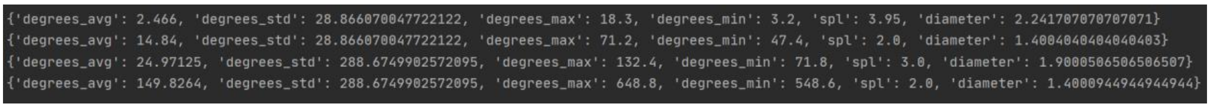
HW1

1.4

```
a = random_networks_generator(100, 0.1, False, 20)
b = random_networks_generator(100, 0.6, False, 20)
c = random_networks_generator(1000, 0.1, False, 10)
d = random_networks_generator(1000, 0.6, False, 10)
gnp_nets = [a,b,c,d]

for i in gnp_nets:
    print(networks_avg_stats(i))
```



```
{'degrees_avg': 2.466, 'degrees_std': 28.866070047722122, 'degrees_max': 18.3, 'degrees_min': 3.2, 'spl': 3.95, 'diameter': 2.241707070707071}
{'degrees_avg': 14.84, 'degrees_std': 28.866070047722122, 'degrees_max': 71.2, 'degrees_min': 47.4, 'spl': 2.0, 'diameter': 1.4004040404040403}
{'degrees_avg': 24.97125, 'degrees_std': 288.6749902572095, 'degrees_max': 132.4, 'degrees_min': 71.8, 'spl': 3.0, 'diameter': 1.9000506506506507}
{'degrees_avg': 149.8264, 'degrees_std': 288.6749902572095, 'degrees_max': 648.8, 'degrees_min': 548.6, 'spl': 2.0, 'diameter': 1.4000944944944944}
```

1.5

The greater the probability value, the greater the number of links.

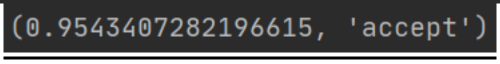
The statistics increased by every value, besides, diameter and spl (will be decrease).

Since more nodes are connected.

If the number of the nodes is increased, the statistics increased by every value, besides, diameter and spl.

2.3

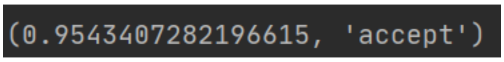
```
p = most_probable_p(rand_net[0])
print(rand_net_hypothesis_testing(rand_net[0], p))
```



```
(0.9543407282196615, 'accept')
```

2.4

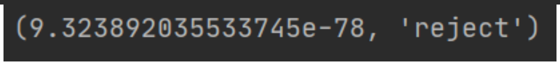
```
print(rand_net_hypothesis_testing(rand_net[0], p))  #theoretical_p 10% down
```



```
(0.9543407282196615, 'accept')
```

Still accept.

```
print(rand_net_hypothesis_testing(max(rand_net,key= len), p)) #much bigger 
network
```



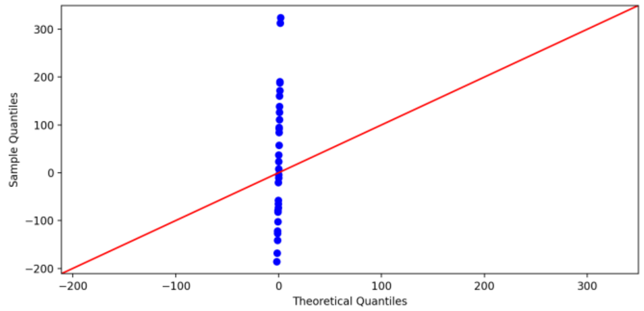
```
(9.323892035533745e-78, 'reject')
```

In bigger network, returned reject.

The range of the mistake in the P Value is increase the bigger the net is get.

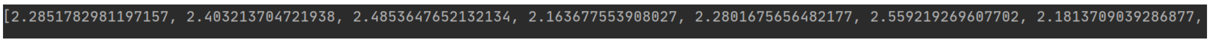
2.5

```
test = 
np.random.normal(network_stats(rand_net[0])['degrees_avg'],network_stats(ra
nd_net[0])['degrees_std'], 30)
sm.qqplot(test, line= '45')
py.show()
```

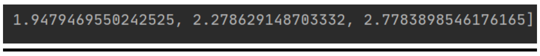


3.3

```
network_gammas = []
for i in range(len(scalefree_nets)):
    network_gammas.append(find_opt_gamma(scalefree_nets[i]))
print(network_gammas)
```

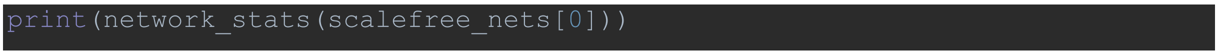


```
[2.2851782981197157, 2.403213704721938, 2.4853647652132134, 2.163677553908027, 2.2801675656482177, 2.559219269607702, 2.1813709039286877,
```

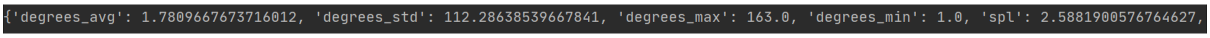


```
 1.9479469550242525, 2.278629148703332, 2.7783898546176165]
```

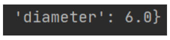
3.4



```
print(network_stats(scalefree_nets[0]))
```



```
{'degrees_avg': 1.7809667673716012, 'degrees_std': 112.28638539667841, 'degrees_max': 163.0, 'degrees_min': 1.0, 'spl': 2.5881900576764627,
```

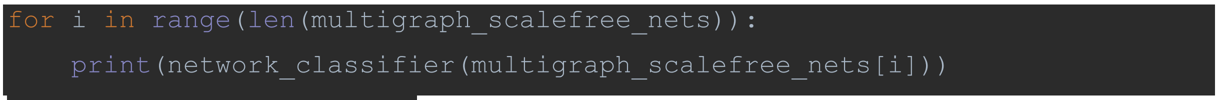


```
 'diameter': 6.0}
```

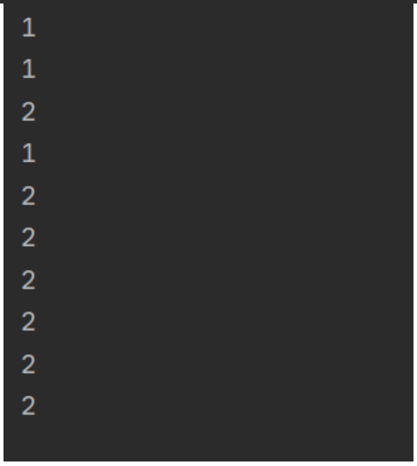
Yes it make sense, because in Scale-Free Networks we are expecting to see results that point on Hubs in the net.

Bigger max degree, std and lower spl compared to Random Networks

4.3



```
for i in range(len(multigraph_scalefree_nets)):
    print(network_classifier(multigraph_scalefree_nets[i]))
```



```
1
1
2
1
2
2
2
2
2
2
```