CPTS 591 Elements of network science
Jinyang Ruan
011696096
Assignment 1

```
---
title: "Assignment 1"
author: "Jinyang Ruan"
date: "2/15/2021"
output:
  pdf_document: default
  html_document: default
---

```{r setup, include=FALSE}
library('igraph')
```

```{r load Political blogs}
Political_gragh <- read_graph('Political_blogs.gml',format='gml')
```
```{r load Neural network}
Neural_graph <- read_graph('Neural_network.gml',format='gml')
```
```{r load Internet}
Internet_graph <- read_graph('Internet.gml',format='gml')
```
```{r generate three random graphs}
Random_graph1 <- sample_gnp(n=2000,p=0.01)
Random_graph2 <- sample_gnp(n=2000,p=0.005)
Random_graph3 <- sample_gnp(n=2000,p=0.0025)
```
```

Question 1: Code part

```
Question1:

```{R}
data.frame( Network = c("Political blogs","Neural network","Internet","Random_graph1","Random_graph2","Random_graph3"),
            Type = c(is.directed(Political_gragh),is.directed(Neural_graph), is.directed(Internet_graph),
                     is.directed(Random_graph1),is.directed(Random_graph2), is.directed(Random_graph3)),
            n = c(vcount(Political_gragh),vcount(Neural_graph),vcount(Internet_graph),
                  vcount(Random_graph1),vcount(Random_graph2),vcount(Random_graph3)),
            m = c(ecount(Political_gragh),ecount(Neural_graph),ecount(Internet_graph),
                  ecount(Random_graph1),ecount(Random_graph2),ecount(Random_graph3)),
            c = c(count_components(Political_gragh),count_components(Neural_graph),count_components(Internet_graph),
                  count_components(Random_graph1),count_components(Random_graph2),count_components(Random_graph3)),
            d = c(max(degree(Political_gragh)),max(degree(Neural_graph)),max(degree(Internet_graph)),
                  max(degree(Random_graph1)),max(degree(Random_graph2)),max(degree(Random_graph3))),
            l = c(average.path.length(Political_gragh),average.path.length(Neural_graph),average.path.length(Internet_graph),
                  average.path.length(Random_graph1),average.path.length(Random_graph2),average.path.length(Random_graph3)),
            L = c(diameter(Political_gragh),diameter(Neural_graph),diameter(Internet_graph),
                  diameter(Random_graph1),diameter(Random_graph2),diameter(Random_graph3)),
            CCl = c(transitivity(Political_gragh, type=c("localaverageundirected")), transitivity(Neural_graph, type=c("localaverageundirected")),
                    transitivity(Internet_graph,type=c("localaverageundirected")),transitivity(Random_graph1,type=c("localaverageundirected")),
                    transitivity(Random_graph2,type=c("localaverageundirected")),transitivity(Random_graph3,type=c("localaverageundirected"))),
            CCg = c(transitivity(Political_gragh,type=c("undirected")),transitivity(Neural_graph, type=c("undirected")),
                    transitivity(Internet_graph,type=c("undirected")),transitivity(Random_graph1,type=c("undirected")),
                    transitivity(Random_graph2,type=c("undirected")),transitivity(Random_graph3,type=c("undirected"))))
```
```

Question 1: Solution part

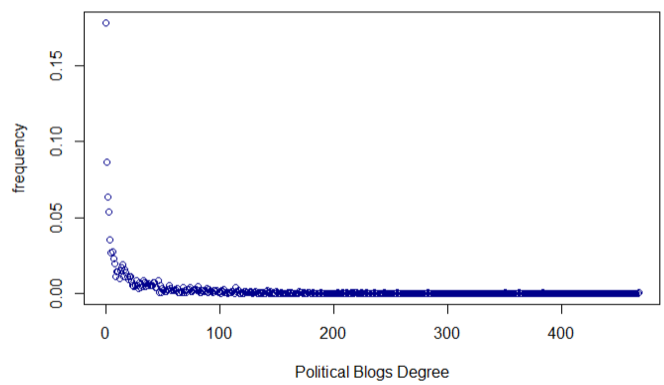
Network <chr>	Type <lgl>	n <int>	m <dbl>	c <dbl>	d <dbl>	l <dbl>	L <dbl>	CCl <dbl>	CCg <dbl>
Political blogs	TRUE	1490	19090	268	468	3.390184	9	0.360028652	0.225958517
Neural network	TRUE	297	2359	1	139	3.991884	14	0.307914537	0.180711471
Internet	FALSE	22963	48436	1	2390	3.842426	11	0.349915358	0.011146384
Random_graph1	FALSE	2000	20014	1	39	2.829963	4	0.010462223	0.010505406
Random_graph2	FALSE	2000	9924	2	22	3.565713	6	0.004632432	0.004592178
Random_graph3	FALSE	2000	5172	10	15	4.799116	10	0.003333303	0.002355757

6 rows

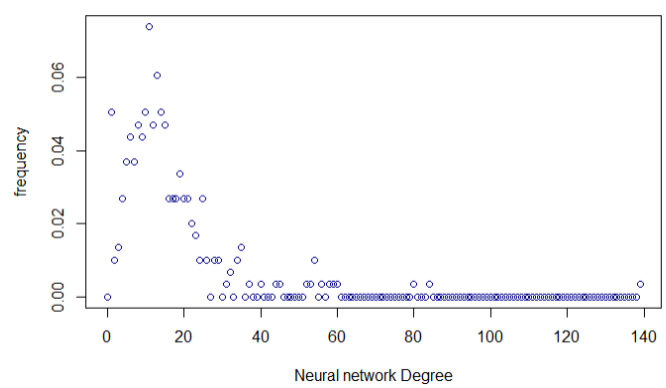
Note: Under "Type" column, TRUE means this graph is directed, "FALSE" means it is undirected.

Question 2:

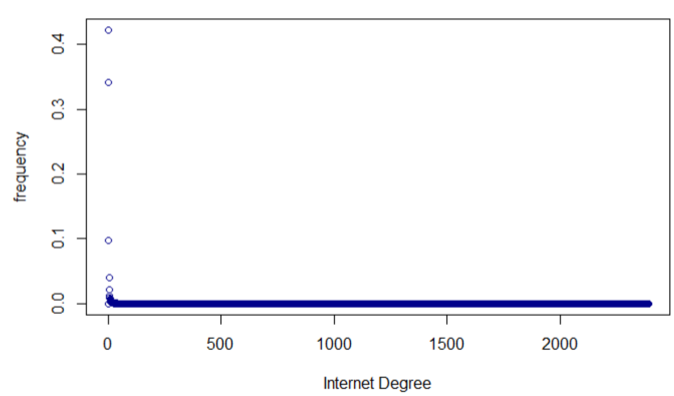
```{R Political blogs degree distribution}
plot(x=0:max(degree(Political_gragh)),y=degree_distribution(Political_gragh, cumulative = FALSE),
     xlab = "Political Blogs Degree", ylab = "Frequency", col="Dark Blue")
```



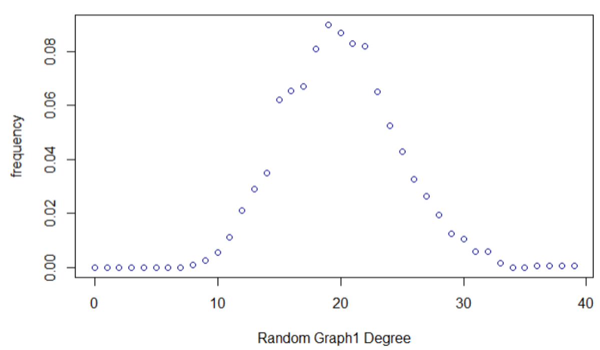

```{R Neural network degree distribution}
plot(x=0:max(degree(Neural_graph)),y=degree_distribution(Neural_graph, cumulative = FALSE),
     xlab = "Neural network Degree", ylab = "Frequency", col="Dark Blue")
```



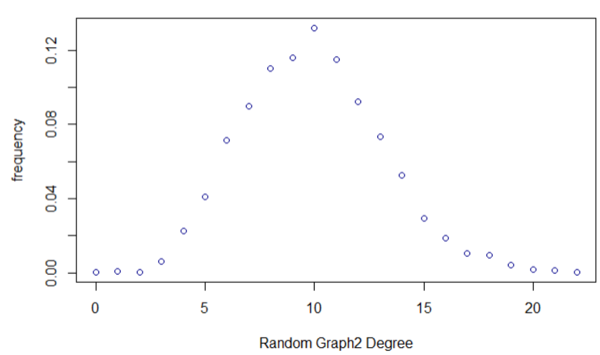

```{R Internet degree distribution}
plot(x=0:max(degree(Internet_graph)),y=degree_distribution(Internet_graph, cumulative = FALSE),
     xlab = "Internet Degree", ylab = "Frequency", col="Dark Blue")
```



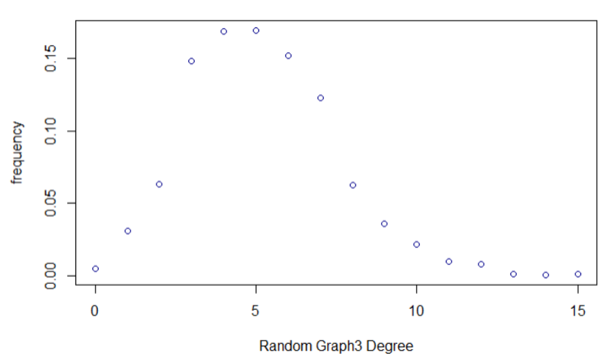
````
```{R Random Graph1 degree distribution}
plot(x=0:max(degree(Random_graph1)),y=degree_distribution(Random_graph1, cumulative = FALSE),
     xlab = "Random Graph1 Degree", ylab = "Frequency", col="Dark Blue")
```
````




````
```{R Random Graph2 degree distribution}
plot(x=0:max(degree(Random_graph2)),y=degree_distribution(Random_graph2, cumulative = FALSE),
     xlab = "Random Graph2 Degree", ylab = "Frequency", col="Dark Blue")
```
````




````
```{R Random Graph3 degree distribution}
plot(x=0:max(degree(Random_graph3)),y=degree_distribution(Random_graph3, cumulative = FALSE),
     xlab = "Random Graph3 Degree", ylab = "Frequency", col="Dark Blue")
```
````

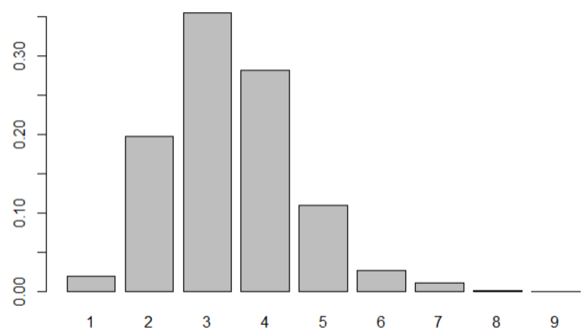


Observation:

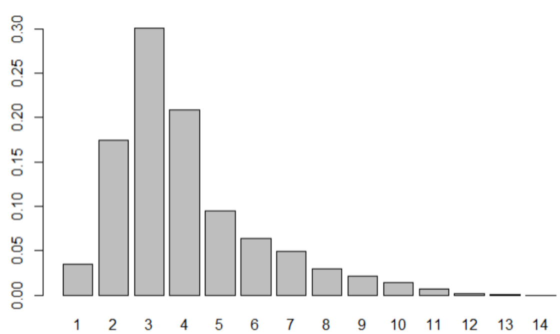
In real world graphs, for example, Political blogs, neural network, and Internet network, the degrees of nodes are at a relatively low level, that is, most nodes have a low degree, especially in the Internet graph, most nodes have a degree less than 10 when few nodes have degrees that over 2000. However, in random graphs, the distribution of degree accords with normal distribution. Degrees of most nodes are at middle level, few nodes' degrees at very low or very high level.

Question 3:

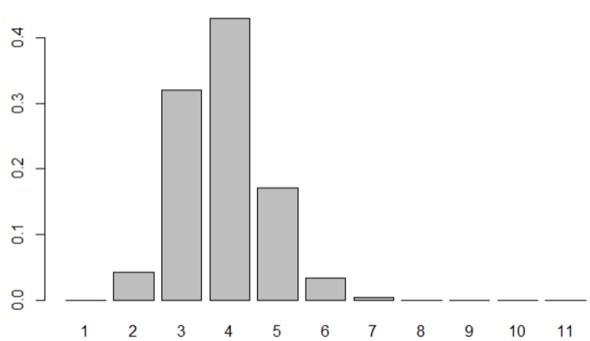
~~~~
```{R Political Blogs pathlength distribution}
paths_political <- path.length.hist(Political_gragh)$res
names(paths_political) <- 1:length(paths_political)
barplot(paths_political/sum(paths_political))
```
~~~~



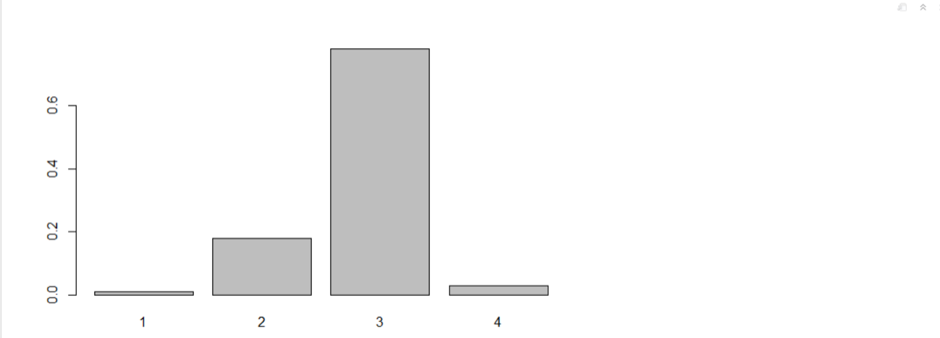

~~~~
```{R Neural Network Pathlength distribution}
paths_neural <- path.length.hist(Neural_graph)$res
names(paths_neural) <- 1:length(paths_neural)
barplot(paths_neural/sum(paths_neural))
```
~~~~



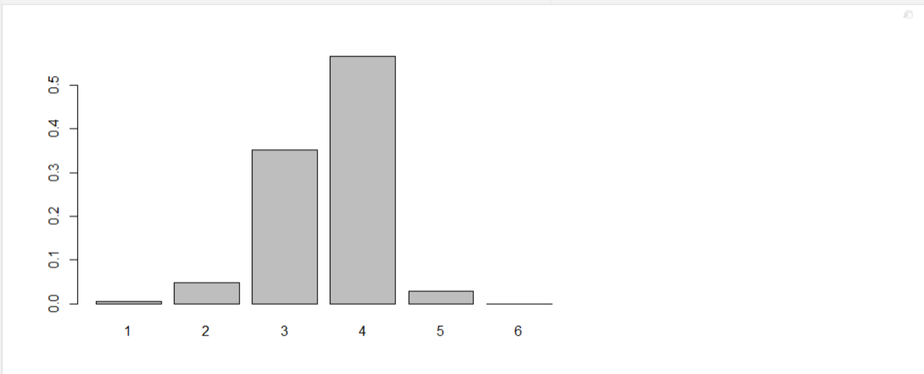

~~~~
```{R Internet pathlength distribution}
paths_internet <- path.length.hist(Internet_graph)$res
names(paths_internet) <- 1:length(paths_internet)
barplot(paths_internet/sum(paths_internet))
```
~~~~



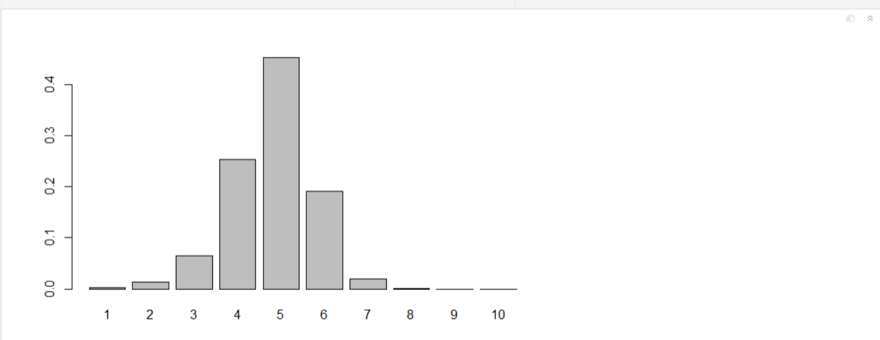
````
```{R Random graph1 pathlength distribution}
paths_randomgraph1 <- path.length.hist(Random_graph1)$res
names(paths_randomgraph1) <- 1:length(paths_randomgraph1)
barplot(paths_randomgraph1/sum(paths_randomgraph1))
```
````




````
```{R Random graph2 distribution}
paths_randomgraph2 <- path.length.hist(Random_graph2)$res
names(paths_randomgraph2) <- 1:length(paths_randomgraph2)
barplot(paths_randomgraph2/sum(paths_randomgraph2))
```
````




````
```{R Random graph3 distribution}
paths_randomgraph3 <- path.length.hist(Random_graph3)$res
names(paths_randomgraph3) <- 1:length(paths_randomgraph3)
barplot(paths_randomgraph3/sum(paths_randomgraph3))
```
````



Observation:

No matter the three selected real-world graphs or the three randomly generated graphs, the path length is distributed around 3 and 4, that is, if we randomly choose a pair of nodes, it has a shortest length which is most likely 3 or 4. In Random Graph 3, the most frequent shortest path length is 5, a little bit bigger than 3 and 4. In my perspective, it is because the edge existence probability between nodes is too low, this graph is kind of loose, so the most frequent shortest path length is bigger than other graphs.

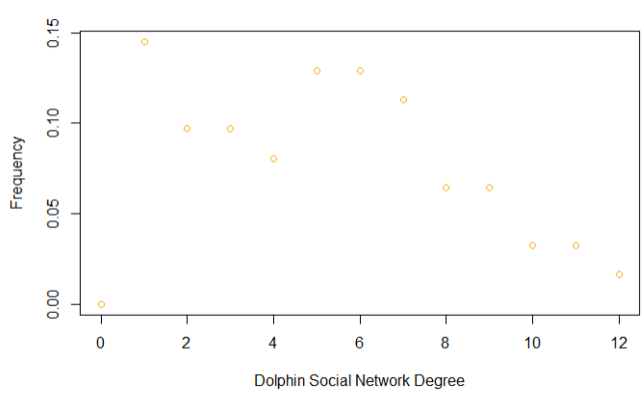
Question 4:
Description:
I found this graph from Mark Newman's collection of Network Data. I am interested in this topic which is about dolphin social network. Dolphin social network a undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand.
Reference:
D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behavioral Ecology and Sociobiology 54, 396-405 (2003)

````
```{R load Dolphin Social network}
Dolphin_network <- read_graph('Dolphins.gml', format='gml')
```
```{R Network characteristics}
data.frame( Network = c("Dolphin social network"),
            Type = c(is.directed(Dolphin_network)),
            n = c(vcount(Dolphin_network)),
            m = c(ecount(Dolphin_network)),
            c = c(count_components(Dolphin_network)),
            d = c(max(degree(Dolphin_network))),
            l = c(average.path.length(Dolphin_network)),
            L = c(diameter(Dolphin_network)),
            CCl = c(transitivity(Dolphin_network, type=c("localaverageundirected"))),
            CCg = c(transitivity(Dolphin_network,type=c("undirected"))))
```
````

Network <chr>	Type <lgl>	n <int>	m <dbl>	c <dbl>	d <dbl>	l <dbl>	L <dbl>	CCl <dbl>	CCg <dbl>
Dolphin social network	FALSE	62	159	1	12	3.356954	8	0.3029323	0.3087757

1 row

````
```{R Dolphin social network degree distribution}
plot(x=0:max(degree(Dolphin_network)),y=degree_distribution(Dolphin_network, cumulative = FALSE),
     xlab = "Dolphin Social Network Degree", ylab = "Frequency", col="orange")
```
````




````
```{R Dolphin Social Network pathlength distribution}
paths_Dolphins <- path.length.hist(Dolphin_network)$res
names(paths_Dolphins) <- 1:length(paths_Dolphins)
barplot(paths_Dolphins/sum(paths_Dolphins))
```
````