KATHMANDU UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB REPORT- 5

(COMP-314)

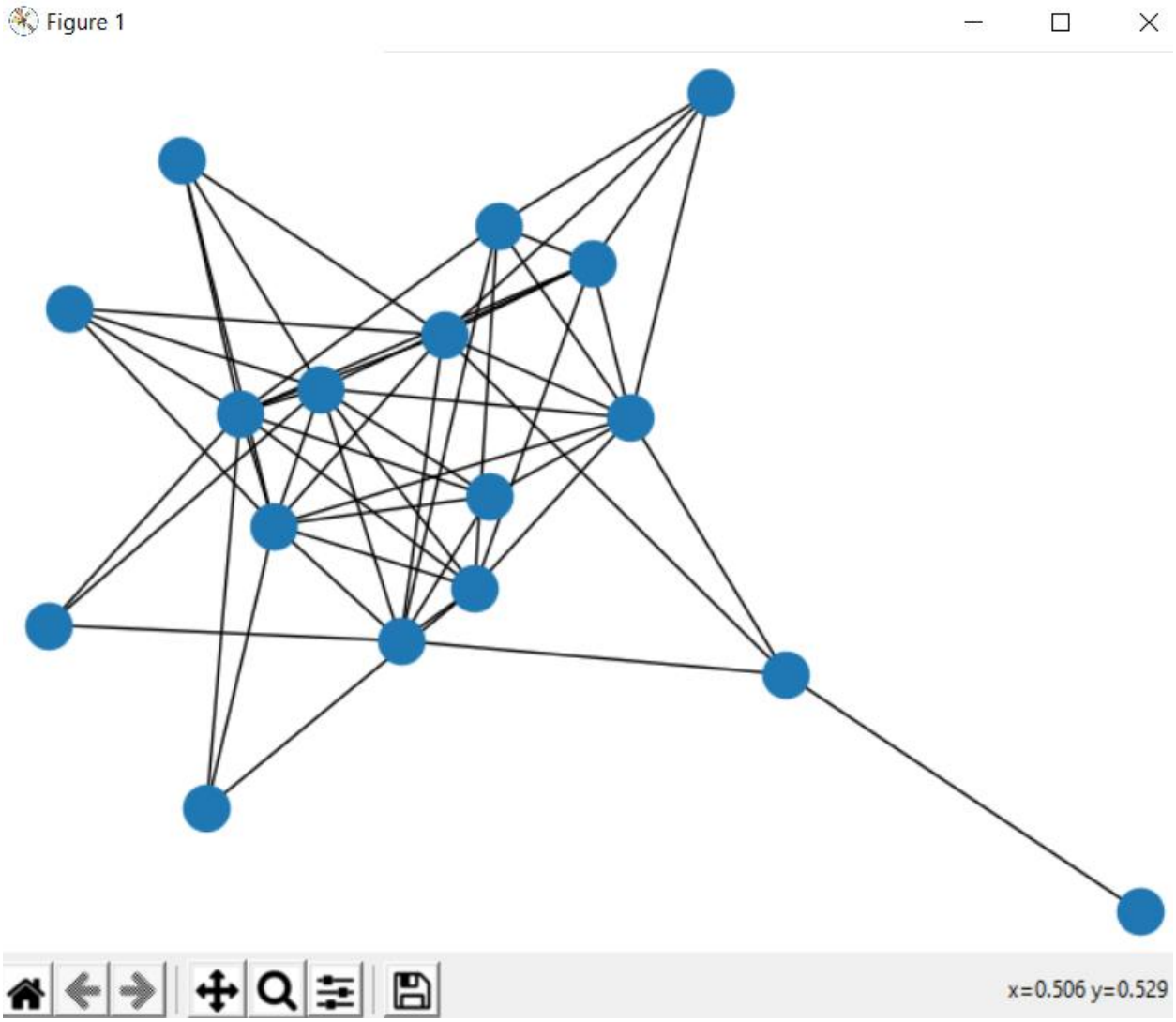
Submitted To:

Mrs. Rajani Chulyadyo

Submitted by:

Allen Maharjan (07)

1. **Download a network data from Network Repository5. Choose a small network, e.g. a graph with about 50 nodes and less than 1000 edges. There are several small networks in Animal Social Networks6 section. Import this graph using Networkx or another library. Draw this graph, e.g., using matplotlib. Then, compute the following network properties:**



```
(algorithm) E:\third year\Algorithm\Lab_5>python question1.py
The number of nodes of the given grahh is 12
The number of edges of the given graph is 36
The average degree of the graph is :6.0
The Density of the graph is: 0.5454545454545454
The diameter of the given graph is  5
The clustering coefficient of the graph is  0.6485449735449735
```

Source Code:

a. Number of nodes and edges:

```python
import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
import string
import pandas as pd
from statistics import mean
import scipy
class Graphdetail():

    def __init__(self,name):
            header_list = ["a","b","w"]
            E = pd.read_csv (name, sep= " ", header=None, names = header_list)
            graph = nx.from_pandas_edgelist(E, "a","b",["w"])
            print(f"The number of nodes of the given grahh is {graph.number_of_nodes()}\nThe number of edges of the given graph
```

b. Average degree:

```python
    def _average_degree(self,degree):
        filter_degree = []
        for i in degree:
            filter_degree.append(i[1])
        print(f"The average degree of the graph is :{(sum(filter_degree)/len(filter_degree))}")
```

c. Density:

```python
def _density(slef,edges,nodes):
    if nodes <1:
        return 0
    else:
        D = (2*edges)/(nodes*(nodes-1))
        print("The Density of the graph is:",D)
```

d. Diameter:

```
def _diameter(self,edges):
    temp = list(edges)
    BFS_dict = dict()
    temp_list = []
    for i,j in edges:
        for l,m in edges:
            if (i==l):
                temp_list.append(m)
        else:
            BFS_dict[i] = temp_list
            temp_list = []
    # print(BFS_dict)
    traverse = self._BFS_graph(BFS_dict,temp[0][0])
    for i in BFS_dict:
        if (traverse[-1] in BFS_dict[i]):
            a = i
            break
    return len((self._BFS_graph(BFS_dict,a)))+1
```

```
def _BFS_graph(self,graph,start):
    visited = []
    queue = [start]
    neighbours = []
    while queue:
        node = queue.pop(0)
        if node not in visited:
            visited.append(node)
            if node in graph:
                neighbours = graph[node]

            for i in neighbours:
                queue.append(i)
    return(visited)
```

e. Clustering Coefficient:

```
71      def _clustering(self,graph):
72          temp_list = []
73          clustering_list = []
74          for i in graph:
75              count = 0
76              a = list(graph.neighbors(i))
77              for j in a :
78                  for k in a:
79                      if j in graph.neighbors(k):
80                          count +=1
81              temp_list.append([count/2,len(a)])
82          for i,j in temp_list:
83              if j > 1:
84                  clustering_list.append(float((2*i)/(j*(j-1))))
85              else:
86                  clustering_list.append(0)
87          return float(mean(clustering_list))
```

2. **Download 5 other networks (with at least 5000 nodes) from Network Repository7 or Stanford Large Network Dataset Collection8.**

(a) Compute the network properties listed in Task 1 for these networks.

(b) Plot the degree distribution of all of these networks. The degree distribution P(k) of a network is the fraction of nodes in the network with degree k.

A. Answer:

```
(algorithm) E:\third year\Algorithm\Lab_5>python question2.py
The number of nodes of the given grahh is 17
The number of edges of the given graph is 53
The average degree of the graph is :6.235294117647059
The Density of the graph is: 0.3897058823529412
The diameter of the given graph is  3
The clustering coefficient of the graph is  0.5482344452932688


The number of nodes of the given grahh is 2617
The number of edges of the given graph is 2985
The average degree of the graph is :2.281238058846007
The Density of the graph is: 0.000872032897112388
The diameter of the given graph is  3
The clustering coefficient of the graph is  0.008205577480004866


The number of nodes of the given grahh is 636
The number of edges of the given graph is 3959
The average degree of the graph is :12.449685534591195
The Density of the graph is: 0.019605803991482196
The diameter of the given graph is  3
The clustering coefficient of the graph is  0.47123869727317824


The number of nodes of the given grahh is 118
The number of edges of the given graph is 5112
The average degree of the graph is :86.64406779661017
The Density of the graph is: 0.7405475880052151
The diameter of the given graph is  115
The clustering coefficient of the graph is  0.8200808040515938


The number of nodes of the given grahh is 1183
The number of edges of the given graph is 2155
The average degree of the graph is :3.643279797125951
The Density of the graph is: 0.003082301012796913
The diameter of the given graph is  987
The clustering coefficient of the graph is  0.8456343014027841
```

(B).Answer

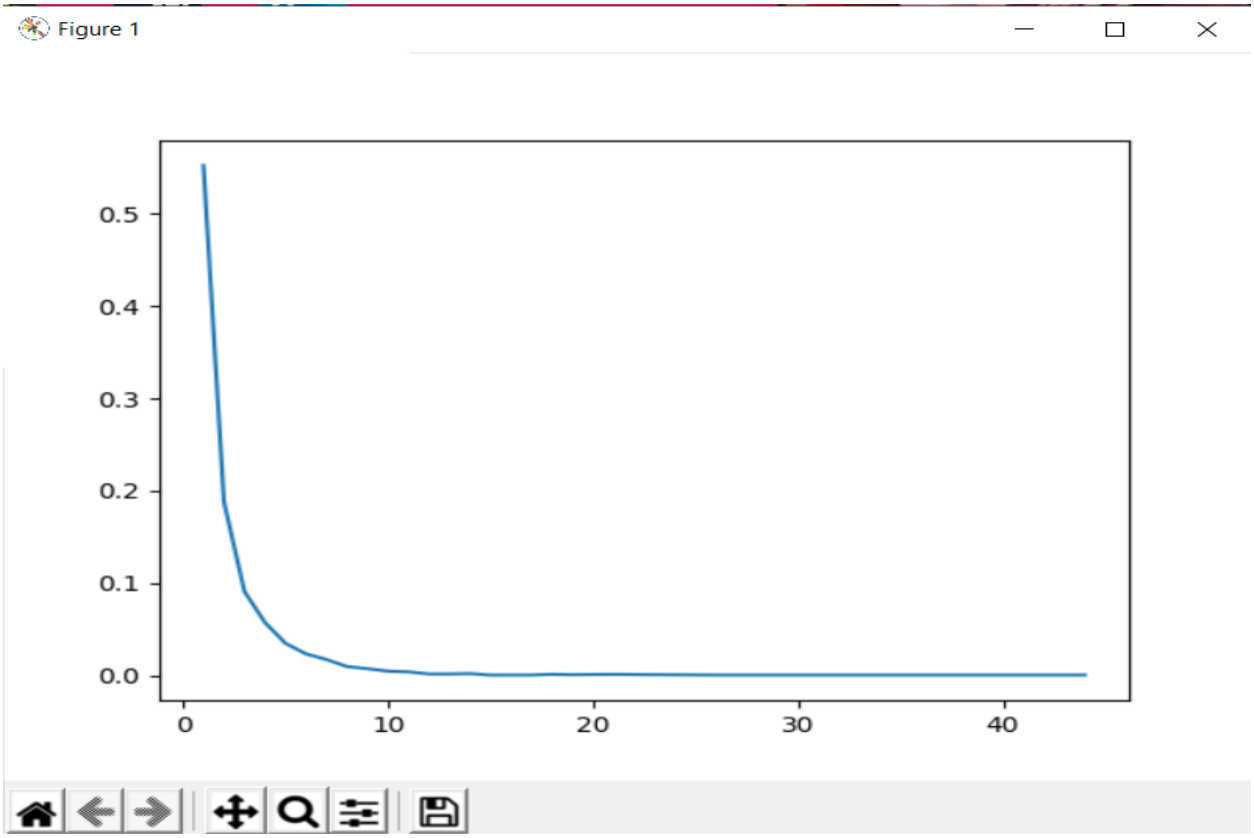


Figure 1
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40

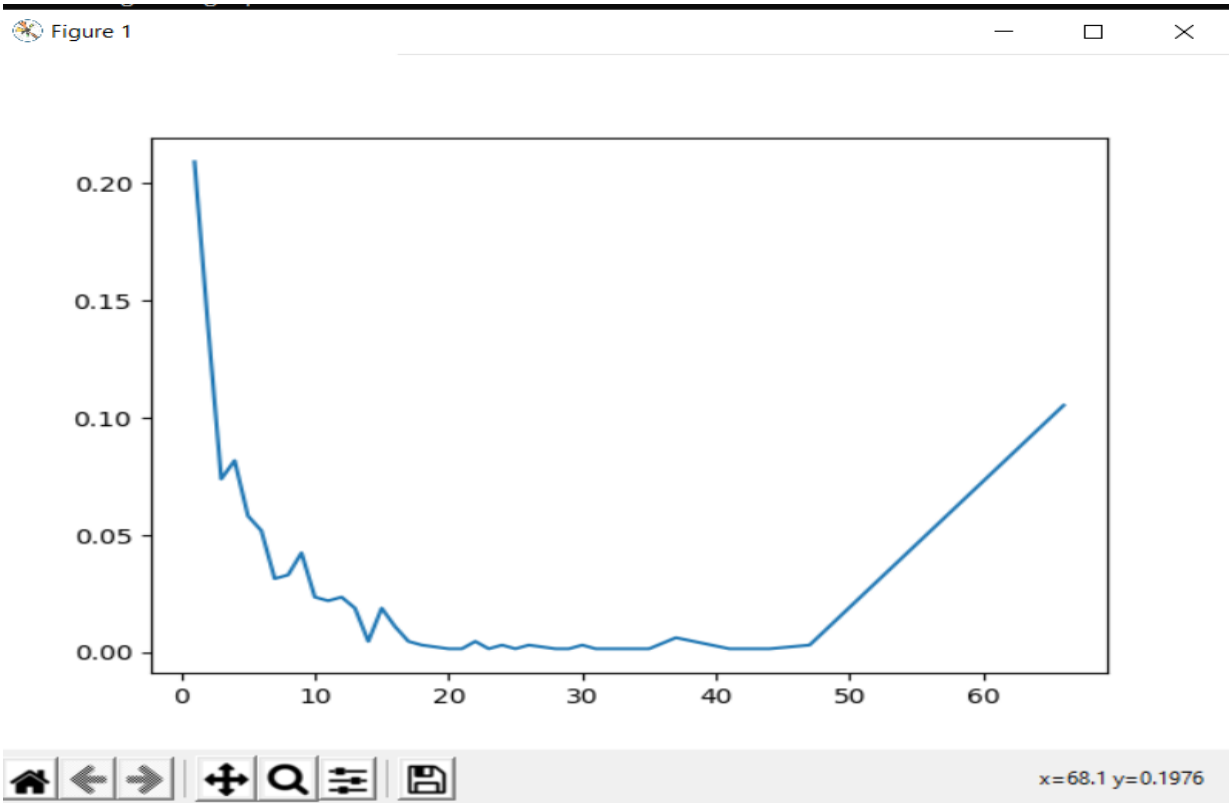


Figure 1
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
x=68.1 y=0.1976

Source code:

```
#selecting 5 graph

from Question1 import Graphdetail
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd

def plotting_graph(name):
    header_list = ["a","b","w"]
    E = pd.read_csv (name, sep= " ", header=None, names = header_list)
    graph = nx.from_pandas_edgelist(E, "a","b",["w"])
    degrees = graph.degree()
    new_tuple = []
    x_cord,y_cord = [],[]
    for i ,j in degrees:
        if j not in x_cord:
            x_cord.append(j)
    for i in x_cord:
        count = 0
        for j,k in degrees:
            if k == i:
                count = count+1
        else:
            y_cord.append(count/graph.number_of_nodes())
    for i,j in zip(x_cord,y_cord):
         new_tuple.append((i,j))
    new_tuple.sort(key=lambda x:x[0])
    plt.plot(*zip(*new_tuple))
    plt.show()


```

```
graph2 = Graphdetail("aves-barn-swallow-contact-network.edges")
plotting_graph("aves-barn-swallow-contact-network.edges")
graph3 = Graphdetail("bio-CE-HT.edges")
plotting_graph("bio-CE-HT.edges")
graph4 = Graphdetail("bio-SC-TS.edges")
plotting_graph("bio-SC-TS.edges")
graph5 = Graphdetail("insecta-ant-colony6-day29.edges")
plotting_graph("insecta-ant-colony6-day29.edges")
graph6 = Graphdetail("mammalia-voles-rob-trapping.edges")
plotting_graph("mammalia-voles-rob-trapping.edges")

```

From the network properties, what can you say about the networks you have selected?

Here we found that the number of edges in the graph1 and graph2 is 2985 and 3959 respectively. The number of nodes is 2617 and 636 respectively. The density of the graph is less i.e., 0.0008720 and 0.01960 which means our graph isn't diverse graph. We found clustering coefficient to be less i.e., 0.00820557 and 0.471238 which is less thus we can conclude that nodes in a graph don't tend to cluster together.

Did you find any pattern in the degree distributions of the networks? In any case, can you come to any conclusion about the networks from their degree distribution?

The relation between the degree of each node (K) to that of frequency of the degree of that node is negatively correlated.