Assignment1

Datasets: the assignment involvesthreenetworks from Mark Newman's collection ofNetwork Data (http://www-personal.umich.edu/~mejn/netdata/). The networks in thecollection are given certain descriptive names. The three networks chosen for this assign-ment are called: (1) Political blogs, (2) Neural network, and (3) Internet. Read the briefdescriptions there so you get an idea for what the networks are modeling.

Your task: Download the three networks. These data sets are in GML format, which is aformat the package igraph understands. "Read" the three networks into igraph. In additionto these three networks, generate three Erdos-Renyi random networksG(n, p) using igraph,the first with the parametersn= 2000 andp= 0.01, the second with the parametersn= 2000 andp= 0.005, and the third with the parametersn= 2000 andp= 0.0025.Perform and report on the following tasks for all six networks.

1. Political Blogs

```
library("igraph")
```

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

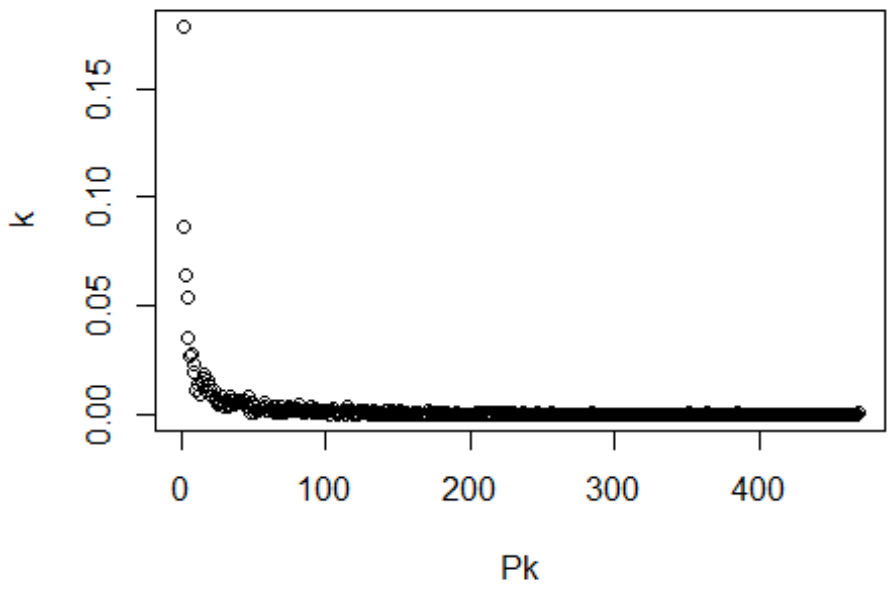
```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

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

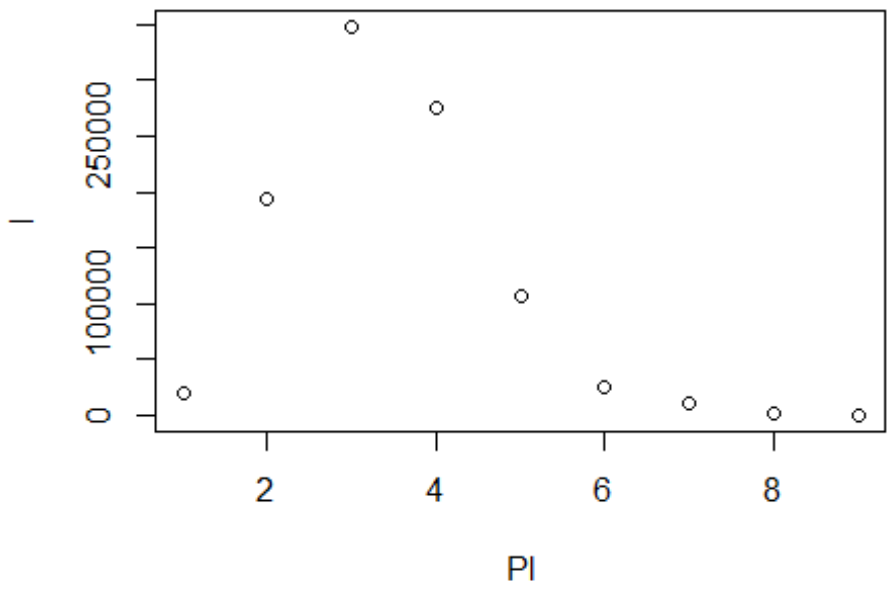
```
po <-read.graph("polblogs.gml",format=c("gml"))
graph = po
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")
```

```
## [1] "Directed:- "
```

```
print(isDirected)

## [1] TRUE

print("Number of Nodes:-" )

## [1] "Number of Nodes:-"

print(n)

## [1] 1490

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 19090

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 468

print("Average Path Length:- ")

## [1] "Average Path Length:- "

print(l)

## [1] 3.390184

print("Diameter:- ")

## [1] "Diameter:- "

print(D)

## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>

print("Average Global Clustering Coefficient:- ")

## [1] "Average Global Clustering Coefficient:- "

print(ccg)

## [1] 0.2259585

plot(degree.distribution(graph),xlab="Pk", ylab="k")
```



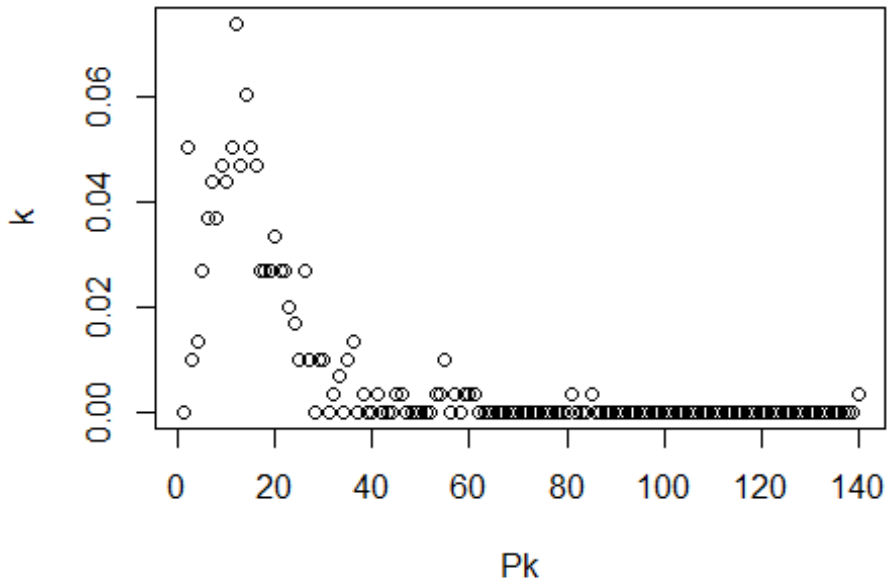

```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



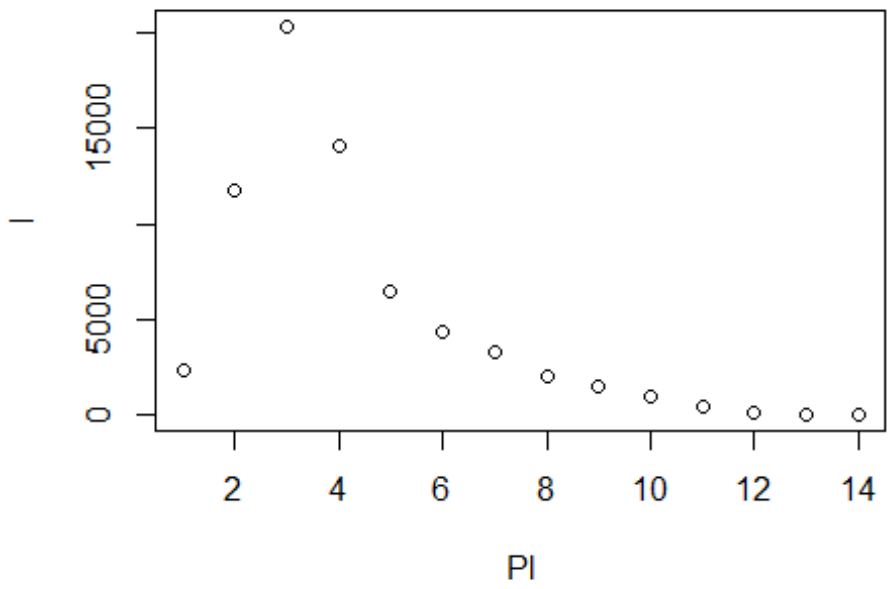
2. Neural Network

```
nn <-read.graph("celegansneural.gml",format=c("gml"))

graph = nn
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")

## [1] "Directed:- "

print(isDirected)

## [1] TRUE

print("Number of Nodes:-" )

## [1] "Number of Nodes:-"

print(n)

## [1] 297

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 2359

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 139

print("Average Path Length:- ")

## [1] "Average Path Length:- "
```

```
print(l)

## [1] 3.991884

print("Diameter:- ")

## [1] "Diameter:- "

print(D)

## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>

print("Average Global Clustering Coefficient:- ")

## [1] "Average Global Clustering Coefficient:- "

print(ccg)

## [1] 0.1807115

plot(degree.distribution(graph),xlab="Pk", ylab="k")
```



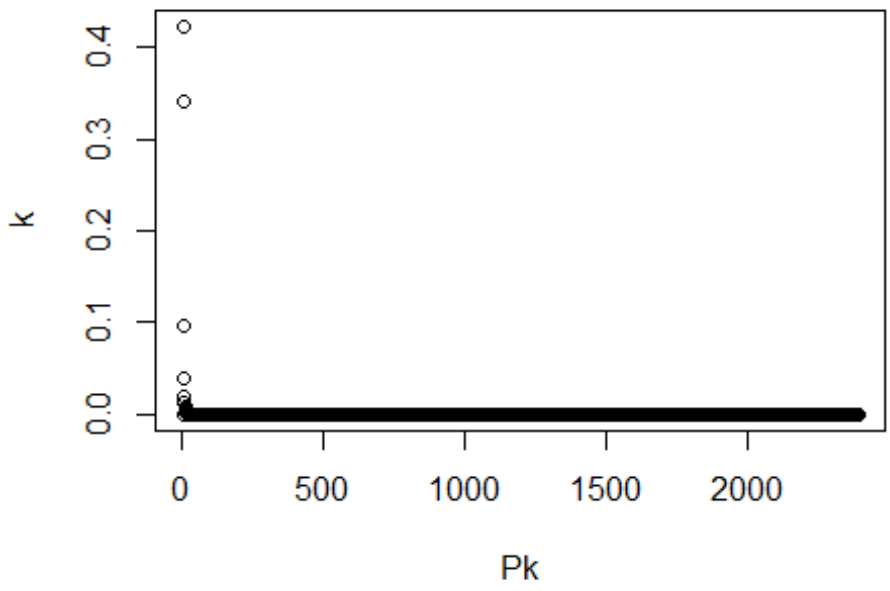

```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



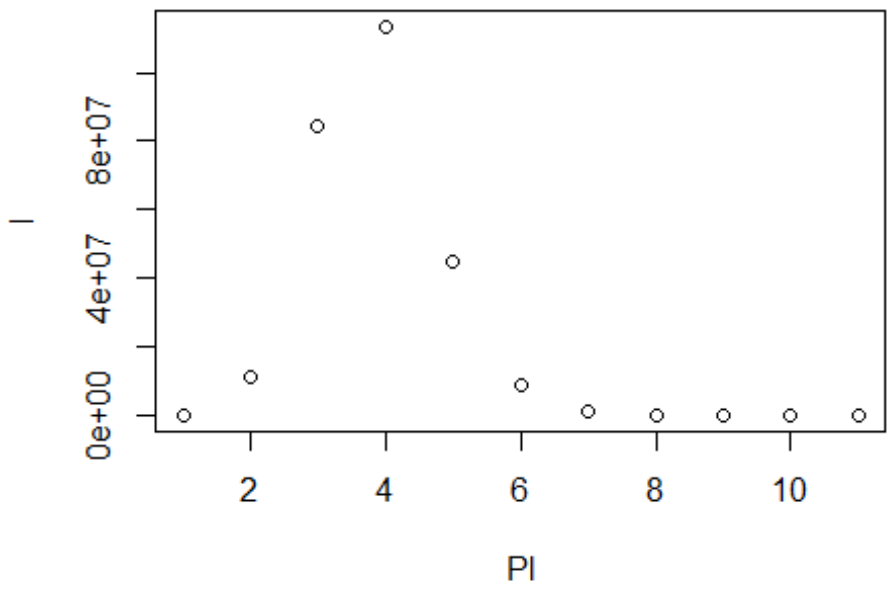
3. Internet

```
internet <-read.graph("as-22july06.gml",format=c("gml"))

graph = internet
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")

## [1] "Directed:- "

print(isDirected)

## [1] FALSE

print("Number of Nodes:-" )
```

```
## [1] "Number of Nodes:-"

print(n)

## [1] 22963

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 48436

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 2390

print("Average Path Length:- ")

## [1] "Average Path Length:- "

print(l)

## [1] 3.842426

print("Diameter:- ")

## [1] "Diameter:- "

print(D)

## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>

print("Average Global Clustering Coefficient:- ")

## [1] "Average Global Clustering Coefficient:- "

print(ccg)

## [1] 0.01114638

plot(degree.distribution(graph),xlab="Pk", ylab="k")
```




```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



4. Erdos-Renyi Network 1

```
e1 <- erdos.renyi.game(2000, 0.01)

graph = e1
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")

## [1] "Directed:- "

print(isDirected)

## [1] FALSE

print("Number of Nodes:-" )

## [1] "Number of Nodes:-"

print(n)

## [1] 2000

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 20225

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 39

print("Average Path Length:- ")

## [1] "Average Path Length:- "
```

```
print(l)
```

```
## [1] 2.822752
```

```
print("Diameter:- ")
```

```
## [1] "Diameter:- "
```

```
print(D)
```

```
## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>
```

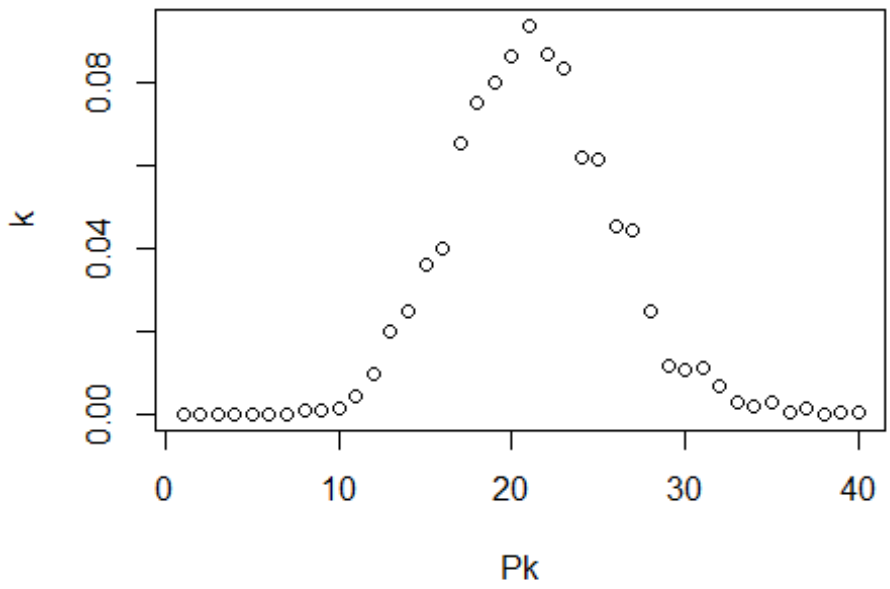
```
print("Average Global Clustering Coefficient:- ")
```

```
## [1] "Average Global Clustering Coefficient:- "
```

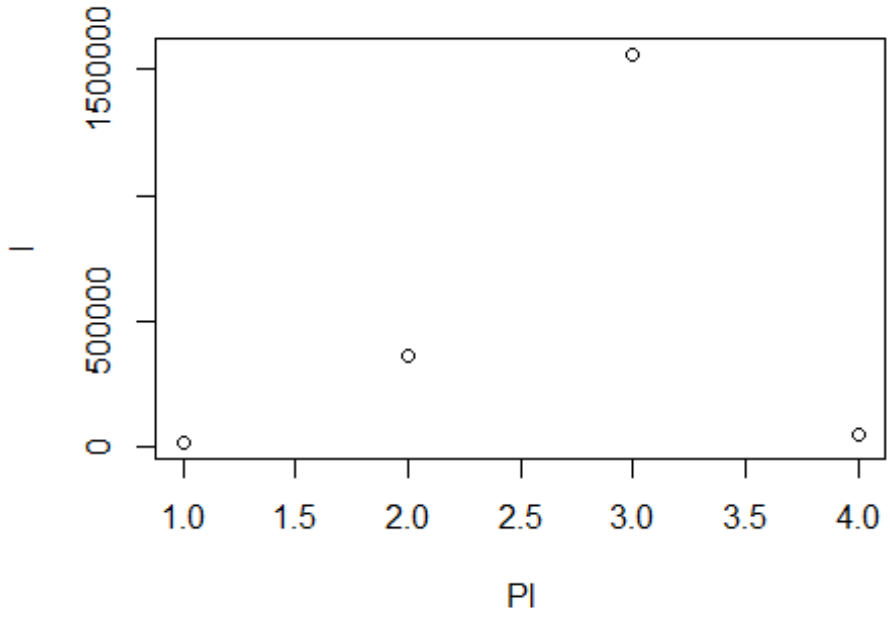
```
print(ccg)
```

```
## [1] 0.009914883
```

```
plot(degree.distribution(graph),xlab="Pk", ylab="k")
```



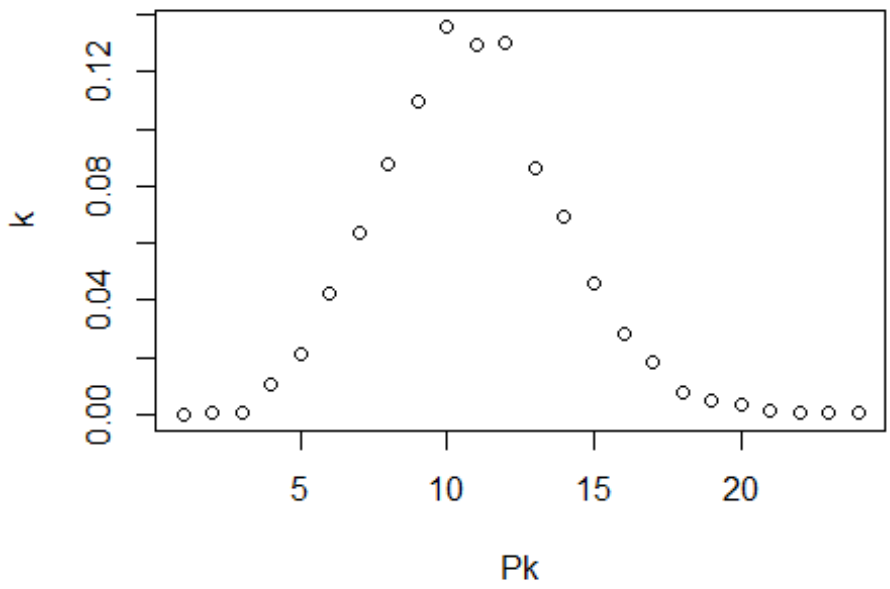

```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



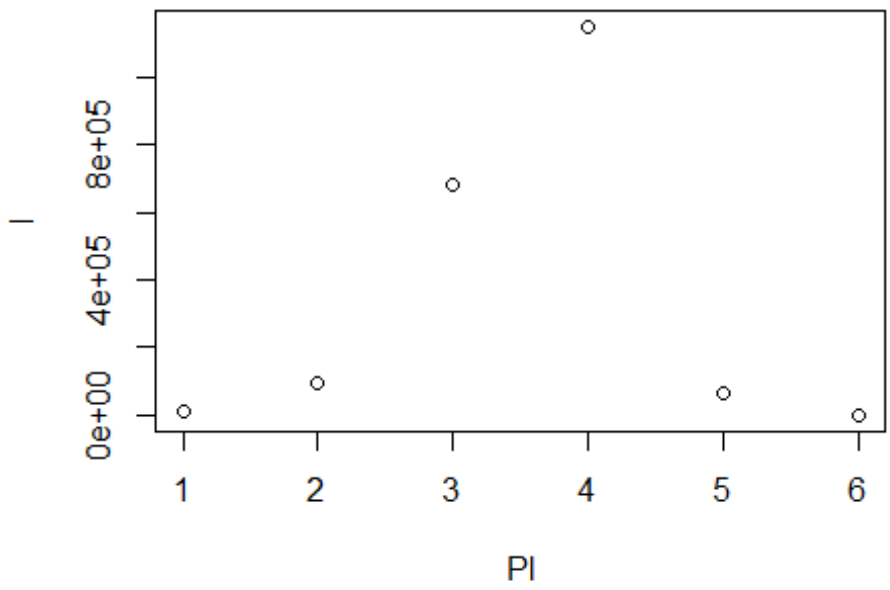
5. Erdos-Renyi Network 2

```
e2 <- erdos.renyi.game(2000, 0.005)

graph = e2
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")

## [1] "Directed:- "

print(isDirected)

## [1] FALSE

print("Number of Nodes:-" )
```

```
## [1] "Number of Nodes:-"

print(n)

## [1] 2000

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 9806

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 23

print("Average Path Length:- ")

## [1] "Average Path Length:- "

print(l)

## [1] 3.586776

print("Diameter:- ")

## [1] "Diameter:- "

print(D)

## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>

print("Average Global Clustering Coefficient:- ")

## [1] "Average Global Clustering Coefficient:- "

print(ccg)

## [1] 0.0046427

plot(degree.distribution(graph),xlab="Pk", ylab="k")
```



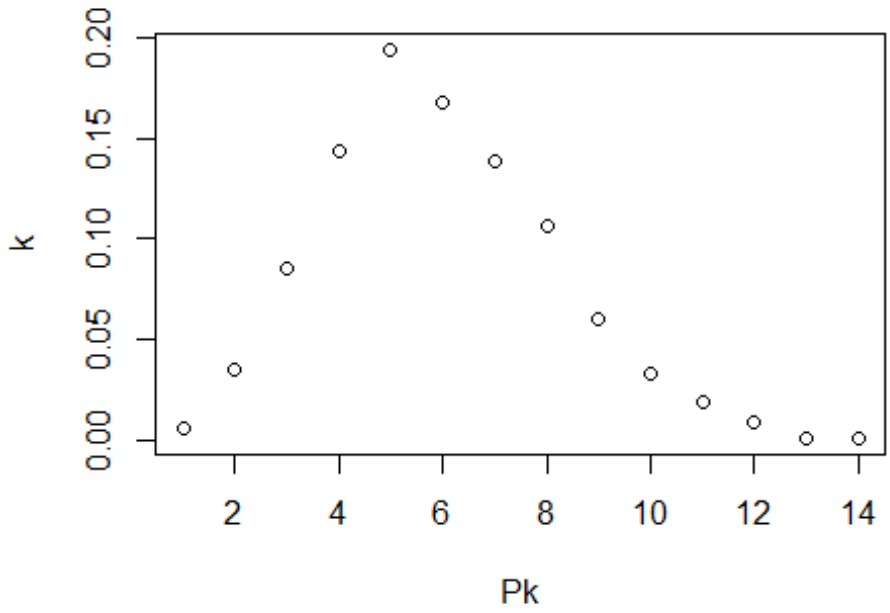

```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



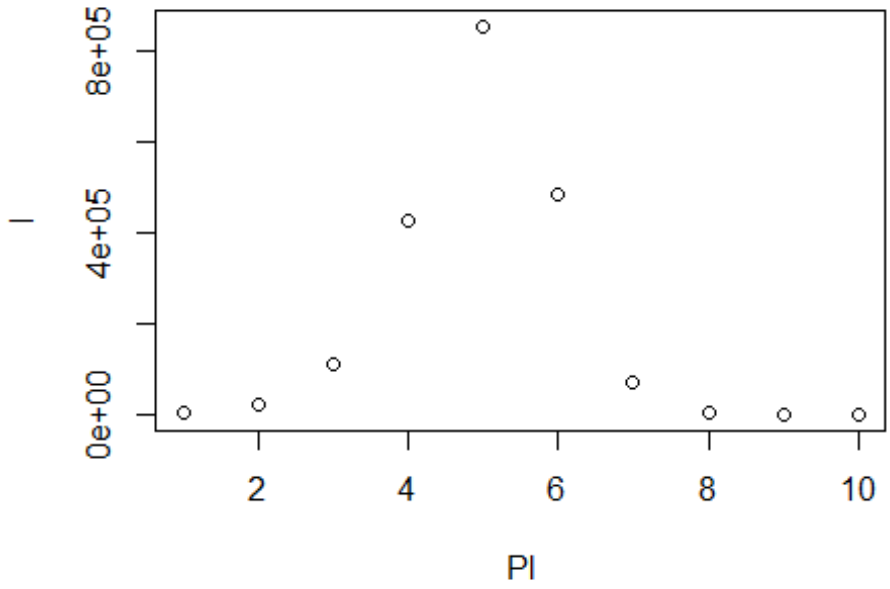
6. Erdos-Renyi Network 3

```
e3 <- erdos.renyi.game(2000, 0.0025)

graph = e3
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")

## [1] "Directed:- "

print(isDirected)

## [1] FALSE

print("Number of Nodes:-" )

## [1] "Number of Nodes:-"

print(n)

## [1] 2000

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 4922

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 13

print("Average Path Length:- ")

## [1] "Average Path Length:- "
```

```
print(l)

## [1] 4.947546

print("Diameter:- ")

## [1] "Diameter:- "

print(D)

## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>

print("Average Global Clustering Coefficient:- ")

## [1] "Average Global Clustering Coefficient:- "

print(ccg)

## [1] 0.002365342

plot(degree.distribution(graph),xlab="Pk", ylab="k")
```



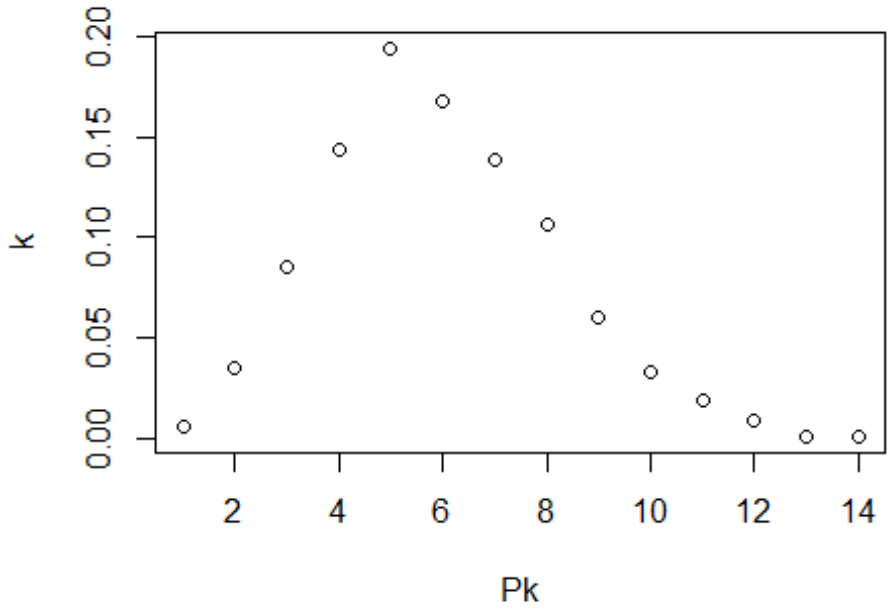

```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



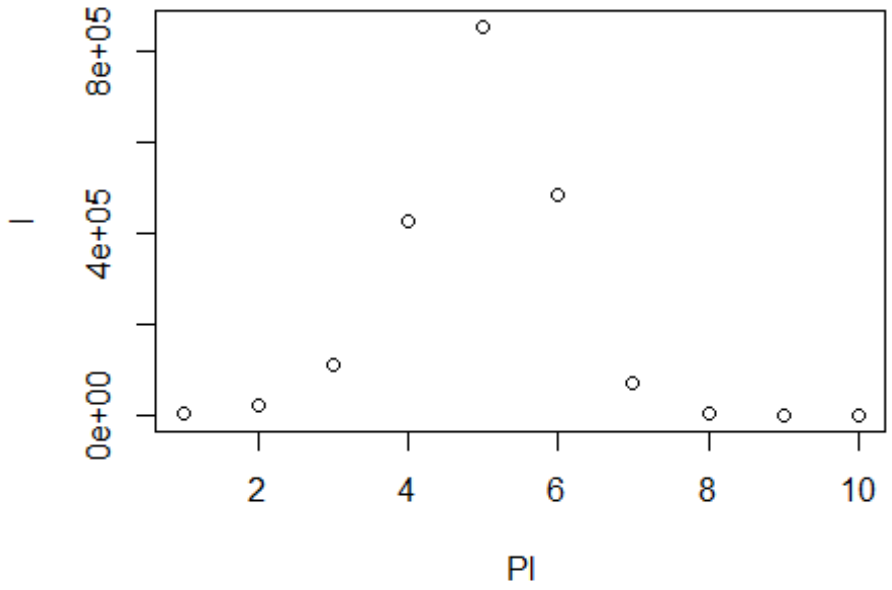
Answer 4 : Dolphin Social Network

```
library("igraph")
dolphins <-read.graph("dolphins.gml",format=c("gml"))

graph = e3
isDirected= is_directed(graph)

n = gorder(graph)
m = gsize(graph)
c = groups(components(graph,mode = c("weak", "strong")))

d = max(degree(graph))
l = average.path.length(graph)
L = diameter(graph)

ccg=transitivity(graph, type = c("global"))
ccl=transitivity(graph, type = c("localundirected"))

print("Directed:- ")

## [1] "Directed:- "

print(isDirected)

## [1] FALSE

print("Number of Nodes:-" )
```

```
## [1] "Number of Nodes:-"

print(n)

## [1] 2000

print("Number of Edges:-" )

## [1] "Number of Edges:-"

print(m)

## [1] 4922

print("Maximum Degree:- ")

## [1] "Maximum Degree:- "

print(d)

## [1] 13

print("Average Path Length:- ")

## [1] "Average Path Length:- "

print(l)

## [1] 4.947546

print("Diameter:- ")

## [1] "Diameter:- "

print(D)

## function (expr, name)
## .External(C_doD, expr, name)
## <bytecode: 0x00000000134b8610>
## <environment: namespace:stats>

print("Average Global Clustering Coefficient:- ")

## [1] "Average Global Clustering Coefficient:- "

print(ccg)

## [1] 0.002365342

plot(degree.distribution(graph),xlab="Pk", ylab="k")
```




```
plot(path.length.hist(graph)$res,xlab="Pl", ylab="l")
```



Analysis:

Above Dolphin network contains undirected social network with number of nodes (number of dolphins) are 2000 and number of edges are 4929. The dataset contains list of of all links where a link represents association between dolphins. Above plot represents the degree distribution and path length distribution of dolphin network.