1. Growing networks in NetLogo: preferential and random growth

   Download the NetLogo applet:
   http://projects.si.umich.edu/netlearn/NetLogo4/RAndPrefAttachment.html

   - Open the model. Click on 'setup' to start out with a cycle of 5 vertices. Click on 'Run' to add vertices one by one, each with m edges. For each of the m-edges:
   - With probability gamma the endpoint is chosen randomly
   - With probability (1-gamma) the endpoint is chosen preferentially according to degree
     - Play with the parameters. Then select m=1 and gamma=0. Add 300 vertices. Click on the 'resize nodes' button to size the vertices by their degree. Repeat the same, but with m=1 and gamma=1. What differences do you observe between the two networks, .e.g. in terms of appearance, the number of vertices with degree 1, and the maximum degree of any vertex?
   - Next generate two networks with 1000 vertices and m=4. (you can run this faster by adjusting the speed slider at the top). For one network select gamma = 0, and for the other gamma = 1. Which degree distribution looks more like a power-law?

2. Generating and fitting power law distributions

   If you are using Matlab to solve this problem please refer to http://www-personal.umich.edu/~ladamic/courses/si614w06/matlab/ for instructions on how to use the various functions, which can be downloaded there. Feel free to use software other than Matlab for this task.

   - Generate 100,000 random integers from a power law distribution with exponent alpha = 2.1
   - What is the largest value in your sample? Is it possible for a node in a network to have a degree this high (assuming you don't allow multiple edges between two nodes)?
   - Construct a histogram of the frequency of occurrence of each integer in your sample. Try both a linear scale plot and a log-log scale plot.
   - What happens to the bins with zero count in the log-log plot?
   - Try a simple linear regression on the log transformation of both variables (if you are using the fitlineonloglog.m Matlab script, you will feed it the binned data, and it will take the log of the x and y for you before doing a linear fit). What is your value of the power-law exponent alpha? Include a plot of the data with the fit superimposed.
   - Now exponentially bin the data and fit with a line. What is your value of alpha?
   - Finally, do a cumulative frequency plot of the original data sample. Fit, plot, and report on the fitted exponent and the corresponding value of alpha.
   - Which method was the most accurate? Which one, in your opinion, gave the best view of the data and the fit?

3. Scale free networks

We're back to Pajek here. There is a Matlab component that you can choose to do in another application.

- Generate a scale-free random network using Pajek's Net>Random Network>Scale Free command. Make the network undirected, with 1000 vertices and an average degree of 4. Select alpha = 0.3 (which will then fix beta), and choose a fully connected network of three vertices (m0 = 3, p = 0.9999) to start with. Let Pajek draw a layout for you (whatever you prefer). Describe briefly a few characteristics you observe about the network. Note that here alpha is the weight given to the degree of the vertex in the model, not the power law exponent!.
- Now generate a larger network of 10,000 with the same parameters (no need to lay it out, unless you want to make Pajek sweat!)
- Calculate the degree of each vertex (Net>Partitions>Degree>All) and save the values to a file by clicking on the save icon next to the partition name (it will be a .clu file).
- Read the (.clu) file into Matlab using the hdrload function.
- Plot the degree frequency distribution on a log-log plot.
- Now fit the distribution using your favorite method from task 2.
- Go back and generate another scale-free network by changing the parameter alpha to a much lower value: alpha = 0.1. Set the network size to 10,000 vertices and repeat the degree distribution fitting procedure above. What do you observe about the degree distribution? Why is lowering the value of alpha having this effect? When fitting, what is the power-law exponent? What value of xmin did you choose?