Elements of Network Science : Assignment1

Sharath Kumar Karnati

2024-02-08

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

```
library(igraph)
```

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

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

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

```
g1=read_graph("C:/Users/shara/OneDrive/Desktop/elemts/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;).
```

```
g2=read_graph("C:/Users/shara/OneDrive/Desktop/elemts/celegansneural/celegansneural.gml","gml")
g3=read_graph("C:/Users/shara/OneDrive/Desktop/elemts/as-22july06/as-22july06.gml","gml")
g4 <- erdos.renyi.game (2000, 0.01)
g5 <- erdos.renyi.game (2000, 0.005)
g6 <- erdos.renyi.game (2000, 0.0025)
#no.of nodes
v1=gorder(g1)
v2=gorder(g2)
v3=gorder(g3)
v4=gorder(g4)
v5=gorder(g5)
v6=gorder(g6)
#no.of links
m1=gsize(g1)
m2=gsize(g2)
m3=gsize(g3)
m4=gsize(g4)
m5=gsize(g5)
m6=gsize(g6)
#type of graph
is_directed(g1)
```

```
## [1] TRUE
```

```
is_directed(g2)
```

```
## [1] TRUE
```

```
is_directed(g3)
```

```
## [1] FALSE
```

```
is_directed(g4)
```

```
## [1] FALSE
```

```
is_directed(g5)
```

```
## [1] FALSE
```

```
is_directed(g6)
```

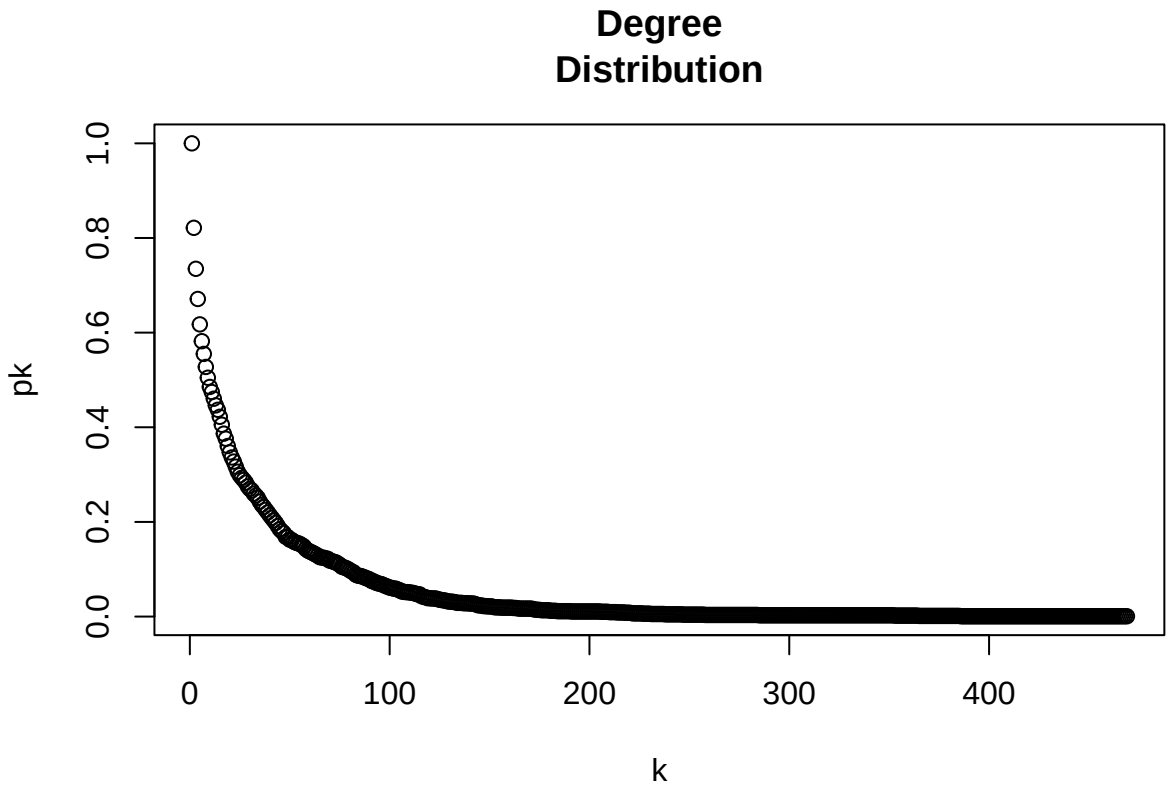
```
## [1] FALSE
```

```
#strong or weak
c1_strong= count_components(g1, mode = c("strong"))
c1_weak= count_components(g1, mode = c("weak"))
c2_strong= count_components(g2, mode = c("strong"))
c2_weak= count_components(g2, mode = c("weak"))
c3= count_components(g3)
c4= count_components(g4)
c5= count_components(g5)
#degree
d1=max(degree(g1))
d2=max(degree(g2))
d3=max(degree(g3))
d4=max(degree(g4))
d5=max(degree(g5))
d6=max(degree(g6))
#average path length
l1= mean_distance(g1)
l2= mean_distance(g2)
l3= mean_distance(g3)
l4= mean_distance(g4)
l5= mean_distance(g5)
l6= mean_distance(g6)
#Diameter
L1 =diameter(g1)
L2 =diameter(g2)
L3 =diameter(g3)
L4 =diameter(g4)
L5 =diameter(g5)
L6 =diameter(g6)
#average local clustering coefficient
ccl1=transitivity(g1, type = "localaverage")
ccl2=transitivity(g2, type = "localaverage")
ccl3=transitivity(g3, type = "localaverage")
ccl4=transitivity(g4, type = "localaverage")
ccl5=transitivity(g5, type = "localaverage")
ccl6=transitivity(g6, type = "localaverage")
#global clustering coefficient
ccg1=transitivity(g1, type = "global")
ccg2=transitivity(g2, type = "global")
```

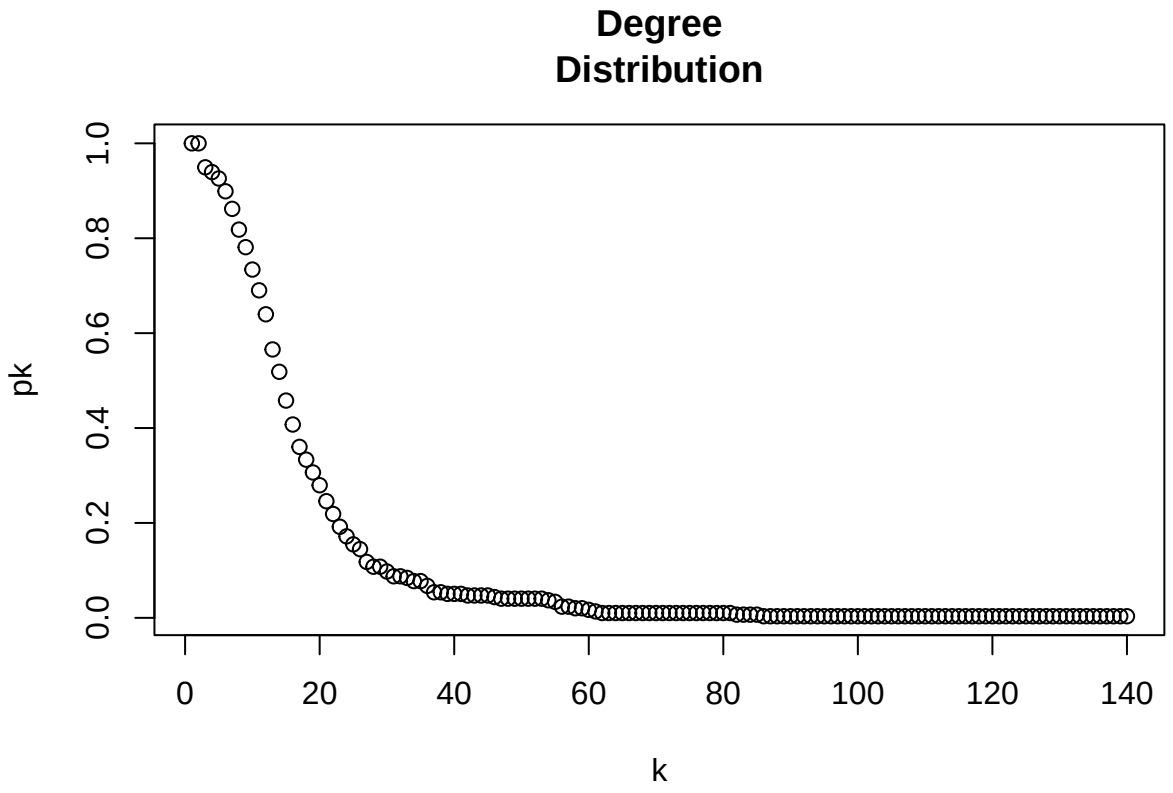
```r
ccg3=transitivity(g3, type = "global")
ccg4=transitivity(g4, type = "global")
ccg5=transitivity(g5, type = "global")
ccg6=transitivity(g6, type = "global")
graph_data <- data.frame(
  Graph = c("polblogs", "celegansneural", "as-22july06", "erdos_renyi_0.01", "erdos_renyi_0.005", "erdos
  Nodes = c(v1, v2, v3, v4, v5, v6),
  Links = c(m1, m2, m3, m4, m5, m6),
  Directed = c(is_directed(g1), is_directed(g2), is_directed(g3), is_directed(g4), is_directed(g5), is_
  StrongComponents = c(c1_strong, c2_strong, NA, NA, NA, NA), # NA for non-directed graphs
  WeakComponents = c(c1_weak, c2_weak, c3, NA, NA, NA), # NA for non-directed graphs
  MaxDegree = c(d1, d2, d3, d4, d5, d6),
  AvgPathLength = c(l1, l2, l3, l4, l5, l6),
  Diameter = c(L1, L2, L3, L4, L5, L6),
  LocalClusteringCoefficient = c(ccl1, ccl2, ccl3, ccl4, ccl5, ccl6),
  GlobalClusteringCoefficient = c(ccg1, ccg2, ccg3, ccg4, ccg5, ccg6)
)

# Display the table
print(graph_data)
```

```
##                Graph Nodes Links Directed StrongComponents WeakComponents
## 1           polblogs  1490 19090     TRUE              688            268
## 2     celegansneural   297  2359     TRUE               57              1
## 3        as-22july06 22963 48436    FALSE               NA              1
## 4   erdos_renyi_0.01  2000 20067    FALSE               NA             NA
## 5  erdos_renyi_0.005  2000 10005    FALSE               NA             NA
## 6 erdos_renyi_0.0025  2000  5008    FALSE               NA             NA
##   MaxDegree AvgPathLength Diameter LocalClusteringCoefficient
## 1       468      3.390184        9                0.360028652
## 2       139      3.991884       14                0.307914537
## 3      2390      3.842426       11                0.349915358
## 4        35      2.828409        4                0.010142245
## 5        22      3.559600        6                0.004530896
## 6        14      4.879600        9                0.003571504
##   GlobalClusteringCoefficient
## 1                 0.225958517
## 2                 0.180711471
## 3                 0.011146384
## 4                 0.010224929
## 5                 0.004583308
## 6                 0.003000840
```

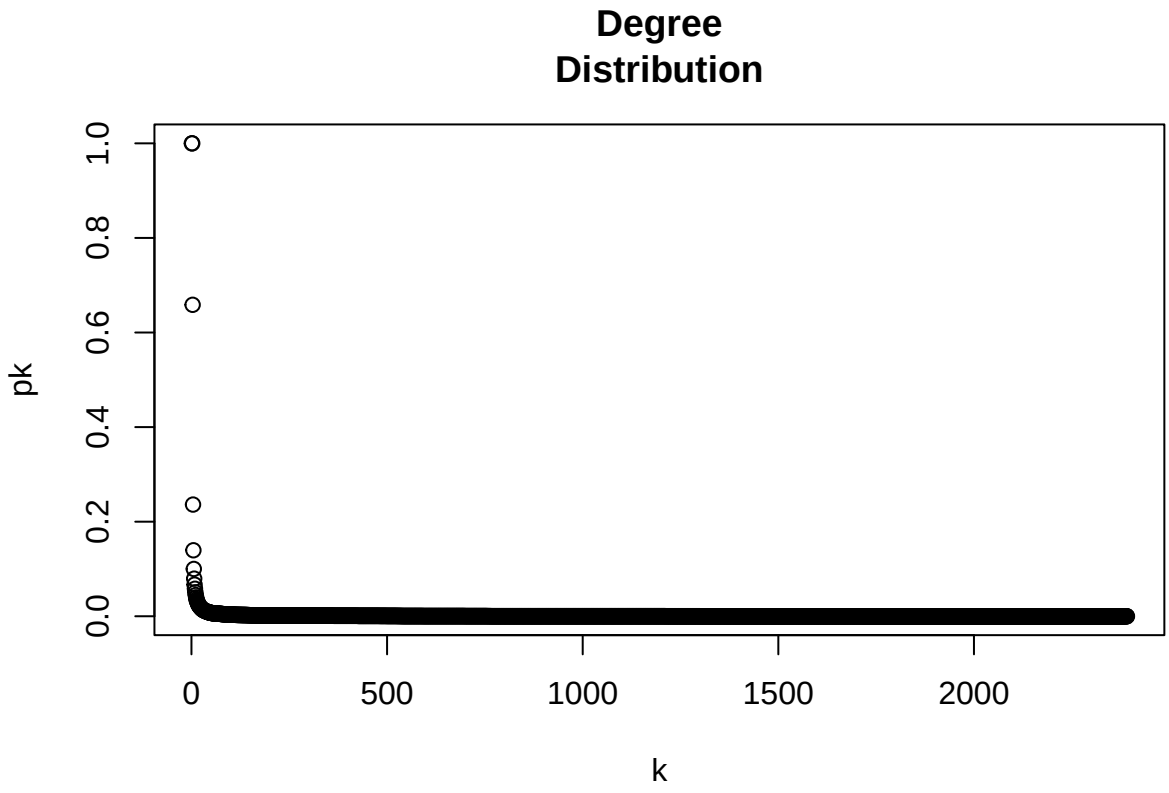
```r
#Question : 2

#degree distibuiton
plot(degree_distribution(g1, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```



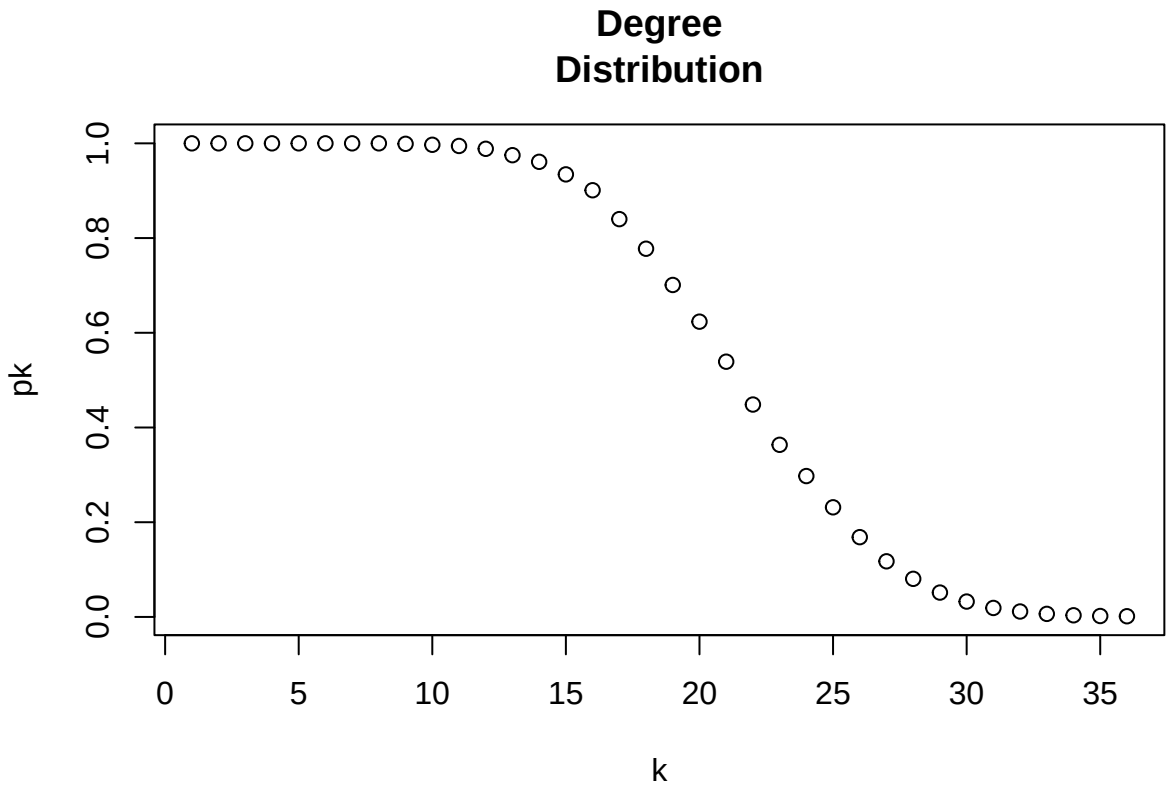

```
plot(degree_distribution(g2, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```



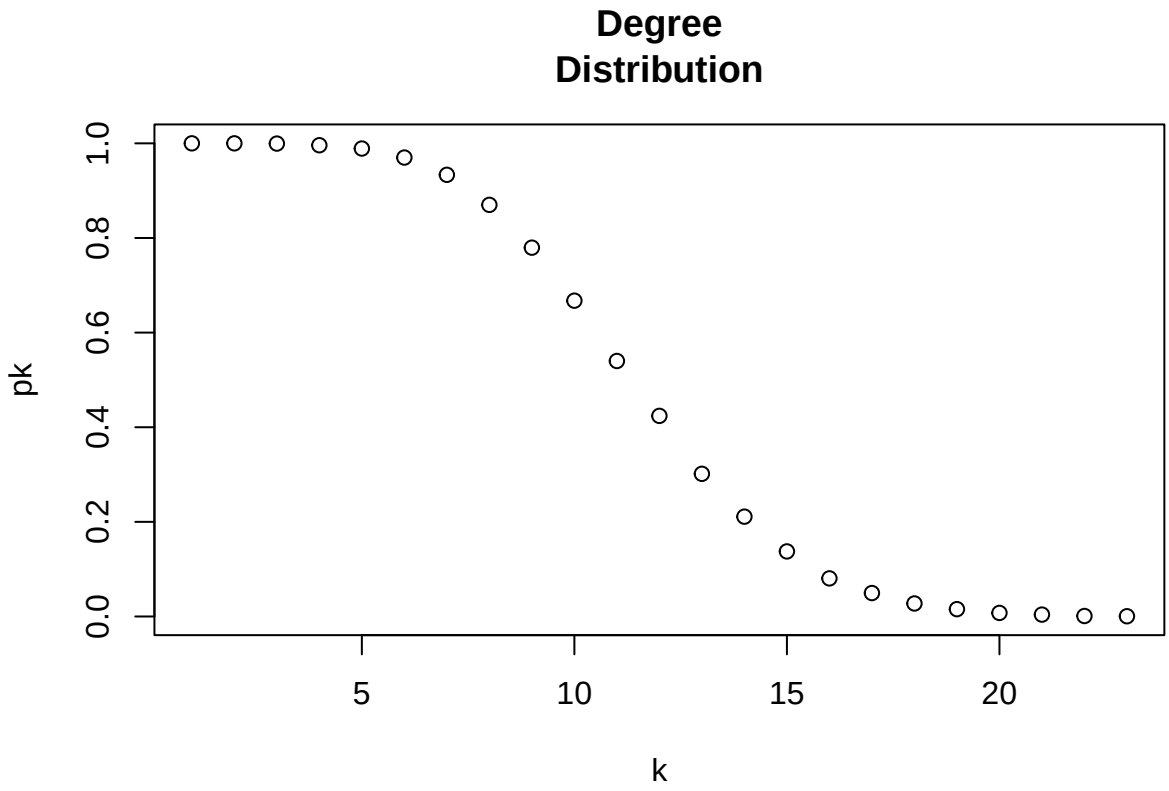

```
plot(degree_distribution(g3, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```



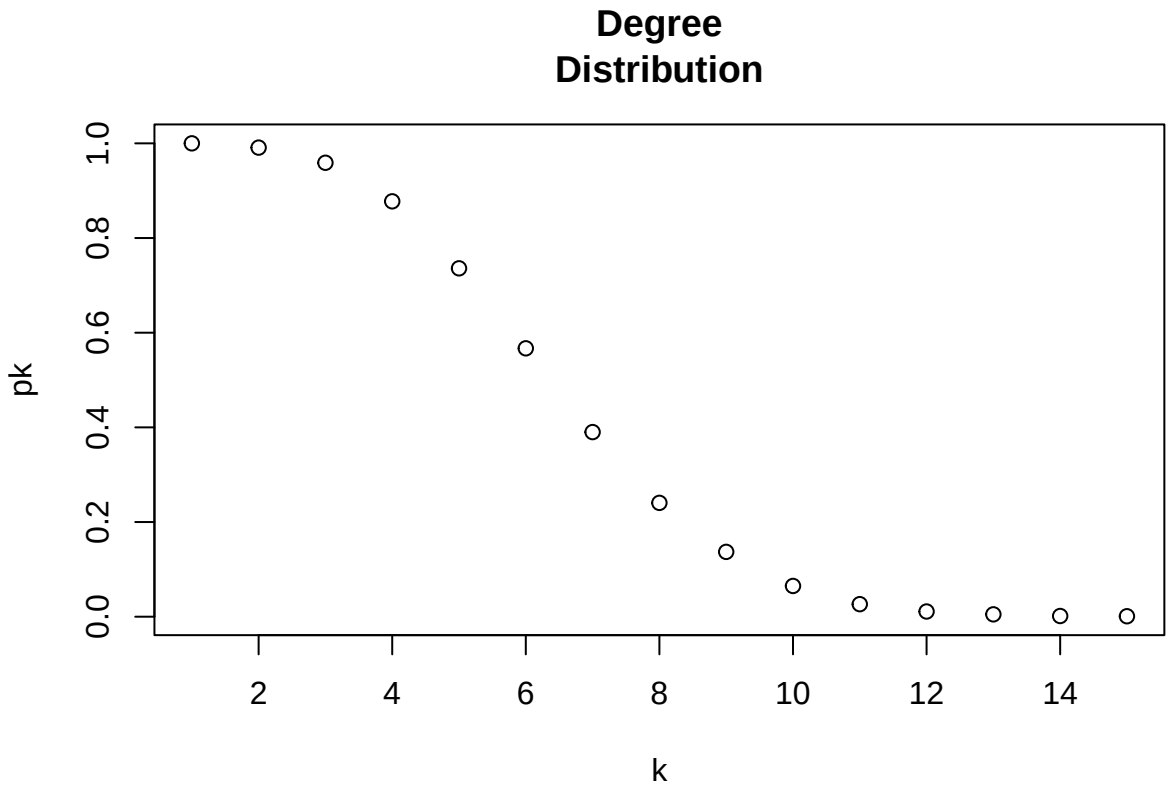

```
plot(degree_distribution(g4, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```




```
plot(degree_distribution(g5, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```




```
plot(degree_distribution(g6, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```



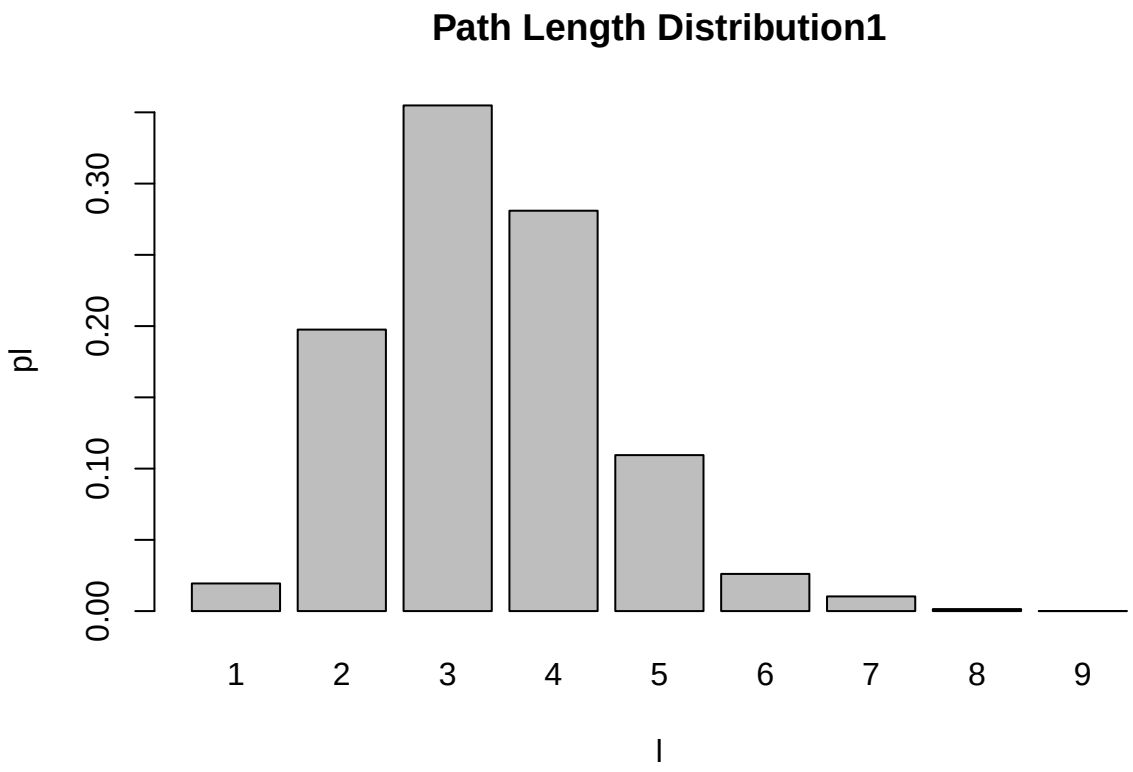
"Observation: In Graphs 1, 2, and 3, the probability that a randomly chosen node has degree k initially decreases as k increases. However, beyond a certain point, the value of pk stabilizes and remains constant as k continues to increase.

On the other hand, in Graphs 4, 5, and 6, the probability that a randomly chosen node has degree k is constant for the first half of the degree distribution. Subsequently, the value of pk decreases, and then it stabilizes once again, maintaining a constant value as k further increases."

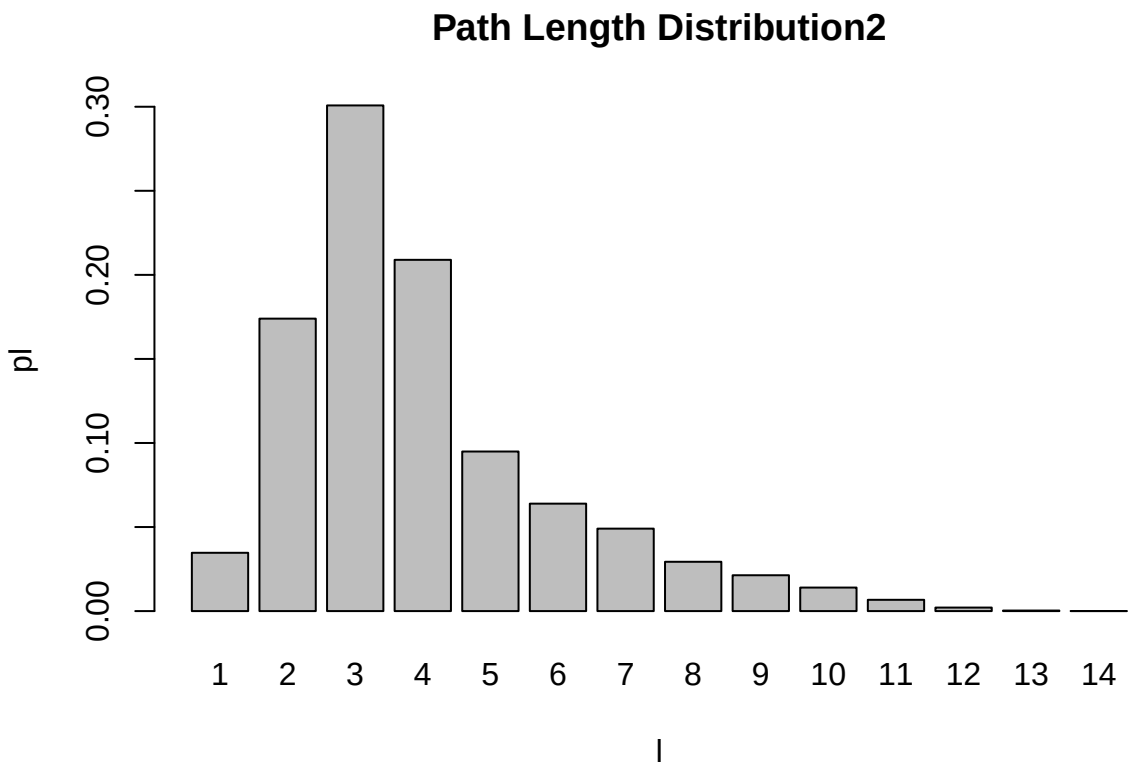
```
# Question : 3
#Path length Distribution
table1 <- as.table(path.length.hist(g1)$res)
```

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

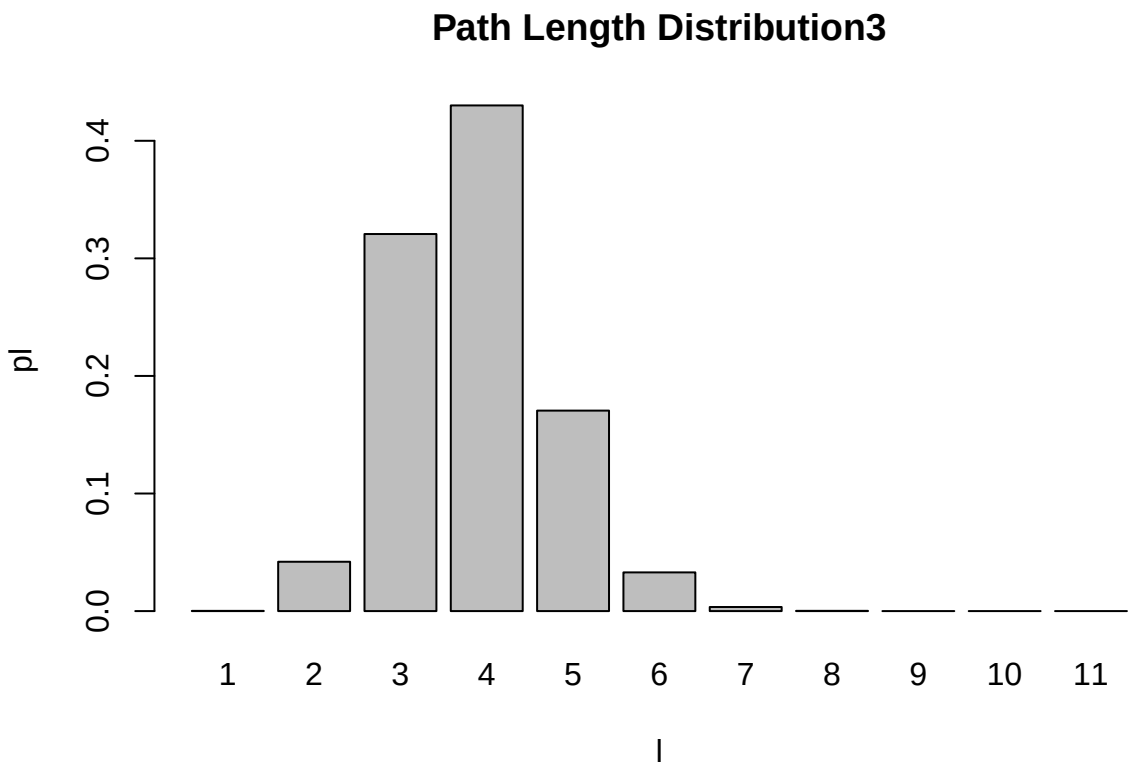
```
names(table1) <- 1:length(table1)
barplot(table1/sum(table1),xlab="l",ylab="pl",main="Path Length Distribution1")
```



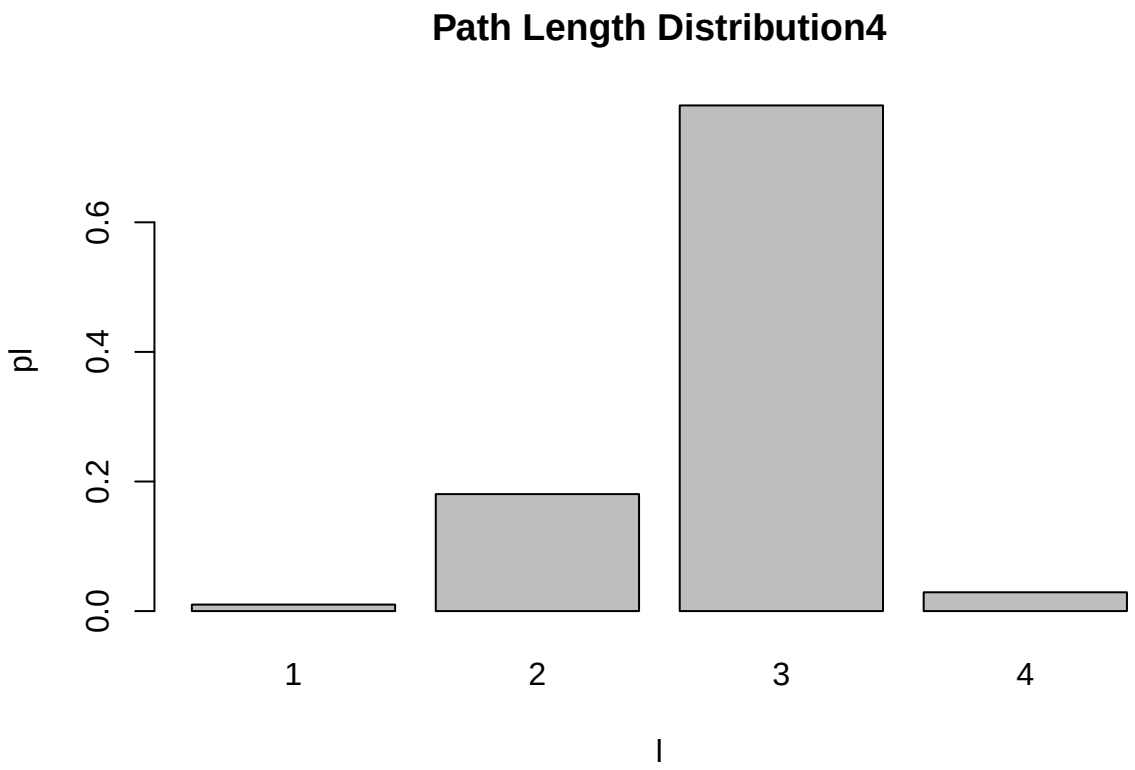

```
table2 <- as.table(path.length.hist(g2)$res)
names(table2) <- 1:length(table2)
barplot(table2/sum(table2),xlab="l",ylab="pl",main="Path Length Distribution2")
```



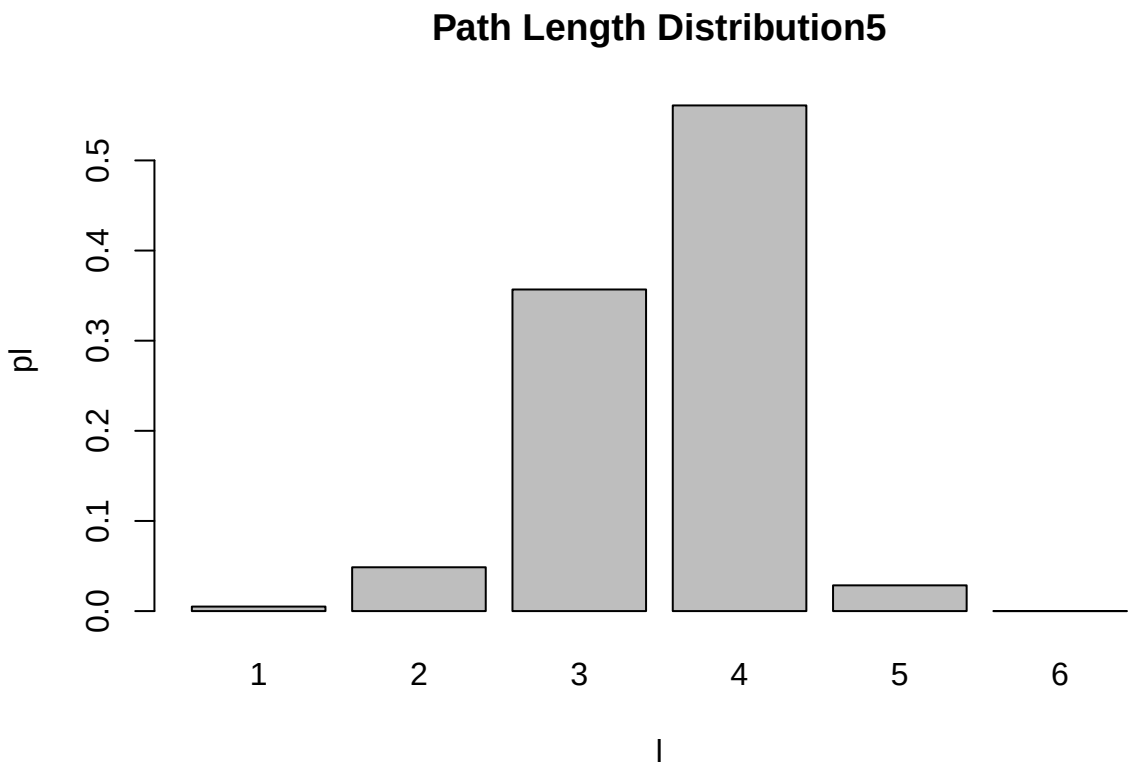

```
table3 <- as.table(path.length.hist(g3)$res)
names(table3) <- 1:length(table3)
barplot(table3/sum(table3),xlab="l",ylab="pl",main="Path Length Distribution3")
```



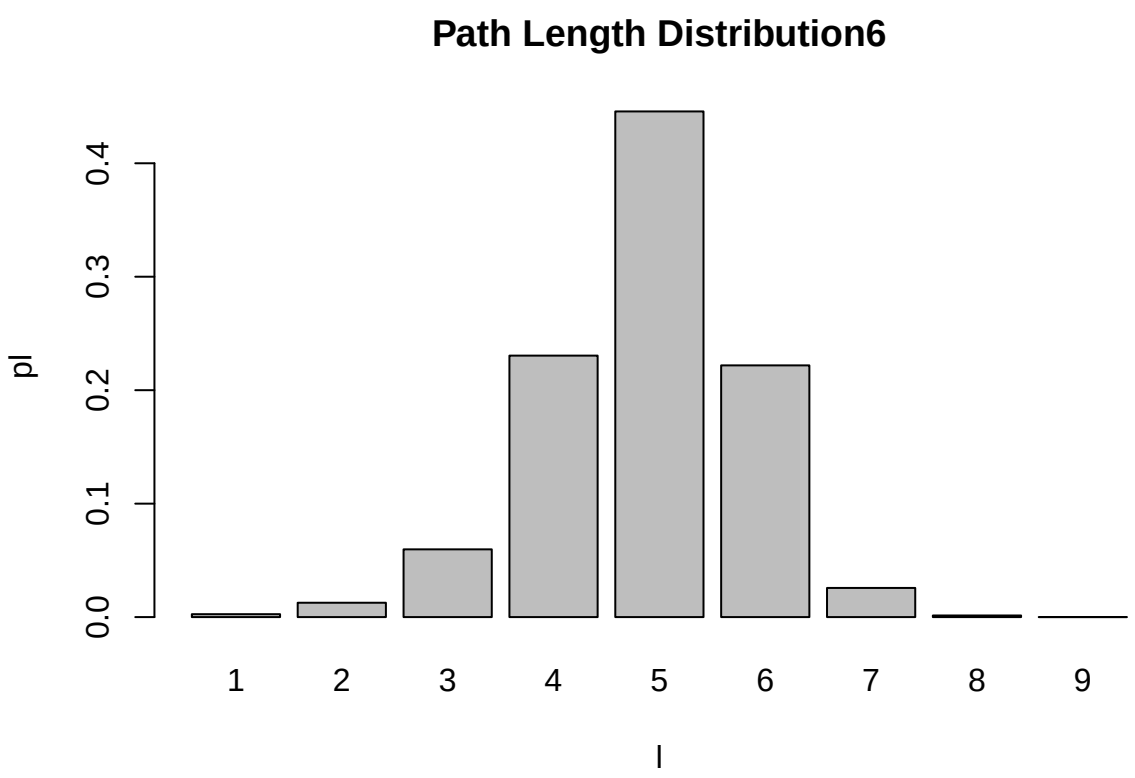

```
table4 <- as.table(path.length.hist(g4)$res)
names(table4) <- 1:length(table4)
barplot(table4/sum(table4),xlab="l",ylab="pl",main="Path Length Distribution4")
```




```
table5 <- as.table(path.length.hist(g5)$res)
names(table5) <- 1:length(table5)
barplot(table5/sum(table5),xlab="l",ylab="pl",main="Path Length Distribution5")
```




```
table6 <- as.table(path.length.hist(g6)$res)
names(table6) <- 1:length(table6)
barplot(table6/sum(table6),xlab="l",ylab="pl",main="Path Length Distribution6")
```



"Observation :

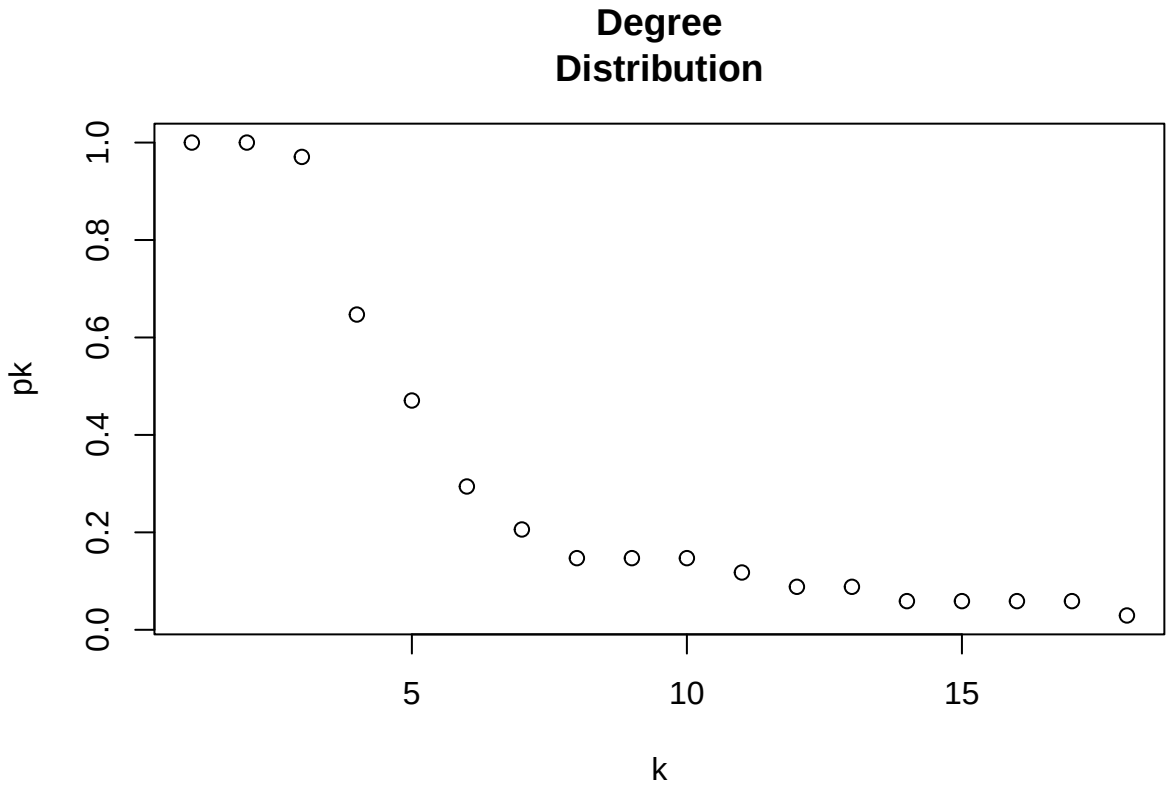
In all the graphs, the probability distribution reaches its highest point before gradually decreasing. There is an uneven distribution of the fraction of node-pairs connected by the shortest path of a specific length. Specifically, the probability of having a short path between nodes is initially low, peaks at a certain length, and then decreases gradually. Notably, Graph 6 exhibits a more balanced distribution compared to the other graphs, where the probability of different path lengths is more evenly spread."

```
#Question : 4
g=read_graph("C:/Users/shara/OneDrive/Desktop/elemts/karate/karate.gml","gml")
#no.of nodes
v=gorder(g)
# Type of the graph
is_directed(g)
```

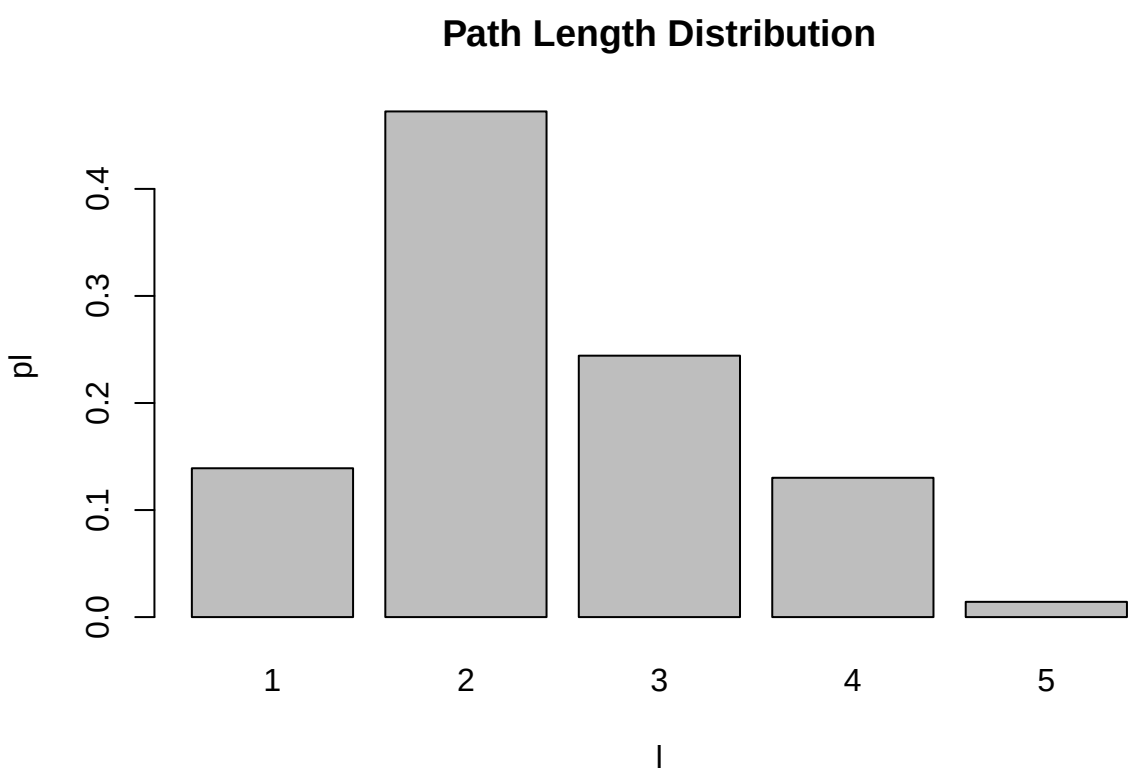
```
## [1] FALSE
```

```
# no. of Links
m=gsize(g)
#Components
c= count_components(g)
# Maximum Degree
d=max(degree(g))
# Average Path length
l= mean_distance(g)
# Diameter
L=diameter(g)
# Local Clustering coefficient
```

```
ccl=transitivity(g, type = "localaverage")
# Global Clustering coefficient
ccg=transitivity(g, type = "global")
#Degree Distribution
plot(degree_distribution(g, cumulative=TRUE),xlab="k",ylab = "pk",main = "Degree
Distribution")
```




```
#Path length Distribution
table <- as.table(path.length.hist(g)$res)
names(table) <- 1:length(table)
barplot(table/sum(table),xlab="l",ylab="pl",main="Path Length Distribution")
```



"Observation: Network Choosen : Zachary's Karate Club The probability that a randomly selected node has a degree of k gradually decreases and eventually becomes nearly constant after reaching a certain point.

The plot illustrates an uneven distribution of the fraction of node pairs connected by the shortest path of a given length. Initially, the graph rapidly ascends to a peak, but subsequently, it experiences a decline in probability, resulting in a lower overall value."