Quiz 12

1. Plot (a) Erdos-Renyi graph and (b) Barabasi-Albert graph of 10,000 nodes and about 50,000 edges.

2. Show the following metrics of the above both graphs.

- Number of nodes
- Number of edges
- Average degree
- Number of connected components
- Number of triangles
- Transitivity (clustering coefficient)
- Maximum degree
- Minimum degree

- Submit from Tokyo Tech OCW-i
- Deadline: ??:??(Japan Standard Time) on Jan. 27(Sun)
- Files should be MS Word, PDF or Zipped Jupyter notebook.

degree distribution with R+igraph

synthetic network based on Barabasi-Albert model (n = 10000, c=3, undirected)

```
> library(igraph)
> ba <- barabasi.game(10000,m=3,directed =
FALSE)
> summary(ba)
```

summary of the network

```
Vertices: 10000
Edges: 29997
Directed: FALSE
No graph attributes.
No vertex attributes.
No edge attributes.
> no.clusters(ba)
```

the number of clusters

```
[1] 1
> average.path.length(ba)
[1] 3.170916
```

average path length

```
> transitivity(ba)
```

clustering coefficient

```
[1] 0.001810504
> mean(degree(ba))
```

average degree

```
[1] 5.9994
> max(degree(ba))
```

the maximum degree

```
[1] 4184
> min(degree(ba))
```

the minimum degree

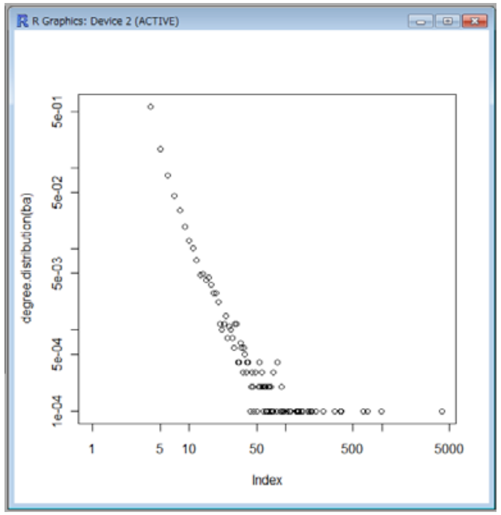
```
[1] 3
> power.law.fit(degree(ba))
Loading required package: stats4
Call:
```

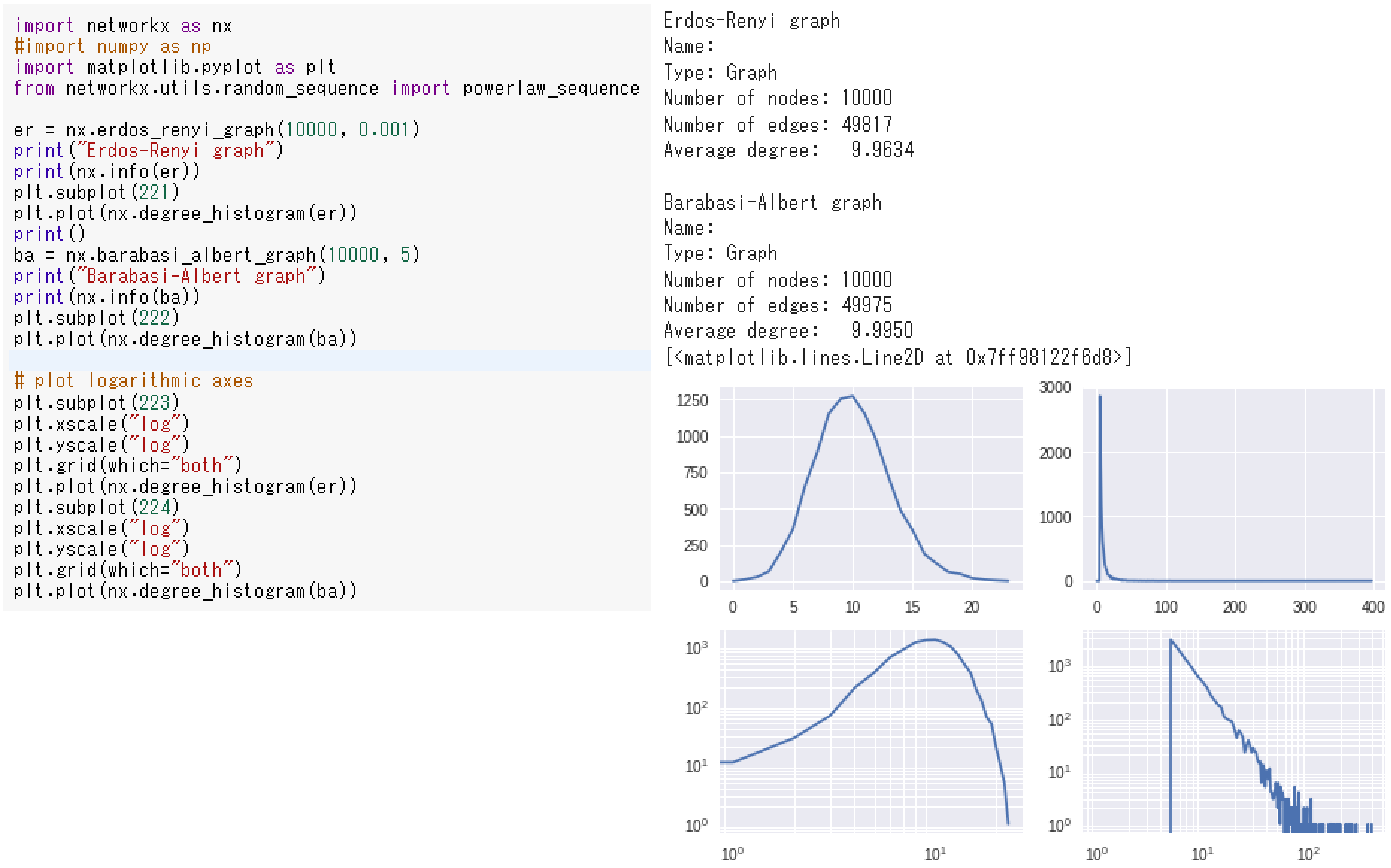
fits a power-law distribution

```
mle(minuslogl = mlogl, start = list(alpha = start))
Coefficients:
  alpha
3.155893
```

power-law distribution

```
> plot(degree.distribution(ba),log="xy")
```





```
import networkx as nx
#import numpy as np
import matplotlib.pyplot as plt
from networkx.utils.random_sequence import powerlaw_sequence

er = nx.erdos_renyi_graph(10000, 0.001)
print("Erdos-Renyi graph")
print(nx.info(er))
plt.subplot(221)
plt.plot(nx.degree_histogram(er))
print()
ba = nx.barabasi_albert_graph(10000, 5)
print("Barabasi-Albert graph")
print(nx.info(ba))
plt.subplot(222)
plt.plot(nx.degree_histogram(ba))

# plot logarithmic axes
plt.subplot(223)
plt.xscale("log")
plt.yscale("log")
plt.grid(which="both")
plt.plot(nx.degree_histogram(er))
plt.subplot(224)
plt.xscale("log")
plt.yscale("log")
plt.grid(which="both")
plt.plot(nx.degree_histogram(ba))
```

```
Erdos-Renyi graph
Name:
Type: Graph
Number of nodes: 10000
Number of edges: 49817
Average degree:   9.9634

Barabasi-Albert graph
Name:
Type: Graph
Number of nodes: 10000
Number of edges: 49975
Average degree:   9.9950
[<matplotlib.lines.Line2D at 0x7ff98122f6d8>]
```