# Librarys Import

In [243…

```
#Import librarys
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
```

# Importing data from a .tsv file

In [244…

```
#Open and extract the network edge list from the tsv file.
fh=open("/content/net1 - net1.tsv", 'rb')
GN1=nx.read_edgelist(fh)
fh.close()

fh=open("/content/net2 - net2.tsv", 'rb')
GN2=nx.read_edgelist(fh)
fh.close()
```

# Suport Functions

### degree_histogram_plot_x_and_y: Get the probability from all elements from the network

### cumulative_function: Get the Cumulative values for the probabilities

In [245…

```
def degree_histogram_plot_x_and_y(g, normalized=True):
    aux_y = nx.degree_histogram(g)
    aux_x = np.arange(0,len(aux_y)).tolist()
    return aux_x[1:], aux_y[1:]
```

In [246…

```
def cumulative_function(network_y):
  network_y_list = []
  for i in range(len(network_y)):
    network_y_list.append(sum((network_y[0:i])))
  return network_y_list
```

# Get the probabilities and the cumulative values to the network

In [247…

```
hist_GN1_x, hist_GN1_y = degree_histogram_plot_x_and_y(GN1)
hist_GN2_x, hist_GN2_y = degree_histogram_plot_x_and_y(GN2)

cumulative_value_GN1 = cumulative_function(hist_GN1_y)
cumulative_value_GN2 = cumulative_function(hist_GN2_y)
```

# For the purpose of study, various types of graphs will be plotted, namely:

## Linear (must be avoided)

## Log-Log binning (With only 1 bin, due to the little data)

## Cumulative log-log

---

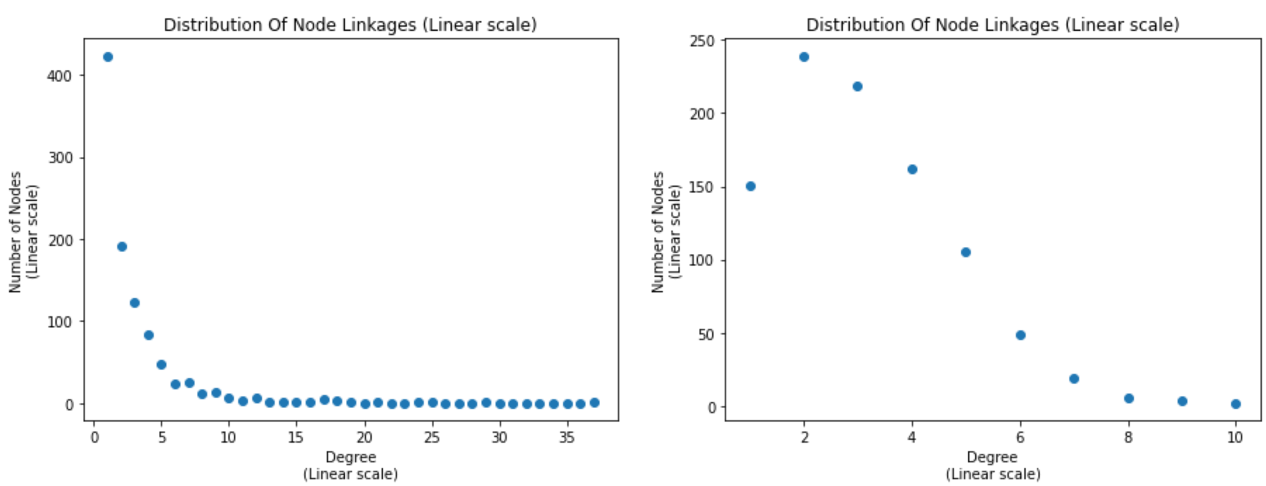
## Linear (must be avoided)

In [248…

```
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
plt.plot(hist_GN1_x, hist_GN1_y,'o')
plt.title('\nDistribution Of Node Linkages (Linear scale)')
plt.xlabel('Degree\n(log scale)')
plt.ylabel('Number of Nodes\n(log scale)')

plt.subplot(1, 2, 2)
plt.plot(hist_GN2_x, hist_GN2_y,'o')
plt.title('\nDistribution Of Node Linkages (Linear scale)')
plt.xlabel('Degree\n(log scale)')
plt.ylabel('Number of Nodes\n(log scale)')

plt.show()
```




.

.

.

.

.

.

.

.

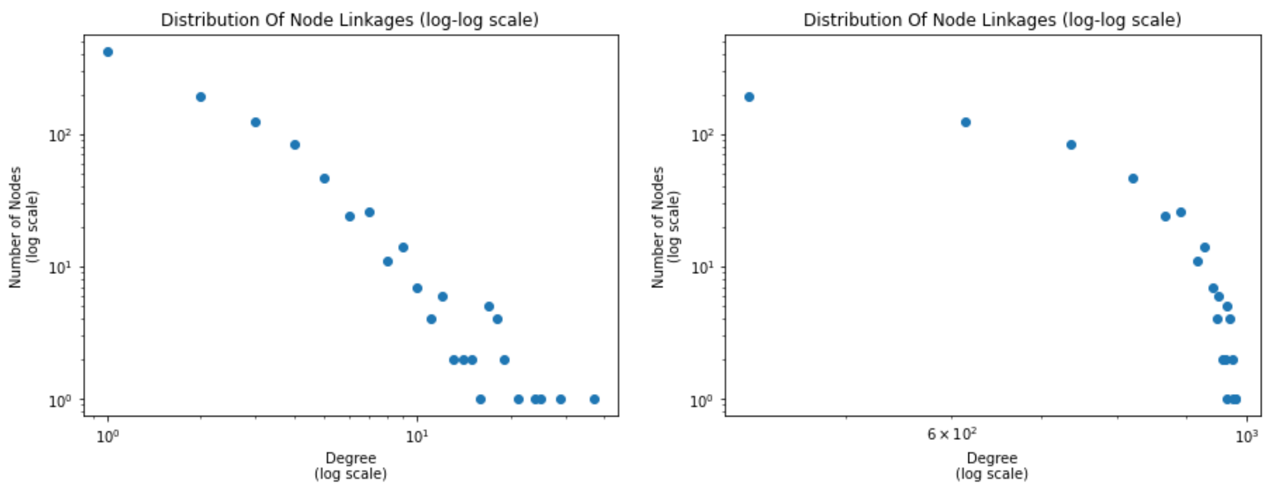
## Log-Log binning (With only 1 bin, due to the little data)

In [249…

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 1)
plt.plot(hist_GN1_x, hist_GN1_y, 'o')
plt.title('\nDistribution Of Node Linkages (log-log scale)')
plt.xlabel('Degree\n(log scale)')
plt.ylabel('Number of Nodes\n(log scale)')
plt.xscale("log")
plt.yscale("log")

plt.subplot(1, 2, 2)
plt.plot(cumulative_value_GN1,hist_GN1_y,'o')
plt.title('\nDistribution Of Node Linkages (log-log scale)')
plt.xlabel('Degree\n(log scale)')
plt.ylabel('Number of Nodes\n(log scale)')
plt.xscale("log")
plt.yscale("log")

plt.show()
```




.

.

.

.

.

.

.

.

.

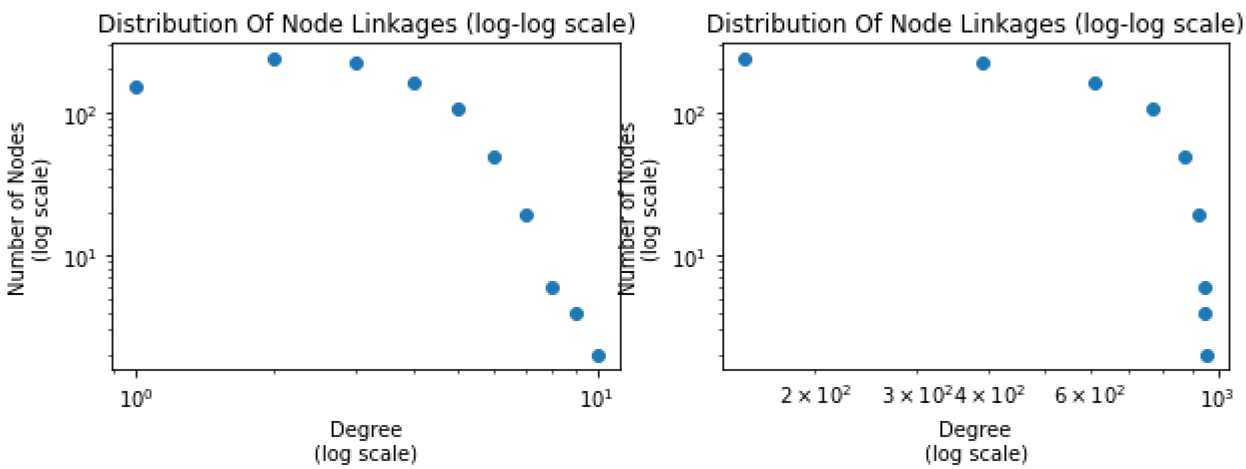
.

## Cumulative log-log

In [251…

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 1)
plt.plot(hist_GN2_x, hist_GN2_y, 'o')
plt.title('\nDistribution Of Node Linkages (log-log scale)')
plt.xlabel('Degree\n(log scale)')
plt.ylabel('Number of Nodes\n(log scale)')
plt.xscale("log")
plt.yscale("log")

plt.subplot(1, 2, 2)
plt.plot(cumulative_value_GN2,hist_GN2_y,'o')
plt.title('\nDistribution Of Node Linkages (log-log scale)')
plt.xlabel('Degree\n(log scale)')
plt.ylabel('Number of Nodes\n(log scale)')
plt.xscale("log")
plt.yscale("log")
```




---

# Which of the two is more likely to be a scale-free network?

Net 1 follows a power low, identifying itself as a scale-free network.