Web and Network Science Assignment 2

Sai Krishna Lakshminarayanan

19 February 2019

1.Introduction

The Networks DBS and LYIT considered in Assignment 1 can further be enhanced by using the network models. The network models can be random models like the GNP and others like Small World and Preferential Attachment are to be tried to compare and contrast the properties of the original network to the model.

2. Original Networks

From the previous assignment dataset, further depth was taken in the sites inorder to get more nodes for processing. This is stored as a csv which is then called up here for modelling.

2.1 DBS Network

Initally, all the required packages are loaded and the initial network graph is drawn using the csv file containing the nodes and the degrees.This is done using the igraph package.

```
#Loading the required packages
library(Rcrawler)
```

```
## Warning: package 'Rcrawler' 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(rvest)
```

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

```
## Loading required package: xml2
```

```r
library(igraphdata)
```

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

```r
library(reldist)
```

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

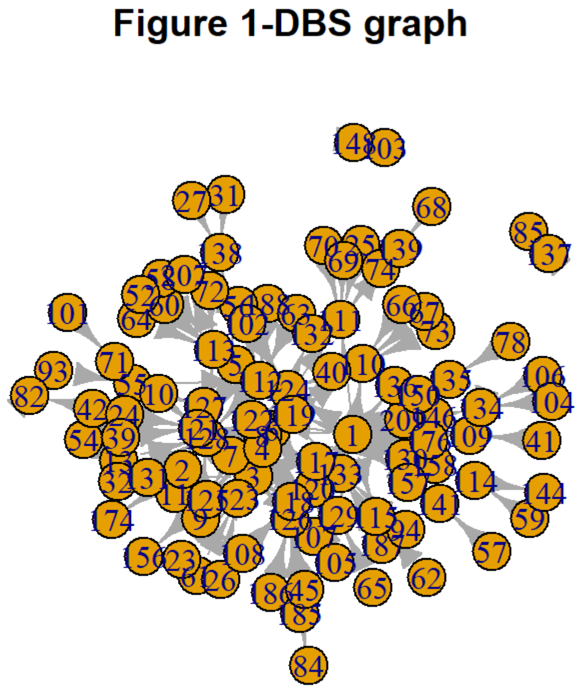
```
## reldist: Relative Distribution Methods
## Version 1.6-6 created on 2016-10-07.
## copyright (c) 2003, Mark S. Handcock, University of California-Los Angeles
##  For citation information, type citation("reldist").
##  Type help(package="reldist") to get started.
```

```r
library(gglorenz)
```

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

```
## Loading required package: ggplot2
```

```r
dbs<-read.csv("dbs.csv")#Network graph of DBS
dbsIndex<-data.frame(IN=dbs$IN,OUT=dbs$OUT)
dbsgraph <- graph.data.frame(dbsIndex, directed=T)
dbsgraph<-simplify(dbsgraph, remove.multiple = TRUE, remove.loops = TRUE,
                     edge.attr.comb = igraph_opt("edge.attr.comb"))
plot(dbsgraph,main="Figure 1-DBS graph")
```



Once that is done, then the properties of the network is analysed by obtaining its average path length, diameter and clustering coefficient.

```
cat("The average path length  of dbs is",average.path.length(dbsgraph))
```

```
## The average path length  of dbs is 1.726592
```

```
cat("\nThe diameter of dbs is",diameter(dbsgraph))
```

```
##
## The diameter of dbs is 6
```

```
cat("\nThe average clustering coefficient of dbs is",transitivity(dbsgraph, type="average"))
```

```
##
## The average clustering coefficient of dbs is 0.02250152
```

```
cat("\nThe global clustering coefficient of dbs is",transitivity(dbsgraph, type="global"))
```

```
##
## The global clustering coefficient of dbs is 0.0170778
```

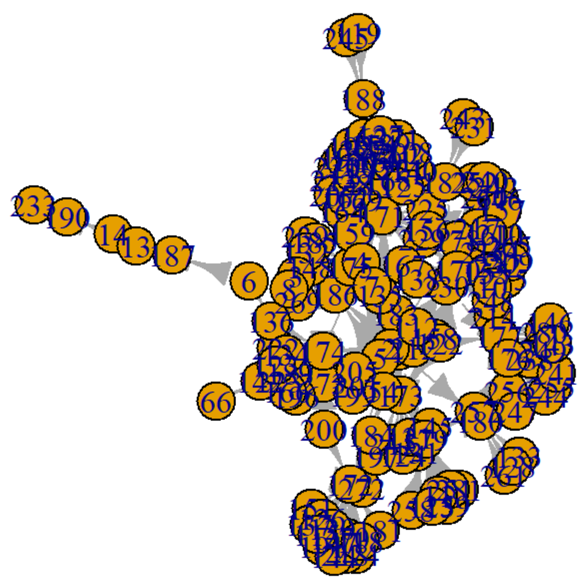
2.2LYIT Graph

The same steps are repeated again for the second network namely LYIT Network.

```
lyit<-read.csv("lyit.csv")
lyitIndex<-data.frame(IN=lyit$IN,OUT=lyit$OUT)
lyitgraph <- graph.data.frame(lyitIndex, directed=T)

lyitgraph<-simplify(lyitgraph, remove.multiple = TRUE, remove.loops = TRUE,
                   edge.attr.comb = igraph_opt("edge.attr.comb"))
plot(lyitgraph,main="Figure 2-LYIT Graph")
```

Figure 2-LYIT Graph



The properties of LYIT are also analysed like DBS.

```
cat("The average path length  of LYIT is",average.path.length(lyitgraph))
```

```
## The average path length  of LYIT is 1.334507
```

```
cat("\nThe diameter of LYIT is",diameter(lyitgraph))
```

```
##
## The diameter of LYIT is 3
```

```
cat("\nThe average clustering coefficient of LYIT is",transitivity(lyitgraph, type="average"))
```

```
##
## The average clustering coefficient of LYIT is 0.1409049
```

```
cat("\nThe global clustering coefficient of LYIT is",transitivity(lyitgraph, type="global"))
```

```
##
## The global clustering coefficient of LYIT is 0.03736921
```

3. Degree Distribution for the Networks

Now, the degree distribution of both the networks are studied. ###3.1 DBS Degree Distribution

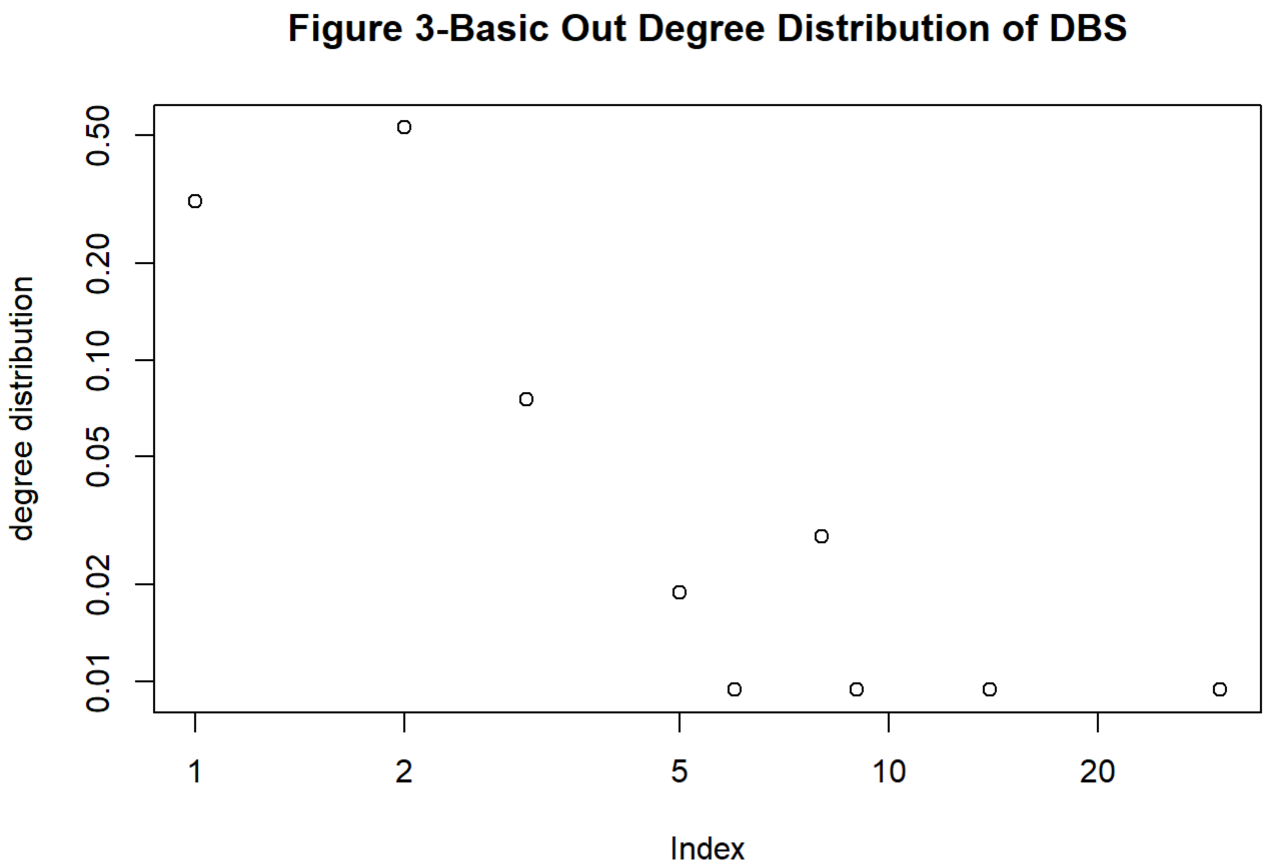
Initially, the degree distribution is obtained for the DBS network. It is done separately for both In and Out degree using the mode in the degree.distribution function. It is seen that the low degrees tend to have alot of probability occuring.

```
g.power.degree <-degree(dbsgraph, mode="out", loops=FALSE)
sort(g.power.degree, decreasing=TRUE)
```

```
##   1   3   6   8   5   4   2   9   7 185  63  54  11  10  42  93 102  85
##  29  13   8   7   7   7   5   4   4   2   2   2   2   2   2   2   2   1
## 188 187 186  64  62  61  60  59  58  57  56  55  13  52 133  32  31 128
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 125 124  94  27  26  25  24 115 114 113 112 111 110 109 108 107 106 105
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 104 103  23 101  78  74  73  72  71  70  69  68  67  66  65  45  41  40
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  39 137  84 126 120 135 146 209 130 141 136 117 158 122 134 150 176 123
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 131 127 138  82 121 119 156 132 144 207 139 118 148 174 129 157
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

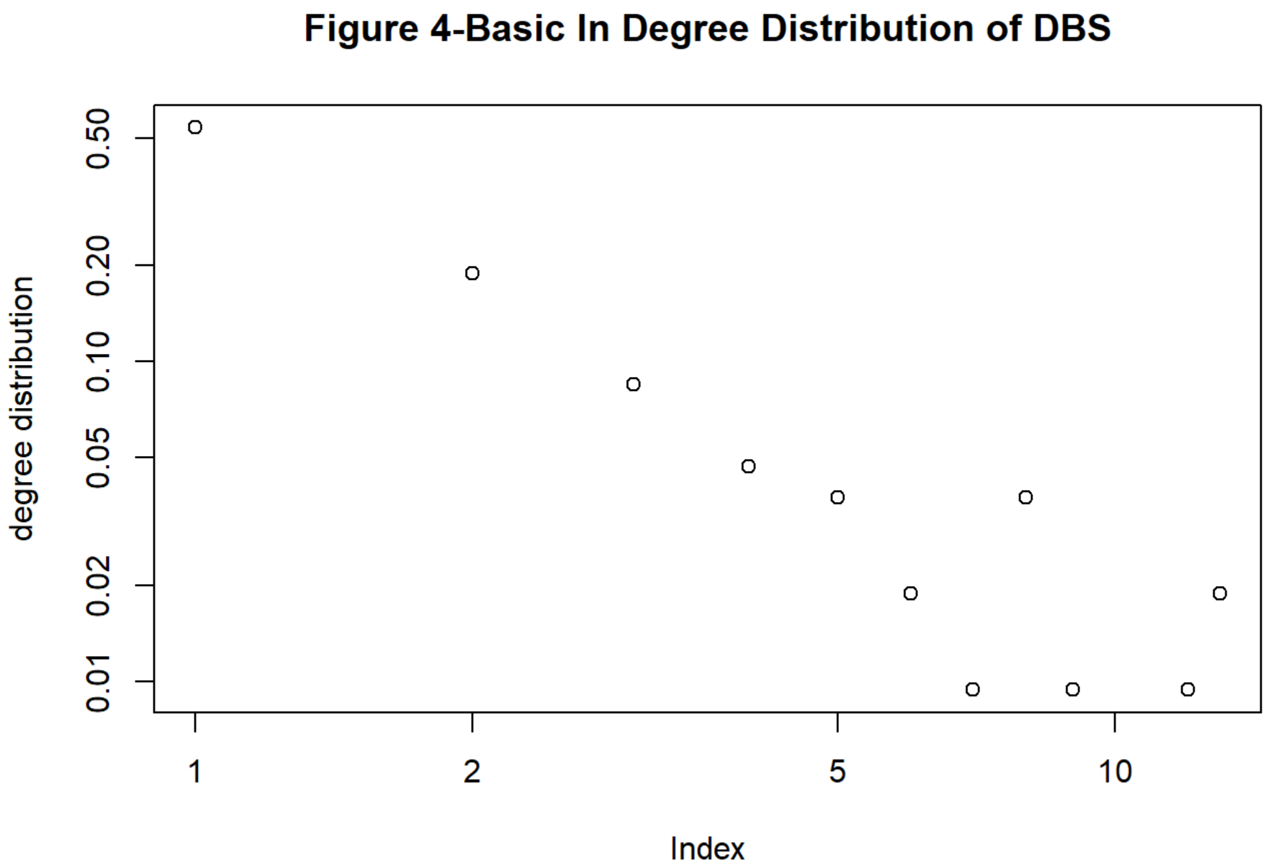
```
g.power.degree.distribution1 <-degree_distribution(dbsgraph, cumulative = FALSE,mode="out")
plot(g.power.degree.distribution1, log="xy",main = "Figure 3-Basic Out Degree Distribution of DB
S",ylab  ="degree distribution")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 21 y values <= 0 omitted
## from logarithmic plot
```



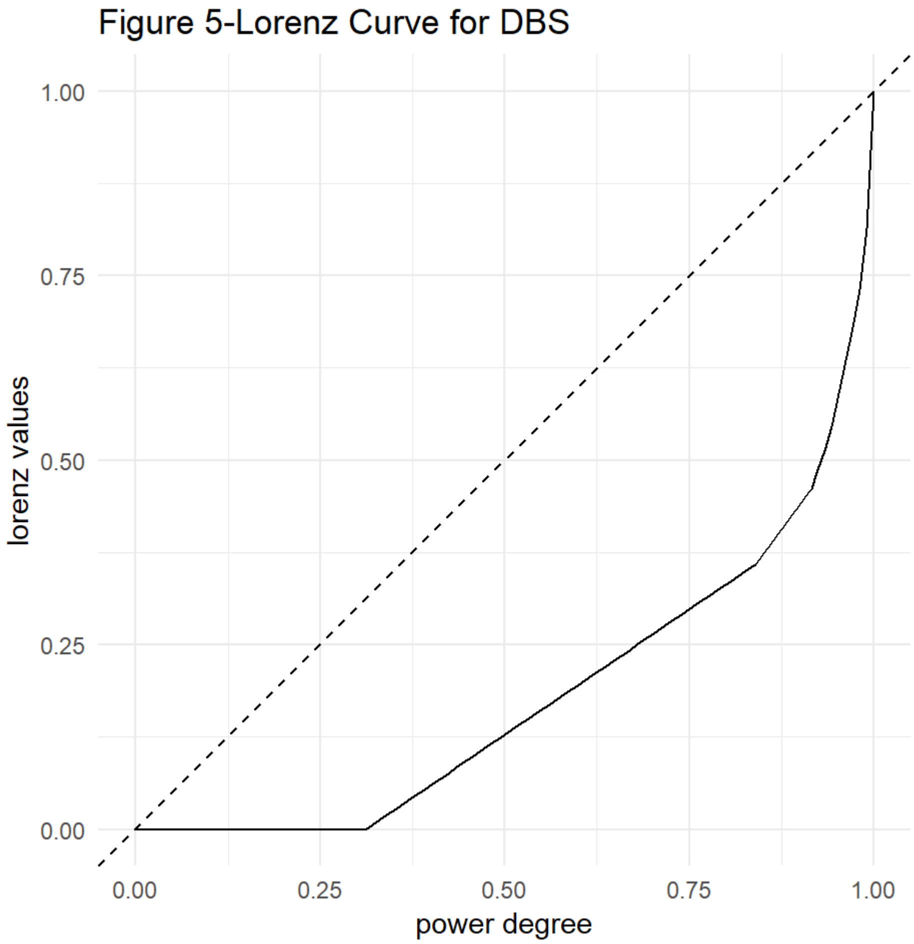

```
g.power.degree.distribution <-degree_distribution(dbsgraph, cumulative = FALSE,mode="in")
plot(g.power.degree.distribution, log="xy",main = "Figure 4-Basic In Degree Distribution of DBS"
,ylab  ="degree distribution")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 2 y values <= 0 omitted
## from logarithmic plot
```



The lorenz curve is then developed inorder to check the inequality in the degree of distribution in the DBS Network. It is seen that there occurs some inequality at the bottom.

```
ggplot(as.data.frame(g.power.degree), aes(x=g.power.degree)) + stat_lorenz() + coord_fixed() + g
eom_abline(linetype="dashed") + theme_minimal()+
xlab("power degree")+
ylab("lorenz values")+
ggtitle("Figure 5-Lorenz Curve for DBS")
```



3.2 LYIT Degree Distribution

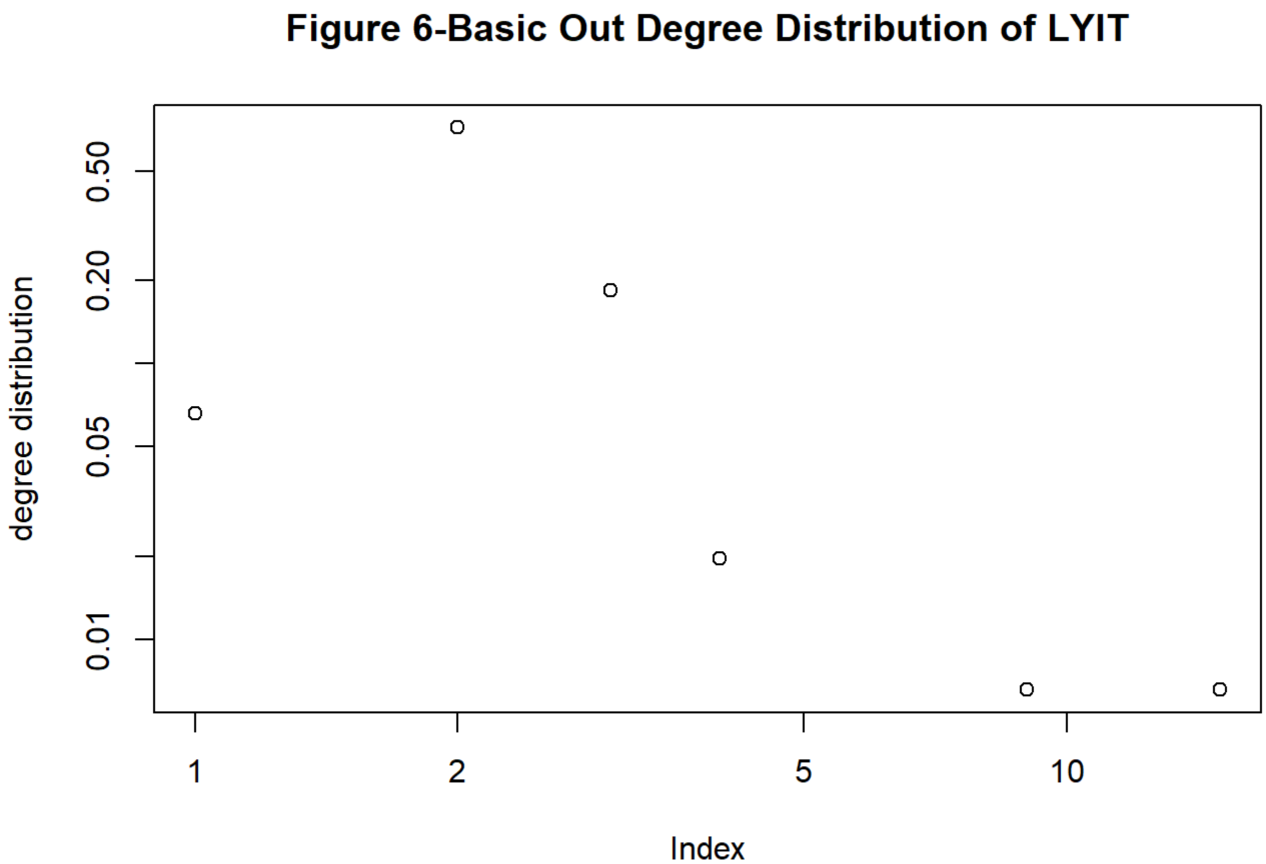
The same process is repeated again for the second network namely LYIT.Here, the In degrees are higher in proportion to the out degrees and are having high probability for low degrees.

```
g.power.degree <-degree(lyitgraph, mode="out", loops=FALSE)
sort(g.power.degree, decreasing=TRUE)
```

```
##   1   2   4   6   5 214 273 272 258 257 256 248 247 238 230 229 225 201
##  14   8   3   3   3   2   2   2   2   2   2   2   2   2   2   2   2   2
## 200 199 197 183 182 178 167 159 158 156 122  10   7   3  14 281 280 271
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   1   1
## 270 275 269  66 265 261 260 259 255 252 251 244 249 246 245 243 242 241
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 240 239 231 234 233 232 228 227 220 224 223 222 221 213 218 217 216 215
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 207 212 211 210 209 202 198 196 189 188 193 192 191 184 181 186 185 175
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 174 177 170 169 168 166 164 163 162 160 157 161 154 151 155 148 147 146
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 127 126 125 124 123 121 120 119 118 115  11 108 113 112  13   8 173 171
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   0   0
## 176 179 172 205 180 190 187  54
##   0   0   0   0   0   0   0   0
```

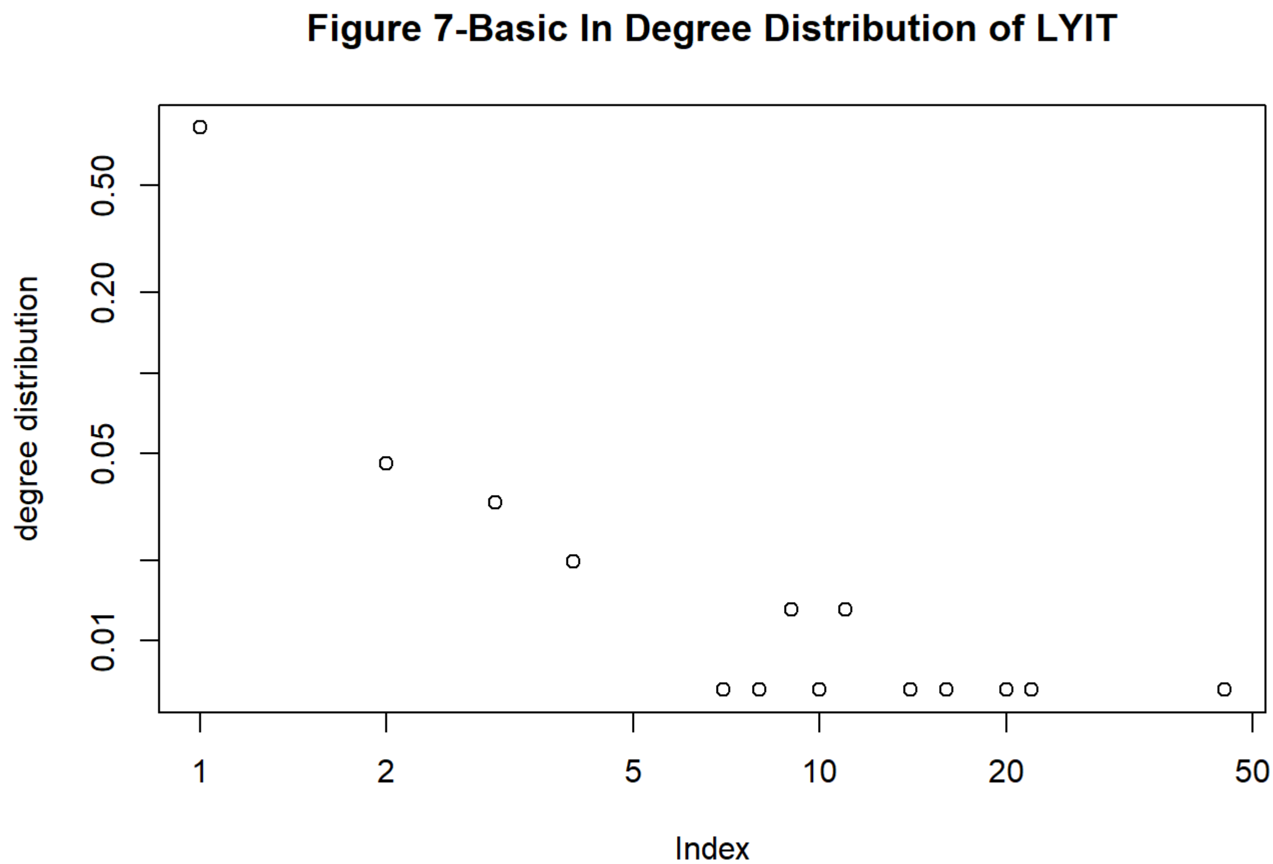
```
g.power.degree.distribution1 <-degree_distribution(lyitgraph, cumulative = FALSE,mode="out")
plot(g.power.degree.distribution1, log="xy",main = "Figure 6-Basic Out Degree Distribution of LY
IT",ylab  ="degree distribution")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 9 y values <= 0 omitted
## from logarithmic plot
```



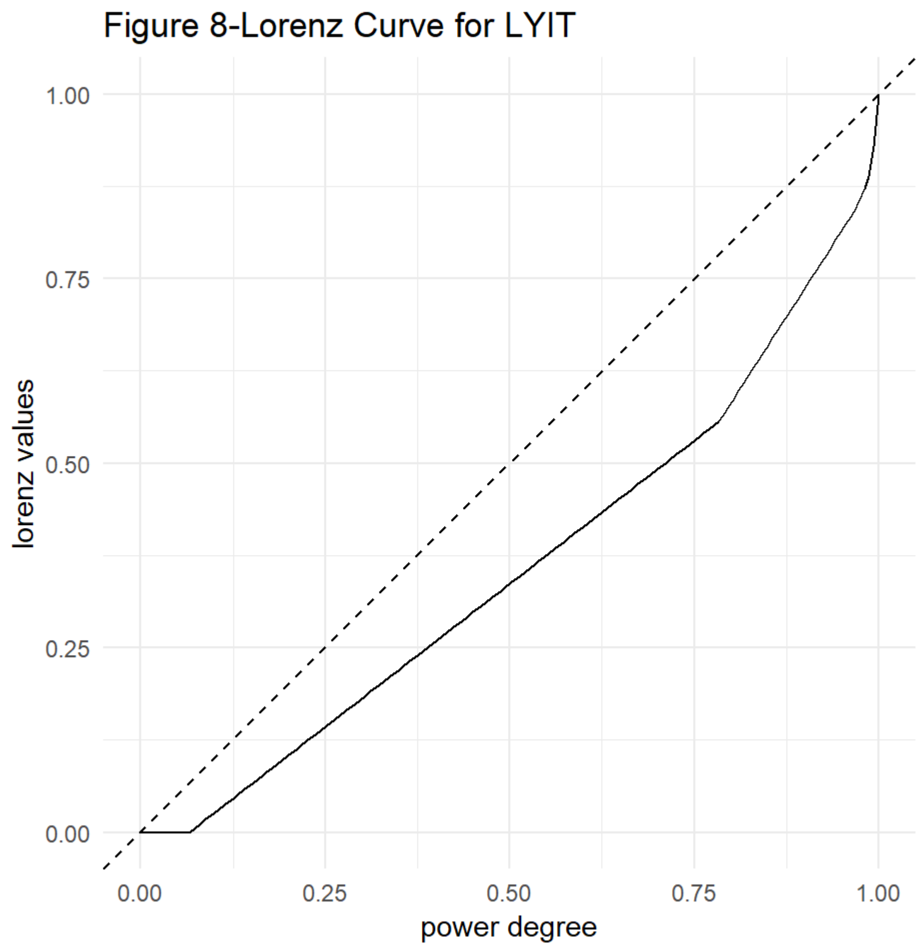

```
g.power.degree.distribution <-degree_distribution(lyitgraph, cumulative = FALSE,mode="in")
plot(g.power.degree.distribution, log="xy",main = "Figure 7-Basic In Degree Distribution of LYI
T",ylab  ="degree distribution")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 31 y values <= 0 omitted
## from logarithmic plot
```



The lorenz curve for LYIT is comparitively better than DBS which shows that it has less inequality in distribution.

```
ggplot(as.data.frame(g.power.degree), aes(x=g.power.degree)) + stat_lorenz() + coord_fixed() + g
eom_abline(linetype="dashed") + theme_minimal()+
xlab("power degree")+
ylab("lorenz values")+
ggtitle("Figure 8-Lorenz Curve for LYIT")
```



4. GNP Model

4.1 GNP Model for DBS

4.1.1 GNP Model Graph for DBS

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 dbs network. k is calculated as the rounded value of the average degree of the dbs network. p is taken as the value of k/(n-1) which are obtained previously.

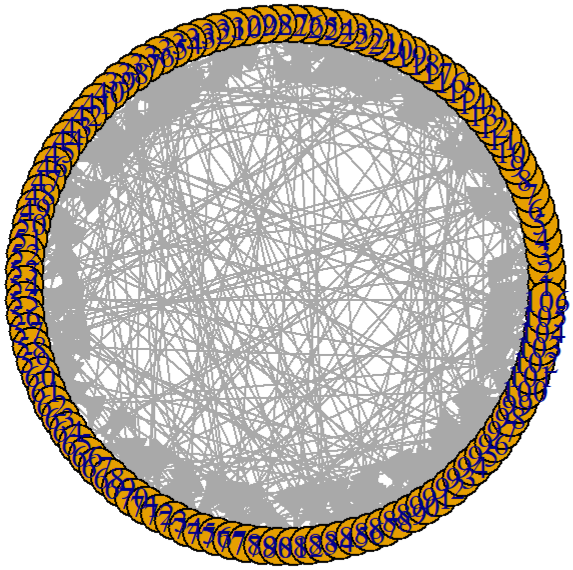
```
k<-round(mean(degree(dbsgraph)))
n<-vcount(dbsgraph)

p<-k/(n-1)
```

The GNP model is done by using the sample_gnp() function. The direction is given as true and it is seen that there are alot of connections and wirings between the nodes that are present.

```
gnp <-sample_gnp(n, p, directed = TRUE, loops = FALSE)
plot(gnp, layout=layout.circle(gnp), main="Figure 9- GNP Model for DBS ")
```

Figure 9- GNP Model for DBS



4.1.2 GNP Model based values for DBS

Now, the average path length,diameter,average and clustering coefficent for the model are obtained. It is seen that the values are not near to the original network values of the DBS Network.

```
cat("The average path length of GNP Model of DBS is",average.path.length(gnp))
```

```
## The average path length of GNP Model of DBS is 4.177469
```

```
cat("\nThe diameter of GNP Model of DBS is",diameter(gnp))
```

```
##
## The diameter of GNP Model of DBS is 11
```

```
cat("\nThe average clustering coefficient of GNP Model of DBS is",transitivity(gnp, type="averag
e"))
```

```
##
## The average clustering coefficient of GNP Model of DBS is 0.04755446
```

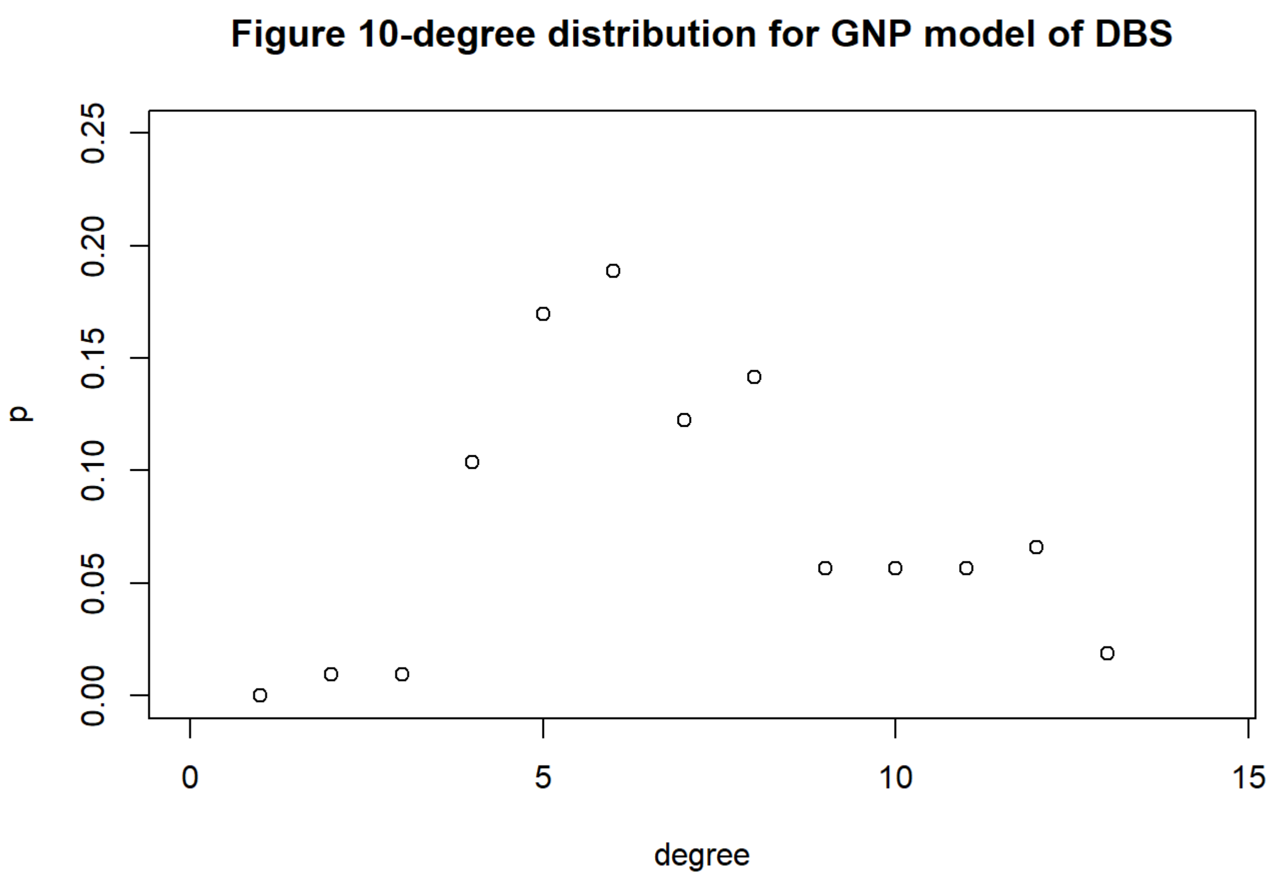
```
cat("\nThe global clustering coefficient of GNP Model of DBS is",transitivity(gnp, type="global"
))
```

```
##
## The global clustering coefficient of GNP Model of DBS is 0.0559194
```

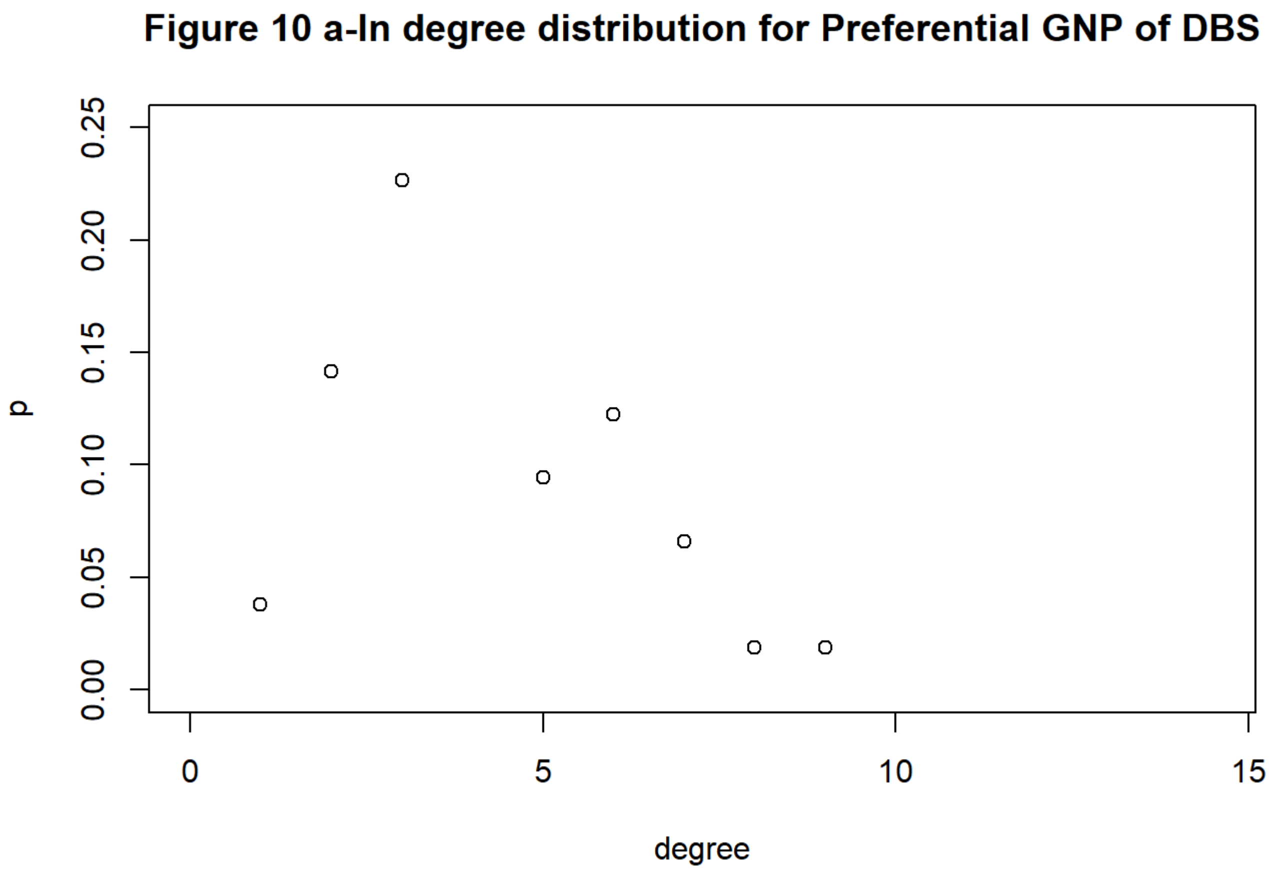
4.1.3 Degree Distribution for GNP Model of DBS

The degree distribution for each in and out degree in the model is obtained. It is seen that there is no difference between the two and the complete degree distribution. It is seen that the middle ranged degree values are having higher probability when compared to the low degree of the original network.It is also seen that it satisfies the poisson distribution and can be normalised so if needed which is a property of GNP.

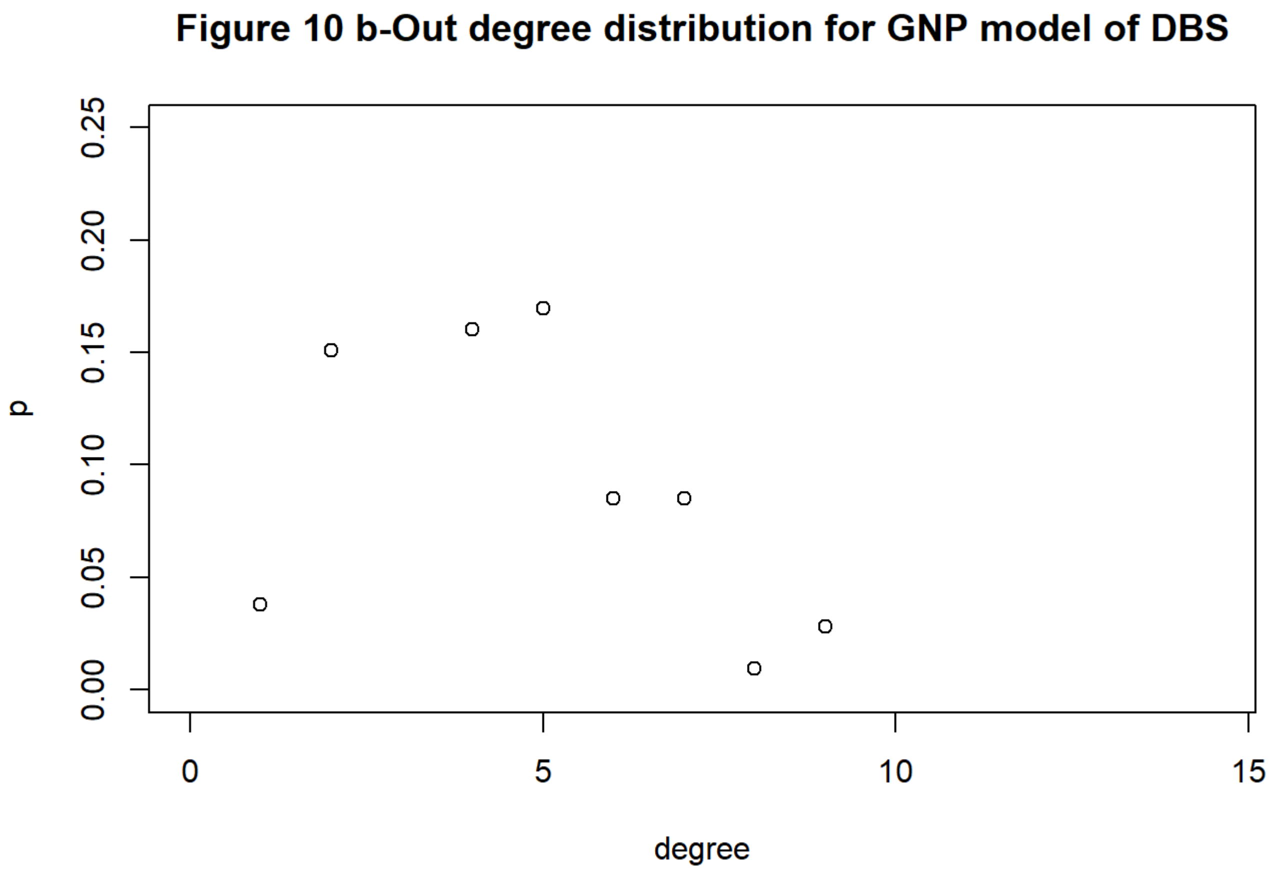
```
#degree distribution
gnp.possible.max_degree <- n-1
gnp.actual.max_degree <- max(degree(gnp))
degree_to_plot = gnp.actual.max_degree + sd(degree(gnp))
#plot actual degree distribution
plot(degree_distribution(gnp), xlim = c(0,degree_to_plot),ylim=c(0,0.25),xlab="degree", ylab="p"
, main="Figure 10-degree distribution for GNP model of DBS")
```




```
plot(degree.distribution(gnp,mode="in"), xlim = c(0,degree_to_plot),ylim=c(0,0.25),xlab="degree"
, ylab="p", main="Figure 10 a-In degree distribution for Preferential GNP of DBS")
```




```
plot(degree.distribution(gnp,mode="out"), xlim = c(0,degree_to_plot),ylim=c(0,0.25),xlab="degre
e", ylab="p", main="Figure 10 b-Out degree distribution for GNP model of DBS")
```



4.2 GNP Model for LYIT

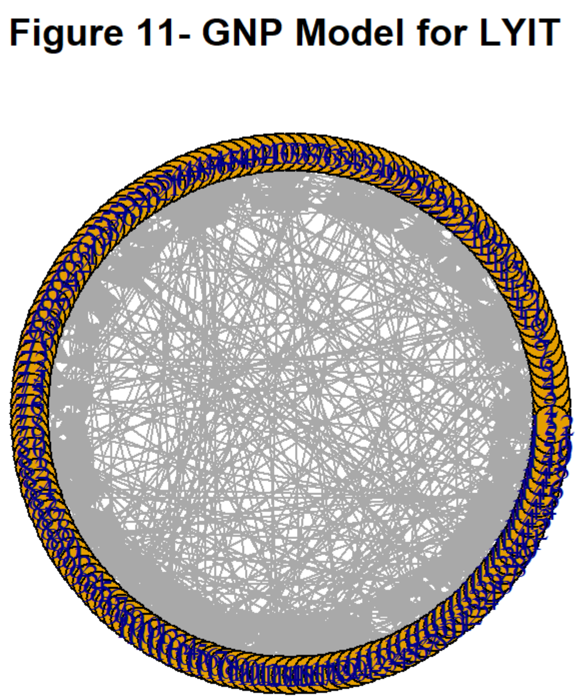
4.2.1 GNP Model Graph for LYIT

The same process is followed for obtaining the k ,n and p values for the LYIT network.

```
k1<-round(mean(degree(lyitgraph)))
n1<-vcount(lyitgraph)
p1<-k1/(n1-1)
```

From the graph, it is seen that the model is more densely mapped when compared to the GNP model for DBS.

```
gnp1 <-sample_gnp(n1, p1, directed = TRUE, loops = FALSE)
plot(gnp1, layout=layout.circle(gnp1), main="Figure 11- GNP Model for LYIT ")
```



4.2.2 GNP Model based values for LYIT

The average path length,diameter ,clustering coefficient of LYIT are obtained. It is seen that it is higher than the LYIT network. The average and global clustering coefficient are very near to each other in this model.

```
cat("The average path length of GNP Model of LYIT is",average.path.length(gnp1))
```

```
## The average path length of GNP Model of LYIT is 4.887505
```

```
cat("\nThe diameter of GNP Model of LYIT is",diameter(gnp1))
```

```
##
## The diameter of GNP Model of LYIT is 12
```

```
cat("\nThe average clustering coefficient of GNP Model of LYIT is",transitivity(gnp1, type="aver
age"))
```

```
##
## The average clustering coefficient of GNP Model of LYIT is 0.03490747
```

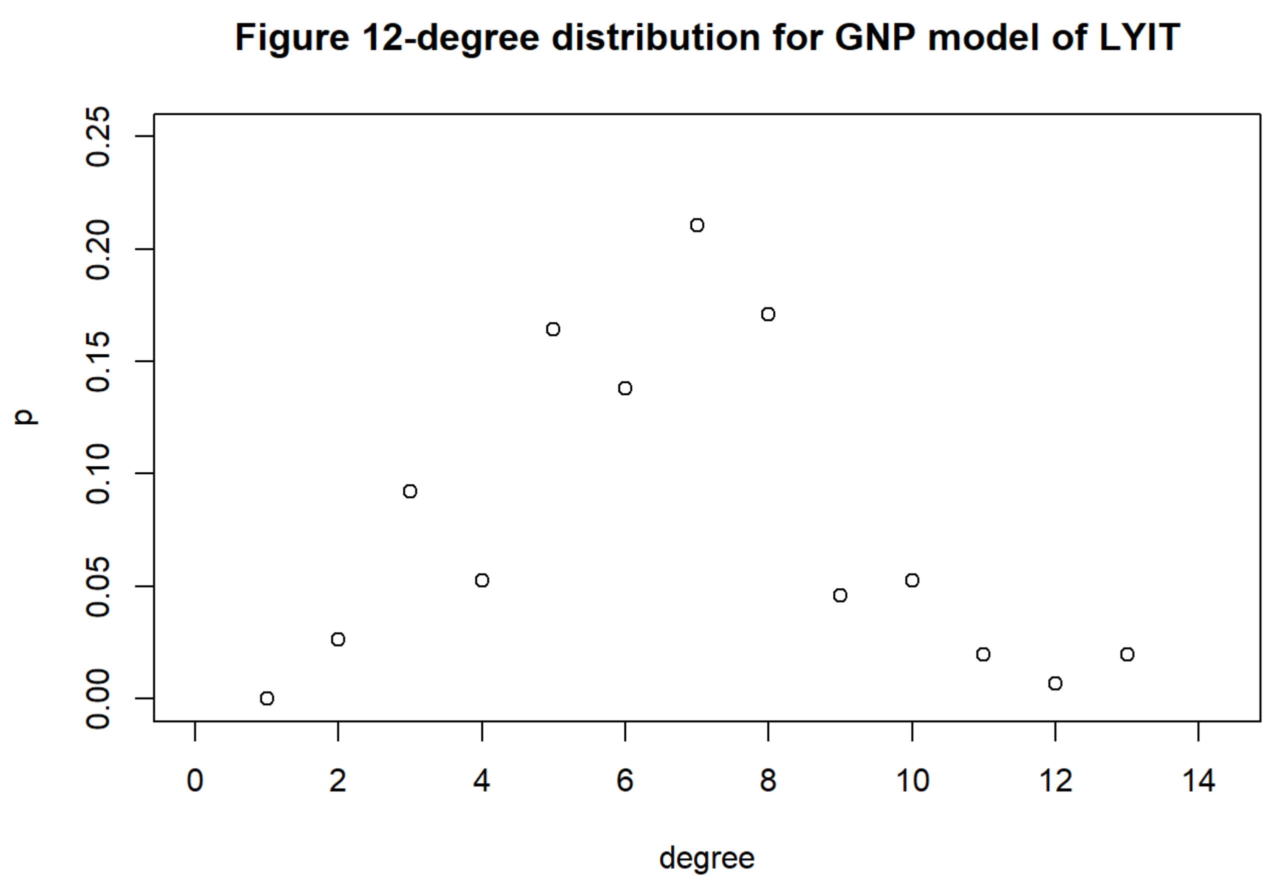
```
cat("\nThe global clustering coefficient of GNP Model of LYIT is",transitivity(gnp1, type="globa
l"))
```

```
##
## The global clustering coefficient of GNP Model of LYIT is 0.04039497
```

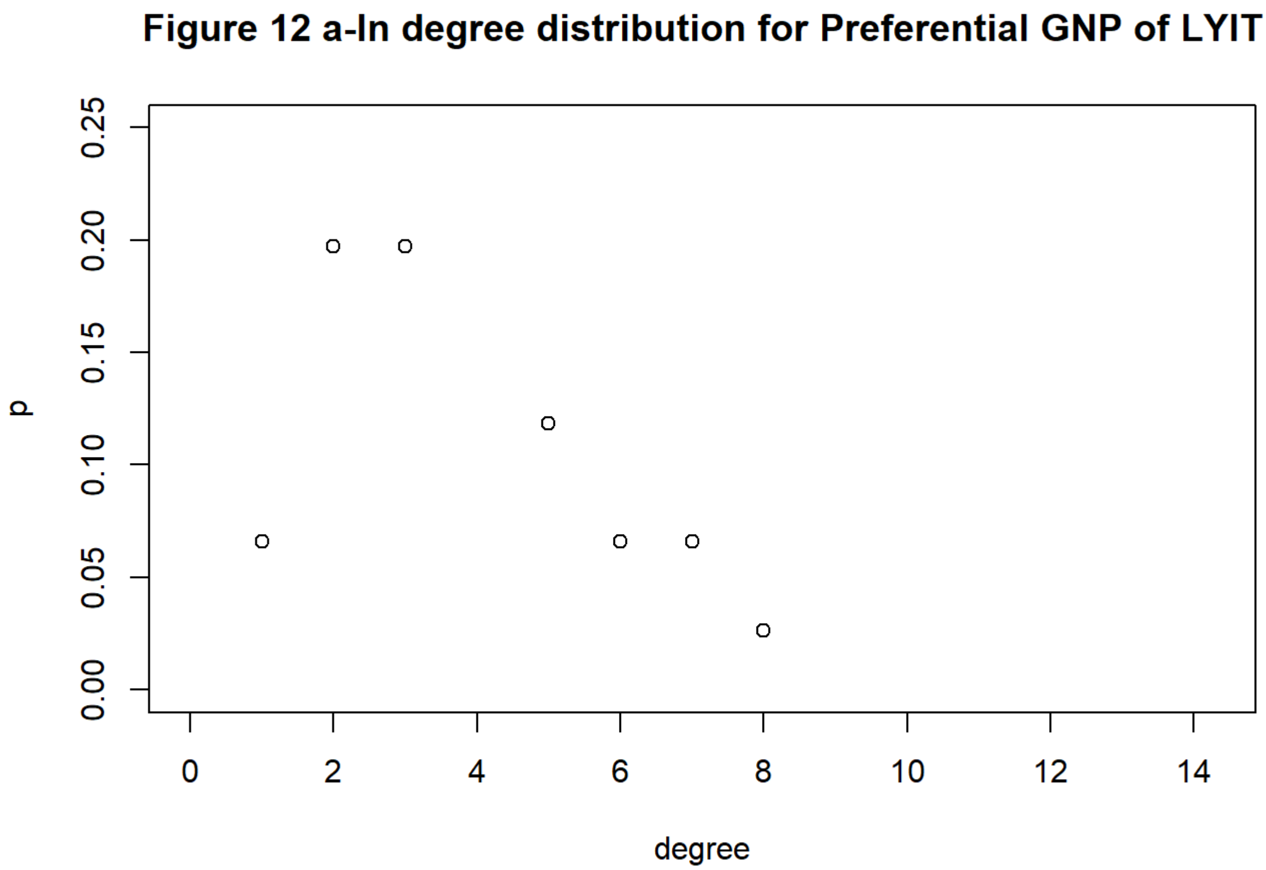
4.2.3 Degree distribution for GNP model of LYIT

Like in the case of DBS, the in and out degree distribution plots are similar to each other. It is having maximum probability when the degree is 4 whereas it is 1 and 2 degrees for the original LYIT Network.

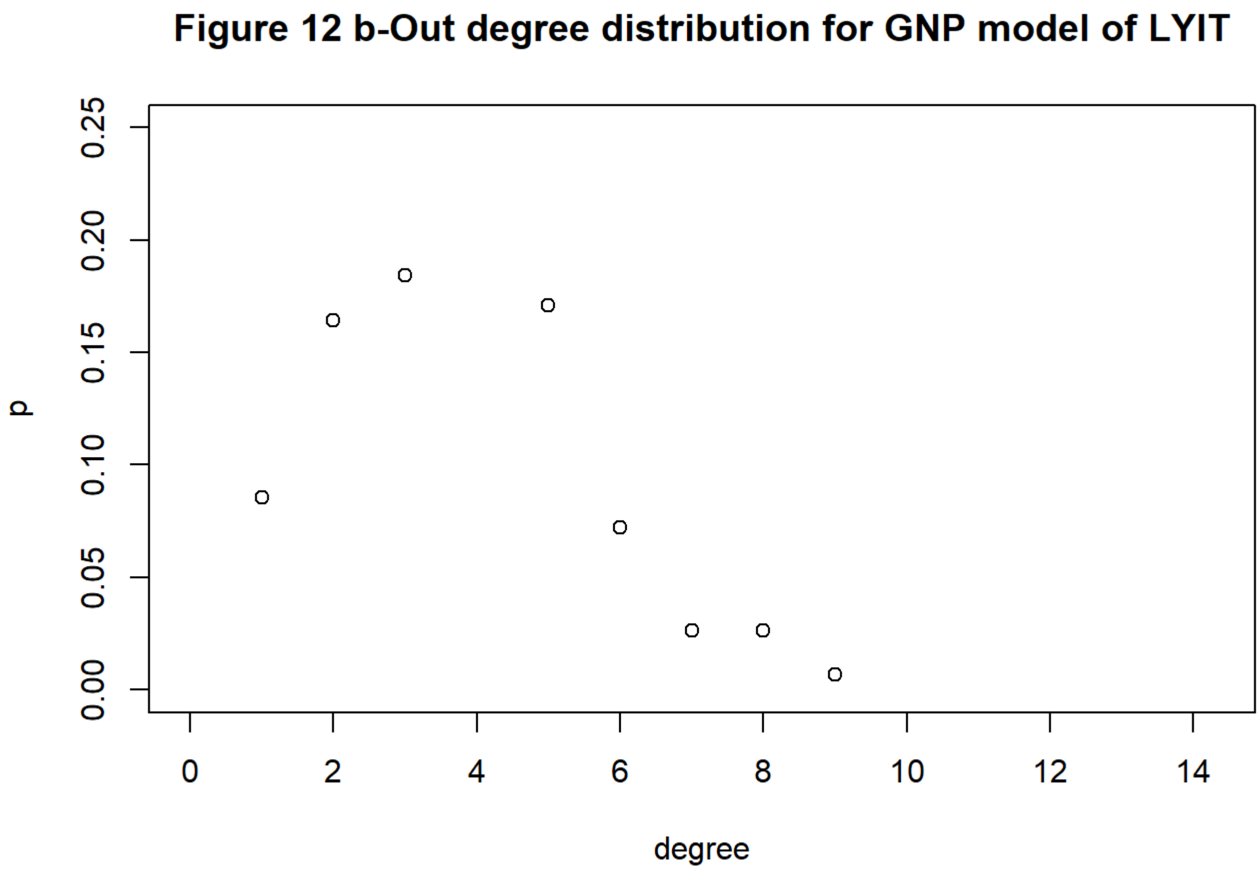
```
gnp.possible.max_degree1 <- n1-1
gnp.actual.max_degree1 <- max(degree(gnp1))
degree_to_plot1 = gnp.actual.max_degree1 + sd(degree(gnp1))
#plot actual degree distribution
plot(degree.distribution(gnp1), xlim = c(0,degree_to_plot1),ylim=c(0,0.25),xlab="degree", ylab=
"p", main="Figure 12-degree distribution for GNP model of LYIT")
```




```
plot(degree.distribution(gnp1,mode="in"), xlim = c(0,degree_to_plot1),ylim=c(0,0.25),xlab="degre
e", ylab="p", main="Figure 12 a-In degree distribution for Preferential GNP of LYIT")
```




```
plot(degree.distribution(gnp1,mode="out"), xlim = c(0,degree_to_plot1),ylim=c(0,0.25),xlab="degr
ee", ylab="p", main="Figure 12 b-Out degree distribution for GNP model of LYIT")
```



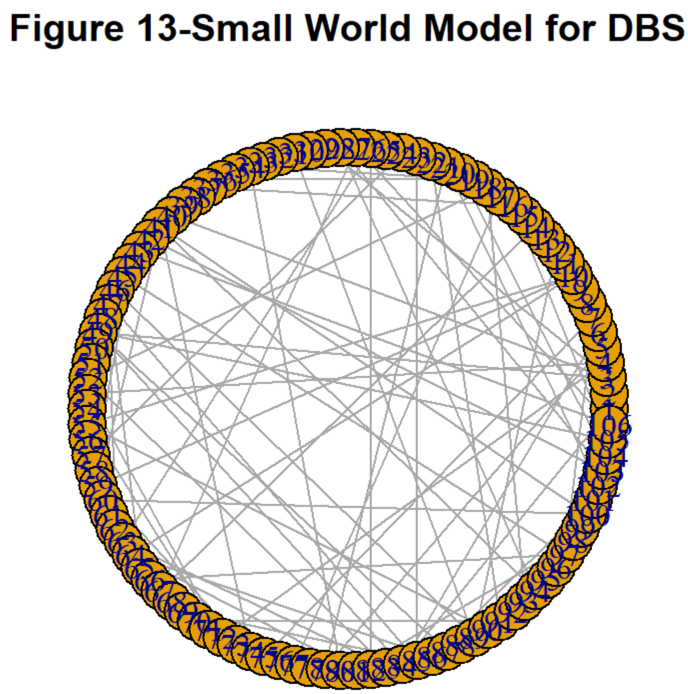
5. Small World Model

5.1 Small World Model for DBS

5.1.1 Small World Model Graph for DBS

For the small world, 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 and found successfully as below. The sample_smallworld() function is used to obtain the small world model and it is seen that the connections are less in this which means there will be high clustering coefficient and average path length.

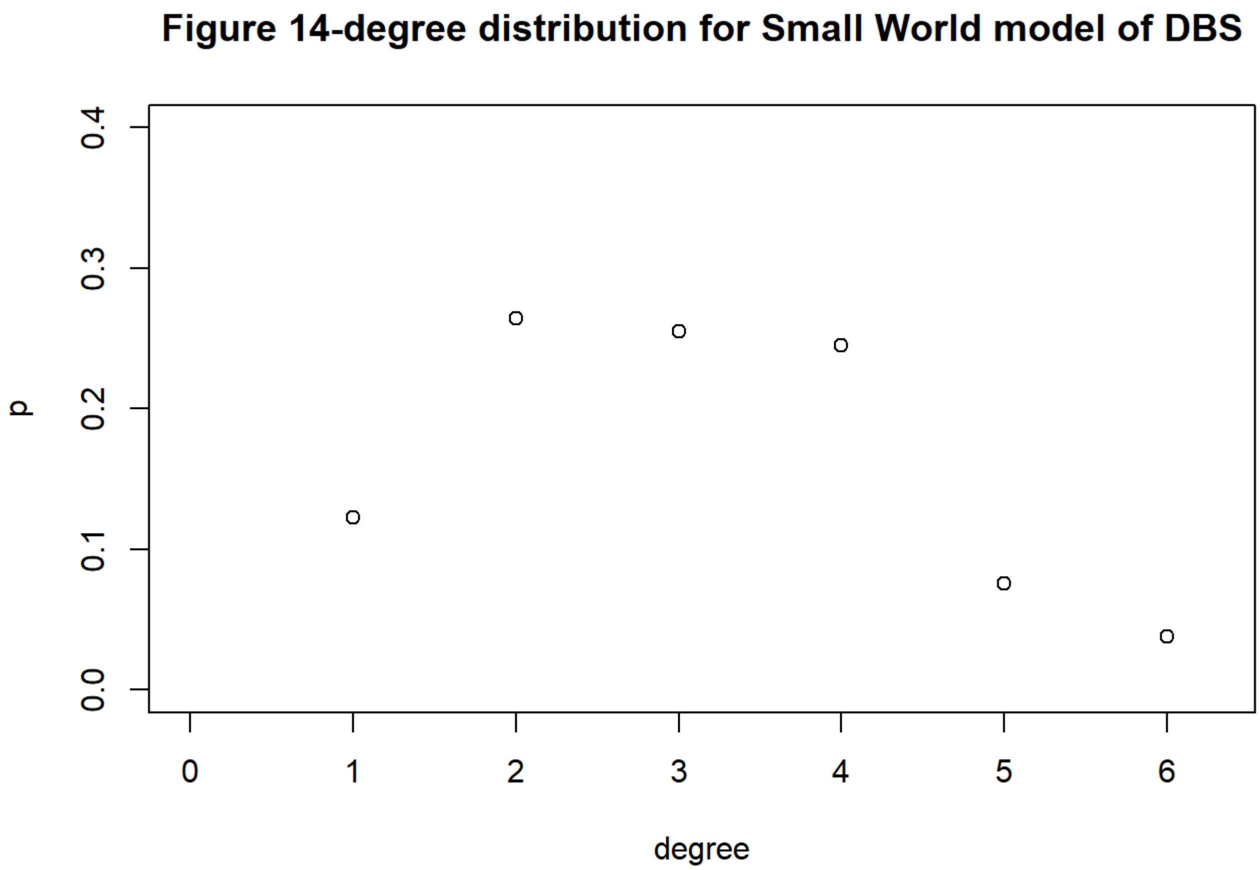
```
clus.coeff<-transitivity(dbsgraph,type="average")
c0<-(3*(k-2))/(4*(k-1))
beta<-1-((clus.coeff)/(c0))^0.33
sw<-sample_smallworld(dim=1,size=n,nei = k/2,p=beta)
sw<- simplify(sw, remove.multiple = TRUE, remove.loops = TRUE)
plot(sw, layout=layout.circle(sw), main="Figure 13-Small World Model for DBS")
```



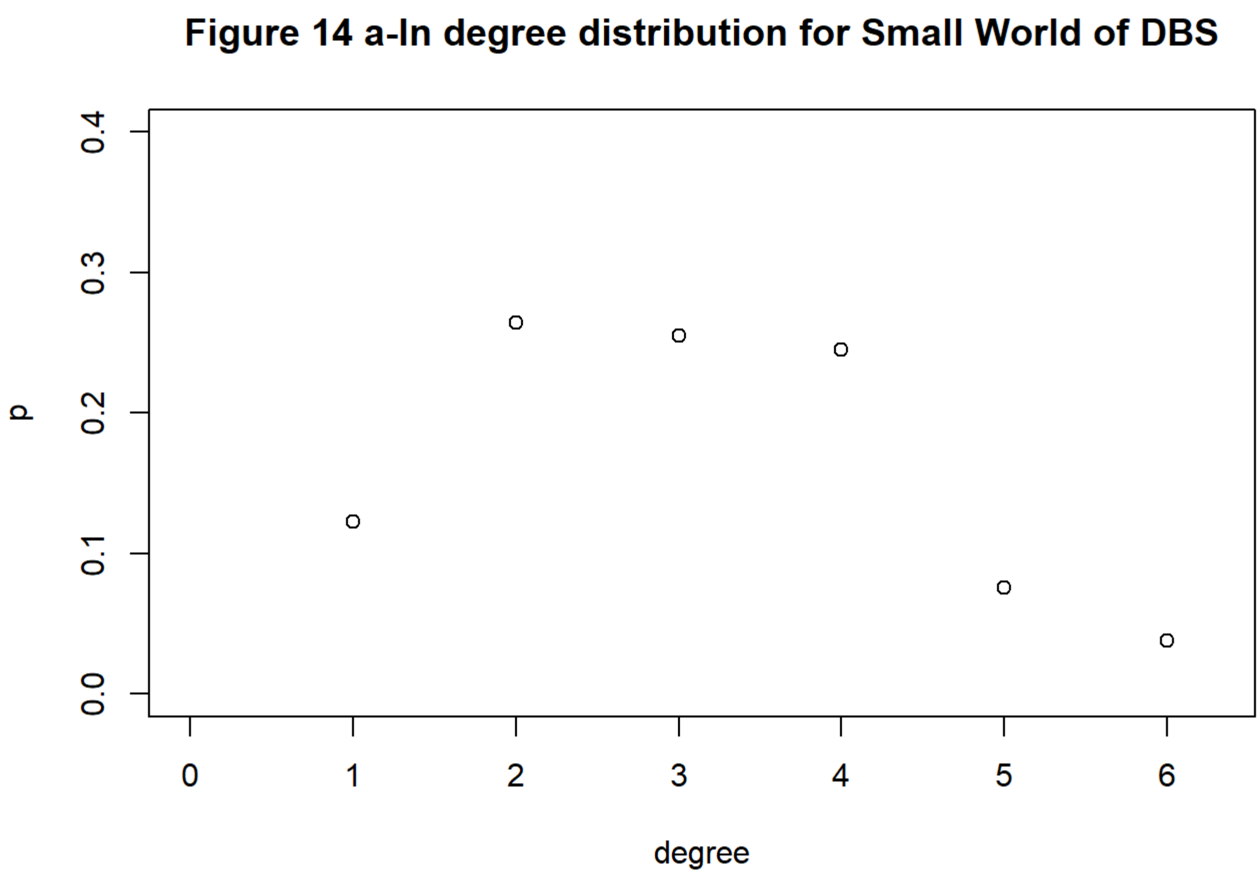
5.1.2 Small World Model Degree Distribution for DBS

Just like in GNP, the in and out degree distribution are similar in Small world model. It is seen that it is trying to follow normal distribution and can be normalised so with the highest probability for 3 degree.

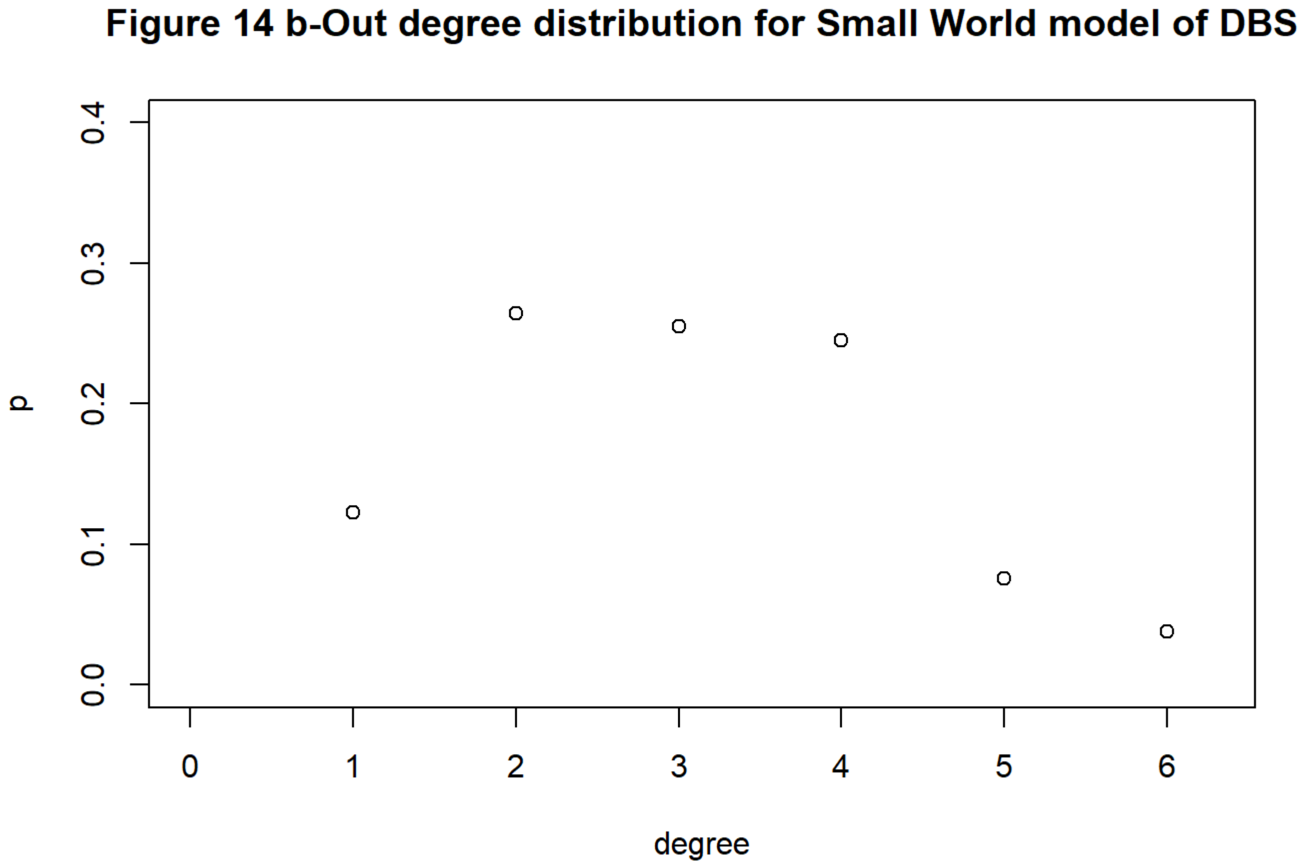
```
#degree distribution
sw.possible.max_degree <- n-1
sw.actual.max_degree <- max(degree(sw))
degree_to_plot = sw.actual.max_degree + sd(degree(sw))
#plot actual degree distribution
plot(degree.distribution(sw), xlim = c(0,degree_to_plot),ylim=c(0,0.40),xlab="degree", ylab="p",
main="Figure 14-degree distribution for Small World model of DBS")
```




```
plot(degree.distribution(sw,mode="in"), xlim = c(0,degree_to_plot),ylim=c(0,0.40),xlab="degree",
ylab="p", main="Figure 14 a-In degree distribution for Small World of DBS")
```




```
plot(degree.distribution(sw,mode="out"), xlim = c(0,degree_to_plot),ylim=c(0,0.40),xlab="degree"
, ylab="p", main="Figure 14 b-Out degree distribution for Small World model of DBS")
```



5.1.3 Small World Model based values for DBS

The values show that there is high average path length and diameter as guessed.

```
cat("The average path length of Small World Model of DBS is",average.path.length(sw))
```

```
## The average path length of Small World Model of DBS is 6.886932
```

```
cat("\nThe diameter of Small World Model of DBS is",diameter(sw))
```

```
##
## The diameter of Small World Model of DBS is 20
```

```
cat("\nThe average clustering coefficient of Small World Model of DBS is",transitivity(sw, type=
"average"))
```

```
##
## The average clustering coefficient of Small World Model of DBS is 0.06564103
```

```
cat("\nThe global clustering coefficient of Small World Model of DBS is",transitivity(sw, type=
"global"))
```

```
##
## The global clustering coefficient of Small World Model of DBS is 0.04663212
```

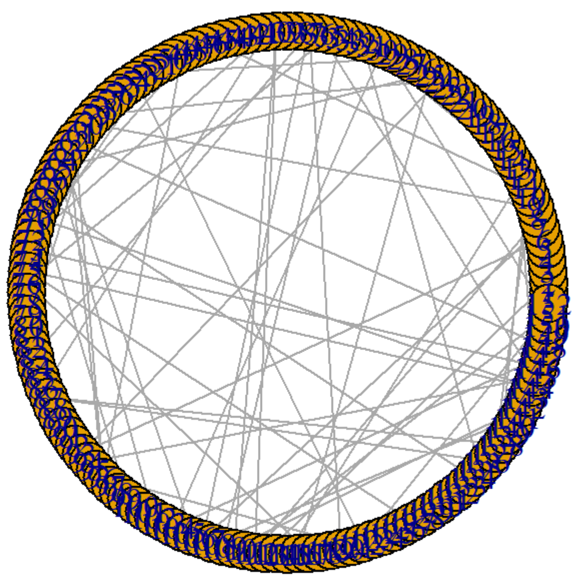
5.2 Small World Model for LYIT

5.2.1 Small World Model Graph for LYIT

The same process is repeated for LYIT. It is seen that the connections are sparse.

```
clus.coeff1<-transitivity(lyitgraph,type="average")
c01<-(3*(k1-2))/(4*(k1-1))
beta1<-1-((clus.coeff1)/(c01))^0.33
sw1<-sample_smallworld(dim=1,size=n1,nei = k1/2,p=beta1)
sw1<- simplify(sw1, remove.multiple = TRUE, remove.loops = TRUE)
plot(sw1, layout=layout.circle(sw1), main="Figure 15-Small World Model for LYIT")
```

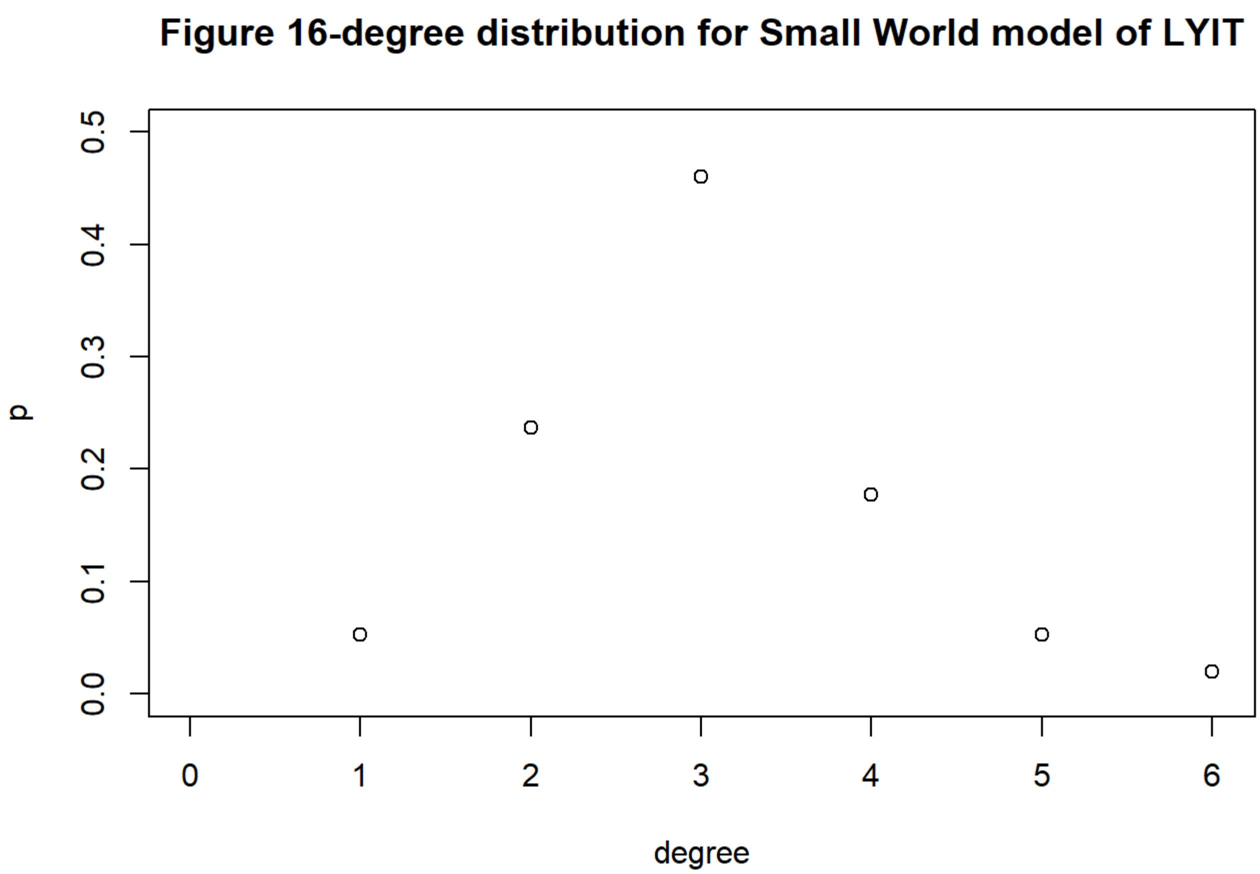
Figure 15-Small World Model for LYIT



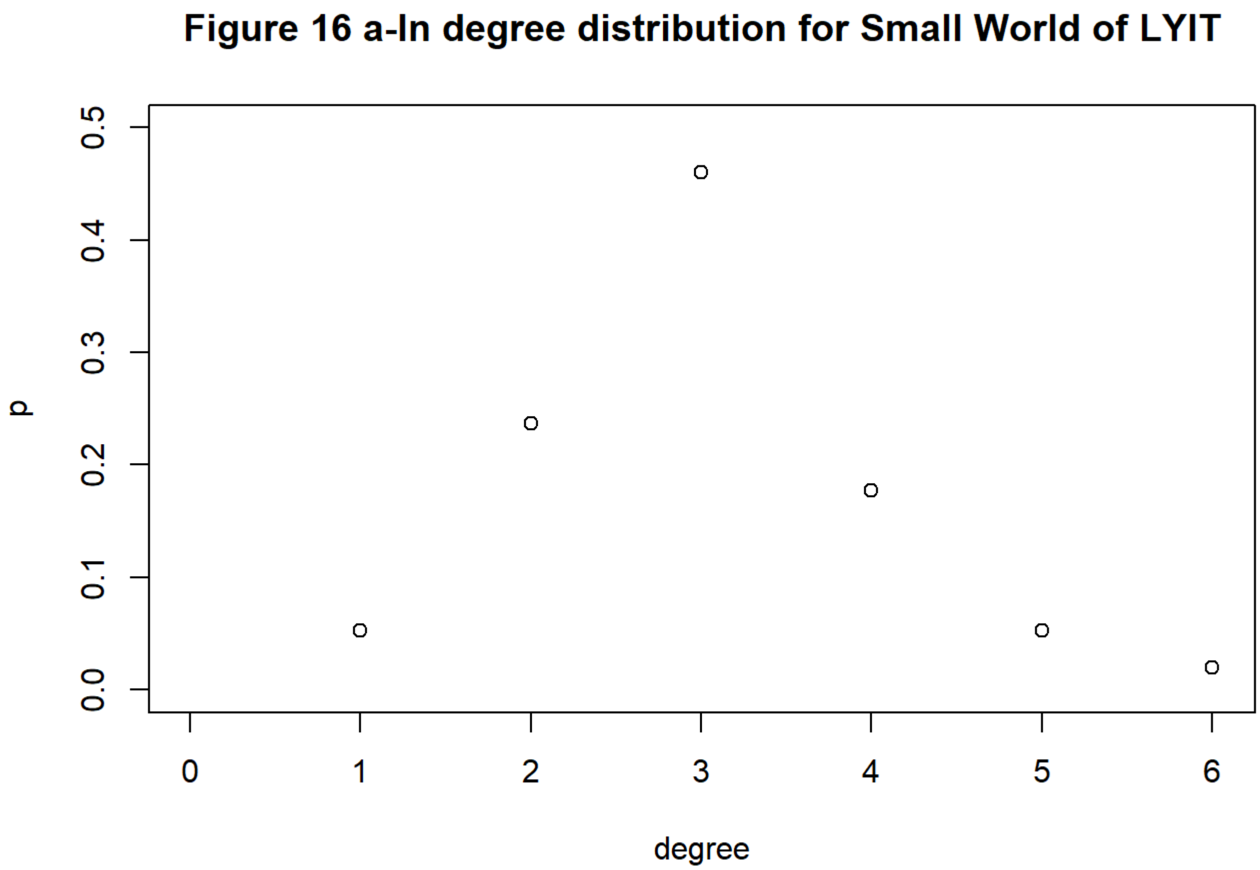
5.2.2 Small World Model Degree Distribution for LYIT

The in and out degree distribution are similar and it is seen that it is trying to follow normal distribution with the highest probablitiy for degree 3.

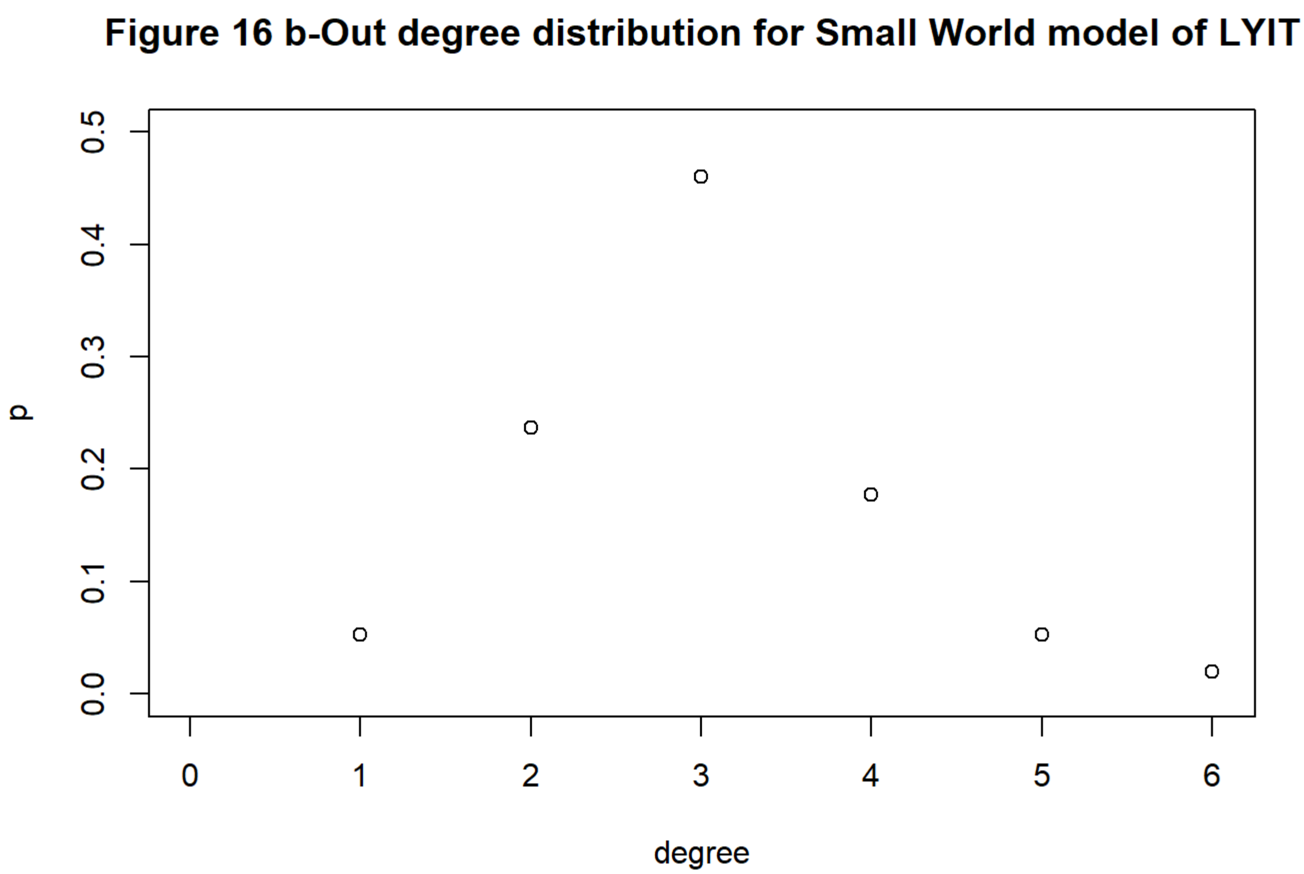
```
sw.possible.max_degree1 <- n1-1
sw.actual.max_degree1 <- max(degree(sw1))
degree_to_plot1 = sw.actual.max_degree1 + sd(degree(sw1))
#plot actual degree distribution
plot(degree.distribution(sw1), xlim = c(0,degree_to_plot1),ylim=c(0,0.50),xlab="degree", ylab=
"p", main="Figure 16-degree distribution for Small World model of LYIT")
```




```
plot(degree.distribution(sw1,mode="in"), xlim = c(0,degree_to_plot1),ylim=c(0,0.50),xlab="degre
e", ylab="p", main="Figure 16 a-In degree distribution for Small World of LYIT")
```




```
plot(degree.distribution(sw1,mode="out"), xlim = c(0,degree_to_plot1),ylim=c(0,0.50),xlab="degre
e", ylab="p", main="Figure 16 b-Out degree distribution for Small World model of LYIT")
```



5.2.3 Small World Model based values for LYIT

The average length and diameter are very high. It confirms that there is it will try and move to a situation where length will be low and clustering coefficient will be high when beta is at 1,

```
cat("The average path length of Small World Model of LYIT is",average.path.length(sw1))
```

```
## The average path length of Small World Model of LYIT is 8.529773
```

```
cat("\nThe diameter of Small World Model of LYIT is",diameter(sw1))
```

```
## 
## The diameter of Small World Model of LYIT is 24
```

```
cat("\nThe average clustering coefficient of Small World Model of LYIT is",transitivity(sw1, typ
e="average"))
```

```
## 
## The average clustering coefficient of Small World Model of LYIT is 0.01388889
```

```
cat("\nThe global clustering coefficient of Small World Model of LYIT is",transitivity(sw1, type
="global"))
```

```
##
## The global clustering coefficient of Small World Model of LYIT is 0.01310044
```

6. Preferential Attachment Model

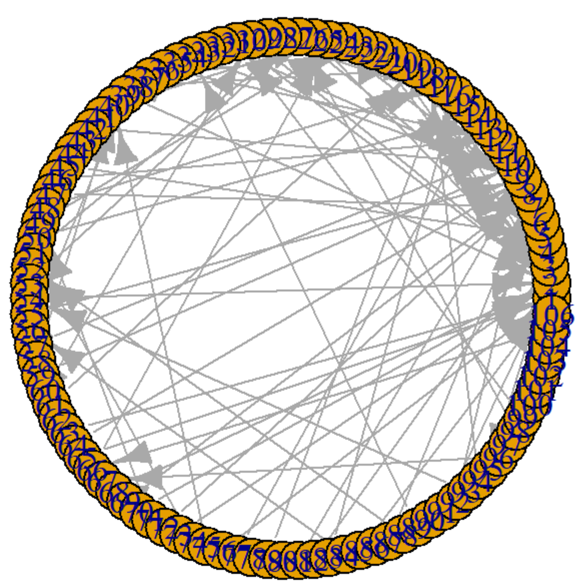
6.1 Preferential Attachment Model for DBS

6.1.1 Preferential Attachment Model Graph for DBS

The preferential attachment model is obtained used the sample_pa() function. The power is given as 1 and m as one so that one is added at each step towards the model. It is directed and it is seen that it is clustered towards some nodes as it joins network by forming edges to other nodes with a probability that is proportional to the node's degree. It is then reinforced till the end.

```
pa<-sample_pa(n,power = 1,m=1)
plot(pa, layout=layout.circle(pa), main="Figure 17 -Preferential Attachment Model for DBS")
```

Figure 17 -Preferential Attachment Model for DBS



6.1.2 Preferential Attachment Model based values for DBS

The average length is obtained and it is seen that it is not high. The clustering coefficient is not applicable for this model and hence leaving it out.

```
cat("The average path length of Preferential Attachment Model of DBS is",average.path.length(p
a))
```

```
## The average path length of Preferential Attachment Model of DBS is 2.327273
```

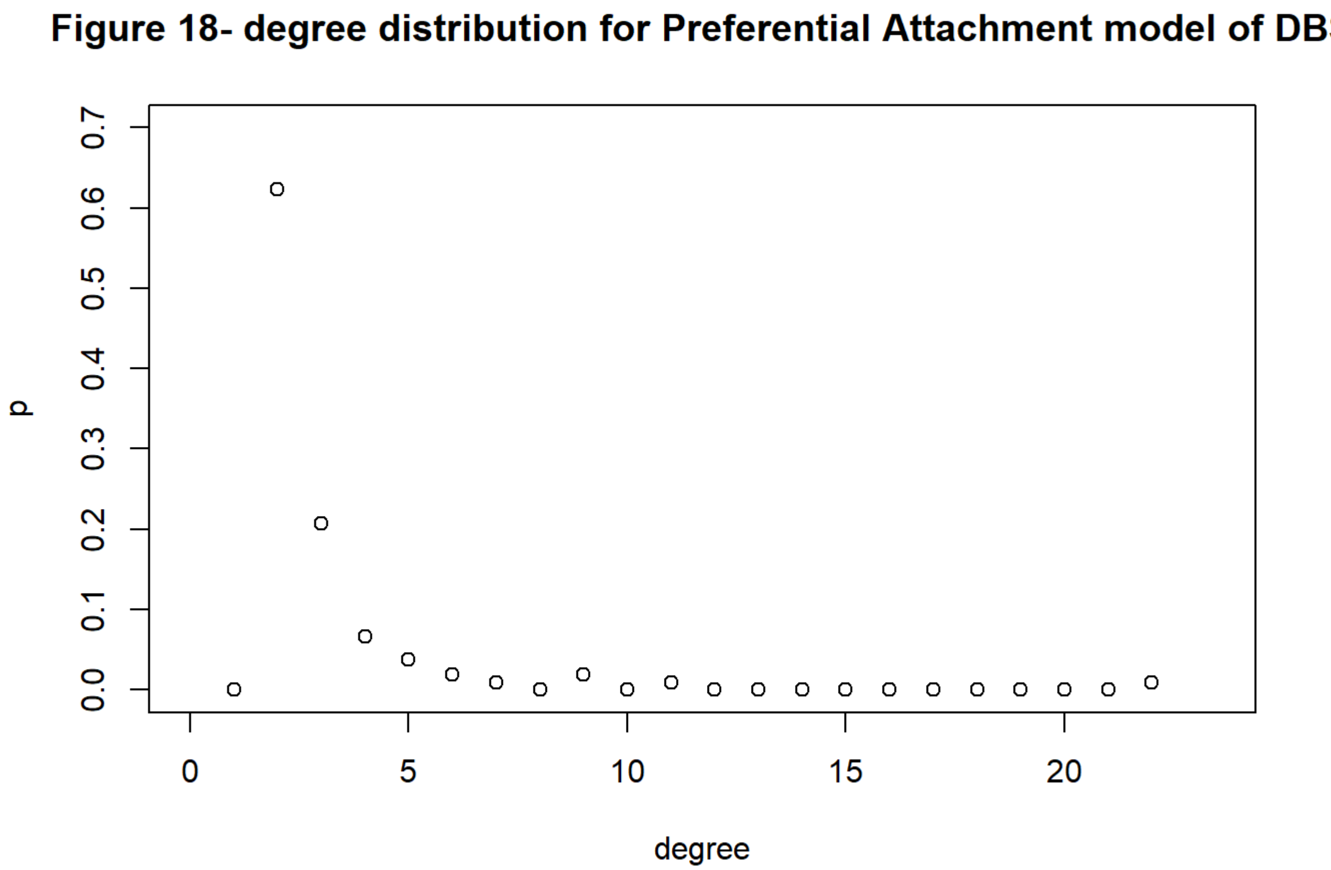
```
cat("\nThe diameter of Preferential Attachment Model of DBS is",diameter(pa))
```

```
##
## The diameter of Preferential Attachment Model of DBS is 7
```

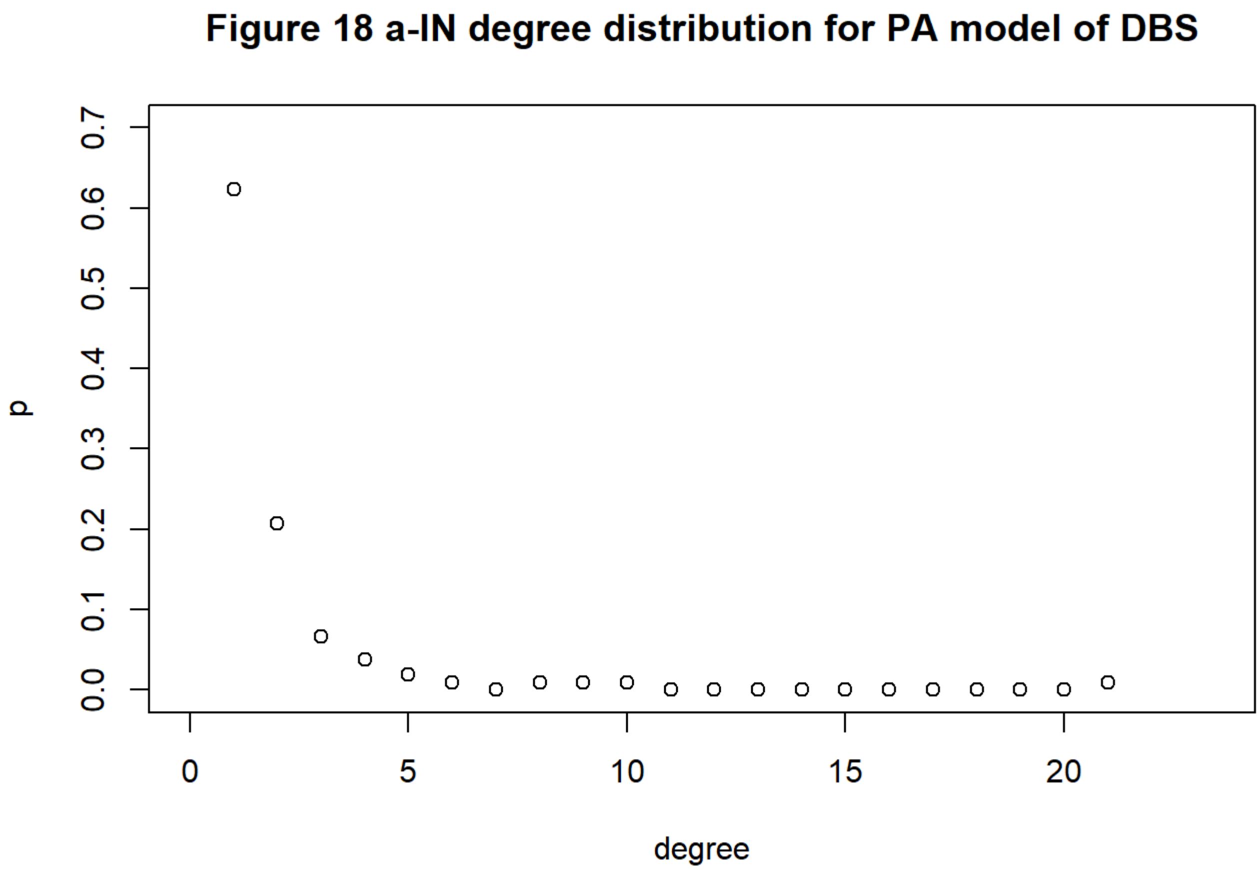
6.1.3 Preferential Attachment Model Degree Distribution for DBS

The degree distribution for the preferential attachment model degree is obtained . It is seen that there is only one out degree distribution and others in the in degree distribution. The highest probability that is present is similar to the original network for DBS.

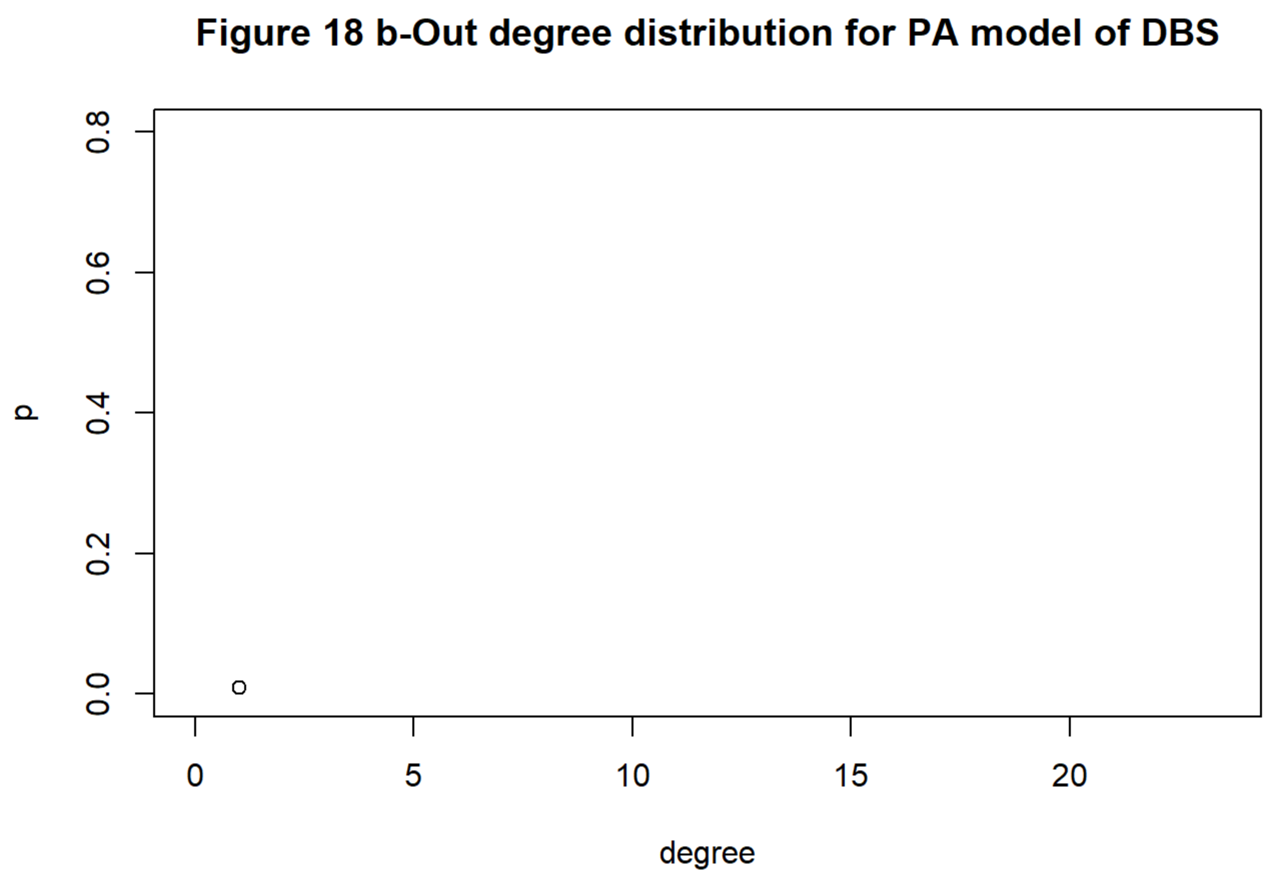
```
#degree distribution
pa.possible.max_degree <- n-1
pa.actual.max_degree <- max(degree(pa))
degree_to_plot = pa.actual.max_degree + sd(degree(pa))
plot(degree.distribution(pa), xlim = c(0,degree_to_plot),ylim=c(0,0.70),xlab="degree", ylab="p",
main="Figure 18- degree distribution for Preferential Attachment model of DBS")
```




```
plot(degree.distribution(pa,mode="in"), xlim = c(0,degree_to_plot),ylim=c(0,0.70),xlab="degree",
ylab="p", main="Figure 18 a-IN degree distribution for PA model of DBS")
```




```
plot(degree.distribution(pa,mode="out"), xlim = c(0,degree_to_plot),ylim=c(0,0.80),xlab="degree"
, ylab="p", main="Figure 18 b-Out degree distribution for PA model of DBS")
```



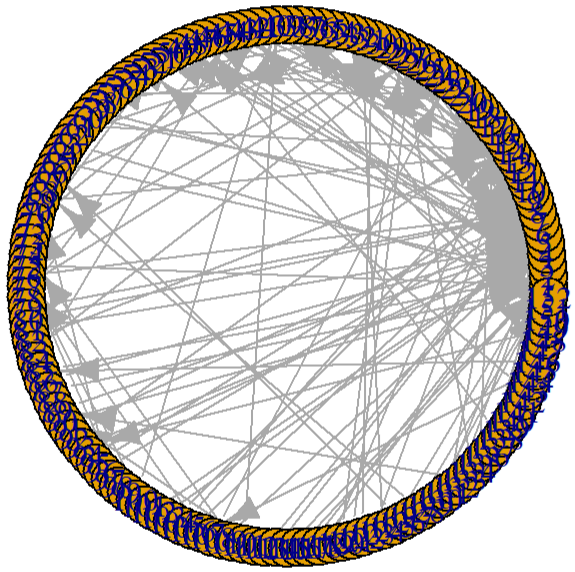
6.2 Preferential Attachment Model for LYIT

6.2.1 Preferential Attachment Model Graph for LYIT

The same process is repeated again for LYIT. It is seen that that there is alot of edges directed towards the initial nodes that are present.

```
pa1<-sample_pa(n1,power = 1,m=1)
plot(pa1, layout=layout.circle(pa1), main="Figure 19 -Preferential Attachment Model for LYIT")
```

Figure 19 -Preferential Attachment Model for LYIT



6.2.2 Preferential Attachment Model based values for LYIT

The average length is low compared to other models. The clustering coefficient is not applicable and so leaving it out.

```
cat("The average path length of Preferential Attachment Model of LYIT is",average.path.length(pa
1))
```

```
## The average path length of Preferential Attachment Model of LYIT is 2.219298
```

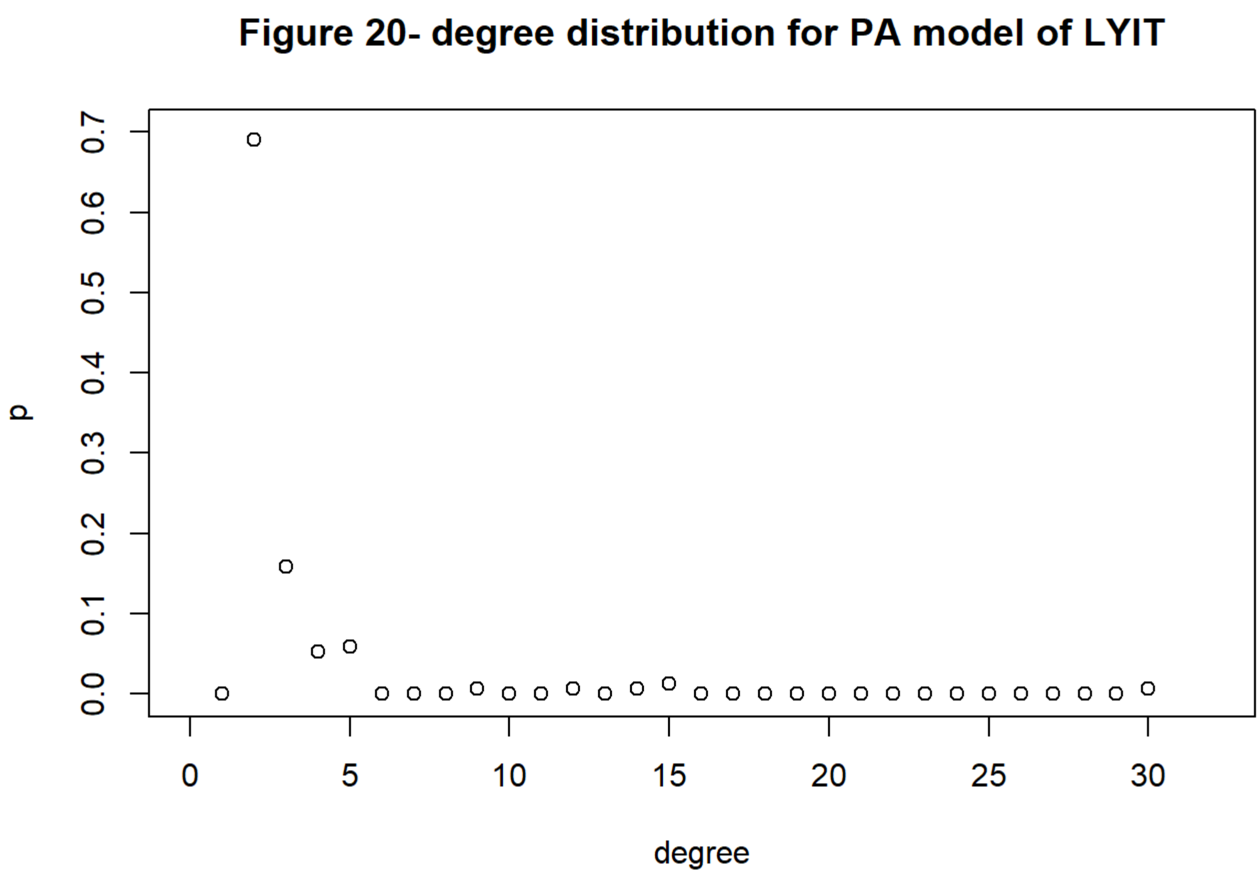
```
cat("\nThe diameter of Preferential Attachment Model of LYIT is",diameter(pa1))
```

```
##
## The diameter of Preferential Attachment Model of LYIT is 7
```

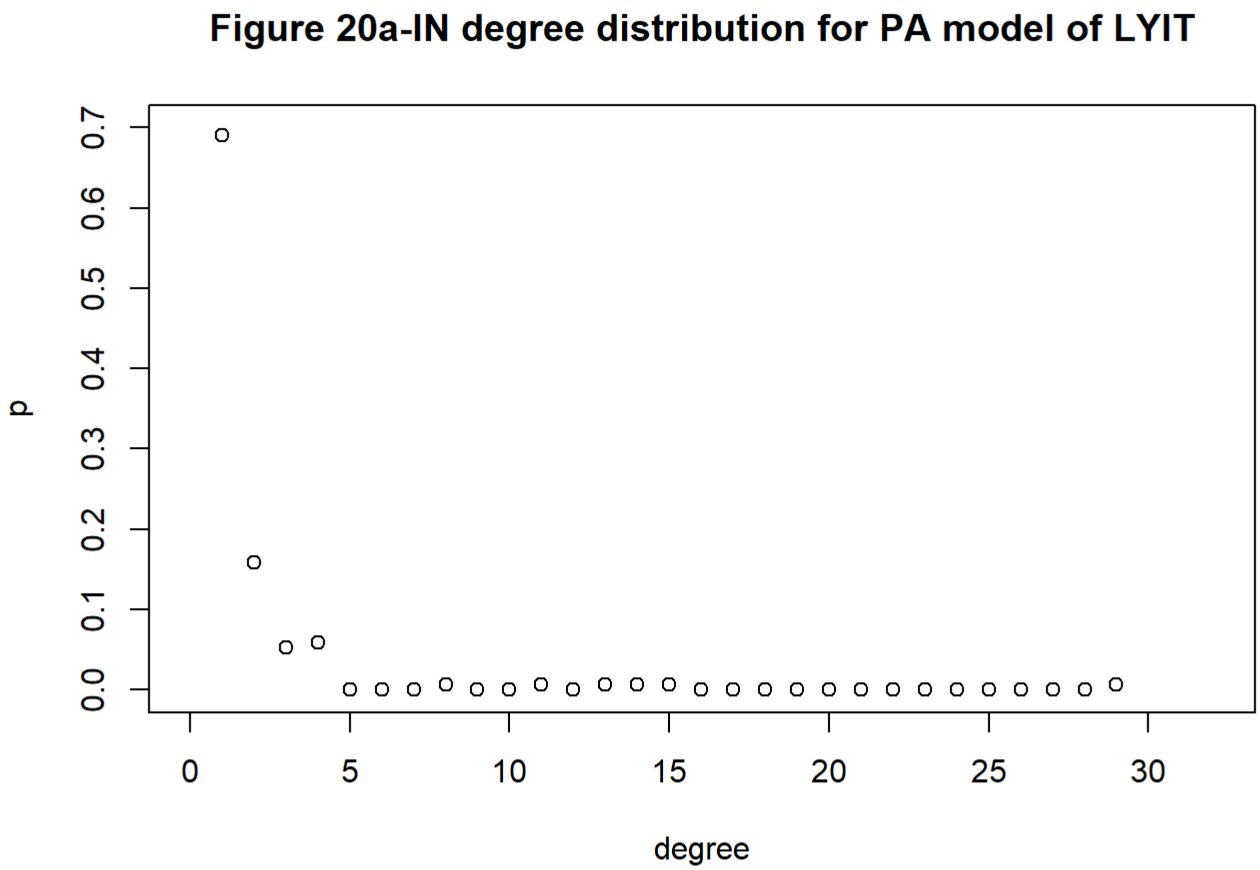
6.2.3 Preferential Attachment Model Degree Distribution for LYIT

The in degree is having the most degrees present with only one present in the out degree. It is seen that the higher probability is present for the lower degree values like in the original network.

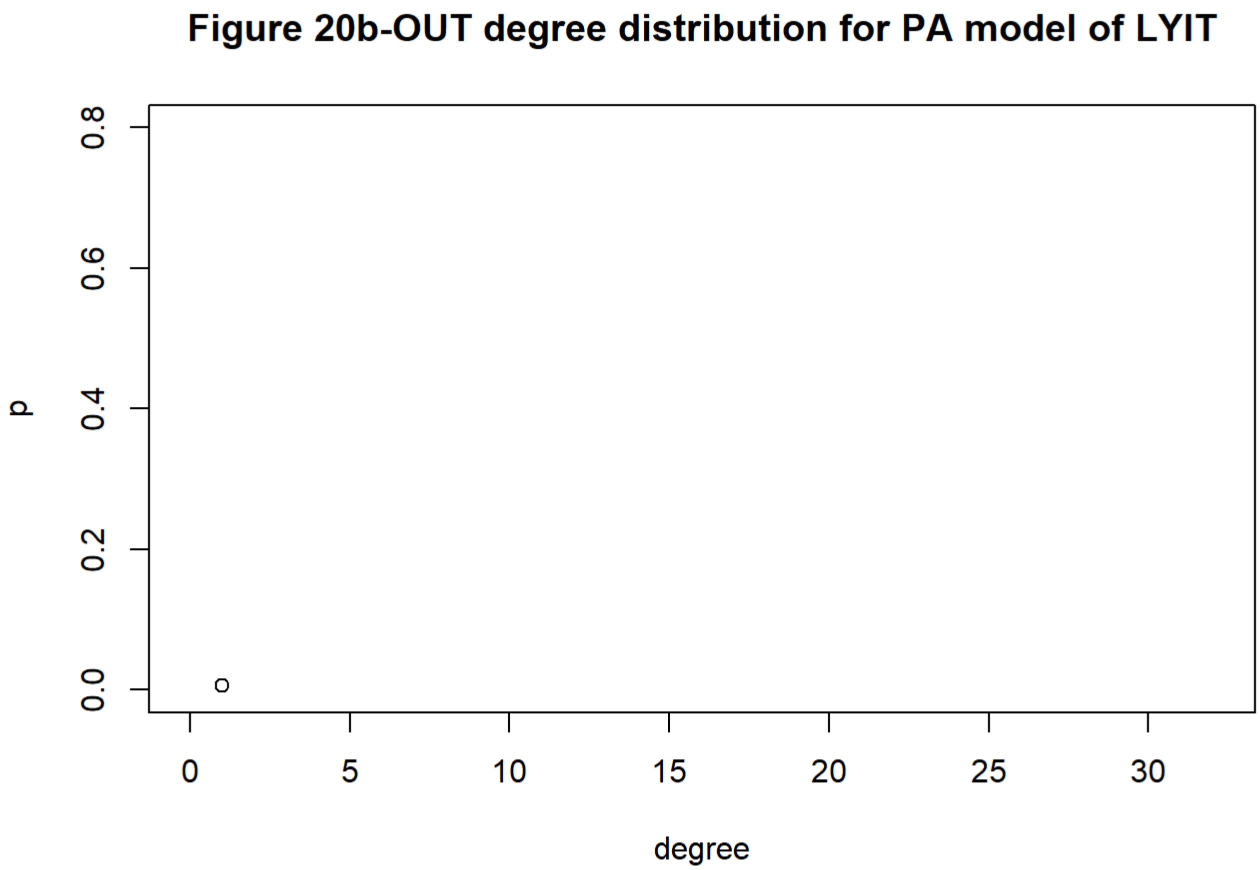
```
pa.possible.max_degree1 <- n1-1
pa.actual.max_degree1 <- max(degree(pa1))
degree_to_plot1 = pa.actual.max_degree1 + sd(degree(pa1))
#plot actual degree distribution
plot(degree.distribution(pa1), xlim = c(0,degree_to_plot1),ylim=c(0,0.70),xlab="degree", ylab=
"p", main="Figure 20- degree distribution for PA model of LYIT")
```




```
plot(degree.distribution(pa1,mode="in"), xlim = c(0,degree_to_plot1),ylim=c(0,0.70),xlab="degre
e", ylab="p", main="Figure 20a-IN degree distribution for PA model of LYIT")
```




```
plot(degree.distribution(pa1,mode="out"), xlim = c(0,degree_to_plot1),ylim=c(0,0.80),xlab="degree", ylab="p", main="Figure 20b-OUT degree distribution for PA model of LYIT")
```



7. Conclusion

1. It is observed that Random Network Model is not suitable for analysing the degree distribution property of a network. This is seen as the degree distribution of DBS and LYIT are different from that of the degree distribution of their random network models.
2. Preferential Distribution is found to be best suitable for finding the property of degree distribution from the given models for the network. Thus, the degree distribution in the DBS shows that there are high outbound and inbound probability for 1 and 2 and it is similar for LYIT.
3. For Clustering Coefficient, it is seen that the Small World Model is giving the nearest values among the model to the original network. The low clustering coefficient in both DBS and LYIT show that there are alot of edges associated with the nodes. In this way, local density is determined successfully.
4. The average path length in cases of LYIT is less than to the DBS because of the huge number of edges that are present in surplus to the nodes when compared to the DBS. The average path lengths are low for both DBS and LYIT due to the huge number of edges present.
5. The In and Out degree distributions show the changes in the distribution present and the varying probability associated with it. It is seen that in the In degree for DBS and LYIT, the higher probability are found for in the lower degrees and in the Out degree, the higher probability are found for degrees usually higher than the In degree ones.
6. Random Network is able to provide only the average path length in its model appropriately in line to the original network. This is confirmed when the model is applied to the DBS and LYIT network.

8. 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. https://en.wikipedia.org/wiki/Web_crawler (https://en.wikipedia.org/wiki/Web_crawler)
7. http://igraph.org/r/doc/erdos.renyi.game.html (http://igraph.org/r/doc/erdos.renyi.game.html)
8. 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
9. 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)
10. Week 3 to 5 Lecture Notes and Tutorials in Blackboard