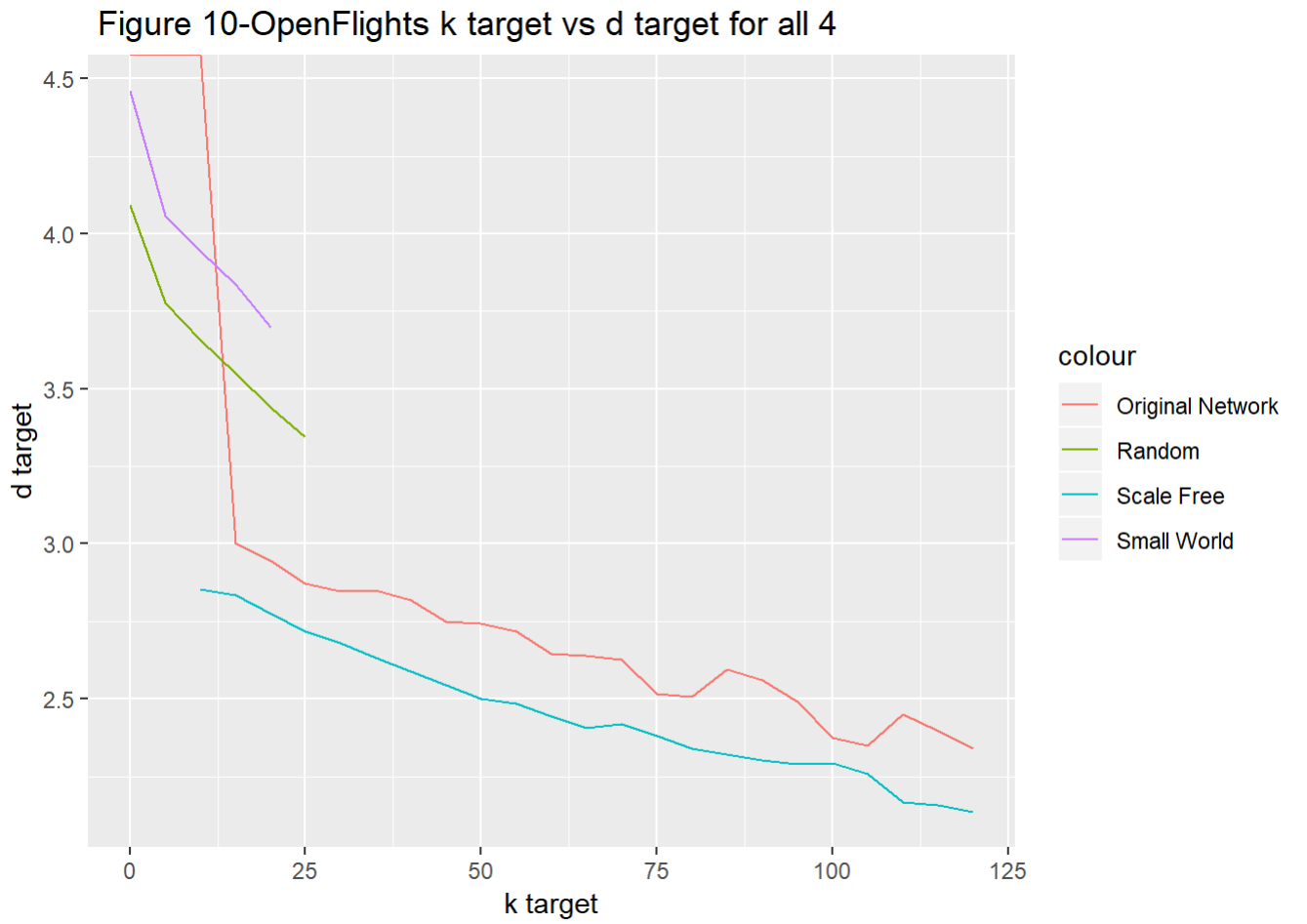
In this, the data is given as the result of the function of k_d_targets(). In this, x axis is the k target and y axis is the d target as given in the question. Therefore, giving x as k target and y as d target and each colour as one of the 4 models. In this way, the plot is obtained.

```
kd_flight<-k_d_targets(flightgraph)#storing the k and d target values for all the 4 models
kd_gnp<-k_d_targets(gnp)
kd_sf<- k_d_targets(sf)
kd_sw<-k_d_targets(sw)
flightplot<- ggplot() +#generating the plot
  geom_line(data =kd_flight , aes(x =k_target, y =d_target,colour = "Original Network"))+
  #giving the x and y details and colours for each line respectively
  geom_line(data =kd_gnp, aes(x = k_target, y = d_target, colour = "Random"))+
  geom_line(data =kd_sf, aes(x = k_target, y = d_target, colour = "Scale Free"))+
  geom_line(data =kd_sw , aes(x = k_target, y = d_target,colour = "Small World"))+
  xlab("k target") +ylab("d target") +
  ggtitle(" Figure 10-OpenFlights k target vs d target for all 4")
flightplot
```

```
## Warning: Removed 19 rows containing missing values (geom_path).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

```
## Warning: Removed 20 rows containing missing values (geom_path).
```



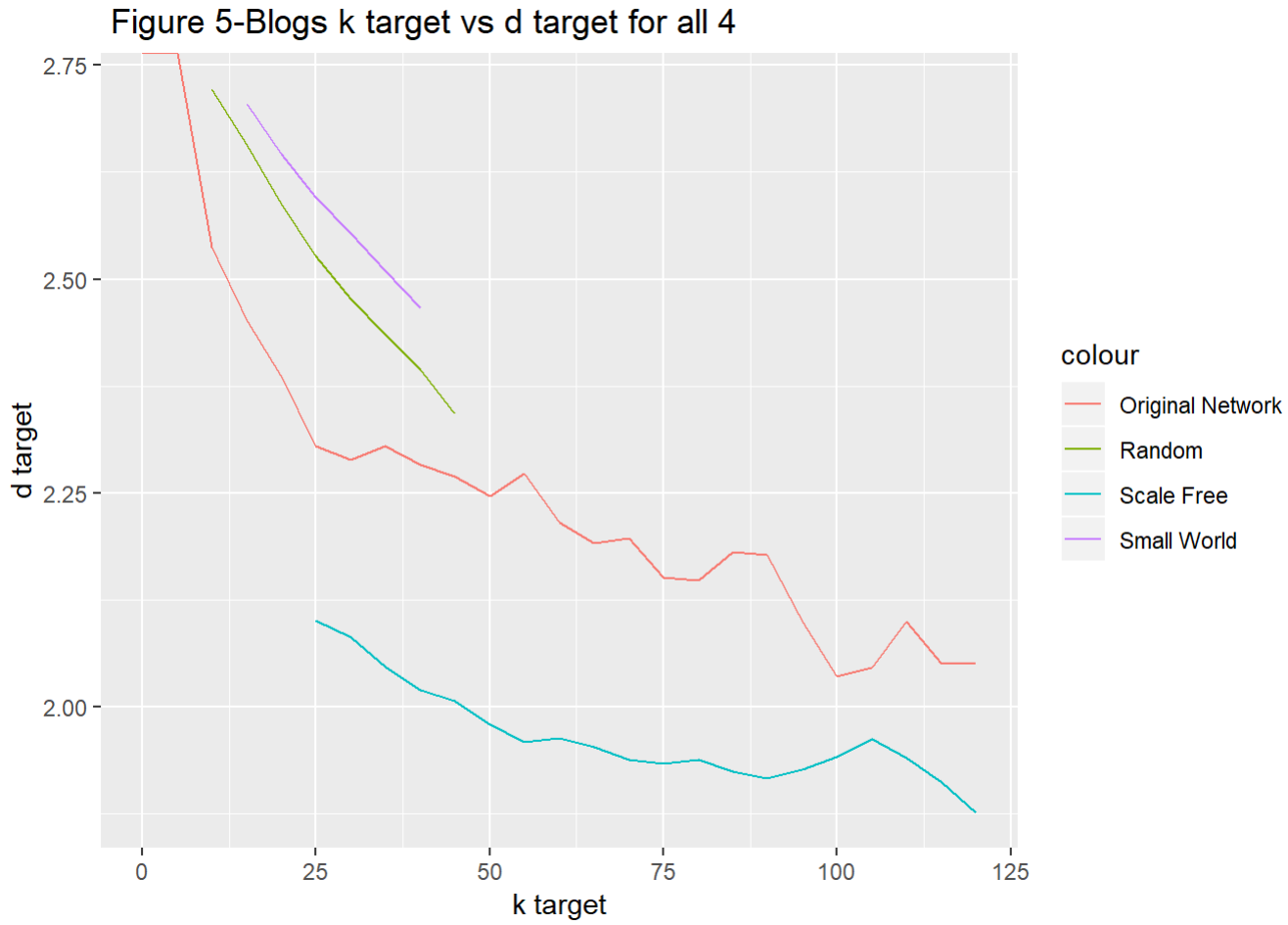
4.Conclusion

```
Blogsplot
```

```
## Warning: Removed 17 rows containing missing values (geom_path).
```

```
## Warning: Removed 5 rows containing missing values (geom_path).
```

```
## Warning: Removed 19 rows containing missing values (geom_path).
```



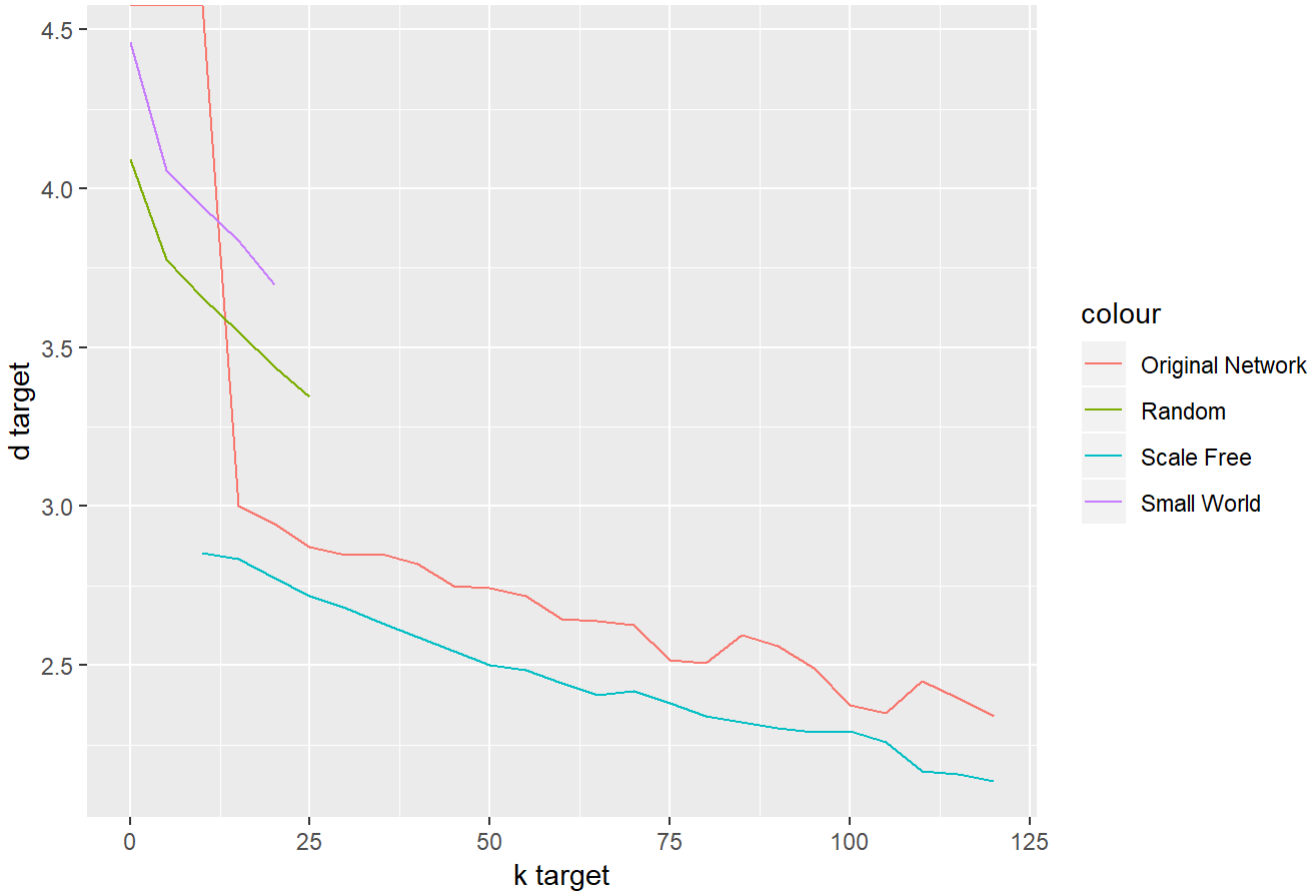

```
flightplot
```

```
## Warning: Removed 19 rows containing missing values (geom_path).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

```
## Warning: Removed 20 rows containing missing values (geom_path).
```

Figure 10-OpenFlights k target vs d target for all 4



1. It is observed that lines denoting the random and short world model are small and the lines denoting Real network and scale free are long. There are also several NA values present for the d target values to the corresponding k values in the models.
2. The random network is seen to be following poisson distribution in both the dataset plots.
3. The scale free network is seen to be following power law distribution in both the dataset plots.
4. The original network has low degree saturation and high degree cut off in both the dataset plots.
5. The scale free networks in both the plots are nearer to the hubs than other ones and there are vertices of high degree present in them.
6. Scale free networks have the lowest average path length among all the models in both the plots and therefore they are the shortest among all the models in both the plots.
7. Therefore, this confirms the hypothesis that the presence of hubs and nodes of high degree has played a significant role in shortening the average path length of the scale free networks in both the datasets compared to other models.

5. References

1. https://www.rdocumentation.org/packages/Rcrawler/versions/0.1.5 (https://www.rdocumentation.org/packages/Rcrawler/versions/0.1.5)
2. https://igraph.org/r/doc/simplify.html (https://igraph.org/r/doc/simplify.html)
3. http://cneurocvs.rmki.kfki.hu/igraph/doc/R/graph.data.frame.html (http://cneurocvs.rmki.kfki.hu/igraph/doc/R/graph.data.frame.html)

4. https://cran.r-project.org/web/packages/Rcrawler/Rcrawler.pdf (https://cran.r-project.org/web/packages/Rcrawler/Rcrawler.pdf)
5. https://en.wikipedia.org/wiki/Distance_(graph_theory) (https://en.wikipedia.org/wiki/Distance_(graph_theory))
6. http://igraph.org/r/doc/erdos.renyi.game.html (http://igraph.org/r/doc/erdos.renyi.game.html)
7. https://stackoverflow.com/questions/33706378/how-do-i-find-the-number-of-vertices-(https://stackoverflow.com/questions/33706378/how-do-i-find-the-number-of-vertices-) in-a-graph-created-by-igraph-in-python
8. https://stackoverflow.com/questions/8345759/how-to-save-a-data-frame-in-r (https://stackoverflow.com/questions/8345759/how-to-save-a-data-frame-in-r)
9. Week 3 to 5 Lecture Notes and Tutorials in Blackboard