# Networks Exercise 1 In 11.4 Version 1.2

zhangtao
ztao1991@gmail.com
github.com/ztao1991

February 21, 2017

## 1 Graph-N=10,p=0.5

### 1.1 Python Source

```
import math
import networkx as nx
import scipy.stats as stats
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

G = nx.random_graphs.erdos_renyi_graph(N,p)
plt.figure(3, figsize=(12, 12), dpi=720)
nx.draw_circular(G, with_labels=False, node_size=30)
plt.savefig(”G.png”)
plt.show()
```

### 1.2 G.png

Figure 1: Graph_N=10_p=0.5

## 2 Matrix-N=10

$$a_{ij} = 0, 1$$

```
A = nx.adjacency_matrix(G)
print(A.todense())
```

**A** = [[0 0 1 0 1 1 1 1 0 0] [0 0 1 0 1 0 0 0 0 1] [1 1 0 0 0 1 0 0 0 0] [0 0 0 0 0 1 0 0 0 0] [1 1 0 0 0 0 0 0 1 0] [1 0 1 1 0 0 1 1 1 1] [1 0 0 0 0 1 0 0 0 1] [1 0 0 0 0 1 0 0 1 0] [0 0 0 0 1 1 0 1 0 1] [0 1 0 0 0 1 1 0 1 0]]

## 3 Degree Distribution

### 3.1 Python Source

```
##Version 1.2
from scipy.optimize import curve_fit
degree = nx.degree_histogram(G)
k = range(len(degree))

#Fit
def poisson(k,lamb):
    return (lamb**k/factorial(k)) * math.exp(-lamb)
popt,pcov = curve_fit(poisson, k, y_1)

# Plot
y_1 = [z / float(sum(degree)) for z in degree]
y_2 = stats.poisson.pmf(k,popt[0])

plt.title('Degree␣Distribution')
plt.ylabel('Probability')
plt.xlabel('Degree')
plt.loglog(k,y_1,color="blue",linewidth=2,marker= 'o')
plt.loglog(k,y_2,color="red",linewidth=2,marker= 'o')
plt.savefig("Degree_Distribution.png")
plt.show()
```

## 3.2 Degree Distribution-N=1000,p=0.1

Figure 2: Degree Distribution-N=1000,p=0.1

## 3.3 Summary

…When

$$N \to \infty, p \to 0$$

the Degree-Distribution approximately poisson.