Homework 04

MO412 - Network Science

Margarita Lacuaña Apaza
RA: 234924

April 2021

Write a program for the following task. Starting from a graph with 1000 nodes and no links, add links randomly, one at a time, until your graph has a connected component that comprises at least 501 nodes. What is the value of hki at this point? Repeat the experiment 50 times, and plot the distributiion of hki values

```
# ---------- importing libraries ---------- #
import networkx as nx
import matplotlib.pyplot as plt
import random
```

Basic functions

```
# ---------- Average degree ---------- #
def averageDegree(graph):
    nodes = len(list(graph.nodes))
    links = len(graph.edges)
    return (2*links)/nodes

# ---------- largest component ---------- #
def maxLenComponent(graph):
    return len(max(nx.connected_components(graph), key=
        len))
```

RandomGraph returns a random graph. If one component has at least 501 nodes the function finish and returns the graph. If the graph try with all possible node connections and there isn't a component with 501 nodes the function try again.

```
# ---------- random Graph ---------- #
def randomGraph(p,nodes,max_nodes):
    bk = False
```

```
    randomG = nx.Graph()
    randomG.add_nodes_from(range(1,nodes+1))
    component_size = maxLenComponent(randomG)
    rep = 0
    while(component_size<=max_nodes):
        rep += 1
        for i in range(1,nodes+1):
            for j in range(i+1,nodes+1):
                u = random.uniform(0, 1)
                if(u>p):
                    randomG.add_edges_from([(i, j)])
                    component_size = maxLenComponent(
                        randomG)
                    if(component_size>=max_nodes):
                        bk = True
                        break
            if(bk):
                break
    return randomG
```

Experiment this function returns a list with distribution of the $\langle k \rangle$ values of n(variable *repeat*) experiments. Where:

- ***p*** is p probability.
- ***max_nodes*** is the number of nodes.
- ***repeat*** is how many times the experiment is going to be repeated.

```
# ———————————— Experiment ———————————— #
def experiment(p,nodes,max_nodes,repeat):
    average_degrees = []
    for i in range(repeat):
        randomGr = randomGraph(p=p,nodes=nodes,max_nodes=
            max_nodes)
        average_degrees.append(averageDegree(randomGr))
    return average_degrees
```

.
Finally we run the code and plot the distribution of $\langle k \rangle$ values of the experiments.

```
p = random.random()
degress = experiment(p,nodes=1000,max_nodes=501,repeat
    =100)

plt.hist(degress,bins=20)
```

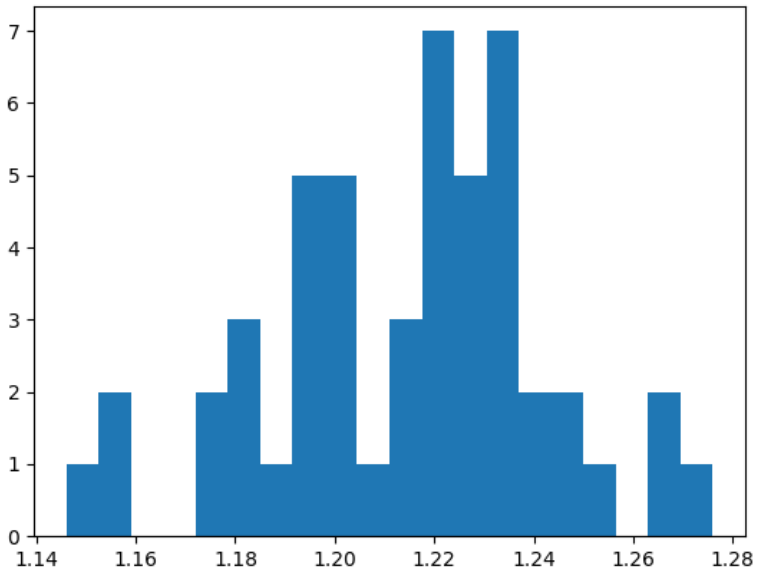
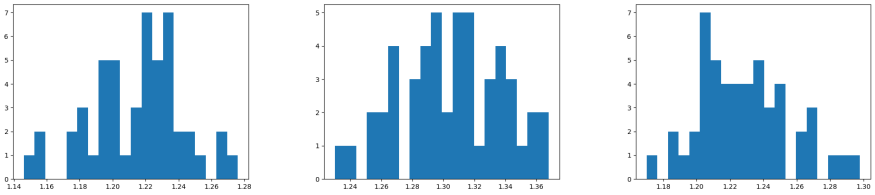


Figure 1: Distribution of $\langle k \rangle$ values

```
plt.show()
```



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

Figure 2: Three different experiments