CptS 591: Elements of Network Science - Assignment 1

Hitesh Bahar

2/4/2019

```
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(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
require(scales)
```

```
## Loading required package: scales
```

```
library(knitr)
```

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

```
setwd("D:/NetworkScience/assignment1/Dataset-20190205T060348Z-001/Dataset")
```

Question-1

```
jazzNetGraph <- read_graph("jazz.net", format = "pajek")
celegansMetabolicGraph <- read_graph("celegans_metabolic.net",format = "pajek")
higgsSocialNetworkGraph <- read_graph("higgs-social_network.edgelist", format = "edgelist")
higgsMenNetworkGraph <- read_graph("higgs-mention_network.edgelist", format = "edgelist")
higgsRetweetNetworkGraph <- read_graph("higgs-retweet_network.edgelist", format = "edgelist")
erdosRenyi01Graph <- erdos.renyi.game(2000, 0.01)
erdosRenyi005Graph <- erdos.renyi.game(2000, 0.005)
erdosRenyi0025Graph <- erdos.renyi.game(2000, 0.0025)

#Create a Data Frame as " Network Type n m c d l L ccl ccg "
dataFrame <- data.frame("Network" = character(8),
                  "Type" = character(8),
                  "n" = integer(8),
                  "m" = integer(8),
                  "cStrong" =integer(8),
                  "cWeak" = integer(8),
                  "d" = integer(8),
                  "l" = double(8),
                  "L" = integer(8),
                  "ccL" = double(8),
```

```
                "ccG" = double(8),
                stringsAsFactors = FALSE)
#Netowk Name
dataFrame$Network[1] <- "Jazz muscians network "
dataFrame$Network[2] <- "C.elegans metabolic network"
dataFrame$Network[3] <- "Higgs Twitter-Soc-Net network"
dataFrame$Network[4] <- "Higgs Twitter-Men-Net network"
dataFrame$Network[5] <- "Higgs Twitter-Retweet-Net network"
dataFrame$Network[6] <- "Erdos Renyi 0.01 network"
dataFrame$Network[7] <- "Erdos Renyi 0.005 network"
dataFrame$Network[8] <- "Erdos Renyi 0.0025 network"

#Type Directed or Un Directed
type <- function(x){
  if(is_directed(x) == TRUE) {
    direction <- "Directed"
  } else{
    direction <- "Un Directed"
  }
  return(direction)
}

dataFrame$Type[1] <- type(jazzNetGraph)
dataFrame$Type[2] <- type(celegansMetabolicGraph)
dataFrame$Type[3] <- type(higgsSocialNetworkGraph)
dataFrame$Type[4] <- type(higgsMenNetworkGraph)
dataFrame$Type[5] <- type(higgsRetweetNetworkGraph)
dataFrame$Type[6] <- type(erdosRenyi01Graph)
dataFrame$Type[7] <- type(erdosRenyi005Graph)
dataFrame$Type[8] <- type(erdosRenyi0025Graph)

# n is the number of nodes
dataFrame$n[1] <- vcount(jazzNetGraph)
dataFrame$n[2] <- vcount(celegansMetabolicGraph)
dataFrame$n[3] <- vcount(higgsSocialNetworkGraph)
dataFrame$n[4] <- vcount(higgsMenNetworkGraph)
dataFrame$n[5] <- vcount(higgsRetweetNetworkGraph)
dataFrame$n[6] <- vcount(erdosRenyi01Graph)
dataFrame$n[7] <- vcount(erdosRenyi005Graph)
dataFrame$n[8] <- vcount(erdosRenyi0025Graph)

# m is the number of links
dataFrame$m[1] <- ecount(jazzNetGraph)
dataFrame$m[2] <- ecount(celegansMetabolicGraph)
dataFrame$m[3] <- ecount(higgsSocialNetworkGraph)
dataFrame$m[4] <- ecount(higgsMenNetworkGraph)
dataFrame$m[5] <- ecount(higgsRetweetNetworkGraph)
dataFrame$m[6] <- ecount(erdosRenyi01Graph)
dataFrame$m[7] <- ecount(erdosRenyi005Graph)
dataFrame$m[8] <- ecount(erdosRenyi0025Graph)

# c is the number of connected components (in the case of a directed network, separately report on the
dataFrame$cStrong[1] <- count_components(jazzNetGraph, mode = "strong")
```

```
dataFrame$cWeak[1] <- count_components(jazzNetGraph, mode = "weak")
dataFrame$cStrong[2] <- count_components(celegansMetabolicGraph, mode = "strong")
dataFrame$cWeak[2] <- count_components(celegansMetabolicGraph, mode = "weak")
dataFrame$cStrong[3] <- count_components(higgsSocialNetworkGraph, mode = "strong")
dataFrame$cWeak[3] <- count_components(higgsSocialNetworkGraph, mode = "weak")
dataFrame$cStrong[4] <- count_components(higgsMenNetworkGraph, mode = "strong")
dataFrame$cWeak[4] <- count_components(higgsMenNetworkGraph, mode = "weak")
dataFrame$cStrong[5] <- count_components(higgsRetweetNetworkGraph, mode = "strong")
dataFrame$cWeak[5] <- count_components(higgsRetweetNetworkGraph, mode = "weak")
dataFrame$cStrong[6] <- count_components(erdosRenyi01Graph, mode = "strong")
dataFrame$cWeak[6] <- NA
dataFrame$cStrong[7] <- count_components(erdosRenyi005Graph, mode = "strong")
dataFrame$cWeak[7] <- NA
dataFrame$cStrong[8] <- count_components(erdosRenyi0025Graph, mode = "strong")
dataFrame$cWeak[8] <- NA

#d is the maximum degree
dataFrame$d[1] <- max(degree(jazzNetGraph))
dataFrame$d[2] <- max(degree(celegansMetabolicGraph))
dataFrame$d[3] <- max(degree(higgsSocialNetworkGraph))
dataFrame$d[4] <- max(degree(higgsMenNetworkGraph))
dataFrame$d[5] <- max(degree(higgsRetweetNetworkGraph))
dataFrame$d[6] <- max(degree(erdosRenyi01Graph))
dataFrame$d[7] <- max(degree(erdosRenyi005Graph))
dataFrame$d[8] <- max(degree(erdosRenyi0025Graph))

# l is the average path length
dataFrame$l[1] <- mean_distance(jazzNetGraph)
dataFrame$l[2] <- mean_distance(celegansMetabolicGraph)
dataFrame$l[3] <- NA
dataFrame$l[4] <- NA
dataFrame$l[5] <- NA
dataFrame$l[6] <- max(degree(erdosRenyi01Graph))
dataFrame$l[7] <- max(degree(erdosRenyi005Graph))
dataFrame$l[8] <- max(degree(erdosRenyi0025Graph))

# L is the diameter
dataFrame$L[1] <- diameter(jazzNetGraph)
dataFrame$L[2] <- diameter(celegansMetabolicGraph)
dataFrame$L[3] <- NA
dataFrame$L[4] <- NA
dataFrame$L[5] <- NA
dataFrame$L[6] <- mean_distance(erdosRenyi01Graph)
dataFrame$L[7] <- mean_distance(erdosRenyi005Graph)
dataFrame$L[8] <- mean_distance(erdosRenyi0025Graph)

# ccl is the average local clustering coefficient
dataFrame$cc_local[1] <- transitivity(jazzNetGraph, type = "localaverageundirected")
dataFrame$cc_local[2] <- transitivity(celegansMetabolicGraph, type = "localaverageundirected")
dataFrame$cc_local[3] <- transitivity(higgsSocialNetworkGraph, type = "localaverageundirected")
dataFrame$cc_local[4] <- transitivity(higgsMenNetworkGraph, type = "localaverageundirected")
dataFrame$cc_local[5] <- transitivity(higgsRetweetNetworkGraph, type = "localaverageundirected")
dataFrame$cc_local[6] <- transitivity(erdosRenyi01Graph, type = "localaverageundirected")
```

```r
dataFrame$cc_local[7] <- transitivity(erdosRenyi005Graph, type = "localaverageundirected")
dataFrame$cc_local[8] <- transitivity(erdosRenyi0025Graph, type = "localaverageundirected")

# ccg is the global clustering coefficient (3 times number of triangles/ number of connected triplets)

dataFrame$cc_global[1] <- transitivity(jazzNetGraph, type = "globalundirected")
dataFrame$cc_global[2] <- transitivity(celegansMetabolicGraph, type = "globalundirected")
dataFrame$cc_global[3] <- transitivity(higgsSocialNetworkGraph, type = "globalundirected")
dataFrame$cc_global[4] <- transitivity(higgsMenNetworkGraph, type = "globalundirected")
dataFrame$cc_global[5] <- transitivity(higgsRetweetNetworkGraph, type = "globalundirected")
dataFrame$cc_global[6] <- transitivity(erdosRenyi01Graph, type = "globalundirected")
dataFrame$cc_global[7] <- transitivity(erdosRenyi005Graph, type = "globalundirected")
dataFrame$cc_global[8] <- transitivity(erdosRenyi0025Graph, type = "globalundirected")


kable(dataFrame, format = "markdown")
```

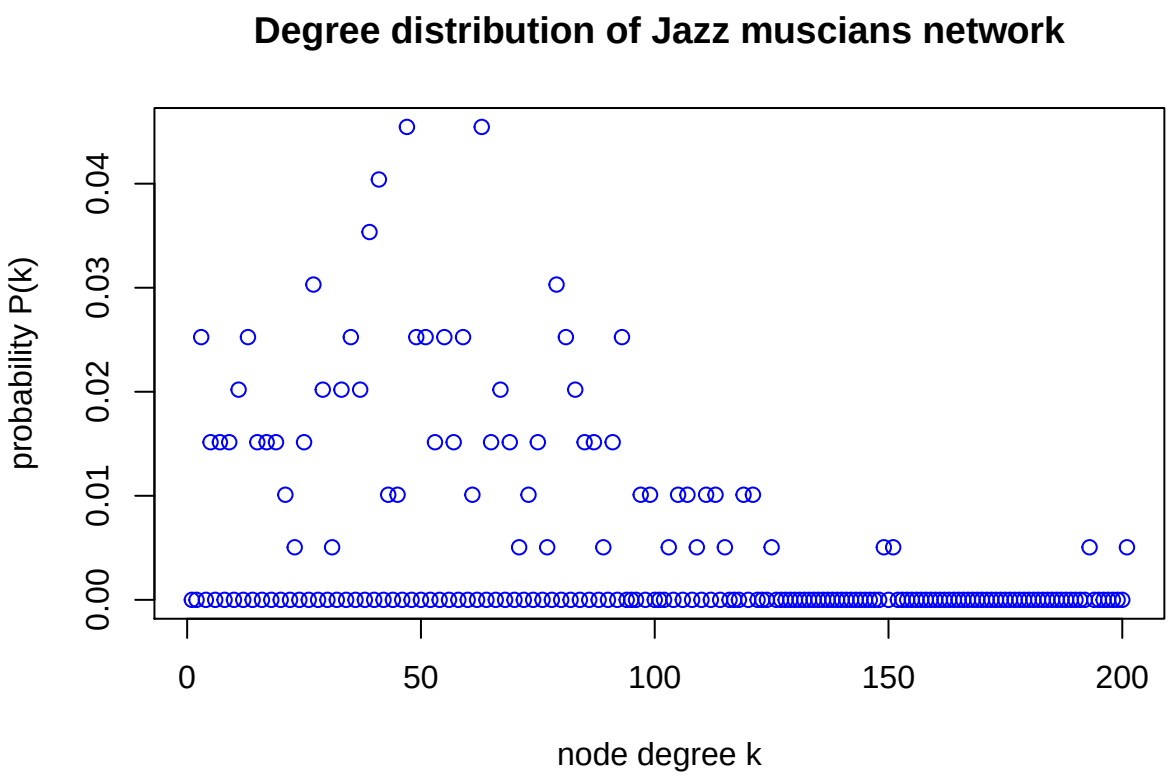
Network	Type	n	m	cStrong	cWeak	d	l	L	ccL	ccG	cc_local	cc_global
Jazz muscians network	Un Directed	198	5484	1	1	200	2.235041	6.000000	0	0	0.6334468	0.5202593
C.elegans metabolic network	Un Directed	453	4596	1	1	644	2.663785	7.000000	0	0	0.6551404	0.1244364
Higgs Twitter-Soc-Net network	Directed	456627	14855842	91665	157	51388	NA	NA	0	0	0.1993803	0.0086522
Higgs Twitter-Men-Net network	Directed	456620	226227	434203	344085	138373	NA	NA	0	0	0.5445226	0.0000472
Higgs Twitter-Retweet-Net network	Directed	456623	492198	417464	205651	311214	NA	NA	0	0	0.4901334	0.0000182
Erdos Renyi 0.01 network	Un Directed	2000	19947	1	NA	37	37.000002	2.831510	0	0	0.0100798	0.0101307
Erdos Renyi 0.005 network	Un Directed	2000	10074	1	NA	22	22.000008	3.552181	0	0	0.0049243	0.0047647
Erdos Renyi 0.0025 network	Un Directed	2000	4988	17	NA	14	14.000004	4.894571	0	0	0.0026262	0.0022864

```r
# Source - https://rpubs.com/yit/display_dataframe_with_Rpres_Rmd
```

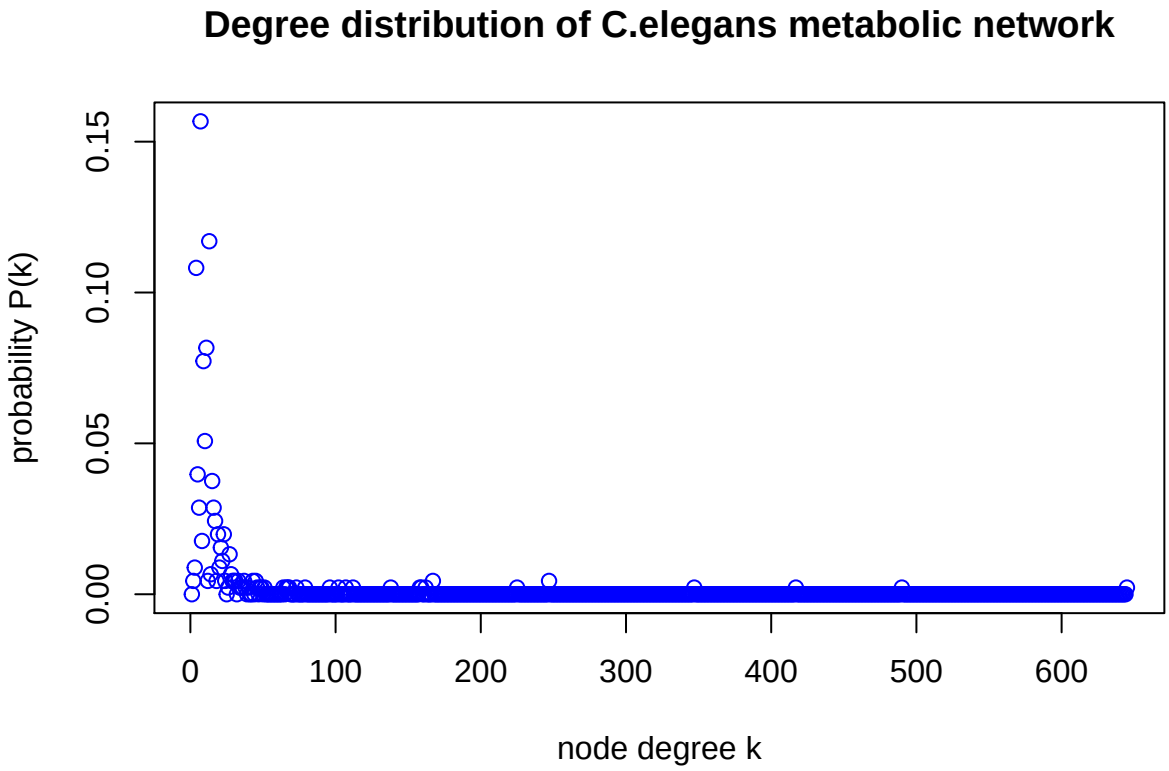
Question-2

```r
# Ref: https://gist.github.com/ruliana/8bac0e980b1e5d5bbd289d7ff34f38ab
# Plotting degree distribution with igraph and ggplot2


#Jazz muscians network
plot(degree.distribution(jazzNetGraph),
   main = "Degree distribution of Jazz muscians network",
   xlab="node degree k",
```

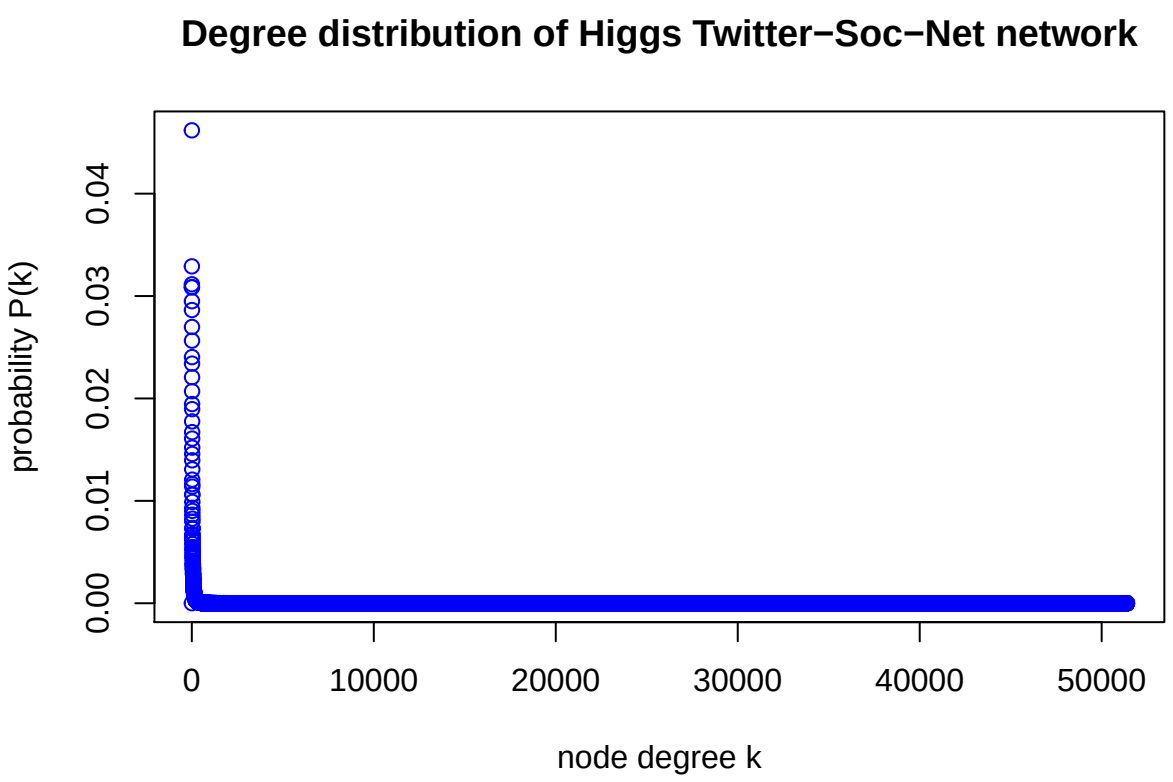
```
   ylab = "probability P(k)",
   col="blue"
   )
```



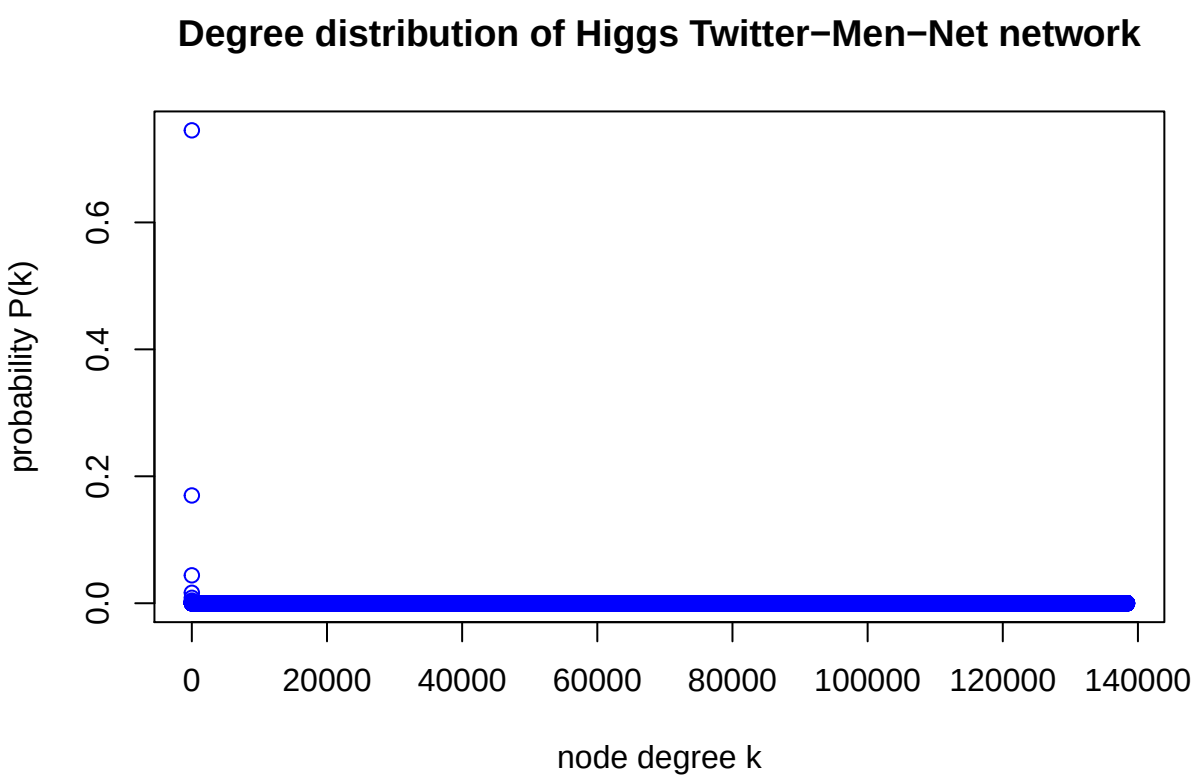

```
#C.elegans metabolic network
plot(degree.distribution(celegansMetabolicGraph),
     main = "Degree distribution of C.elegans metabolic network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```



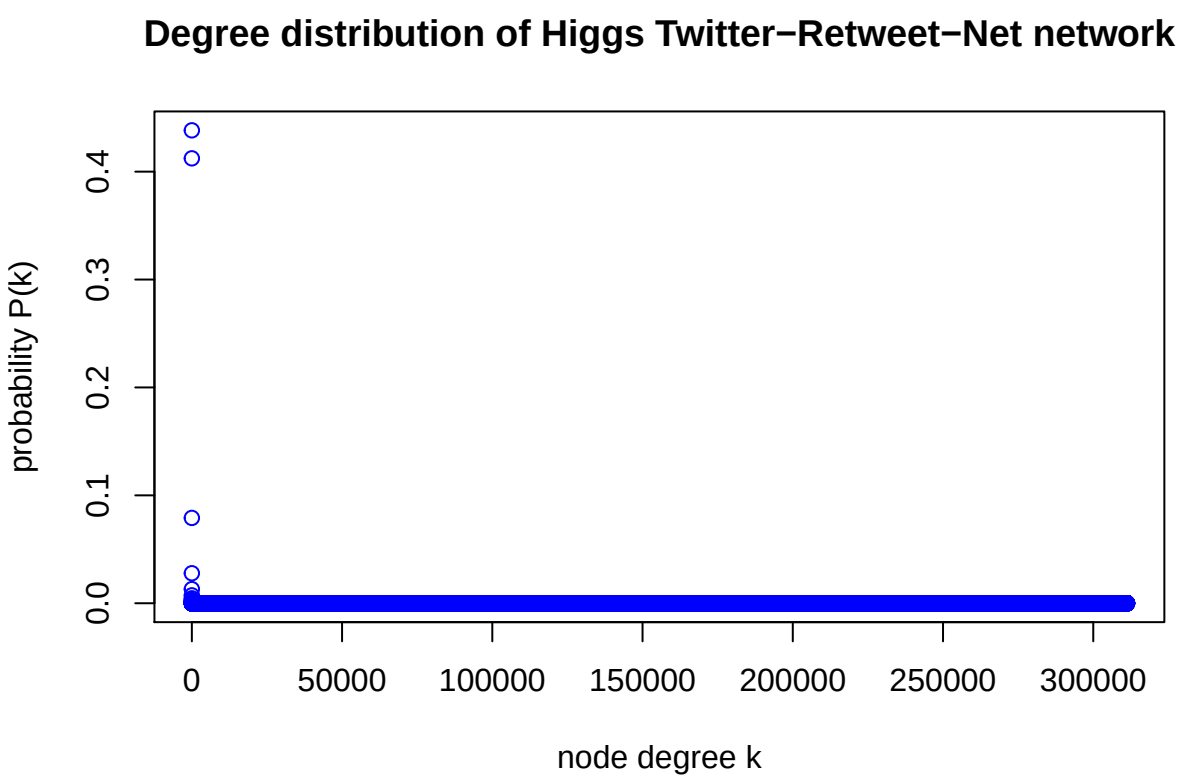

```
#Higgs Twitter-Soc-Net network
plot(degree.distribution(higgsSocialNetworkGraph),
     main = "Degree distribution of Higgs Twitter-Soc-Net network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```



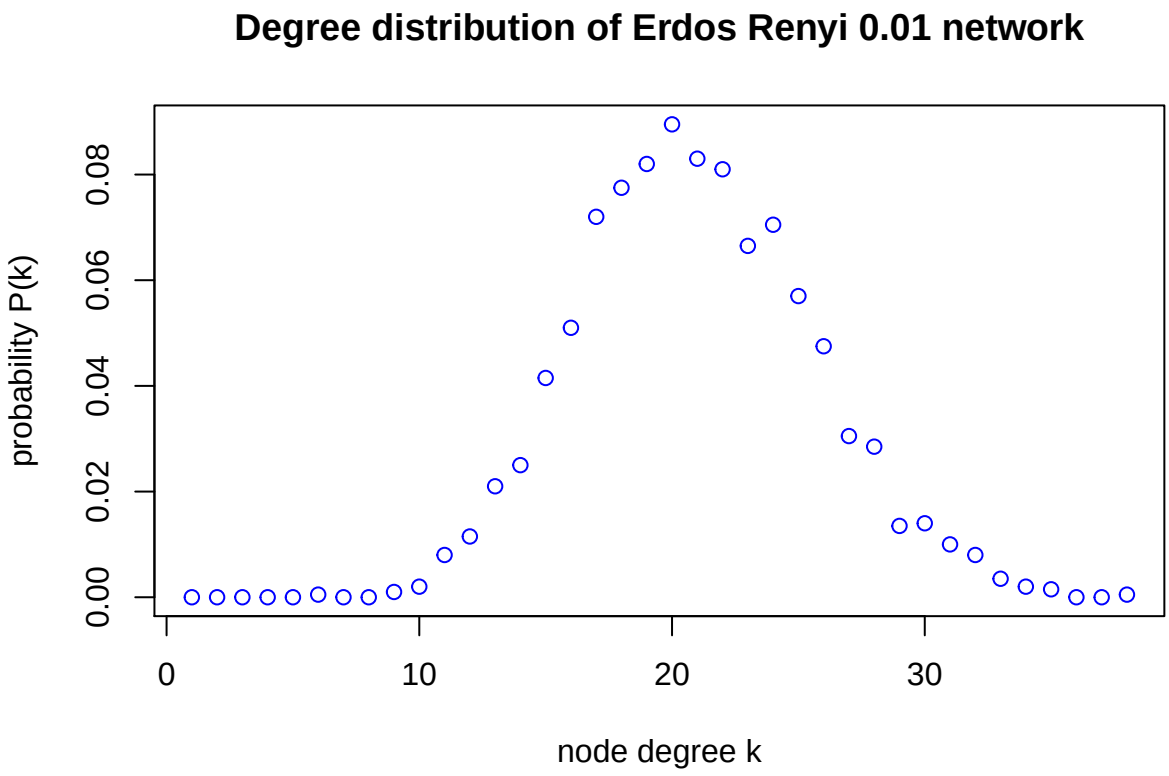

```
#Higgs Twitter-Men-Net network
plot(degree.distribution(higgsMenNetworkGraph),
     main = "Degree distribution of Higgs Twitter-Men-Net network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```



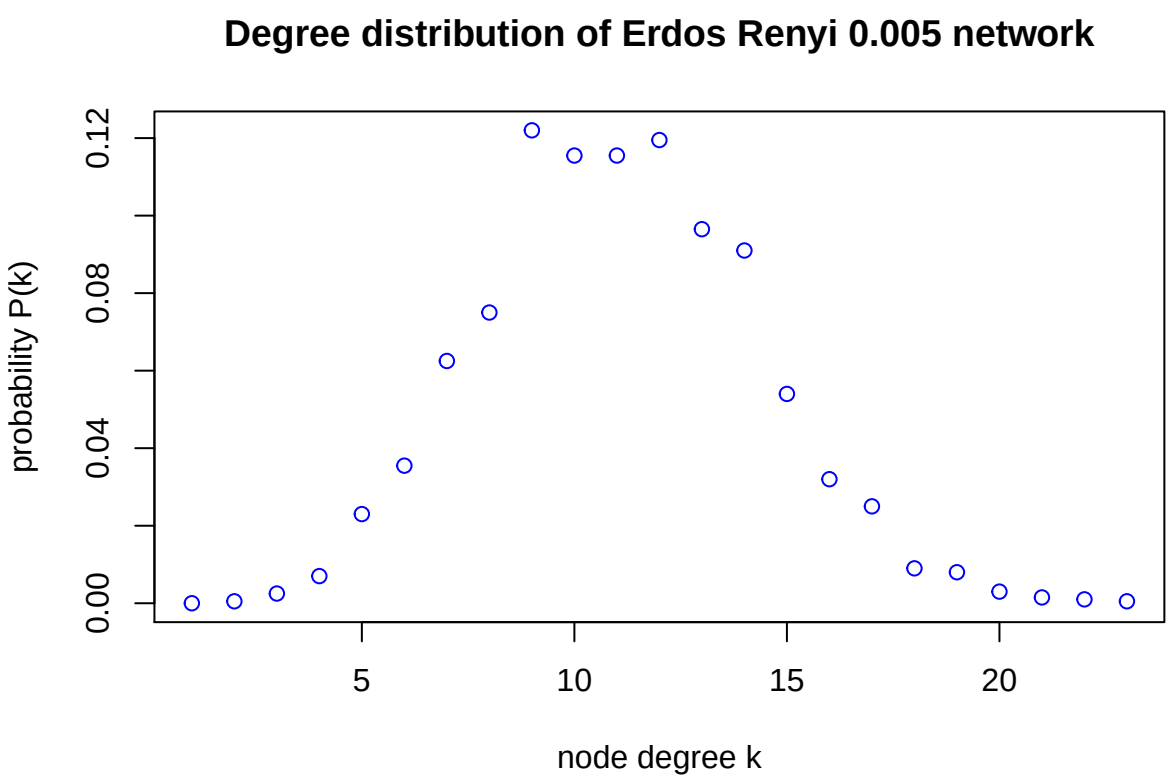

```
#Higgs Twitter-Retweet-Net network
plot(degree.distribution(higgsRetweetNetworkGraph),
     main = "Degree distribution of Higgs Twitter-Retweet-Net network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```



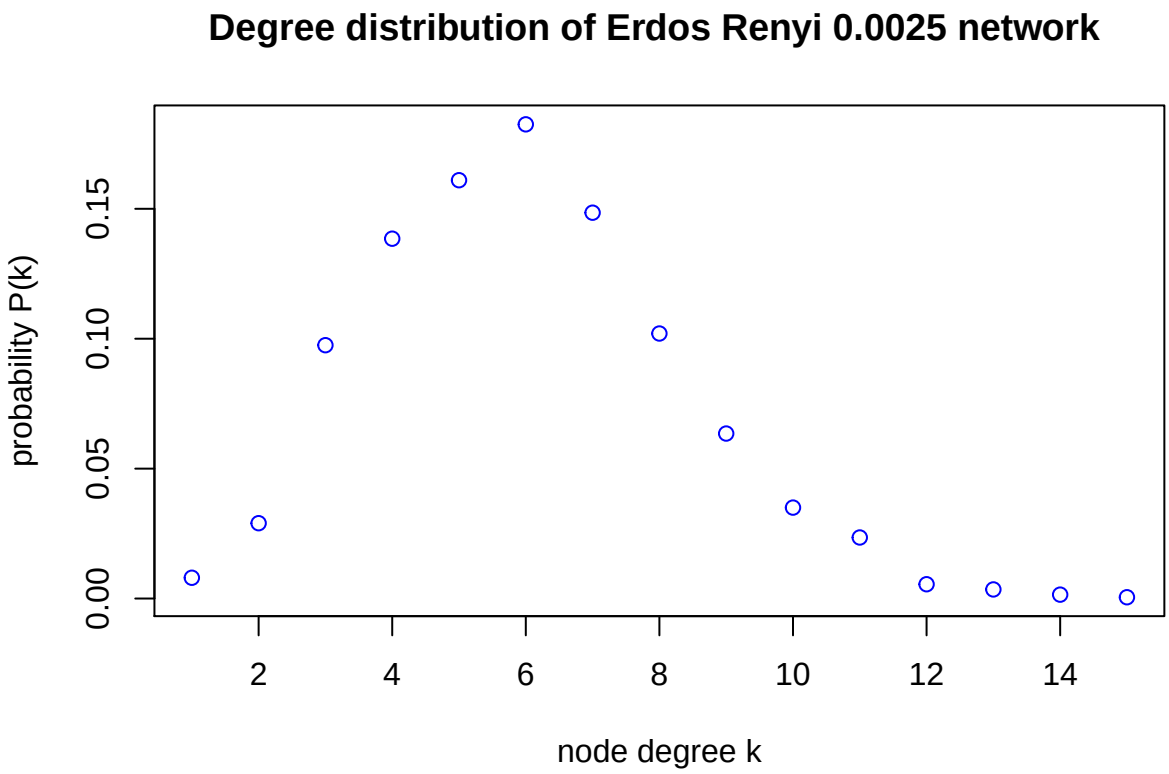

```
#Erdos Renyi 0.01 network
plot(degree.distribution(erdosRenyi01Graph),
     main = "Degree distribution of Erdos Renyi 0.01 network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```




```
#Erdos Renyi 0.005 network
plot(degree.distribution(erdosRenyi005Graph),
     main = "Degree distribution of Erdos Renyi 0.005 network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```




```
#Erdos Renyi 0.0025 network
plot(degree.distribution(erdosRenyi0025Graph),
     main = "Degree distribution of Erdos Renyi 0.0025 network ",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
 )
```



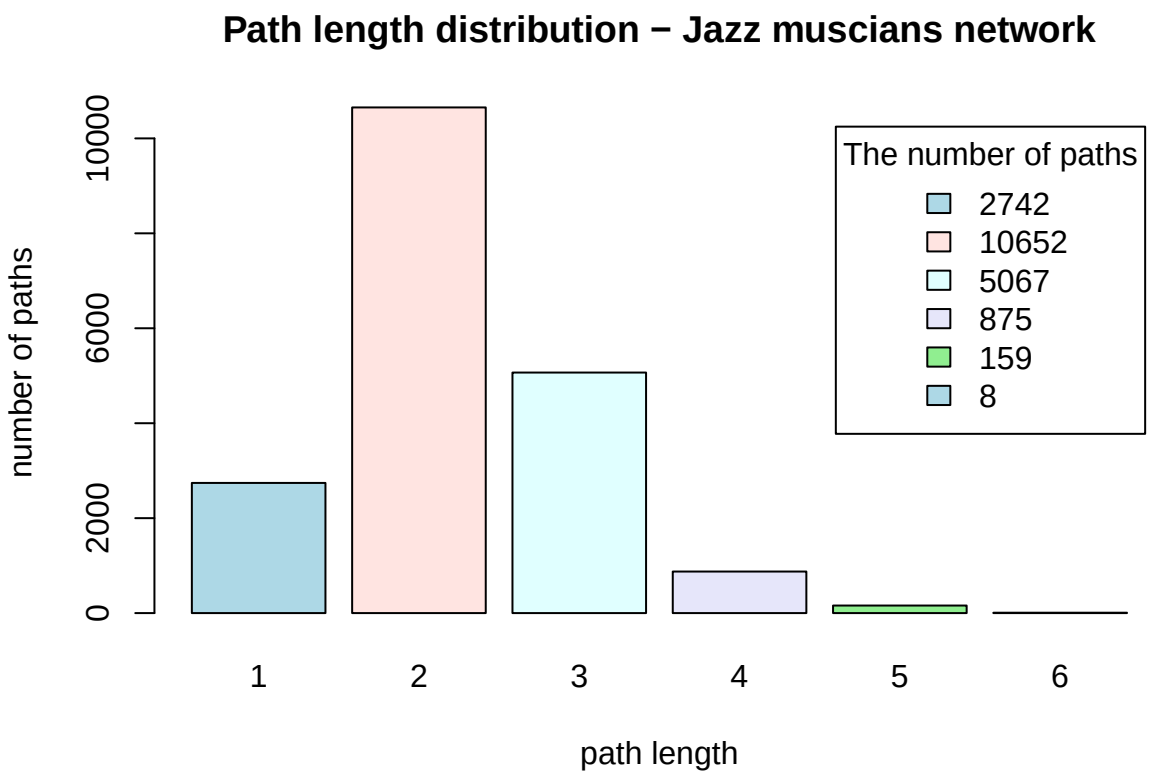
Observation:

From the above degree distribution graphs, a power law distribution is observed in the graphs "Jazz music network, C.elegans metabolic network and Higgs Networks(Social, Retweet, Twitter)", which means that there are many nodes with only few links at the top of the left corner in the plot and at the right bottom of the plot there are few hubs with large number of links as they are real world networks. While a bell cure distribution is observed in the three Erdos-Renyi random networks in which most of the nodes have the same number of links in the middle and there is no high connected nodes at the bottom of the bell curve distribution as they are random genrated networks.

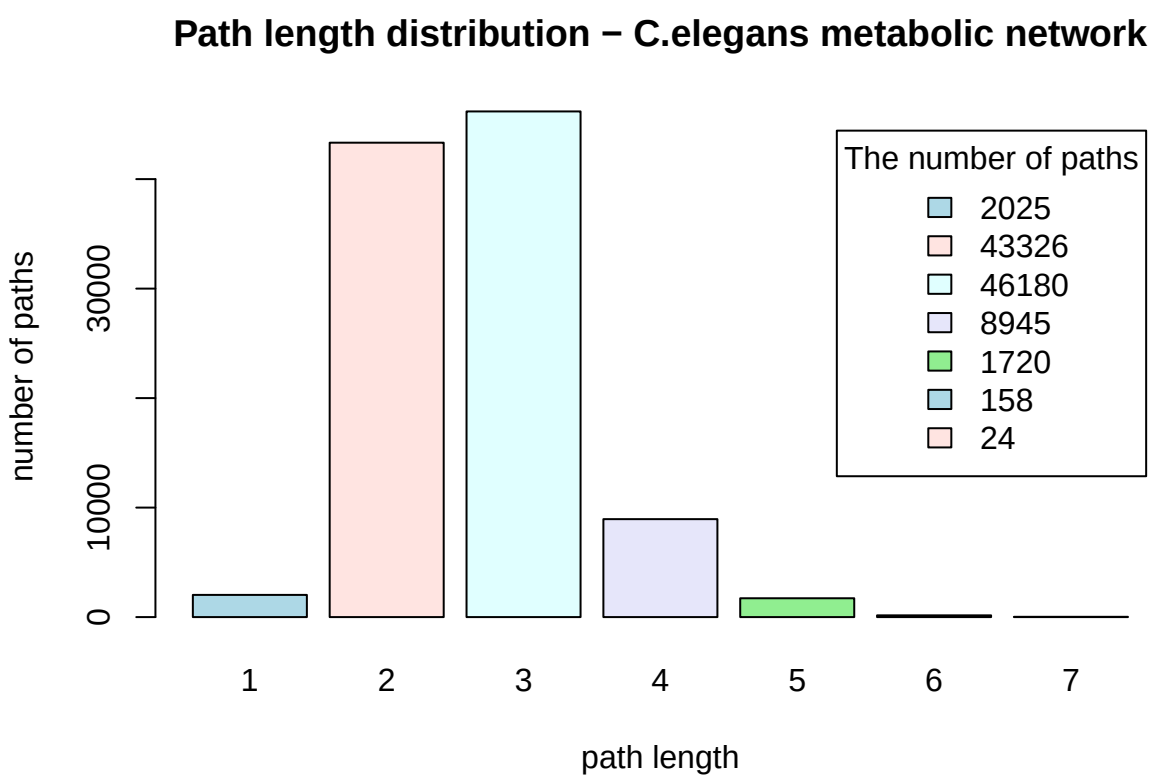
Ref : https://www2.cs.duke.edu/courses/spring14/compsci290/lectures/09-graph.pdf

Question-3

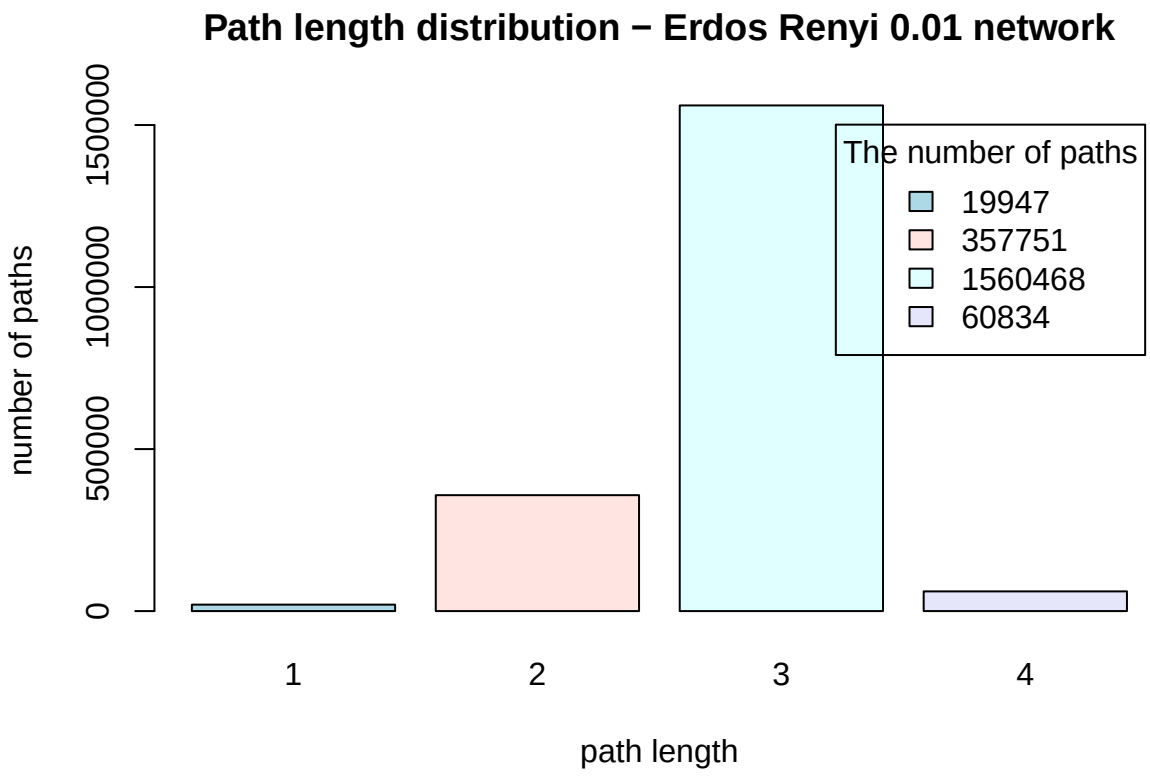
```
#Jazz muscians network
d <- as.table(path.length.hist(jazzNetGraph)$res)
names(d) <- 1 : length(d)
barplot(d,
        main = "Path length distribution - Jazz muscians network",
        legend = c(path.length.hist(jazzNetGraph)$res),
        xlab="path length",
        ylab = "number of paths",
        col = c("lightblue", "mistyrose", "lightcyan","lavender", "lightgreen"),args.legend = list(titl
        names.arg=1:length(d),
        horiz = FALSE
        )
```



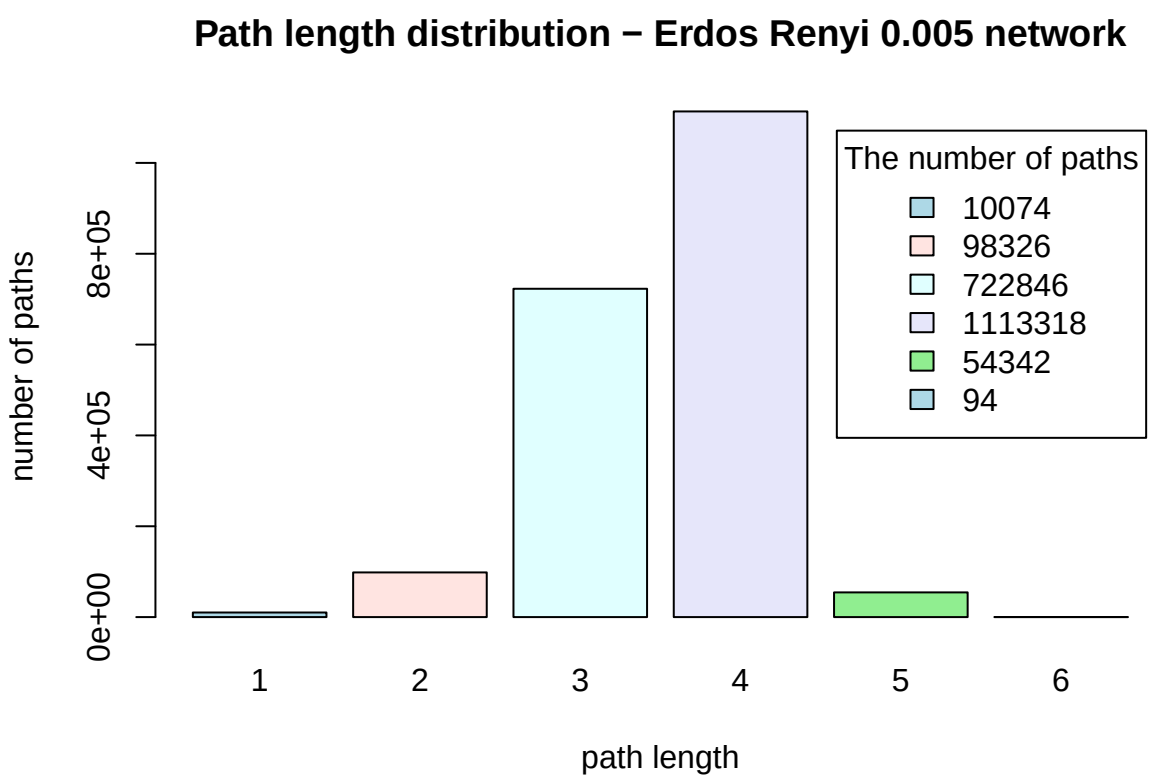

```r
# ref - https://stackoverflow.com/questions/16083522/how-to-create-a-path-length-hist-diagram
# ref - https://rstudio-pubs-static.s3.amazonaws.com/74249_3f838d8d009a417f865d8b431f53f986.html

#C.elegans metabolic network
d <- as.table(path.length.hist(celegansMetabolicGraph)$res)
names(d) <- 1 : length(d)
barplot(d,
        main = "Path length distribution - C.elegans metabolic network",
        legend = c(path.length.hist(celegansMetabolicGraph)$res),
        xlab="path length",
        ylab = "number of paths",
        col = c("lightblue", "mistyrose", "lightcyan","lavender", "lightgreen"),args.legend = list(titl
        names.arg=1:length(d),
        horiz = FALSE
        )
```



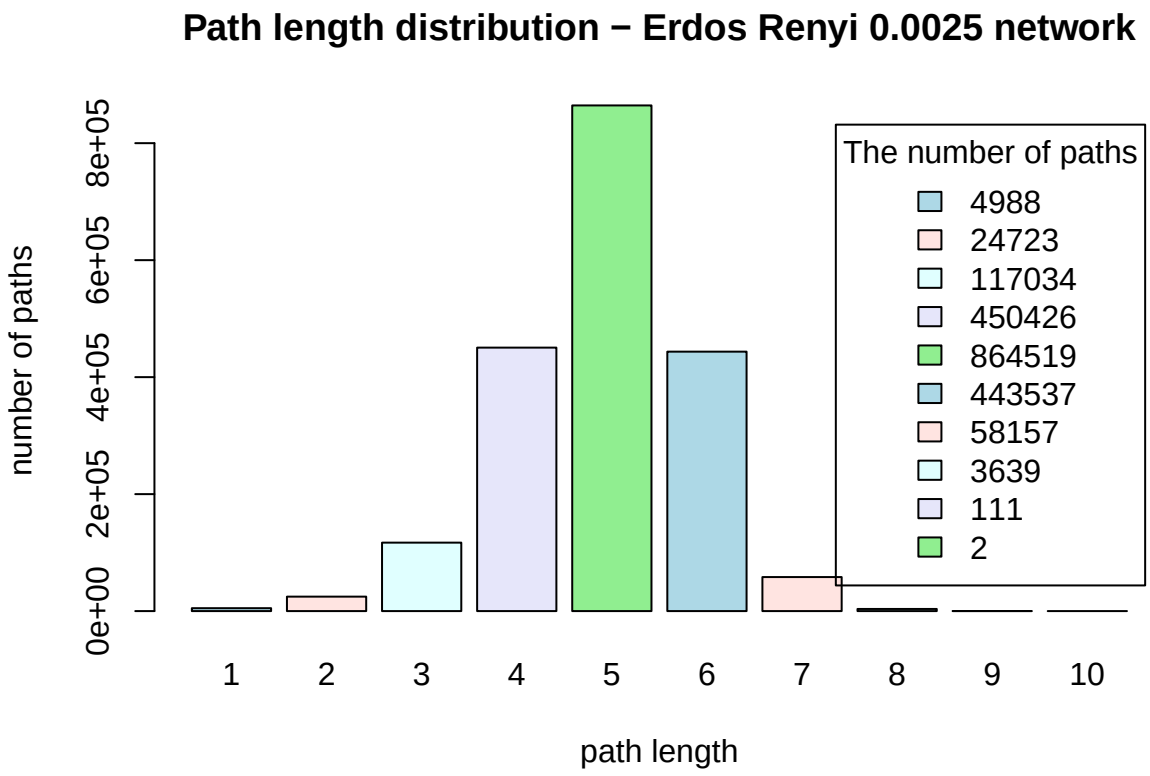

```
#Erdos Renyi 0.01 network
d <- as.table(path.length.hist(erdosRenyi01Graph)$res)
names(d) <- 1 : length(d)
barplot(d,
        main = "Path length distribution - Erdos Renyi 0.01 network",
        legend = c(path.length.hist(erdosRenyi01Graph)$res),
        xlab="path length",
        ylab = "number of paths",
        col = c("lightblue", "mistyrose", "lightcyan","lavender", "lightgreen"),args.legend = list(titl
        names.arg=1:length(d),
        horiz = FALSE
        )
```




```
#Erdos Renyi 0.005 network
d <- as.table(path.length.hist(erdosRenyi005Graph)$res)
names(d) <- 1 : length(d)
barplot(d,
        main = "Path length distribution - Erdos Renyi 0.005 network",
        legend = c(path.length.hist(erdosRenyi005Graph)$res),
        xlab="path length",
        ylab = "number of paths",
        col = c("lightblue", "mistyrose", "lightcyan","lavender", "lightgreen"),args.legend = list(titl
        names.arg=1:length(d),
        horiz = FALSE
        )
```




```r
#Erdos Renyi 0.0025 network
d <- as.table(path.length.hist(erdosRenyi0025Graph)$res)
names(d) <- 1 : length(d)
barplot(d,
        main = "Path length distribution - Erdos Renyi 0.0025 network",
        legend = c(path.length.hist(erdosRenyi0025Graph)$res),
        xlab="path length",
        ylab = "number of paths",
        col = c("lightblue", "mistyrose", "lightcyan","lavender", "lightgreen"),args.legend = list(titl
        names.arg=1:length(d),
        horiz = FALSE
        )
```



Observation:

It can be observed that all the path length distribution plots follow a bell curve distribution. Here in the random graphs i.e Erdos Random network, the path lenghts have lower values when the probability is increased in the edges. And the vice versa can be observed for the same, i.e path lengths have higher values when probability is decreased.

Question-4

For this question I have chosen US airport network(2010 December) dataset, for the the network analysis. It is a network of passenger flights between airports in the United States. It consists of 2010 December data set. It is a directed network where the edges correspond to the fligt directions. In this data set, the nodes denote the airports and the edges denote the flight carriers.

Source : https://www.transportation.gov/research-technology and Igraph library dataset.

```
library(igraphdata)
```

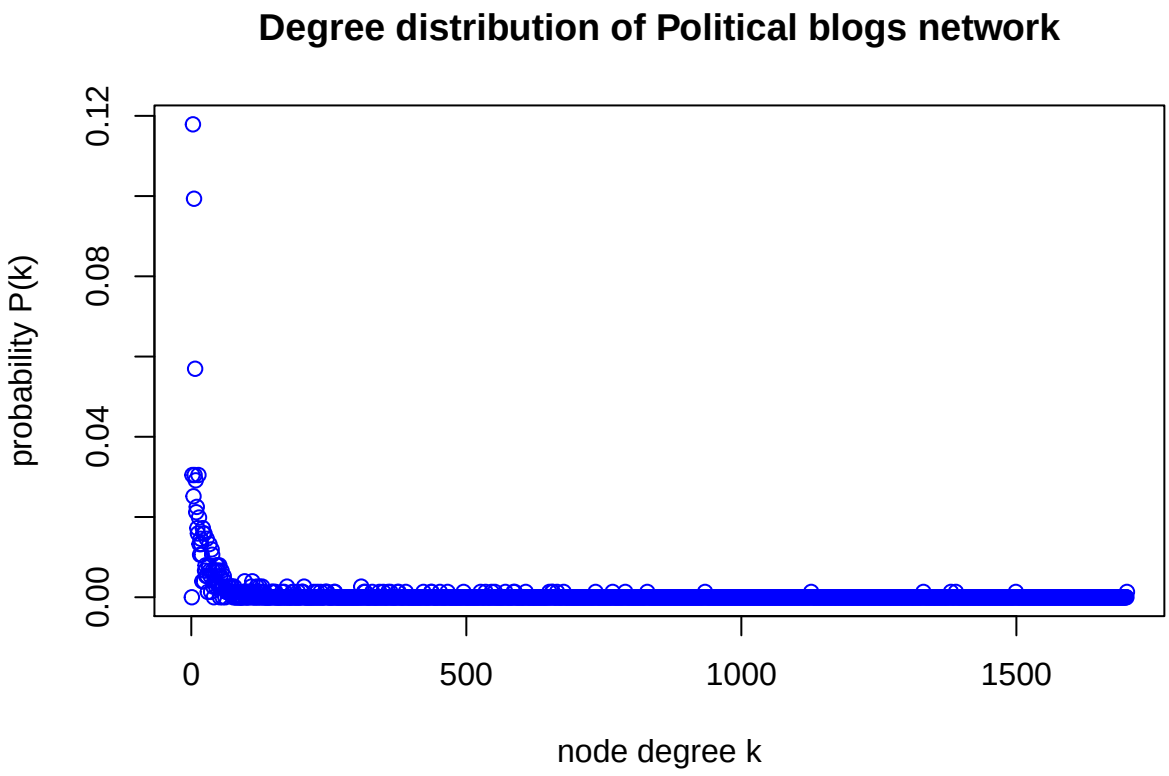
```
## Warning: package 'igraphdata' was built under R version 3.5.2
```

```
data('USairports')

#Create a Data Frame as " Network Type n m c d l L ccl ccg "
usaAirportDataFrame <- data.frame("Network" = character(1),
                  "Type" = character(1),
                  "n" = integer(1),
                  "m" = integer(1),
                  "cStrong" =integer(1),
```

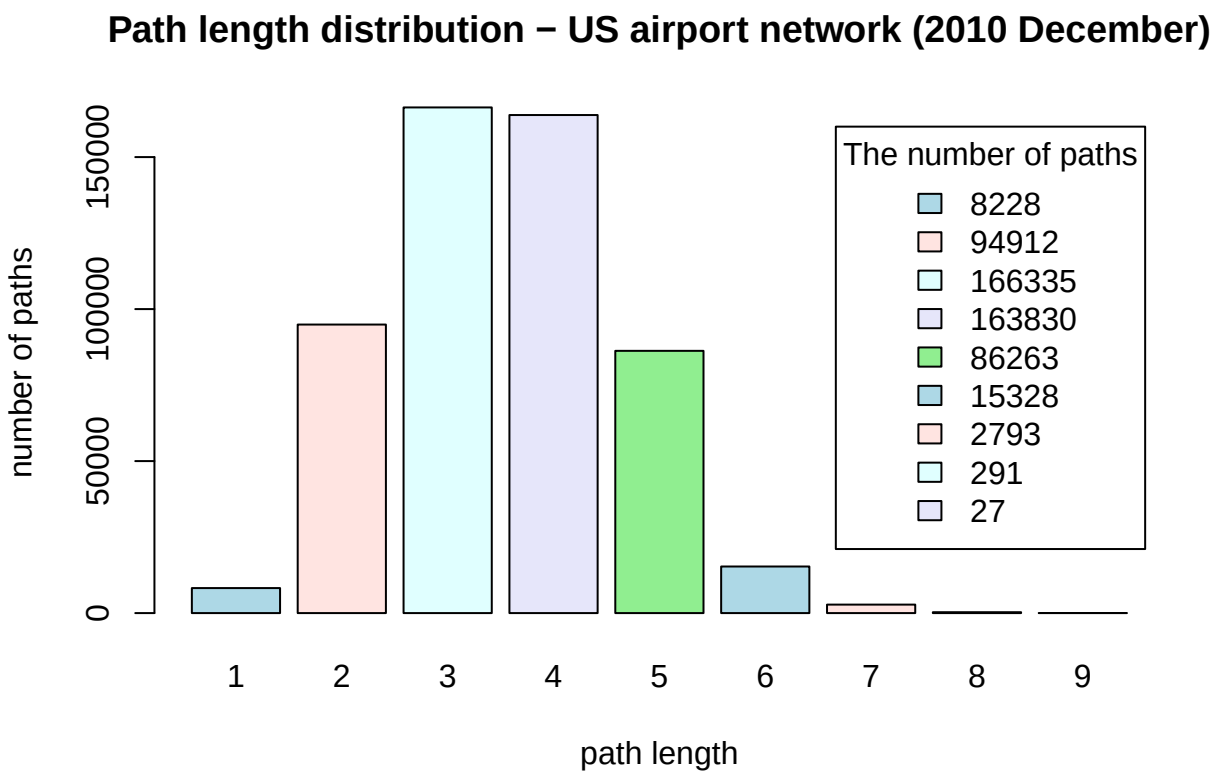
```
                  "cWeak" = integer(1),
                  "d" = integer(1),
                  "l" = double(1),
                  "L" = integer(1),
                  "ccL" = double(1),
                  "ccG" = double(1),
                  stringsAsFactors = FALSE)

usaAirportDataFrame$Network[1] <- "US airport network (2010 December)"
usaAirportDataFrame$Type[1] <- if(is_directed(USairports) == TRUE) "Directed" else "Un Directed"
usaAirportDataFrame$n[1] <- vcount(USairports)
usaAirportDataFrame$m[1] <- ecount(USairports)
usaAirportDataFrame$cStrong[1] <- count_components(USairports, mode = "strong")
usaAirportDataFrame$cWeak[1] <- count_components(USairports, mode = "weak")
usaAirportDataFrame$d[1] <- max(degree(USairports))
usaAirportDataFrame$l[1] <- mean_distance(USairports)
usaAirportDataFrame$L[1] <- diameter(USairports)
usaAirportDataFrame$cc_local[1] <- transitivity(USairports, type = "localaverageundirected")
usaAirportDataFrame$cc_global[1] <- transitivity(USairports, type = "globalundirected")


kable(usaAirportDataFrame, format = "markdown")
```

Network	Type	n	m	cStrong	cWeak	d	l	L	ccL	ccG	cc_local	cc_global
US airport network (2010 December)	Directed	755	23473	30	6	1700	3.52743	9	0	0	0.6452844	0.3384609

```
#US airport network (2010 December)
plot(degree.distribution(USairports),
     main = "Degree distribution of Political blogs network",
     xlab="node degree k",
     ylab = "probability P(k)",
     col="blue"
     )
```




```r
#US airport network (2010 December)
d <- as.table(path.length.hist(USairports)$res)
names(d) <- 1 : length(d)
barplot(d,
        legend = c(path.length.hist(USairports)$res),
        main = "Path length distribution - US airport network (2010 December)",
        xlab="path length",
        ylab = "number of paths",
        col = c("lightblue", "mistyrose", "lightcyan","lavender", "lightgreen"),args.legend = list(titl
        names.arg=1:length(d),
        horiz = FALSE
        )
```


Path length distribution − US airport network (2010 December)
number of paths
0
50000
100000
150000
1
2
3
4
5
6
7
8
9
path length
The number of paths
8228
94912
166335
163830
86263
15328
2793
291
27