Random Graphs

Thomas Breuel

UniKL

Random Graphs

Real-world graphs are often generated in some random fashion:

- social networking users meet randomly and “friend” each other
- social networking users randomly discover friends-of-friends
- people randomly meet on the street
- ...

That is, many graphs are created from underlying random processes.

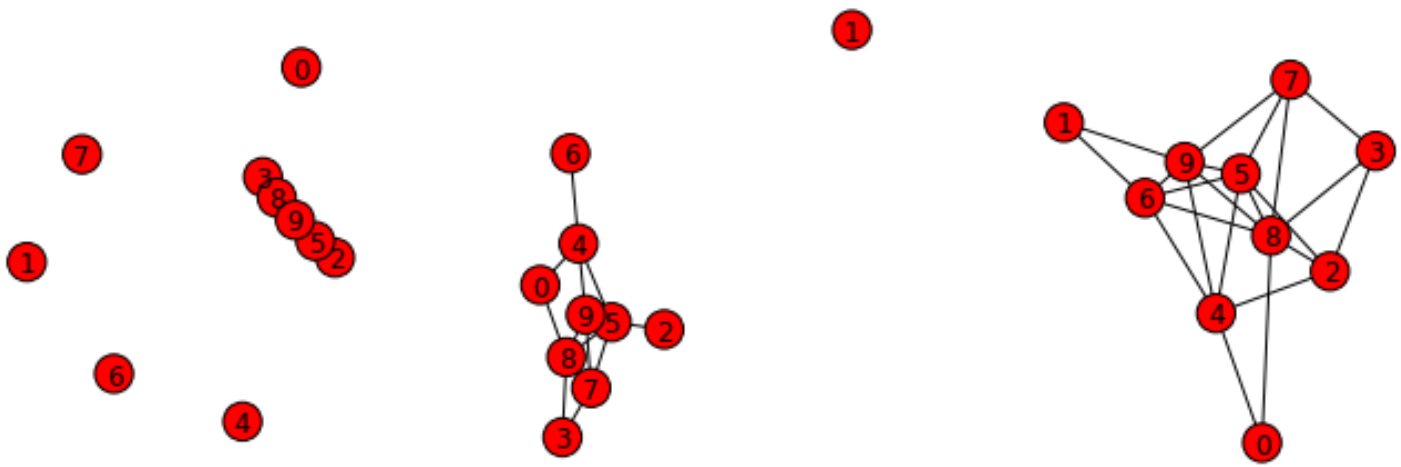
Random Binomial Graph

Simplest form of random graph.

Given a set of n vertices, pick each edge with probability p .

Also the Erds-Renyi random graph.

```
for i,p in enumerate([0.1,0.3,0.5]):
    subplot(1,3,i+1)
    nx.draw(nx.erdos_renyi_graph(10,p,33))
```



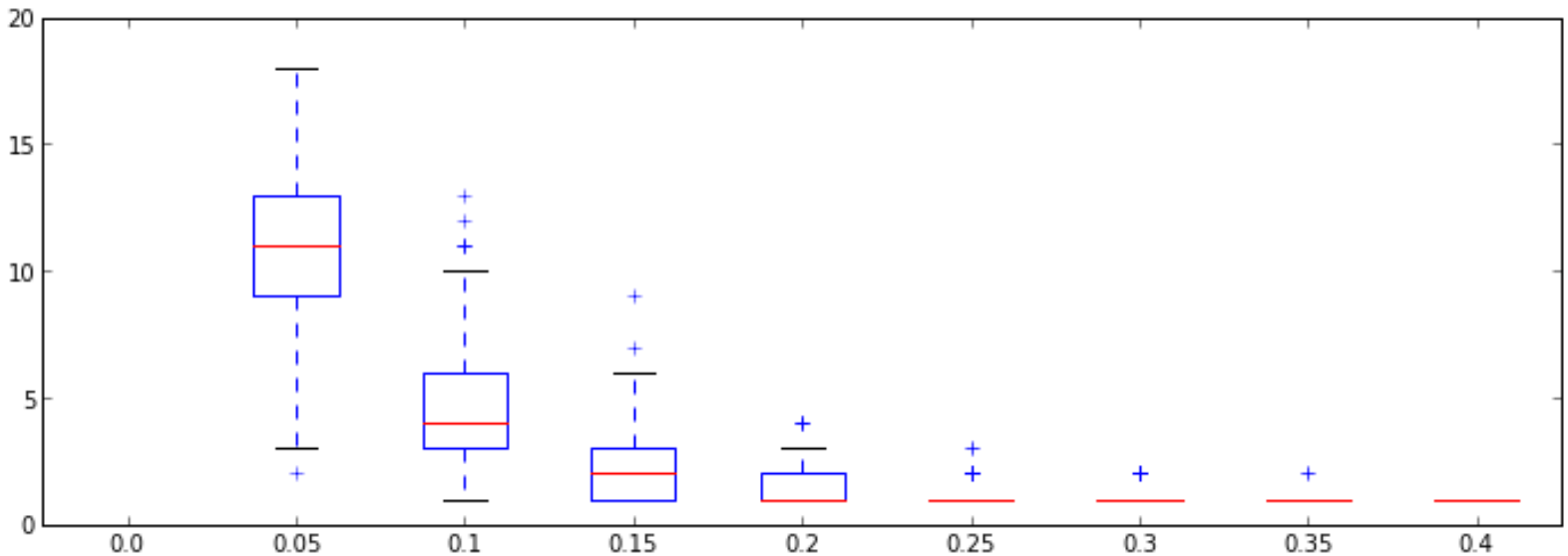
global properties of random graphs

Random graphs have interesting global properties and transitions.

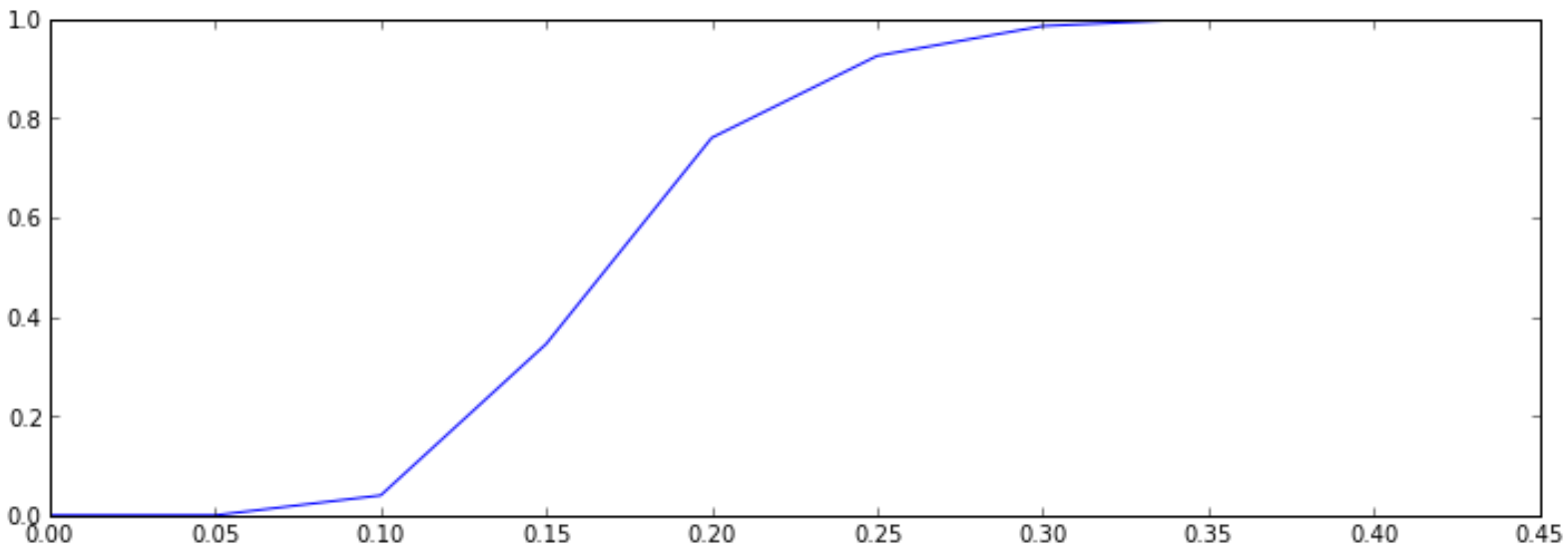
Question: what is the probability that a random Erds-Renyi graph is connected?

Let's try.

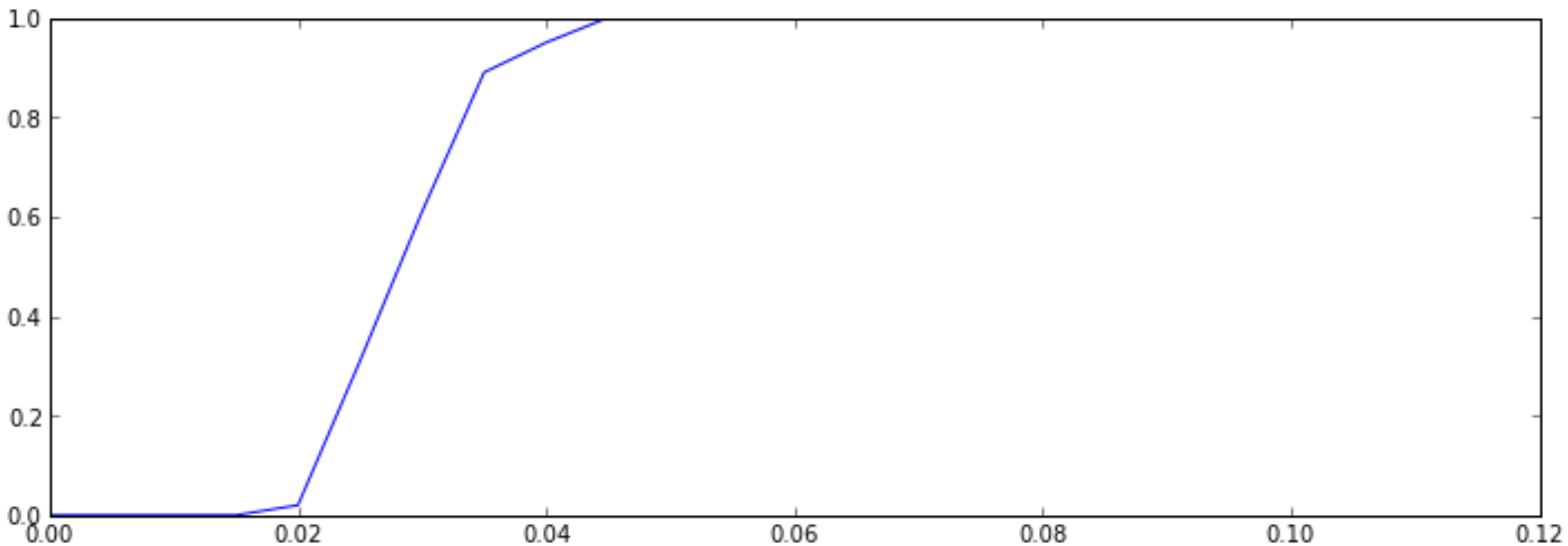
```
ps = []; vs = []
for p in linspace(0.0,0.4,9):
    ps.append(p); ns = []
    for trial in range(1000):
        G = nx.erdos_renyi_graph(20,p,trial)
        n = nx.number_connected_components(G)
        ns.append(n)
    vs.append(array(ns))
xticks(range(1,9),ps); _=boxplot(vs)
```



```
ps = []; ms = []
for p in linspace(0.0,0.4,9):
    connected = 0
    for trial in range(200):
        G = nx.erdos_renyi_graph(20,p,trial)
        n = nx.number_connected_components(G)
        connected += (n<=1)
    ps.append(p); ms.append(connected/200.0)
plot(ps,ms)
```



```
ps = []; ms = []
for p in linspace(0.0,0.1,21):
    connected = 0
    for trial in range(100):
        G = nx.erdos_renyi_graph(200,p,trial)
        n = nx.number_connected_components(G)
        connected += (n<=1)
    ps.append(p)
    ms.append(connected/100.0)
plot(ps,ms)
```



thresholds in random graphs

Many interesting random graphs properties have thresholds and “phase transitions”.

One the probability of connectivity crosses some threshold, a global property (like connectedness) becomes true with very high probability.

Model-Based Random Graph Generation

Randomly Citing Scientists

model-based random graphs

The Erds-Renyi model seems kind of ad hoc (although it actually describes some forms of physical situations).

We can and often should derive a random graph model from an actual functional model.

This is similar to the kinds of simulations we discussed before.

randomly citing scientists

Generative model:

- scientist writes a manuscript
- scientist picks several random papers (random recent papers)
- scientist copies a number of the references out of these papers
- the resulting paper is added to the collection of published papers

This is a kind of *branching process*.

TODO: Implement this (as an exercise).

Barabasi-Albert Model

Barabasi-Albert Model

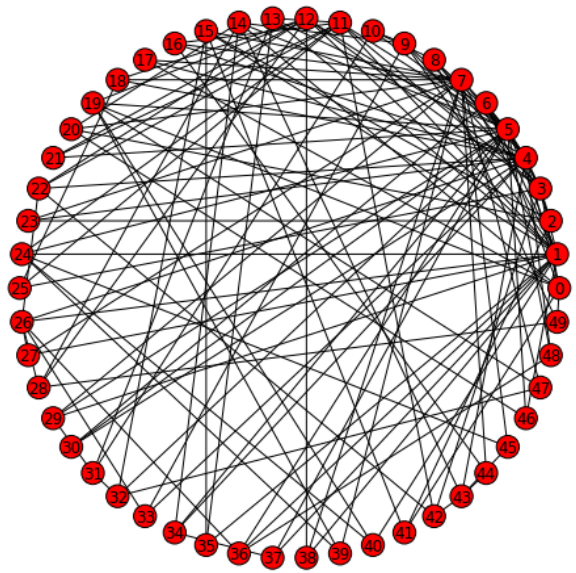
Important property: generates *scale free models* (see later).

Uses *preferential attachment*: the more connected a node is, the more likely it is to receive new connections.

Algorithm:

- begin with a connected network of size m_0
- add a new node, connect it to mm_0 nodes
- the probability of connection to an existing node is proportional to the number of links on the existing nodes

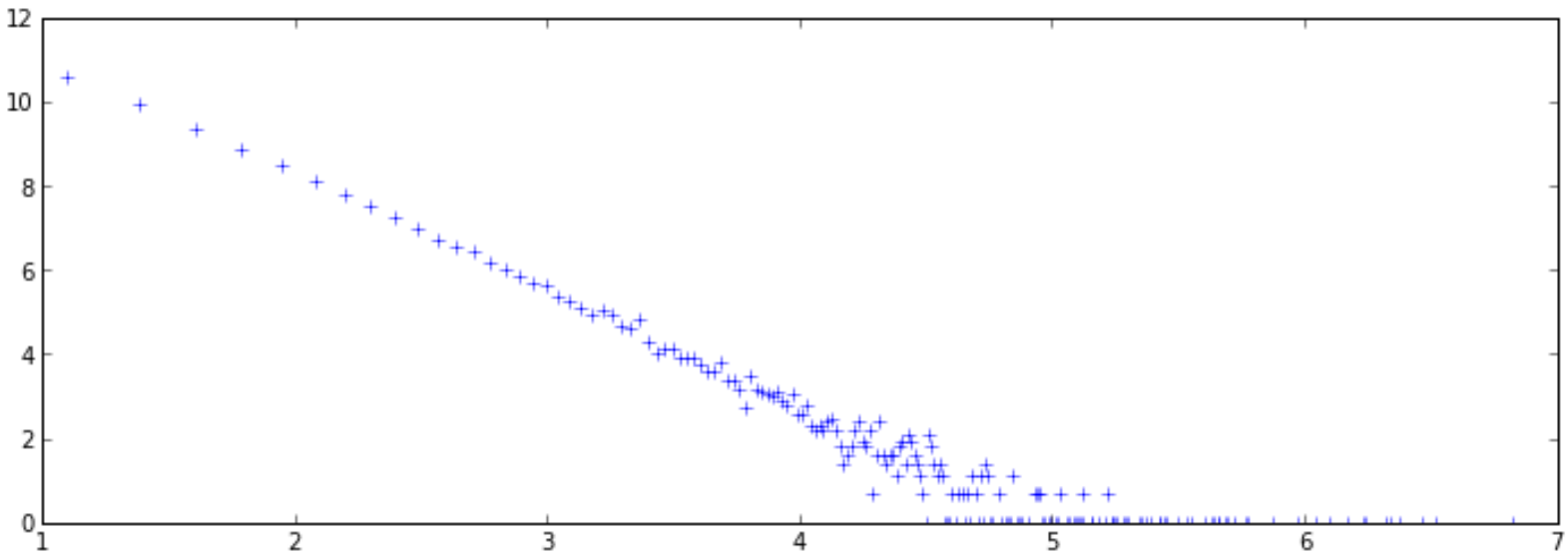
```
G = nx.barabasi_albert_graph(50,3)
nx.draw_circular(G)
```



Power Laws

node degree distribution for Barabasi Albert graph

```
1 G = nx.barabasi_albert_graph(100000,3)
2 counts = Counter([len(G[i]) for i in G])
3 ll = array([(log(n*1.0),log(c*1.0)) for n,c in counts.items()])
4 plot(ll[:,0],ll[:,1],'+')
```



power law

If we plot the frequency of the number of nodes of degree k , we get a power law:

$$P(k) \propto k^{-3}$$

This is a power law.

Graphs obeying such laws are called *scale free* (in some sense, they look "the same" at all scales).

scale free networks

In general, scale-free networks have the form:

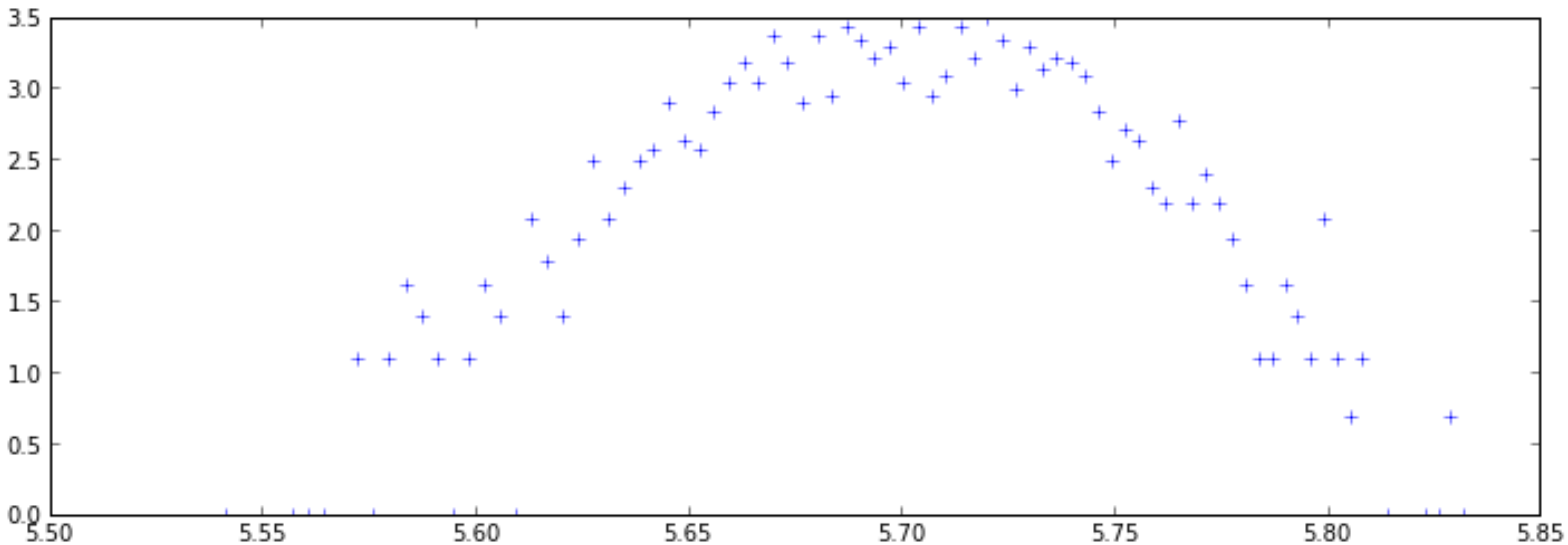
$$P(k) \propto k^{-\gamma}, \gamma \in [2, 3]$$

Many real-world networks have been claimed to be scale free (web, citation networks, social networks, biochemical networks), but it's not clear that this is true (it's statistically difficult to test).

First observed with citations: the number of citations to papers is a Pareto distribution.

node degree distribution for Erdos Renyi graphs

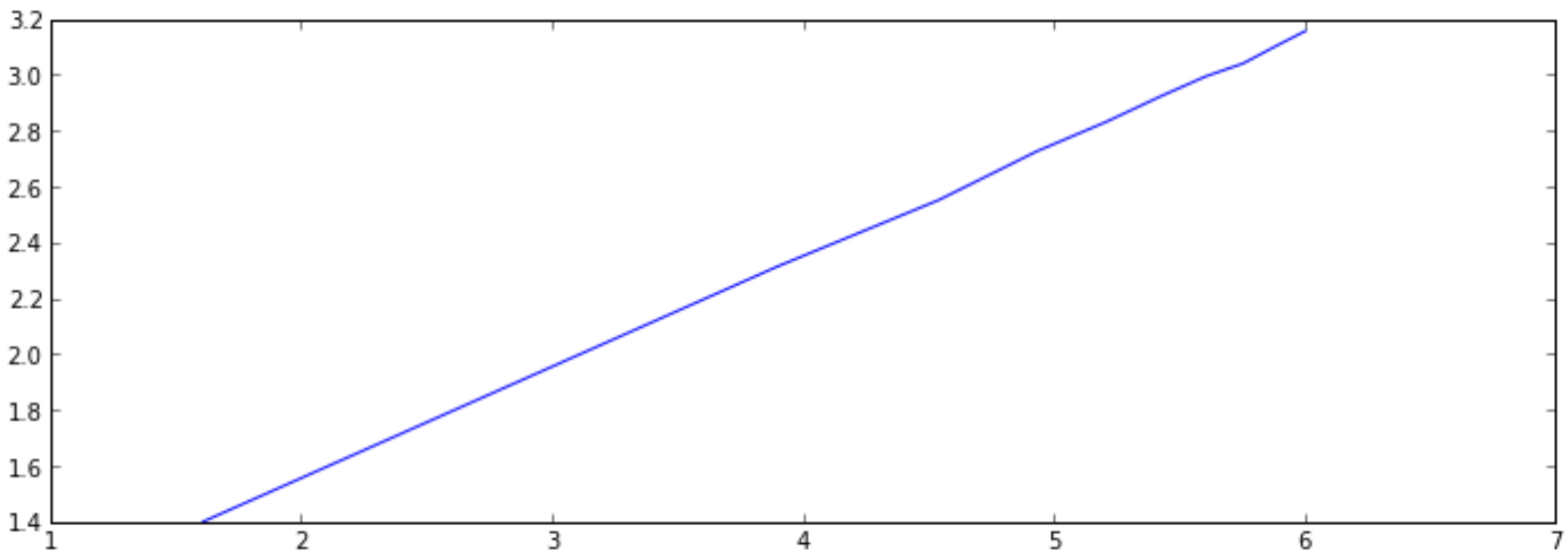
```
1 from collections import Counter
2 G = nx.erdos_renyi_graph(1000,0.3)
3 counts = Counter([len(G[i]) for i in G])
4 ll = array([(log(n*1.0),log(c*1.0)) for n,c in counts.items()])
5 plot(ll[:,0],ll[:,1],'+')
```



Small World Graphs

average shortest path for Barabasi Albert graphs

```
result = []
for N in linspace(5,405,10):
    ds = []
    for trial in range(10):
        G = nx.barabasi_albert_graph(N,3)
        d = nx.average_shortest_path_length(G)
        ds.append(d)
    result.append((N,mean(ds)))
result = array(result); plot(log(result[:,0]),result[:,1])
```



small world network

The Barabasi-Albert network is sort-of an example of a kind of small-world network:

The average distance between nodes is a logarithmic function of the size of the network.

True small world networks also have a high clustering value.

Watts-Strogatz Graph

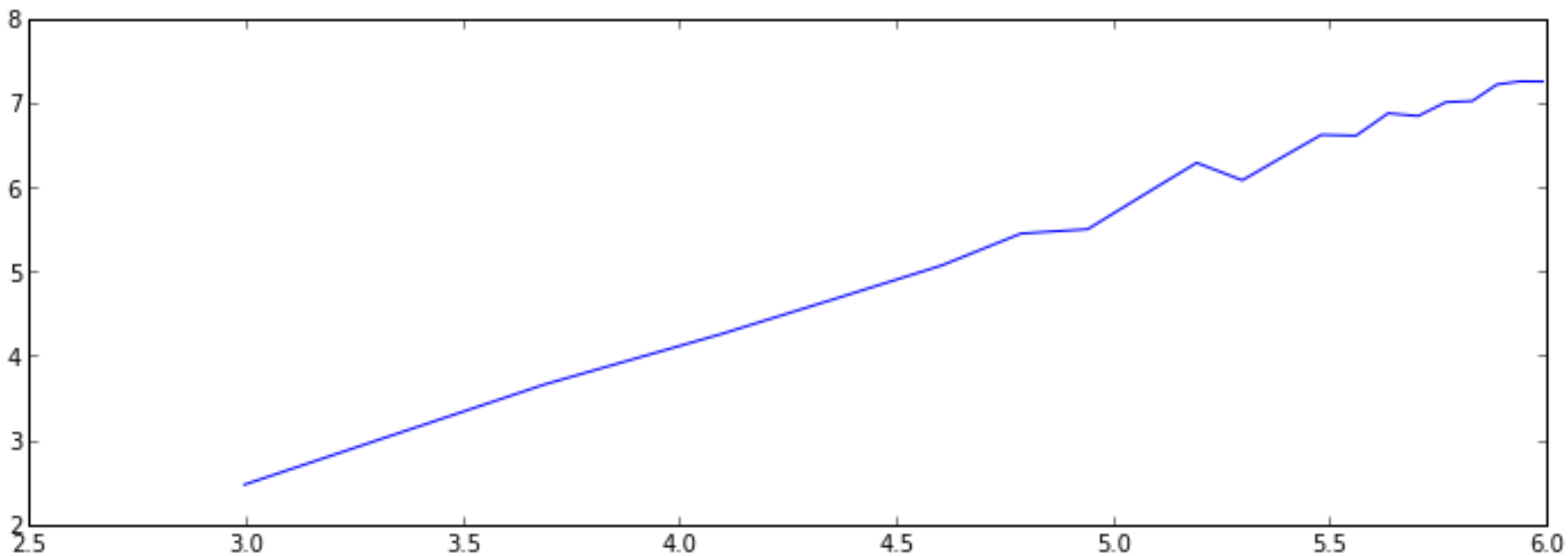
A process that generates a small world graph:

- create a ring over n nodes
- connect each node with its k nearest neighbors
- pick an existing edge (u, v) and replace it with (u, w) where w is chosen randomly from all nodes

(Different variants: add a new edge, replace only when the graph remains connected, ...)

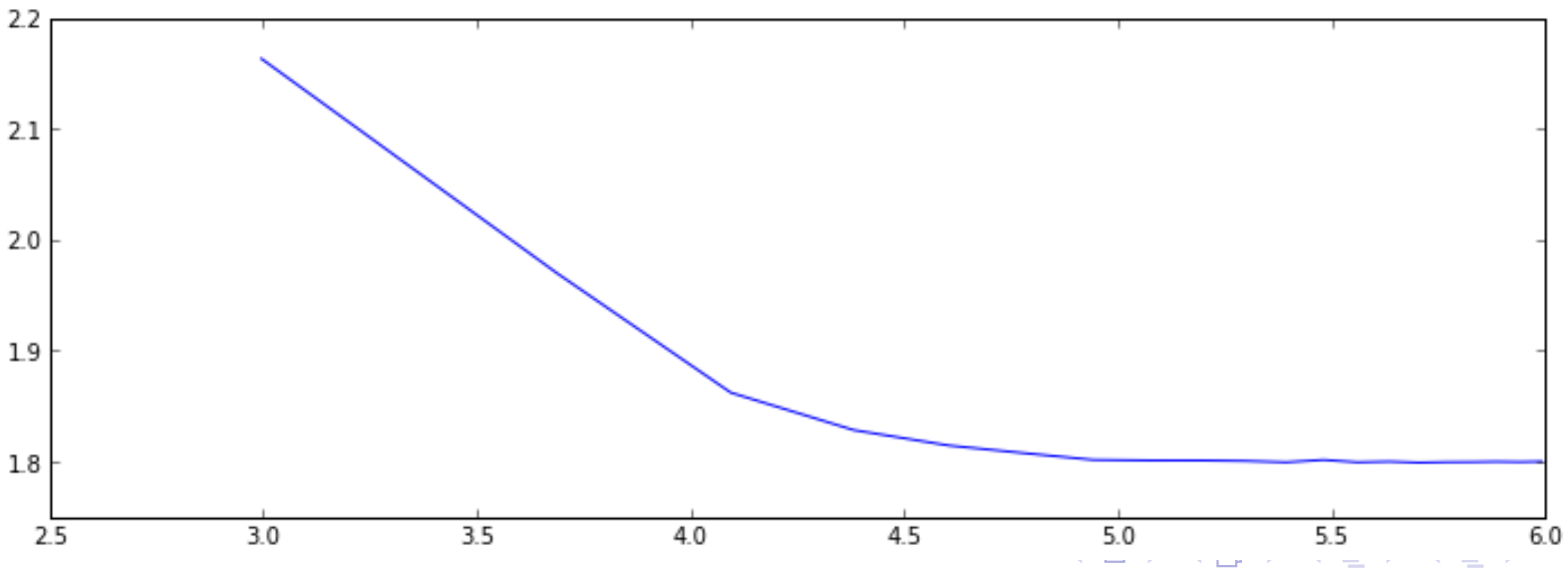
average shortest path for Watts Strogatz

```
result = []
for N in range(20,405,20):
    ds = []
    for trial in range(20):
        G = nx.connected_watts_strogatz_graph(N,5,0.1)
        d = nx.average_shortest_path_length(G)
        ds.append(d)
    result.append((N,mean(ds)))
result = array(result); plot(log(result[:,0]),result[:,1])
```



average shortest path for Erdos Renyi

```
result = []
for N in range(20,405,20):
    ds = []
    for trial in range(20):
        G = nx.erdos_renyi_graph(N,0.2)
        if not nx.is_connected(G): continue
        d = nx.average_shortest_path_length(G)
        ds.append(d)
    result.append((N,mean(ds)))
result = array(result); plot(log(result[:,0]),result[:,1])
```



polynomial growth of number of neighbors

There is another "power law" that ocurs frequently in graphs: the growth in the number of neighbors.

- Consider a graph formed by a planar grid.
- The geodesic graph distance is just the Manhattan distance.
- The number of neighbors of a node with distance r grows like r^2 .

Graphs of this type are often generated by k -nearest neighbor relations in a d -dimensional space. For these, the number of neighbors grows as r^d both for the vector distance and for the geodesic distance.