P1 - Graph Data Mining

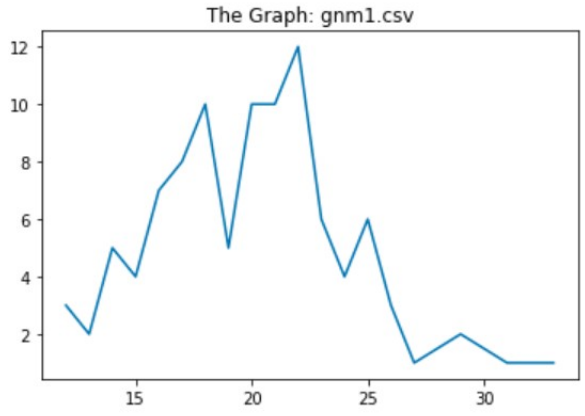
Vivek Reddy Karri (ID: 200315262)

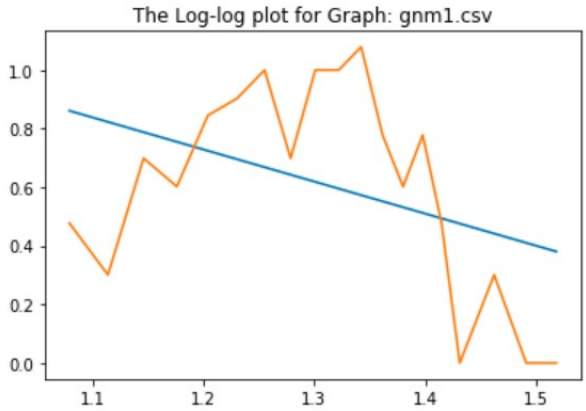
Network Properties in Spark GraphFrames

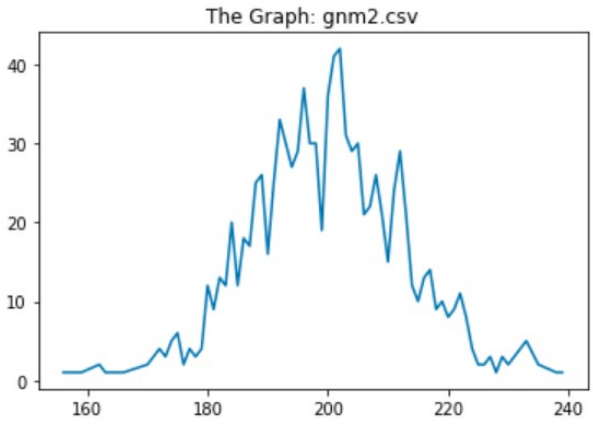
September 22, 2019

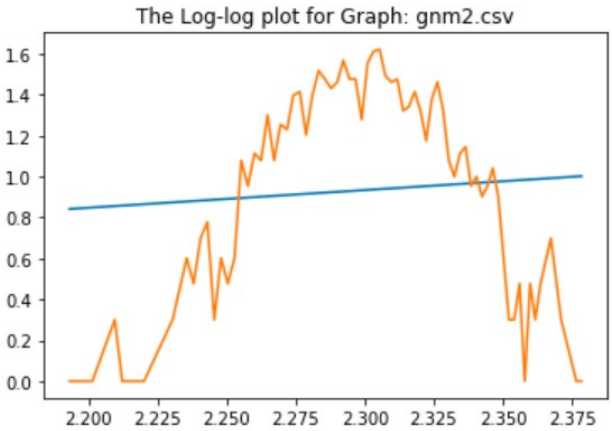
Question 1: Generate a few random graphs. You can do this using networkx's random graph generators. Do the random graphs you tested appear to be scale free?

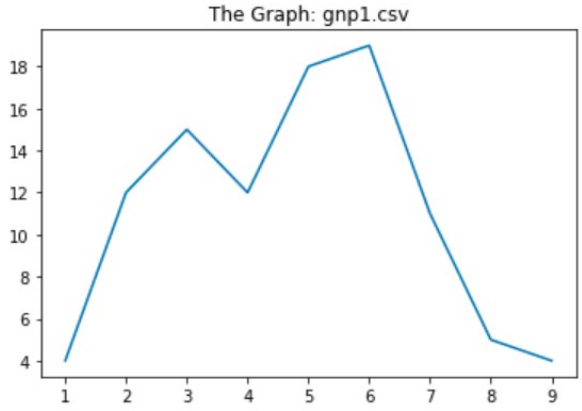
Ans: The four random graphs generated gnp1, gnp1, gnm1, gnm2 follows a near gaussian distribution but not the power law and hence are not scale free. This was inferred from looking at the degree distribution plots and also applying least squares to log-log plot of degree distribution. The log-log plot wasn't linear as the coefficient of determination (R^2) was in the range of (0.01-0.06) and by these evidences, it was inferred that these graphs are not scale free.

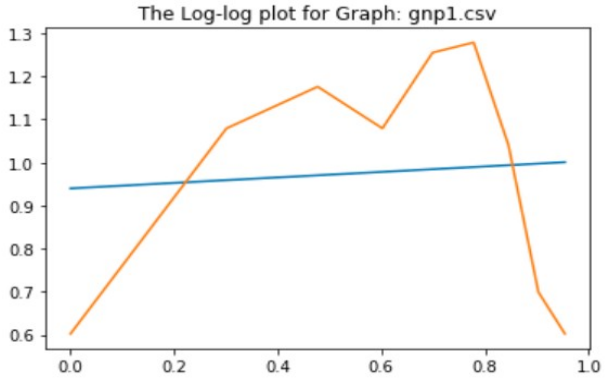


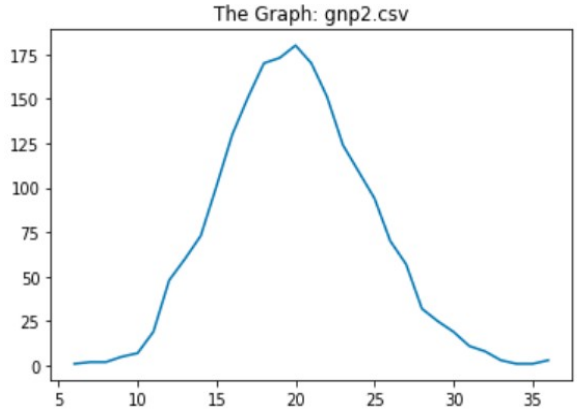


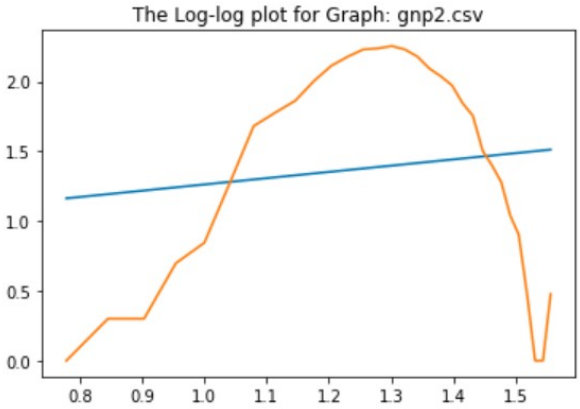






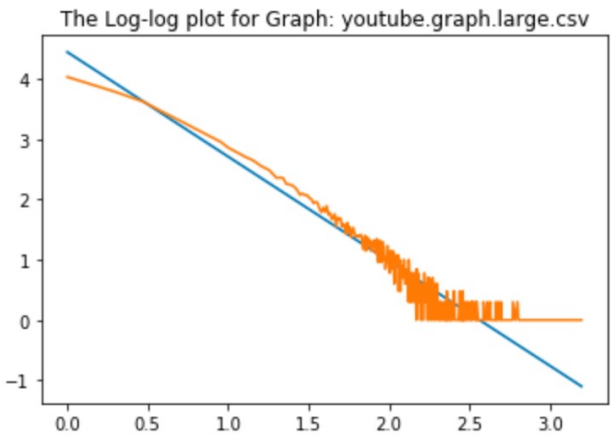


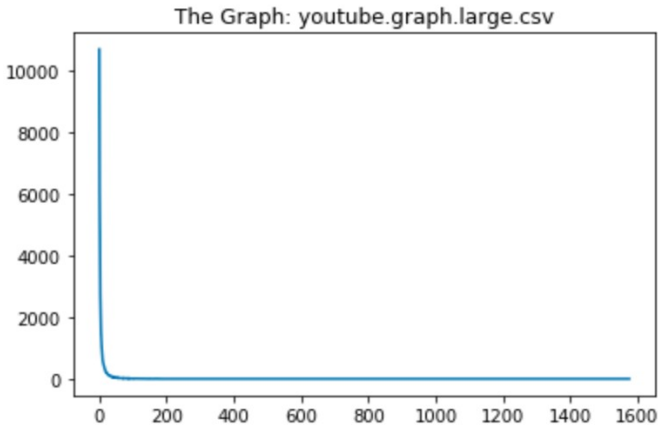


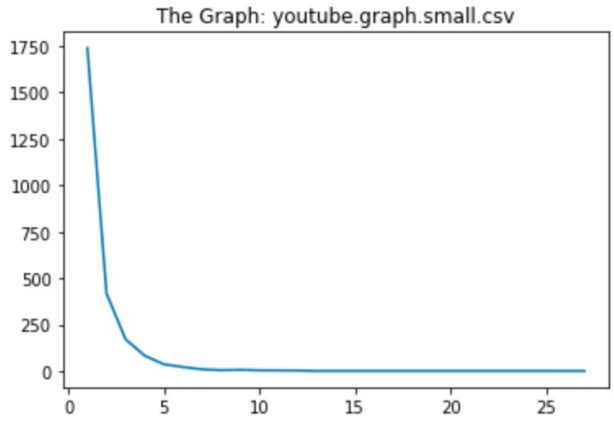


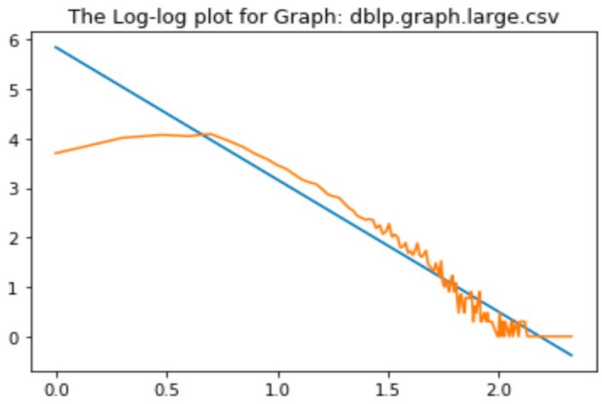
Question 2: Do the Stanford graphs provided to you appear to be scale free?

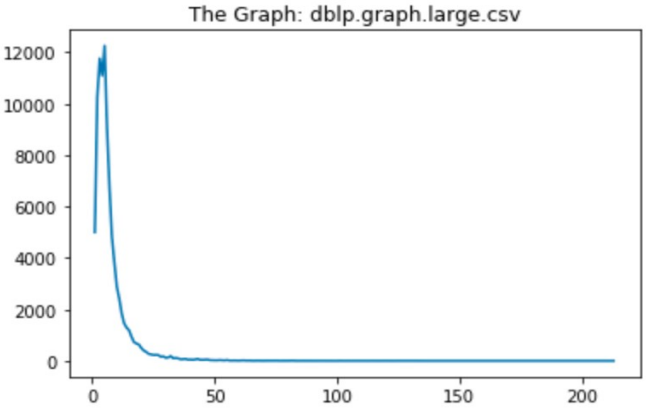
Ans: Clearly, **youtube.graph.large**, **youtube.graph.small, dblp.graph.large** are scale free, they appear to have exponential decay trend in normal plots and have a decent fit in a log-log plots with R^2 values 0.90, 0.96 and 0.92 respectively. The decay parameter has the values of 2.60, 2.54 and 2.67 which are in expected lines and are inferred to be scale free.



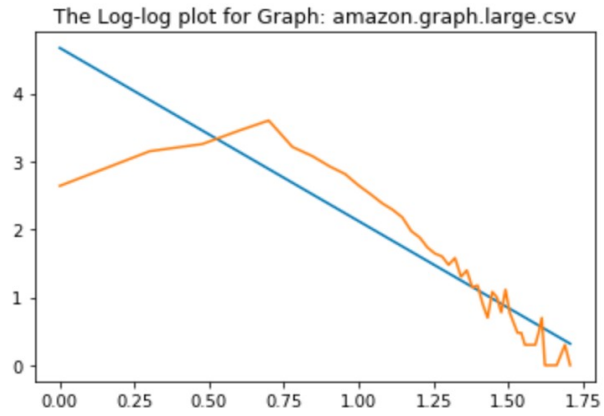


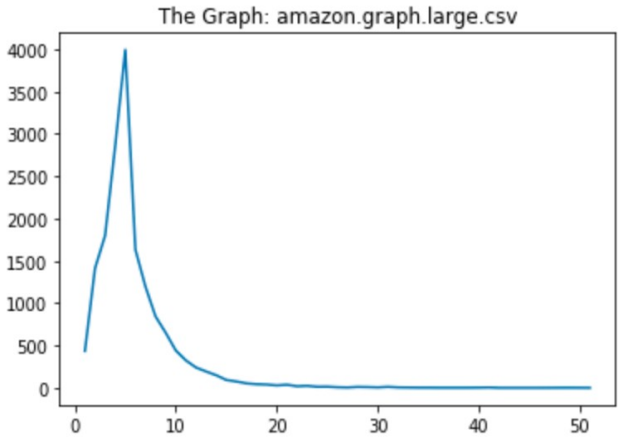




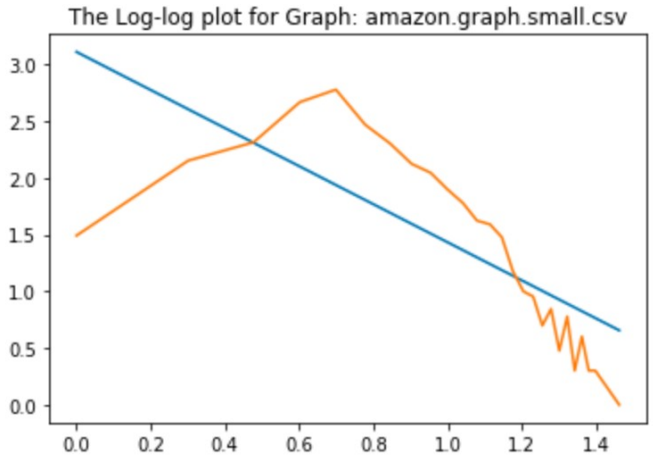


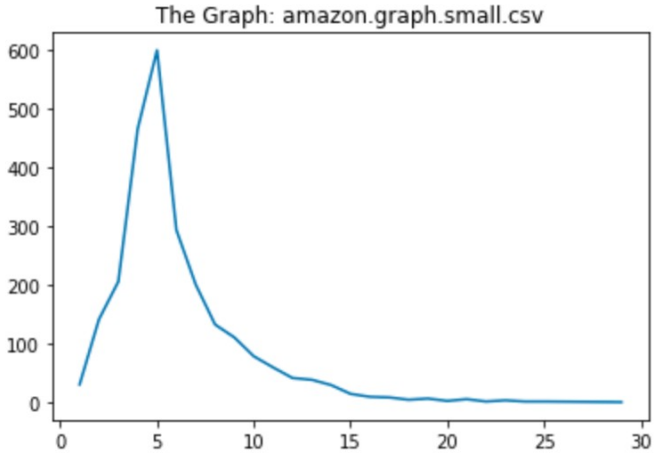
Although, **amazon.graph.large** has comparitively low fit (0.81 R^2) over the other three graphs stated above, the graph appears to have exponential decay with a coefficient of 2.54 after a slight rise at the initial degrees. Hence i'm inferring that this graph is following power law and is scale free.

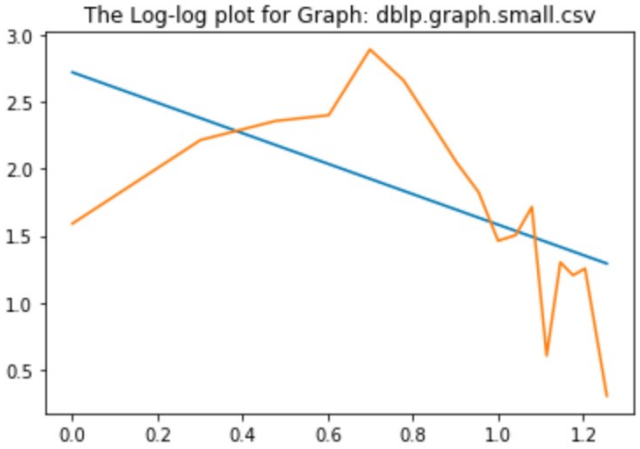


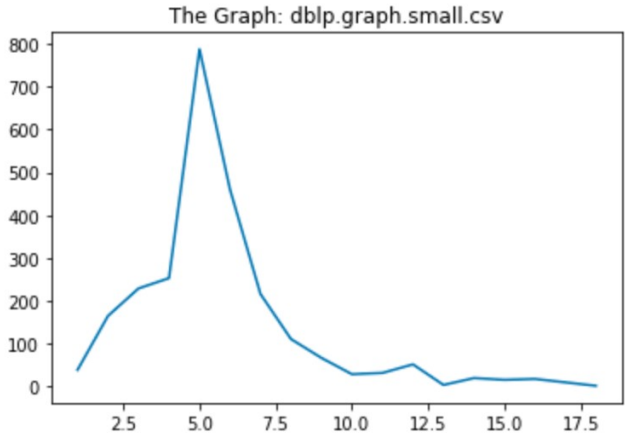


The graphs **dblp.graph.small** and **amazon.graph.small** have plots whom appear to be folowing normal distribution than power law. They have R^2 values 0.31 and 0.55 respectively (definitely did'nt fit the line well) and also has the coefficient values to be 1.14 and 1.67. Also, these graphs are sampled from their larger counter parts which appear to follow power law and the samples following power law is a bit too much to expect and hence i'm inferring that these two are not scale free.









Question 3: Consider a small network of 10 computers, illustrated below, with nodes representing computers and edges representing direct connections of two machines. If two computers are not connected directly, then the information must flow through other connected machines.

3.1 Rank the nodes from highest to lowest closeness centrality.

F	0.071428571428571
C	0.071428571428571
D	0.066666666666667
H	0.066666666666667
E	0.058823529411765
B	0.058823529411765
A	0.055555555555556
G	0.055555555555556
I	0.047619047619048
J	0.03448275862069

3.2 Suppose we had some centralized data that would sit on one machine but would be shared with all computers on the network. Which two machines would be the best candidates to hold this data based on other machines having few hops to access this data?

Ans: **F** and **C** would be two ideal condidates to hold this data. Since both have a Closeness Centrality of 0.0714 i.e higher than other nodes in the graph. So, the overall number of hops required to access the data from any other machine would be low, if the data sits on F or C.

Question 4: In this example, which members should have been targeted to best disrupt communication in the organization?

Ans: Usman Bandukra
Raed Hijazi
Nawaf Alhazmi
Djamal Beghal
Essid Sami Ben Khemais
Mohamed Atta
Mamoun Darkazanli

These members are found out to be the articulation points in the graph and hence these members should be targeted to dirupt the communication.

Instructions to run the Power_Law_Check script:
1) To generate the csv files, run this script given below adjusting for versions and paths depending on your machine.
2) All these programs were run using **Python 2.7, Spark 2.4, Graphframes-0.7.0.**
3) Change paths in Power_Law_check.ipynb and point them towards the generated csv files and run thr ipynb file.

Other:
You may try to run this code on other versions of Spark and Graphframes (1.6 and 0.1.0 as in vcl),. But it may break
1) unionAll in Spark 1.6 was changed to union in Saprk 2.4. (one such identified change)

I do although have a degree.py file written for Spark 1.6 and could send that if required.

A Script to automatically run the degree.py over all the graphs:
Bash Script:

```
FILES=~/GDM_P1/stanford_graphs/*.large
export ROOT=~/GDM_P1/stanford_graphs/
for f in $FILES
do
	$SPARK_HOME/bin/spark-submit --jars graphframes-0.7.0-spark2.4-s_2.11.jar GDM_P1/degree.py $f large
done

FILES=$ROOT/*.small
for f in $FILES
do
	$SPARK_HOME/bin/spark-submit --jars graphframes-0.7.0-spark2.4-s_2.11.jar GDM_P1/degree.py $f
done
```