Assignment 1

Giridhar Manoharan

2/2/2017

Question 1

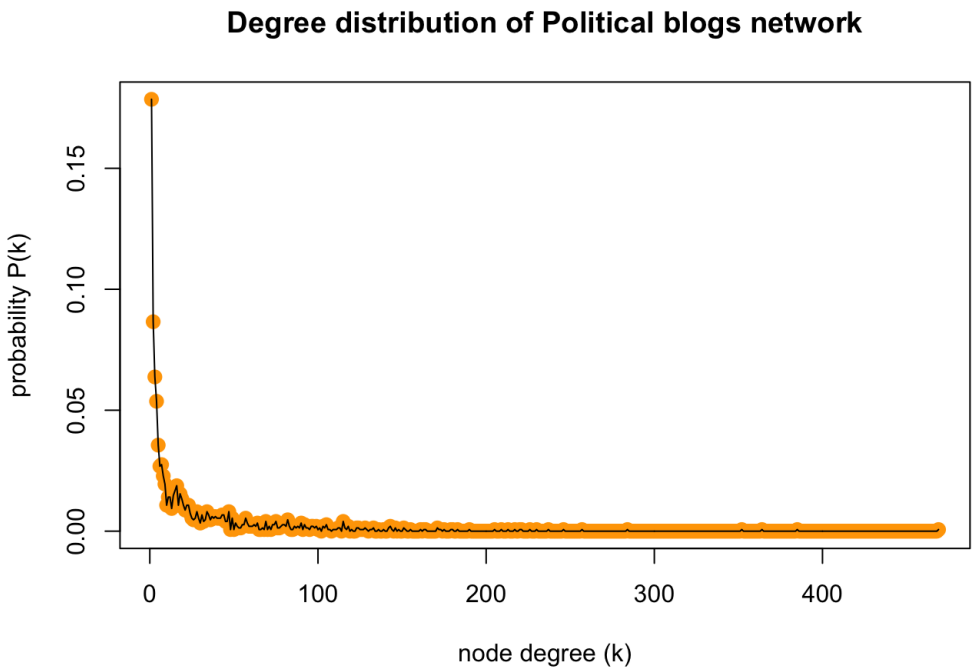
```
internetGraph = read_graph("./as-22july06/as-22july06.gml", format = "gml")
politicalGraph = read_graph("./polblogs/polblogs.gml", format = "gml")
neuralNetGraph = read_graph("./celegansneural/celegansneural.gml", format = "gml")
erdosRenyi01 = erdos.renyi.game(2000, 0.01)
erdosRenyi005 = erdos.renyi.game(2000, 0.005)
erdosRenyi0025 = erdos.renyi.game(2000, 0.0025)
networks = data.frame(Network = character(6), Type = character(6), n = integer(6), m = integer(6), c_strong =
integer(6), c_weak = integer(6), d = integer(6), l = double(6), L = integer(6), cc_local = double(6), cc_globa
l = double(6), stringsAsFactors = FALSE)

#political blogs
networks$Network[1] = "Political blogs"
networks$Type[1] = if(is_directed(politicalGraph) == TRUE) "DIRECTED" else "UNDIRECTED"
networks$n[1] = vcount(politicalGraph)
networks$m[1] = ecount(politicalGraph)
networks$c_strong[1] = count_components(politicalGraph, mode = "strong")
networks$c_weak[1] = count_components(politicalGraph, mode = "weak")
networks$d[1] = max(degree(politicalGraph))
networks$l[1] = mean_distance(politicalGraph)
networks$L[1] = diameter(politicalGraph)
networks$cc_local[1] = transitivity(politicalGraph, type = "localaverageundirected")
networks$cc_global[1] = transitivity(politicalGraph, type = "globalundirected")

#neural network
networks$Network[2] = "Neural network"
networks$Type[2] = if(is_directed(neuralNetGraph) == TRUE) "DIRECTED" else "UNDIRECTED"
networks$n[2] = vcount(neuralNetGraph)
networks$m[2] = ecount(neuralNetGraph)
networks$c_strong[2] = count_components(neuralNetGraph, mode = "strong")
networks$c_weak[2] = count_components(neuralNetGraph, mode = "weak")
networks$d[2] = max(degree(neuralNetGraph))
networks$l[2] = mean_distance(neuralNetGraph)
networks$L[2] = diameter(neuralNetGraph)
networks$cc_local[2] = transitivity(neuralNetGraph, type = "localaverageundirected")
networks$cc_global[2] = transitivity(neuralNetGraph, type = "globalundirected")

#internet
networks$Network[3] = "Internet"
networks$Type[3] = if(is_directed(internetGraph) == TRUE) "DIRECTED" else "UNDIRECTED"
networks$n[3] = vcount(internetGraph)
networks$m[3] = ecount(internetGraph)
networks$c_strong[3] = count_components(internetGraph)
networks$c_weak[3] = NA
networks$d[3] = max(degree(internetGraph))
networks$l[3] = mean_distance(internetGraph)
networks$L[3] = diameter(internetGraph)
networks$cc_local[3] = transitivity(internetGraph, type = "localaverageundirected")
networks$cc_global[3] = transitivity(internetGraph, type = "globalundirected")

#Erdos Renyi 0.01
networks$Network[4] = "Erdos Renyi 0.01"
networks$Type[4] = if(is_directed(erdosRenyi01) == TRUE) "DIRECTED" else "UNDIRECTED"
networks$n[4] = vcount(erdosRenyi01)
networks$m[4] = ecount(erdosRenyi01)
networks$c_strong[4] = count_components(erdosRenyi01)
networks$c_weak[4] = NA
networks$d[4] = max(degree(erdosRenyi01))
networks$l[4] = mean_distance(erdosRenyi01)
networks$L[4] = diameter(erdosRenyi01)
networks$cc_local[4] = transitivity(erdosRenyi01, type = "localaverageundirected")
networks$cc_global[4] = transitivity(erdosRenyi01, type = "globalundirected")

#Erdos Renyi 0.005
networks$Network[5] = "Erdos Renyi 0.005"
networks$Type[5] = if(is_directed(erdosRenyi005) == TRUE) "DIRECTED" else "UNDIRECTED"
```

```
networks$n[5] = vcount(erdosRenyi005)
networks$m[5] = ecount(erdosRenyi005)
networks$c_strong[5] = count_components(erdosRenyi005)
networks$c_weak[5] = NA
networks$d[5] = max(degree(erdosRenyi005))
networks$l[5] = mean_distance(erdosRenyi005)
networks$L[5] = diameter(erdosRenyi005)
networks$cc_local[5] = transitivity(erdosRenyi005, type = "localaverageundirected")
networks$cc_global[5] = transitivity(erdosRenyi005, type = "globalundirected")

#Erdos Renyi 0.0025
networks$Network[6] = "Erdos Renyi 0.0025"
networks$Type[6] = if(is_directed(erdosRenyi0025) == TRUE) "DIRECTED" else "UNDIRECTED"
networks$n[6] = vcount(erdosRenyi0025)
networks$m[6] = ecount(erdosRenyi0025)
networks$c_strong[6] = count_components(erdosRenyi0025)
networks$c_weak[6] = NA
networks$d[6] = max(degree(erdosRenyi0025))
networks$l[6] = mean_distance(erdosRenyi0025)
networks$L[6] = diameter(erdosRenyi0025)
networks$cc_local[6] = transitivity(erdosRenyi0025, type = "localaverageundirected")
networks$cc_global[6] = transitivity(erdosRenyi0025, type = "globalundirected")

colnames(networks) = c("Network", "Type", "Nodes", "Links", "Connected components (Strongly connected - directed)", "Weakly connected components", "Maximum degree", "Average path length", "Diameter", "Average local clustering coefficient", "Global clustering coefficient")

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

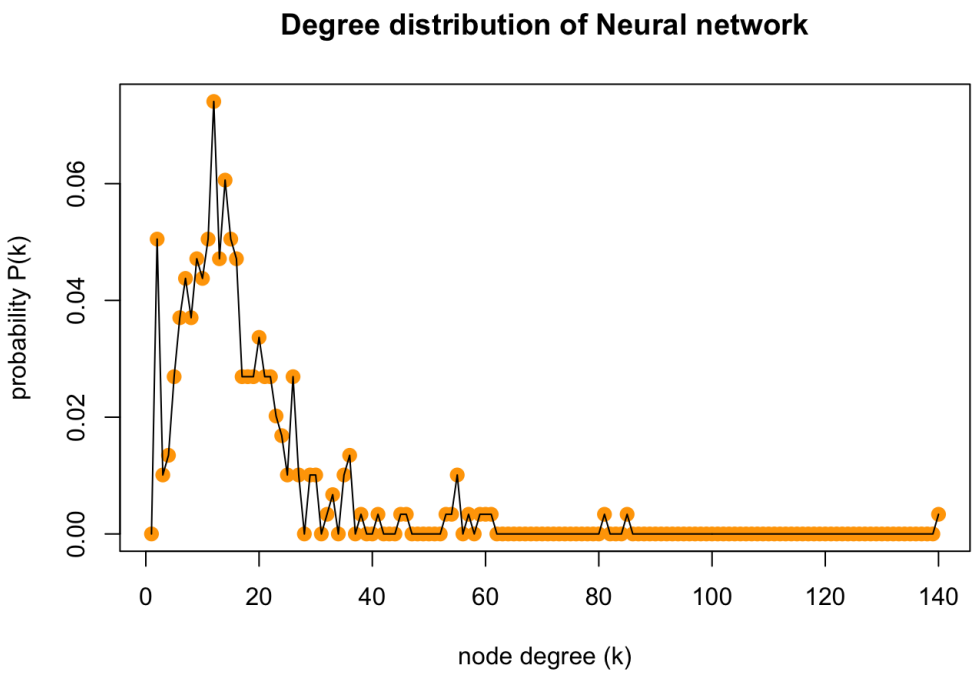
Network	Type	Nodes	Links	Connected components (Strongly connected - directed)	Weakly connected components	Maximum degree	Average path length	Diameter	Average local clustering coefficient	Global clustering coefficient
Political blogs	DIRECTED	1490	19090	688	268	468	3.390184	9	0.3600287	0.2259585
Neural network	DIRECTED	297	2359	57	1	139	3.991884	14	0.3079145	0.1807115
Internet	UNDIRECTED	22963	48436	1	NA	2390	3.842426	11	0.3499154	0.0111464
Erdos Renyi 0.01	UNDIRECTED	2000	19862	1	NA	38	2.834644	4	0.0104363	0.0104980
Erdos Renyi 0.005	UNDIRECTED	2000	9947	1	NA	21	3.566632	6	0.0053518	0.0051179
Erdos Renyi 0.0025	UNDIRECTED	2000	5045	16	NA	14	4.872491	10	0.0020744	0.0025961

Question 2

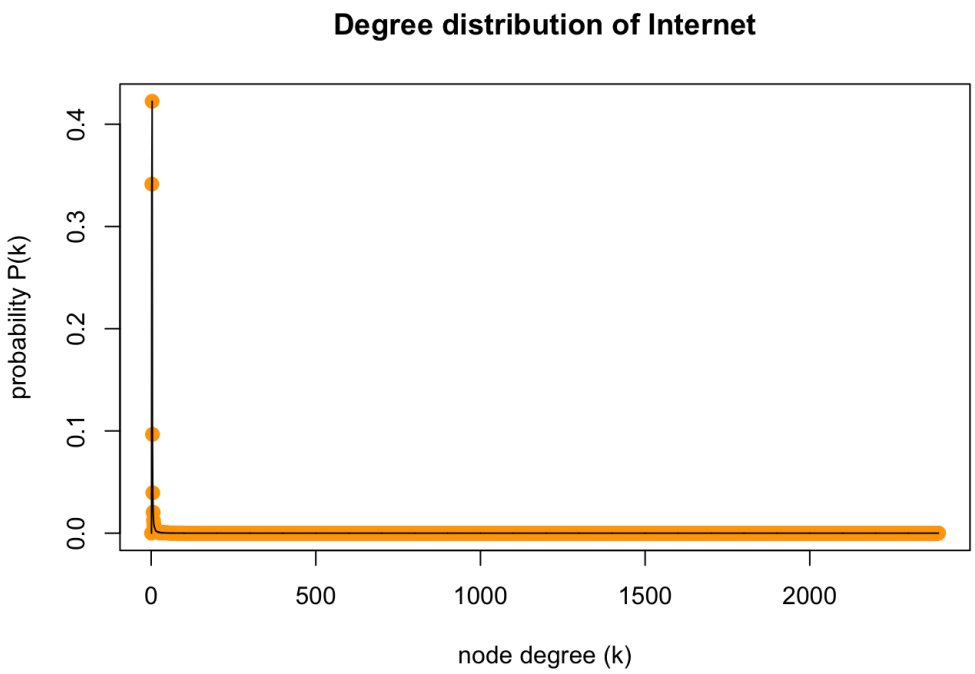
```
#political blogs
plot(degree.distribution(politicalGraph), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "probability P(k)", main = "Degree distribution of Political blogs network")
lines(degree_distribution(politicalGraph))
```



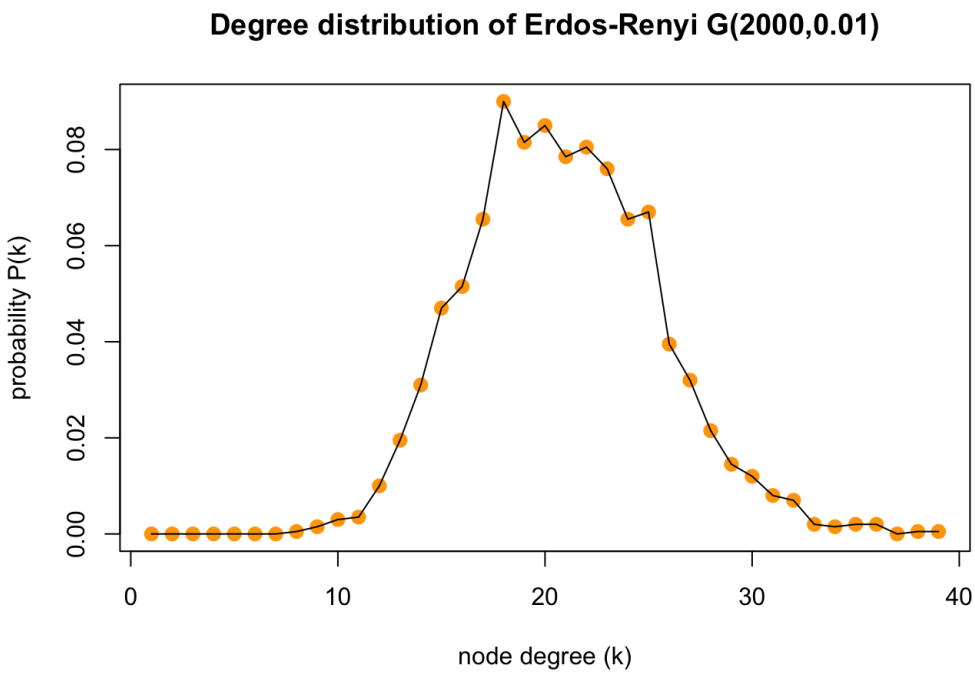

```
#neural network
plot(degree.distribution(neuralNetGraph), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "proba
bility P(k)", main = "Degree distribution of Neural network")
lines(degree_distribution(neuralNetGraph))
```



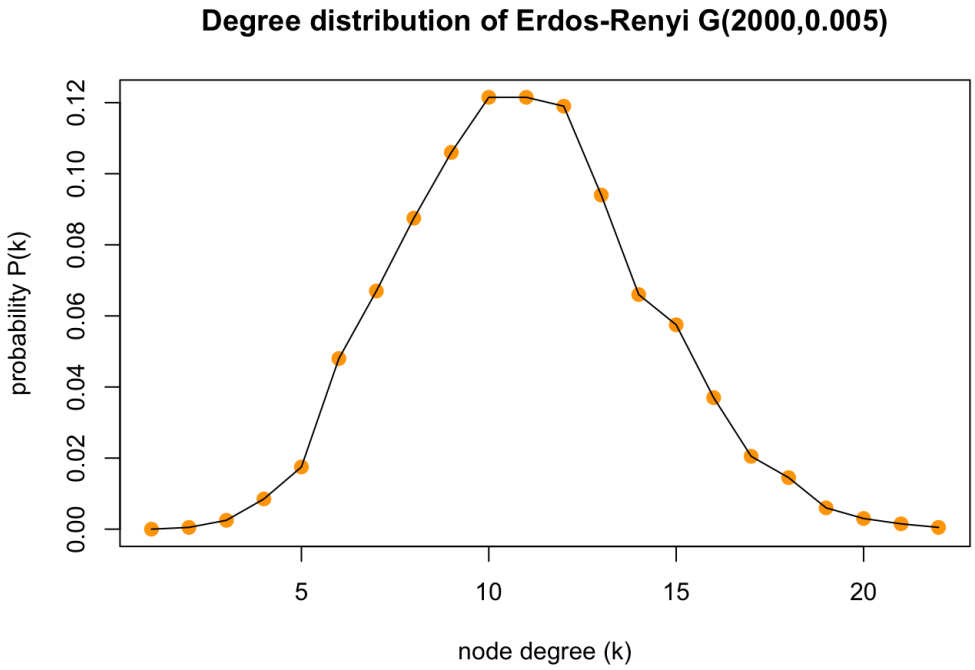

```
#internet
plot(degree.distribution(internetGraph), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "probab
ility P(k)", main = "Degree distribution of Internet")
lines(degree_distribution(internetGraph))
```



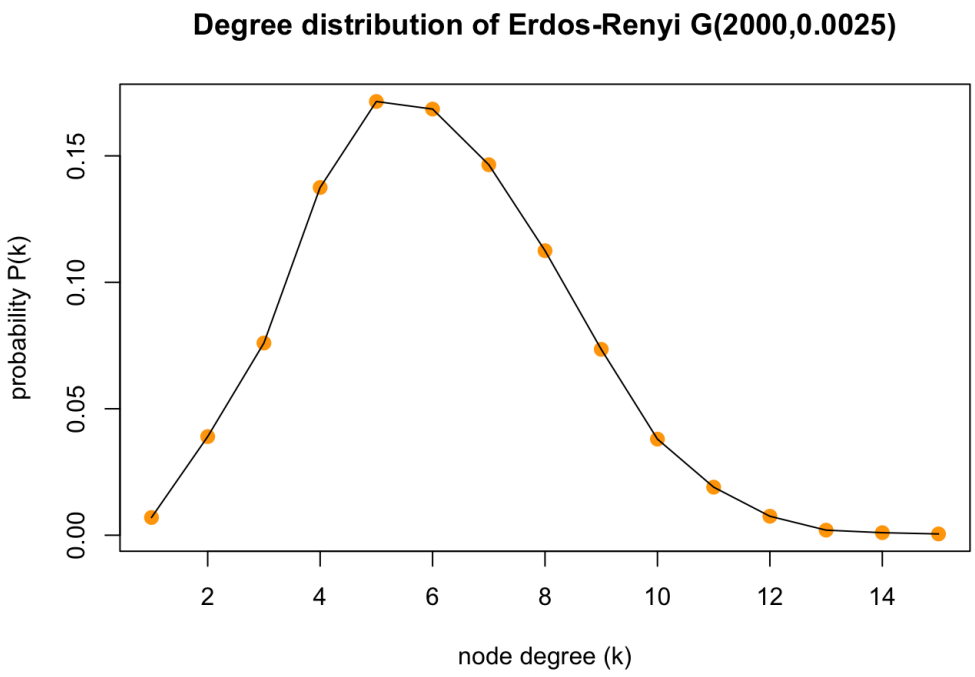

```
#Erdos Renyi 0.01
plot(degree.distribution(erdosRenyi01), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "probabi
lity P(k)", main = "Degree distribution of Erdos-Renyi G(2000,0.01)")
lines(degree_distribution(erdosRenyi01))
```




```
#Erdos Renyi 0.005
plot(degree.distribution(erdosRenyi005), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "probab
ility P(k)", main = "Degree distribution of Erdos-Renyi G(2000,0.005)")
lines(degree_distribution(erdosRenyi005))
```



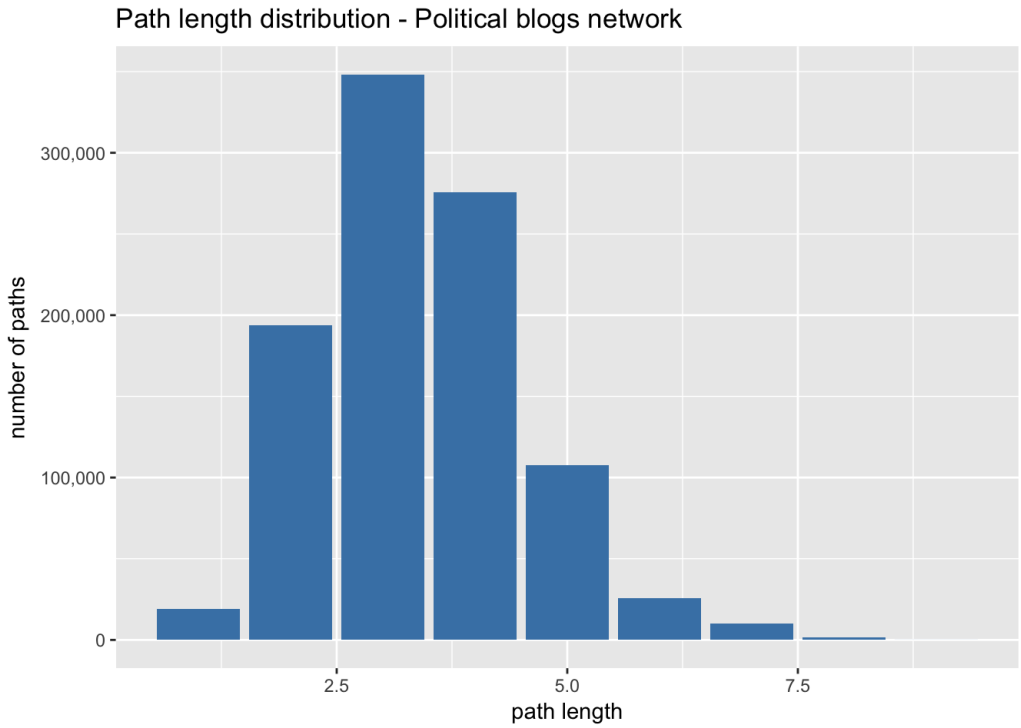

```
#Erdos Renyi 0.0025
plot(degree.distribution(erdosRenyi0025), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "proba
bility P(k)", main = "Degree distribution of Erdos-Renyi G(2000,0.0025)")
lines(degree_distribution(erdosRenyi0025))
```



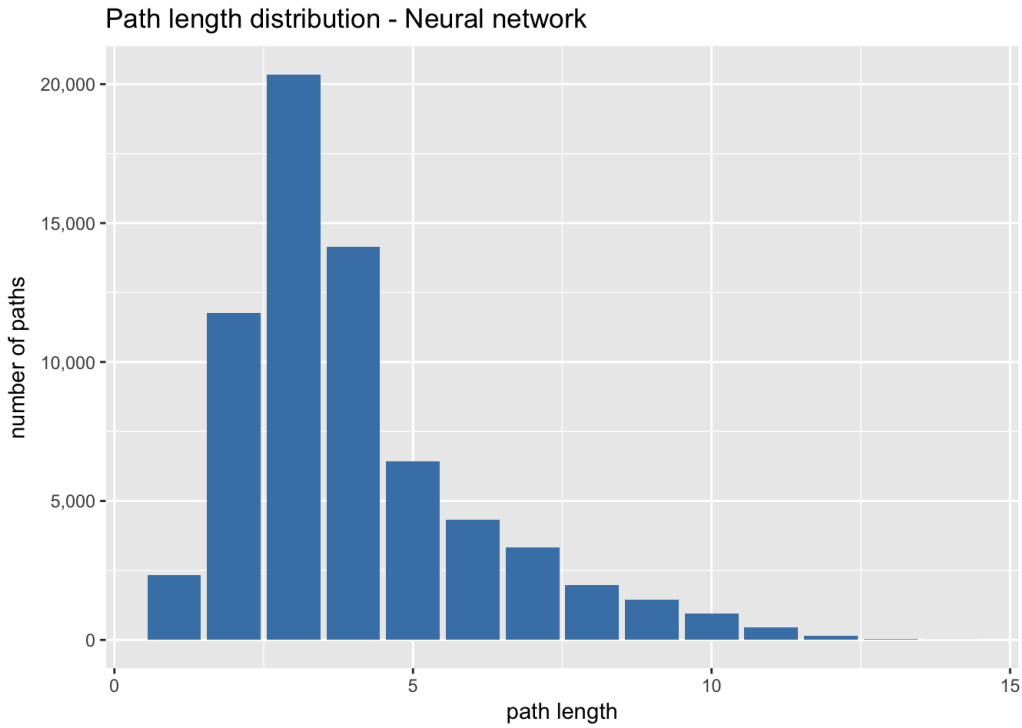
Observations From the above degree distribution graphs we can observe that real world networks like “Political blogs”, “Neural networks”, and “Internet” have power law distribution. Whereas, random graphs generated using Erdos-Renyi model have poisson (bell-shaped) distribution.

Question 3

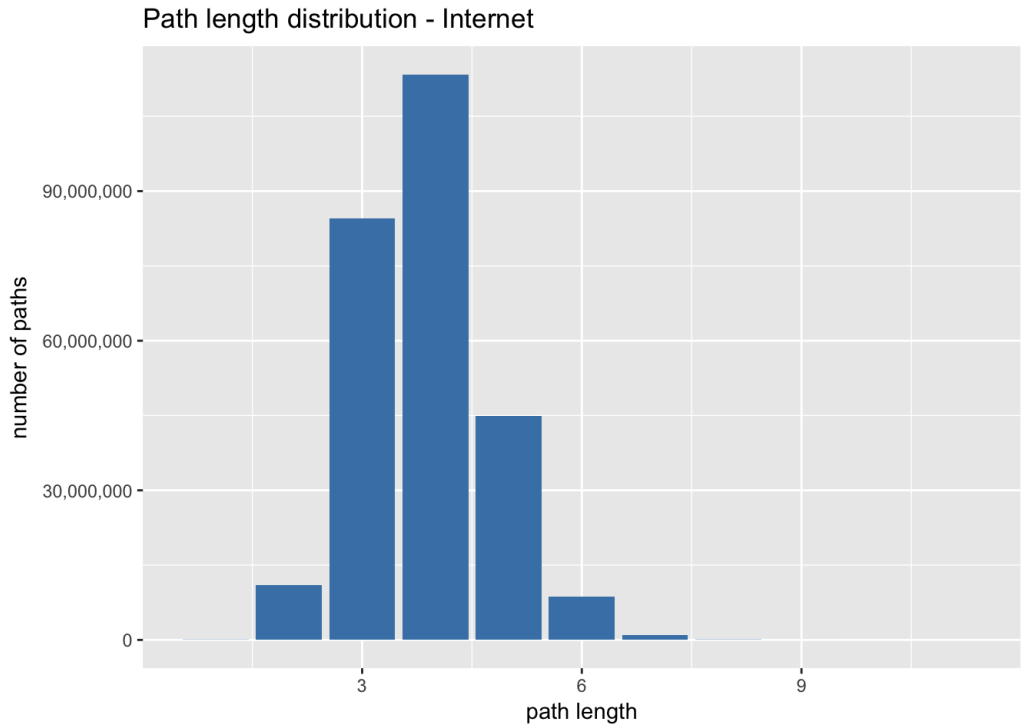
```
#political blogs
d = path.length.hist(politicalGraph)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path
length distribution - Political blogs network", x="path length", y="number of paths") + scale_y_continuous(lab
els = comma)
```



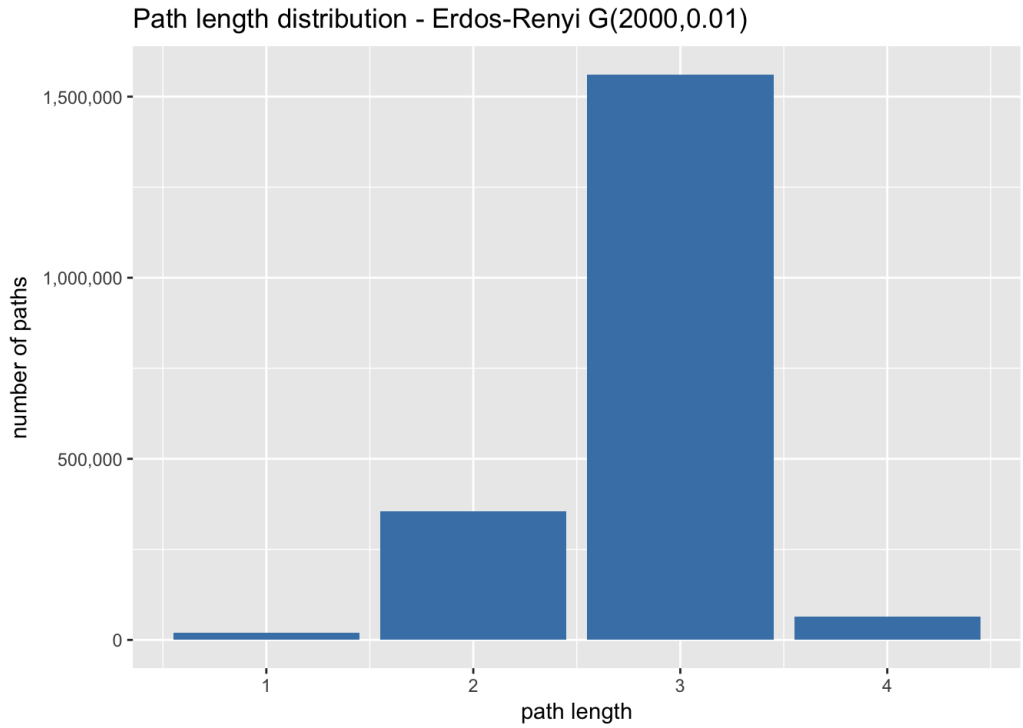

```
#neural network
d = path.length.hist(neuralNetGraph)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path length distribution - Neural network", x="path length", y="number of paths") + scale_y_continuous(labels = comma)
```



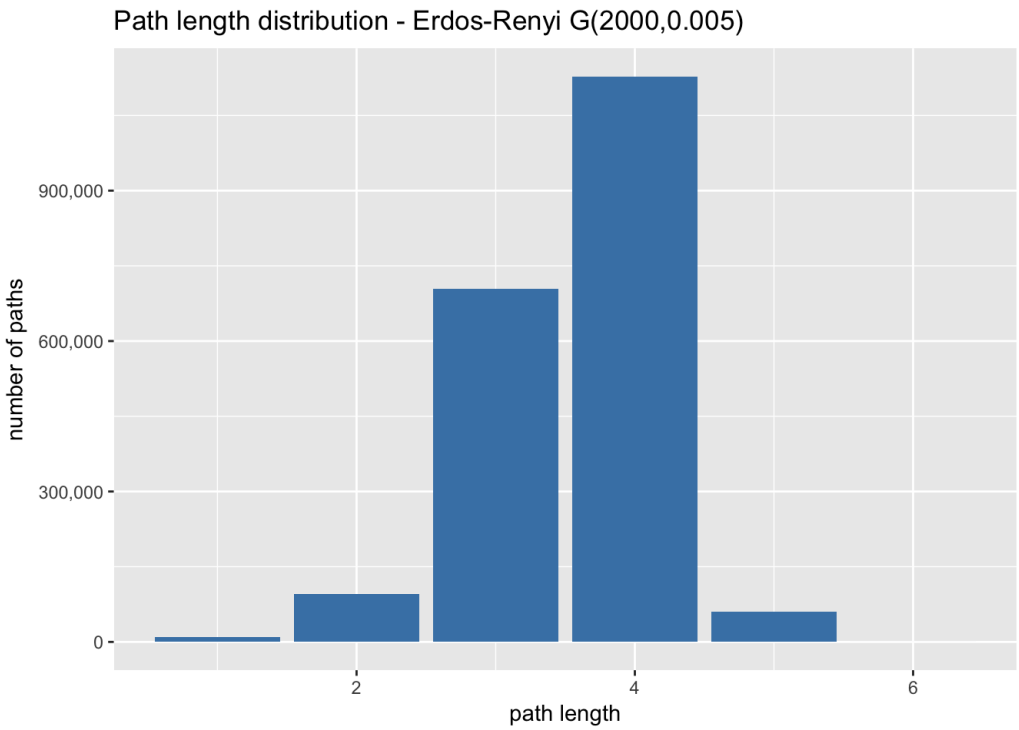

```
#internet
d = path.length.hist(internetGraph)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path length distribution - Internet", x="path length", y="number of paths") + scale_y_continuous(labels = comma)
```



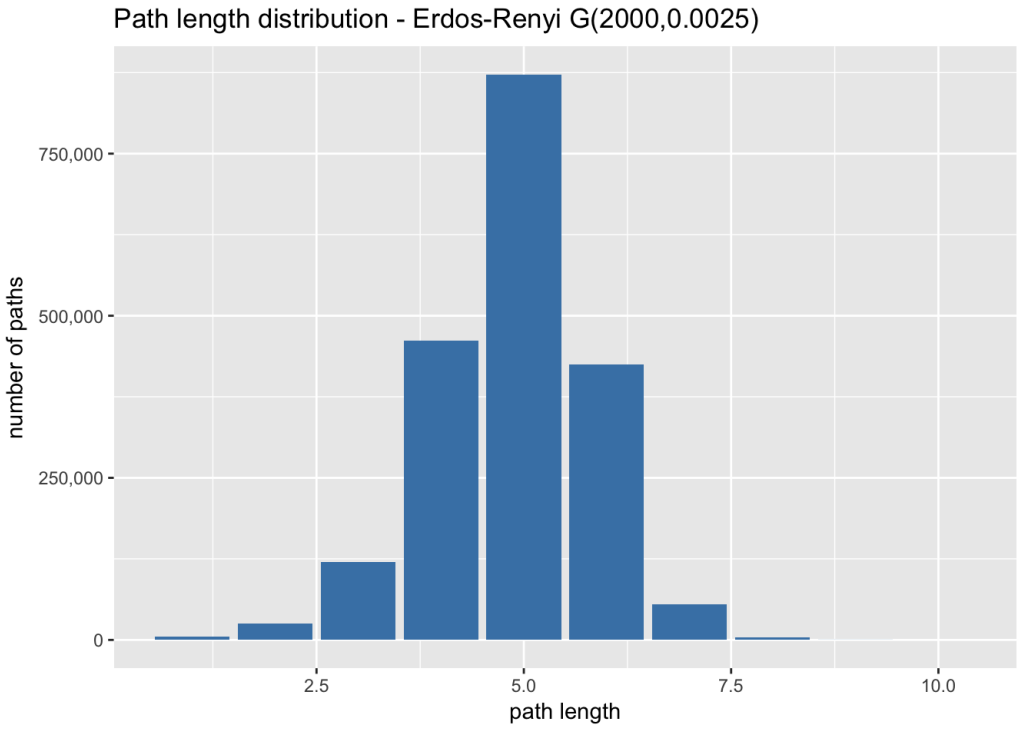

```
#Erdos Renyi 0.01
d = path.length.hist(erdosRenyi01)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path length distribution - Erdos-Renyi G(2000,0.01)", x="path length", y="number of paths") + scale_y_continuous(labels = comma)
```




```
#Erdos Renyi 0.005
d = path.length.hist(erdosRenyi005)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path length distribution - Erdos-Renyi G(2000,0.005)", x="path length", y="number of paths") + scale_y_continuous(labels = comma)
```




```
#Erdos Renyi 0.0025
d = path.length.hist(erdosRenyi0025)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path length distribution - Erdos-Renyi G(2000,0.0025)", x="path length", y="number of paths") + scale_y_continuous(labels = comma)
```



Observations From the above path length distributions, we can observe that all the six networks have bell-shaped path length distribution. We can also observe that in erdos renyi random graphs, as we increase the probability of an edge between any two nodes, path lengths take lower values. For lower probabilities of an edge between two nodes, path lengths with higher values can be found.

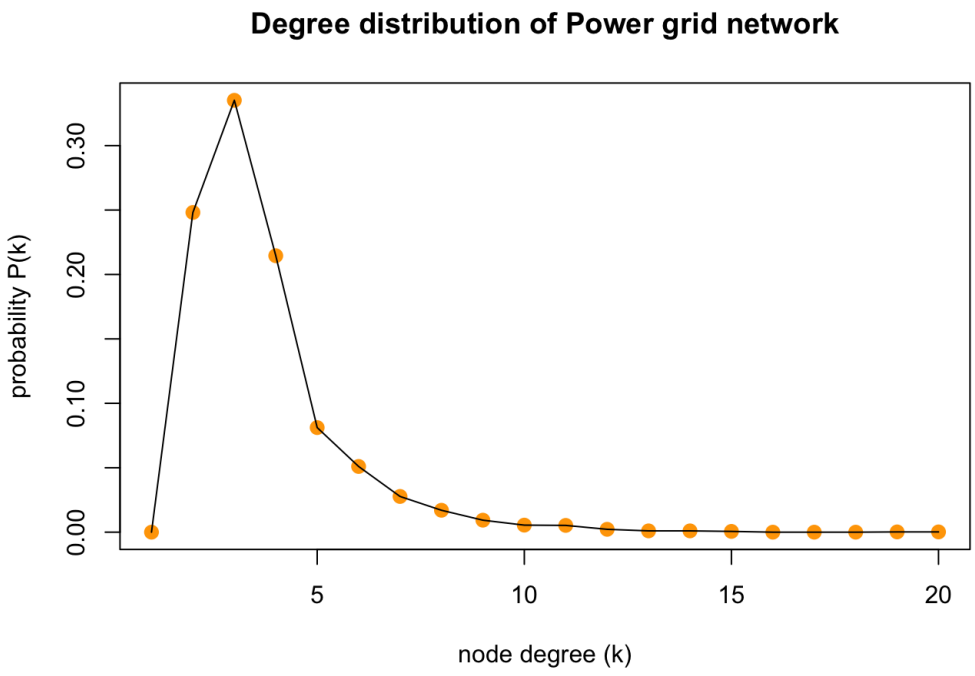
Question 4

I have chosen power grid real world network here to do network analysis. It is an undirected, unweighted network representing the topology of the Western States Power Grid of the United States. The nodes represent power plants and substations, and links represent transmission (high voltage) lines.

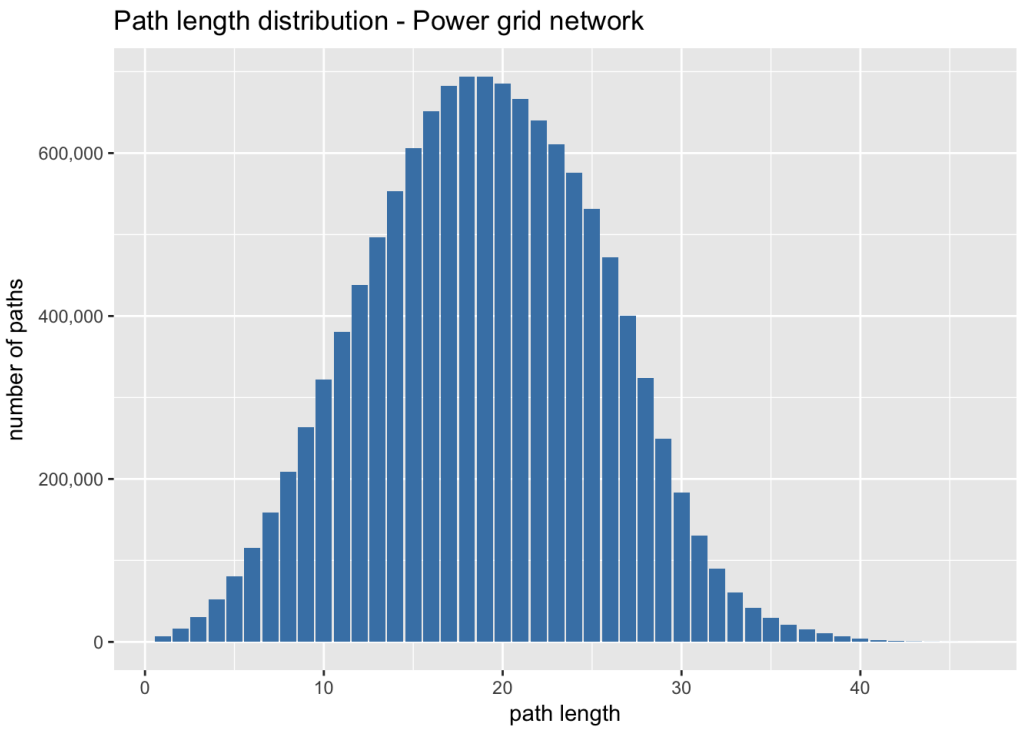
```
powerGridGraph = read_graph("./power/power.gml", format = "gml")
networks = data.frame(Network = character(1), Type = character(1), n = integer(1), m = integer(1), c = integer
(1), d = integer(1), l = double(1), L = integer(1), cc_local = double(1), cc_global = double(1), stringsAsFact
ors = FALSE)
networks$Network[1] = "Power grid network"
networks$Type[1] = if(is_directed(powerGridGraph) == TRUE) "DIRECTED" else "UNDIRECTED"
networks$n[1] = vcount(powerGridGraph)
networks$m[1] = ecount(powerGridGraph)
networks$c[1] = count_components(powerGridGraph)
networks$d[1] = max(degree(powerGridGraph))
networks$l[1] = mean_distance(powerGridGraph)
networks$L[1] = diameter(powerGridGraph)
networks$cc_local[1] = transitivity(powerGridGraph, type = "localaverageundirected")
networks$cc_global[1] = transitivity(powerGridGraph, type = "globalundirected")
colnames(networks) = c("Network", "Type", "Nodes", "Links", "Connected components", "Maximum degree", "Average
path length", "Diameter", "Average local clustering coefficient", "Global clustering coefficient")
kable(networks, format = "markdown")
```

Network	Type	Nodes	Links	Connected components	Maximum degree	Average path length	Diameter	Average local clustering coefficient	Global clustering coefficient
Power grid network	UNDIRECTED	4941	6594	1	19	18.98919	46	0.1065389	0.1031532

```
plot(degree.distribution(powerGridGraph), pch=19, cex=1.2, col="orange", xlab="node degree (k)", ylab = "proba
bility P(k)", main = "Degree distribution of Power grid network")
lines(degree_distribution(powerGridGraph))
```




```
d = path.length.hist(powerGridGraph)$res
plDist = data.frame(pathLength = integer(length(d)), freq = integer(length(d)))
plDist$pathLength = 1:length(d)
plDist$freq = d
ggplot(plDist, aes(x=pathLength, y=freq)) + geom_bar(stat = "identity", fill = "steelblue") + labs(title="Path
length distribution - Power grid network", x="path length", y="number of paths") + scale_y_continuous(labels =
comma)
```


Path length distribution - Power grid network
number of paths
600,000
400,000
200,000
0
0
10
20
30
40
path length