(Note many inconsistencies in given data set, so results are far from ideal)

- Nonsensical gross values (e.g. < 100).
- Bruce Willis is within 1 degree of all actors except 1 (probably nature of scraper starting with him).
- Some actors/movies have no neighbors.

Manual Testing Plan for Visualization portion:

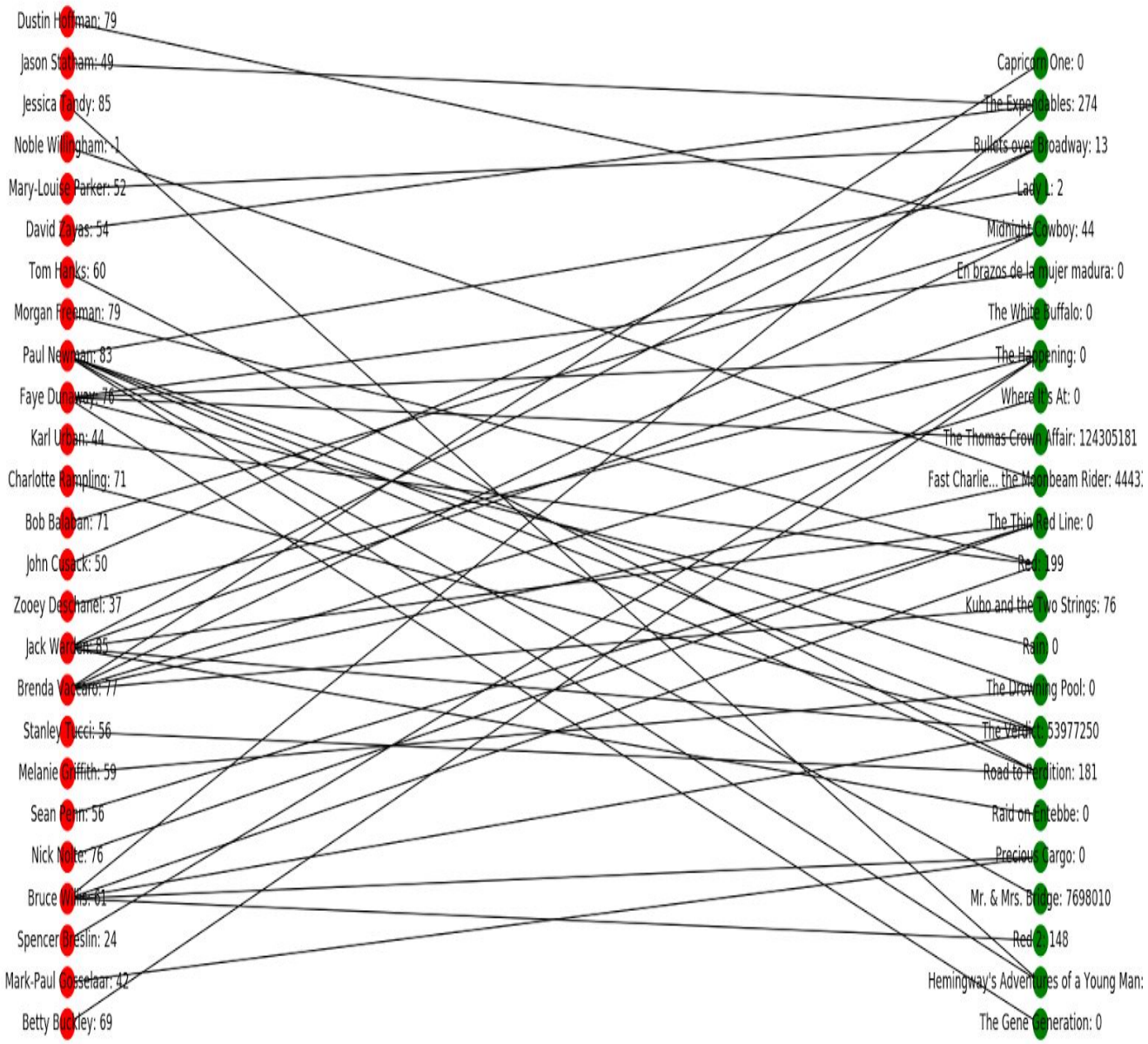
Run “python model/graph/graph_visualization.py” from home directory of project. Visually confirm the following information:

- Four plots should appear. They should roughly match the 4 plots on the following 4 pages.
- The first plot should have text labels for all nodes including name, age (for actors), and box office gross (for movies)
- In the first plot all nodes should have at least 1 neighbor.
- In the second and third plot, no nodes should be labeled
- In the fourth plot, all text labels should appear (as in the first plot).
- In all plots all movies should be colored green and all actors should be colored red
-

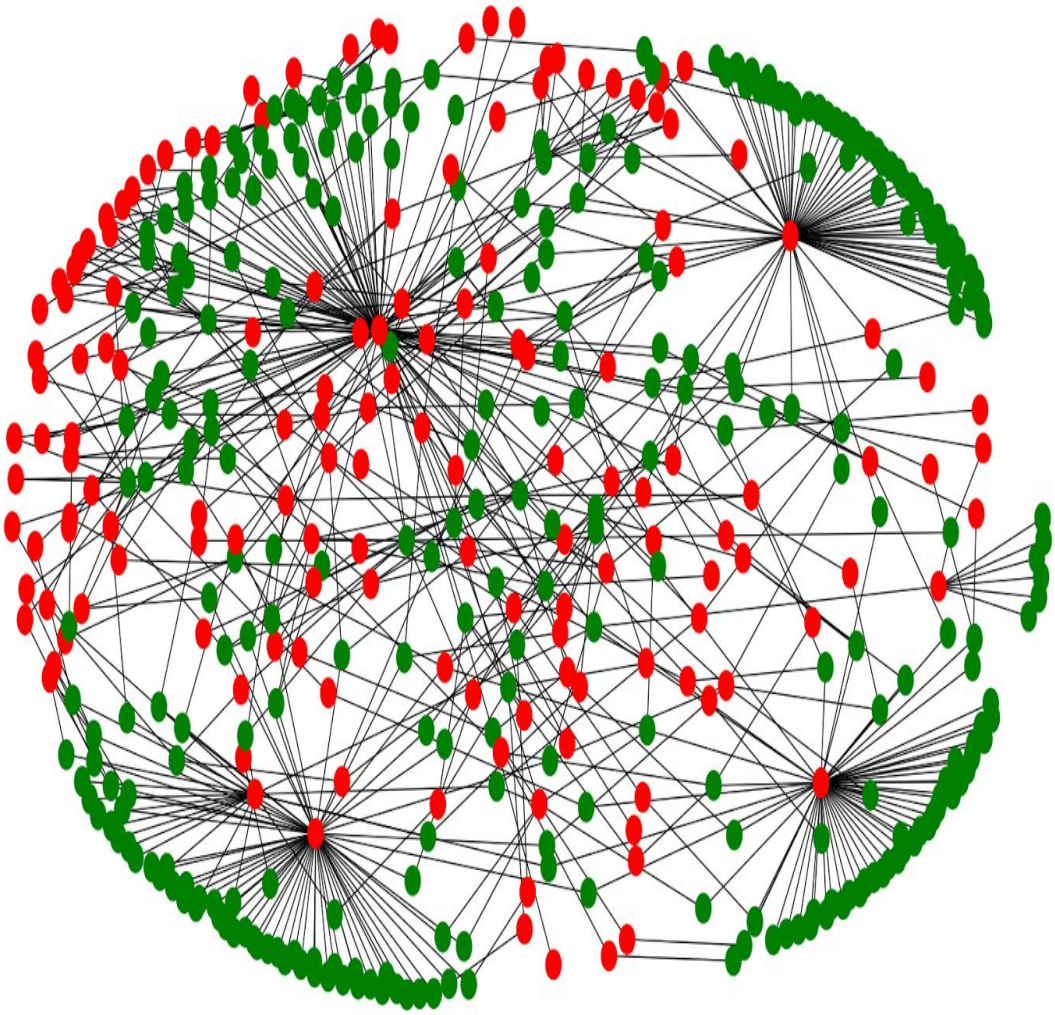
Example plots are shown in-order on the following pages.

(note that since nodes are stored in an unordered set, the functions to generate the plots are nondeterministic - you may see different actors and movies included with each run).

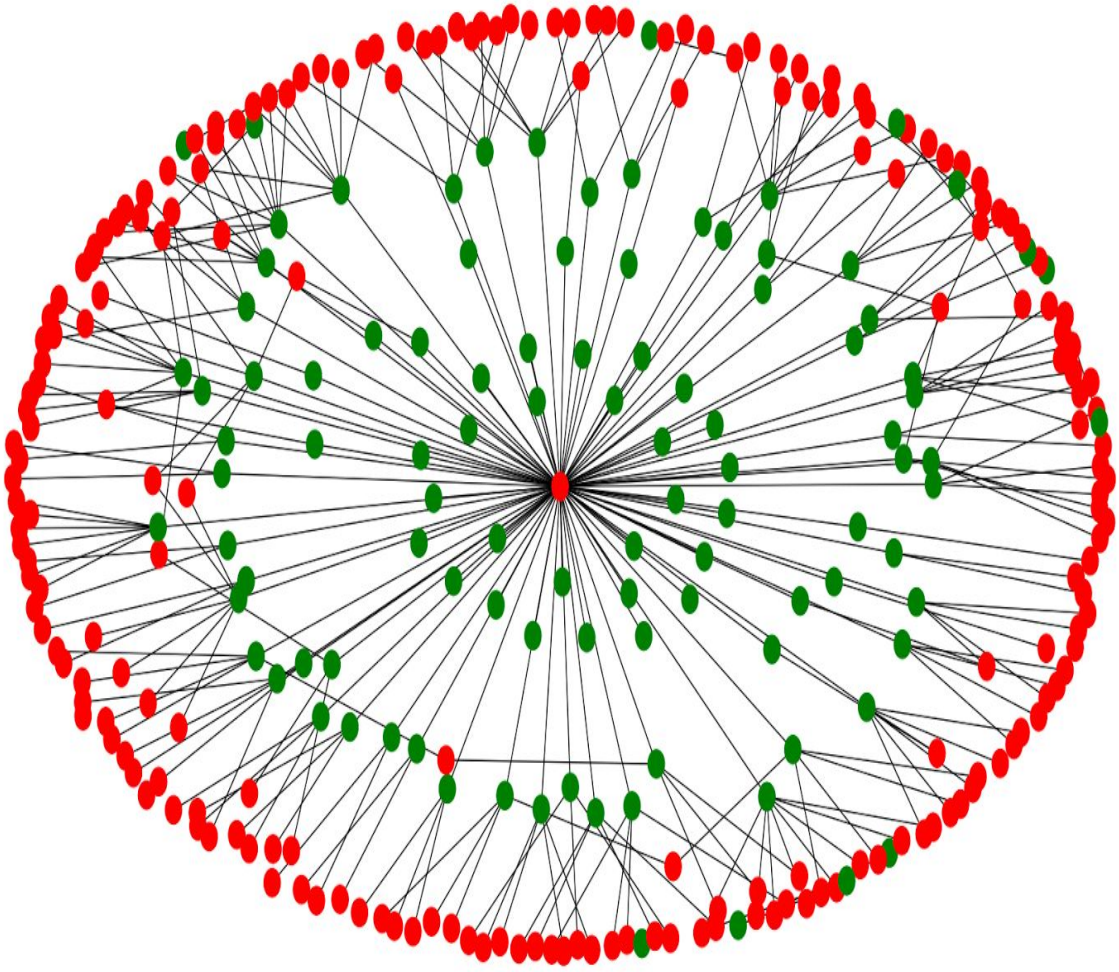
First visualization generated is a graph of a small subsection of the movie-actor graph with all actors positioned in column on left and movies positioned in column on right. This is a natural fit for the data since actors and movies form two disjoint sets (i.e this is a bipartite graph); however, unfortunately the format is not scalable and quickly becomes visually overwhelming with more than 20-30 of each type of node. An example is shown below.



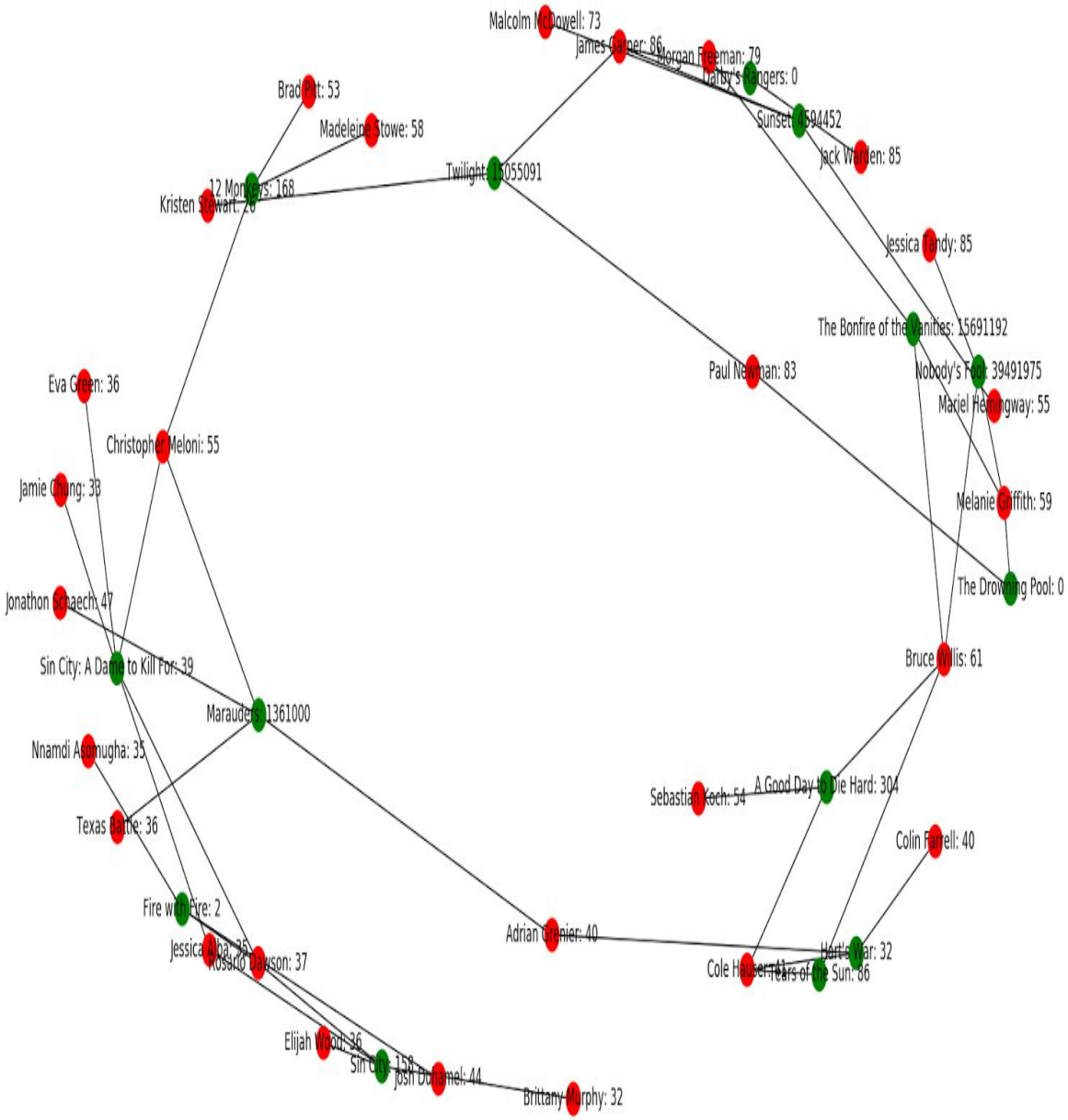
The second visualization is a large display of the top actors (in terms of movies acted in). This gives a high level view of the data set. What is interesting here is that there is very little overlap between many of the actors with the most movie neighbors. Also of note is the the actor in the middle-upper left who is tied to almost all of the other actors - we will see in next section that this is bruce willis.



The third plot places bruce willis in the center and then the bottom 200 actors (in terms of movies acted in). It is interesting to see that Bruce Willis is within 1 degree of all of them, which will make the results of the Six Degrees of Kevin Bacon in the analysis section make more sense.



The fourth plot shows a small subset of the graph, with names and age/gross labels for all actors and movies. This is similar to the first graph, but without actors and movies forced to opposite sides.



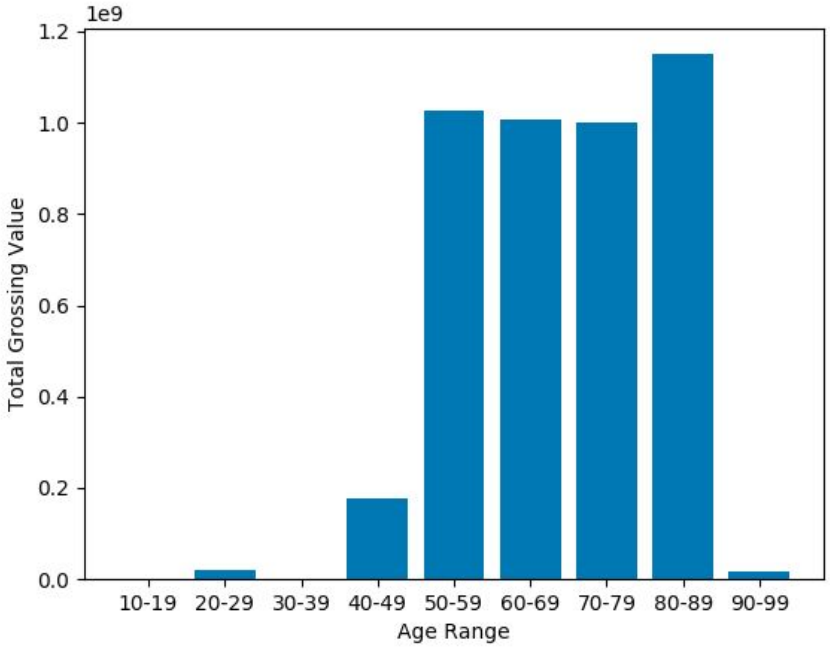
Manual Testing Plan for Analysis portion:

Run “python model/graph/graph_analysis.py” from home directory of project.

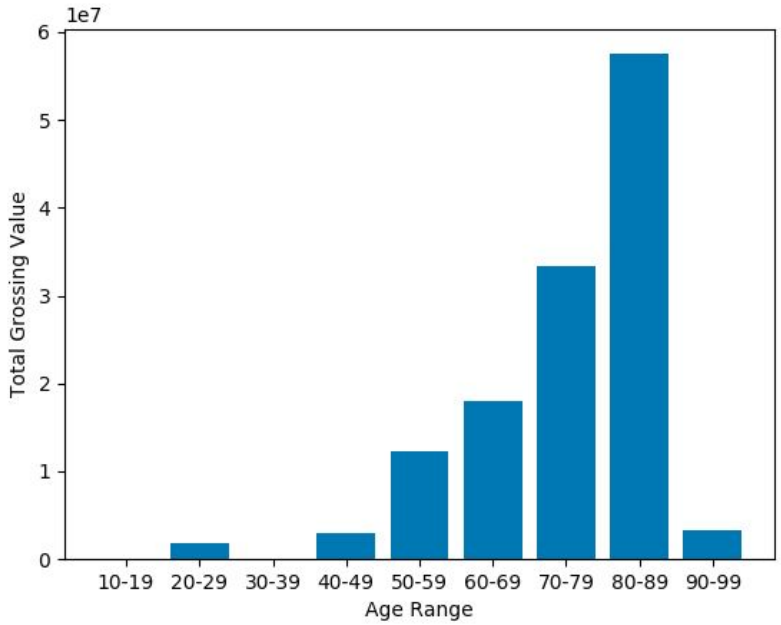
- Variations of the 4 plots on the following pages should appear.
- Data on the average separation degree, max separation degree, and min separation degree should all be printed to console as well.
- After plots, the top 10 hub actors should be printed to console in list format.
- Regardless of data, axis labels and bins (for histograms) should be identical.
- Confirm that data represented in graph makes sense for your data set. This can be done by doing simple CRTL-f searches in the data.json file.

Unit tests are also provided for all methods (other than plot generating ones) in the graph.test file. These tests can be modified to run on whatever data you have.

Age Groups and Grossing Value:



Age Groups and Grossing Value (normalized by counts):



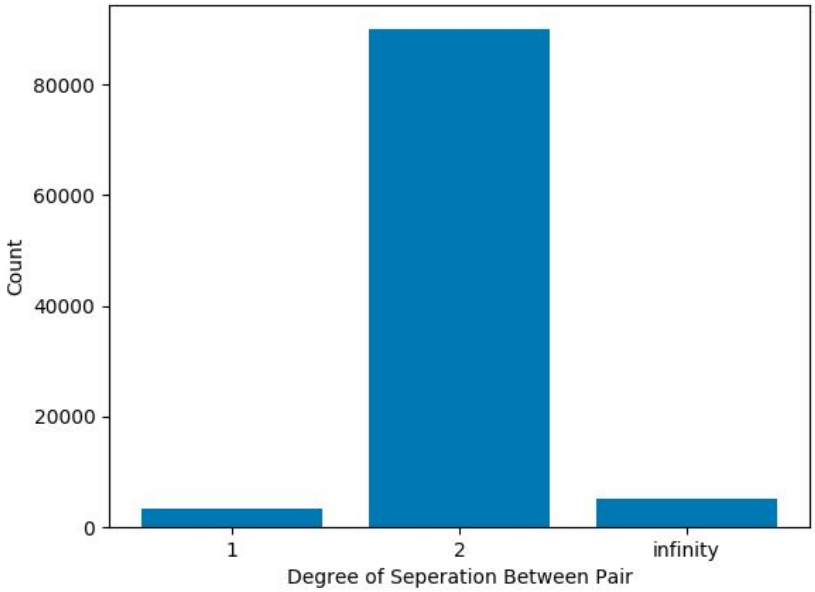
It appears that total gross valuing increases (rather steeply) with age, which intuitively makes sense as actors will continue to accumulate more grossing value over the course of their career. Additionally, actors likely to earn more later in their career, as their name recognition increases. **Likely further compounded by the fact that only popular actors are likely to work late into their lives, who are typically higher earners.**

Degrees of separation found using BFS starting from each actor. Dijkstra's not needed since edges are unweighted.

Degrees of Separation:

Max = Infinity
Min = 1
Average (excluding infinite pairs) = 1.9636

Counts:



As noted above, all but 4 actors in this data set were in a movie with Bruce Willis (1 degree of separation). Therefore, almost all pairs will have separation value 2 in this set, as they can travel through Bruce Willis' node if they had not acted together.

Also, using the scraped data is a horrible strategy for attempting to prove "Six Degrees of Kevin Bacon", as the scraper begins with a single actor and traverses to the movies that actor was in, then the actors that were also in those movies. Therefore, the scraper chooses data that is inherently connected through whoever the "root" of the scraping is (in this case likely Bruce Willis).

Also note, most infinite cases are due to bugs in the given data (i.e actor node has no neighbors). An example would be ""Milo O%27Shea".

Hub Actors:

Name	Connected Actors
Bruce Willis	305
Jack Warden	39
Faye Dunaway	35

Paul Newman	33
Jeremy Piven	32
Steve Buscemi	31
Mickey Rourke	30
Nick Nolte	30
John Cusack	28
Buck Henry	26

Average # of connected Actors: 10

Boxplot with outlier (Bruce Willis) removed: