CS 111: Program Design I

Lecture 22: CS: Network Analysis, Degree distribution, Profiling

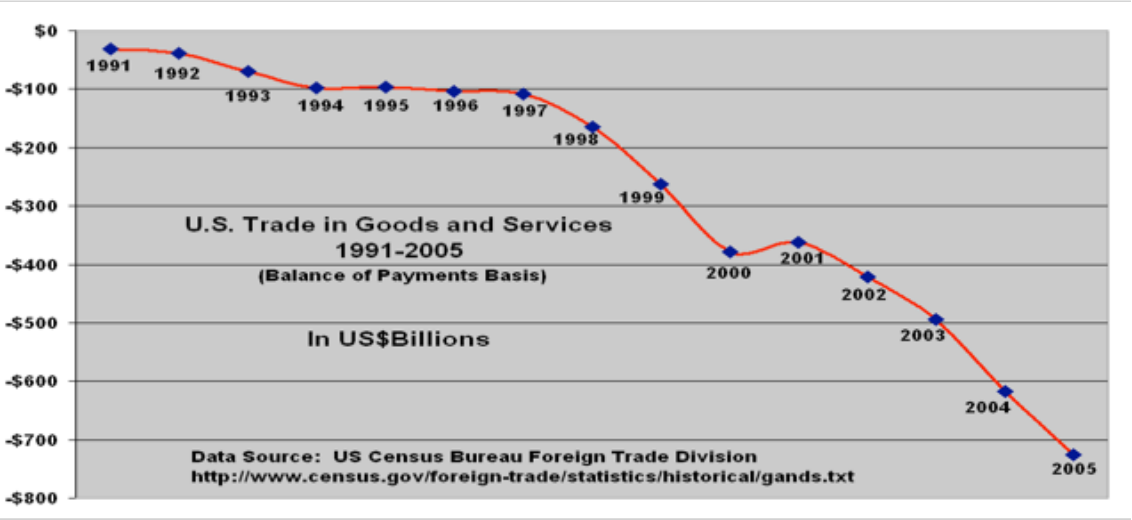
Robert H. Sloan & Richard Warner
University of Illinois at Chicago
November 15, 2018

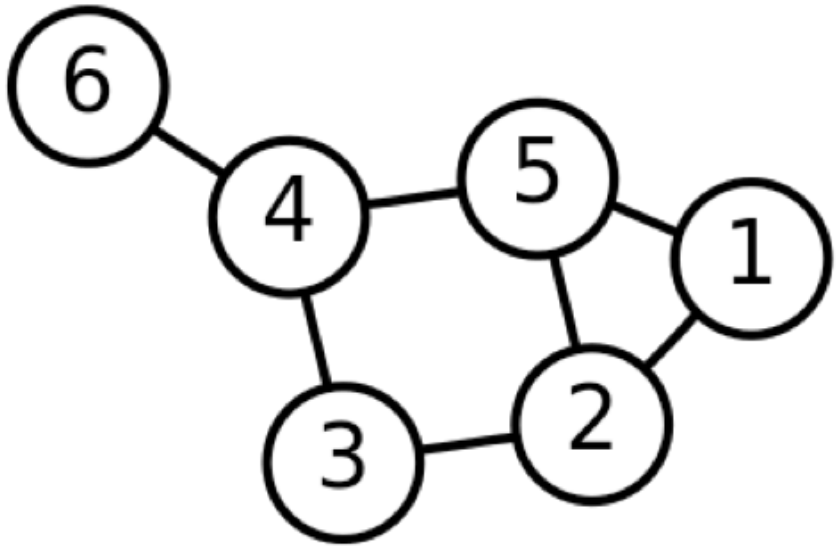
UIC

NETWORK ANALYSIS

Which displays a graph in the sense of graph/network analysis?

A: Left, B. Right, C. Both





Graphs

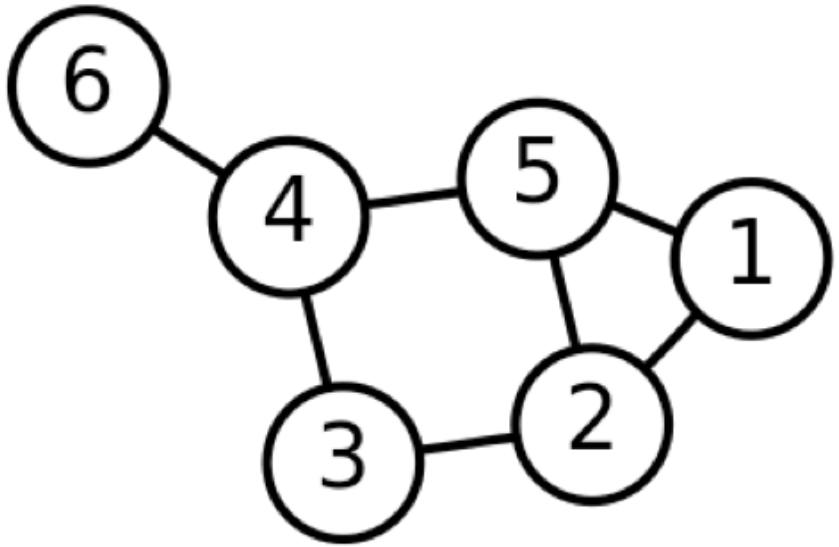
- Graph (or network) is collection of nodes (aka vertices) and links (aka edges).
  - For the formal minded, link is pair of nodes
    - A CS 151 (Discrete Math) textbook might make a formal definition like "A graph is a finite set of nodes N together with a set L of pairs of nodes."

Graphs

- Graph (or network) is collection of nodes (aka vertices) and links (aka edges).
  - For the formal minded, link is pair of nodes
    - ~~A CS 151 (Discrete Math) might make a formal definition like "A graph is a finite set of nodes N together with a set L of pairs of nodes."~~
  - Default case: links are bidirectional, undirected, and what we assume if we just see "graph" or "network"
  - Can also have directed links (e.g., road network with one-way streets): "directed graph"

Example

- Network with 6 nodes and 7 links
- (Undirected)
- Links are:
  - (6, 4), (4, 5), (1, 5), (1,2), (2,3), (2, 5) (3, 4)



Examples

- Social networks
  - people as nodes; friend = (undirected) link
- Web pages (directed)
  - page = node; link = link
  - Web crawler: crawls around this network
- Computer networks
  - nodes = computers
  - links = 2 computers that can communicate directly
- Chicago El
  - Nodes = stops; links between adjacent stops
- Collection of Phone Calls
  - Nodes = phone numbers; links = phone calls

Networkx

- To work with graphs in Python, especially for network analysis:
- `import networkx (as nx)`
- Learn more at:
  - https://networkx.github.io/documentation/networkx-2.1/tutorial.html
  - Spyder almost certainly has version 2.1, almost identical to version 2.2, released in mid-September. (Some important differences from old versions 1.x)
  - networkx provides Graph as basic data type and ways to add nodes and edges and do all sorts of things, including visualize

Simple graph example

```
import networkx as nx

g = nx.Graph()        #Create an empty graph object

#Add several nodes
g.add_node('Alice')
g.add_node('Bob')
g.add_node('Charlie')

g.number_of_nodes()       → 3
g.number_of_edges()       → 0
```

Simple graph example continued

```
# Add a single edge
In [9]: g.add_edge('Alice', 'Bob')          # undirected
In [10]: g.number_of_edges()
Out[10]: 1
In [11]: g.nodes()
Out[11]: ['Alice', 'Charlie', 'Bob']
In [12]: g.edges()
Out[12]: EdgeView([('Alice', 'Bob')])
```

Drawing

- networkx can do simple drawing (working with matplotlib.pyplot under hood):

```
nx.draw(g, with_labels='True')
```

Drawing without node labels

```
nx.draw(g)
```

(or, for control freaks or the pedantic)

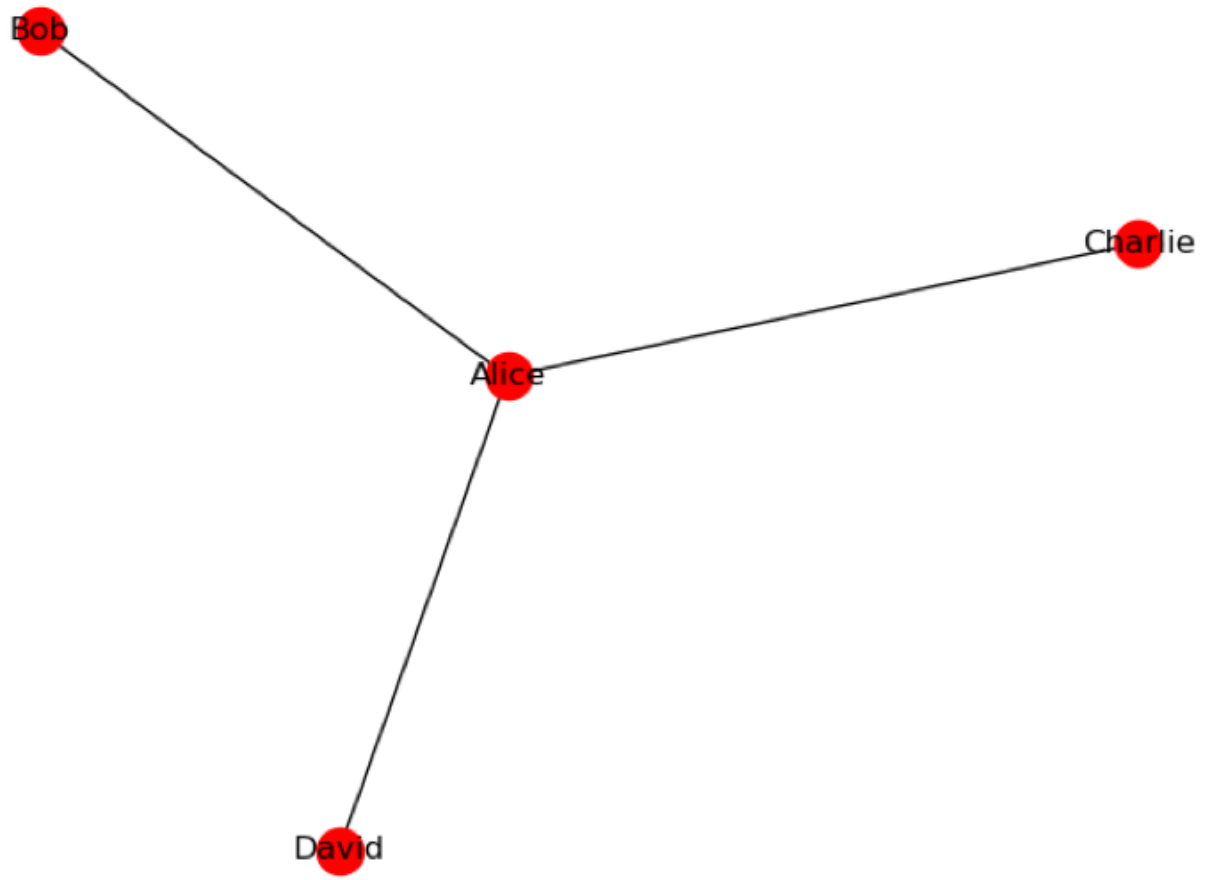
```
nx.draw(g, with_labels=False)
```

Adding a bit to the graph

```
# Add some more edges and nodes g.add_node("David")
g.add_edges_from([("Alice", "Charlie"),
                  ("Alice", "David")])
```

- add_edges_from is a method whose argument should be *list* of pairs of node names, each pair in ()s.

networkx.draw(g) (as updated)



Key graph statistic 1: Degree

- Degree of node = number of neighbors node has
- Range of different degrees discovered in last 20-30 years to vary with nature of graph.
- networkx has graph method degree that gives us special data structure easily converted to a *dictionary* with all the degree information for the graph

degrees from networkx

```
In [5]: d = networkx.degree(g)
In [6]: d
Out[6]: DegreeView({'Alice': 3, 'Bob': 1, 'Charlie': 1,
'David': 1})
In [7]: ddict = dict(d)
In [8]: ddict
Out[8]: {'Alice': 3, 'Bob': 1, 'Charlie': 1, 'David': 1}
```

Will explain dictionaries next class; for now: pandas knows them!

networkx degree data → pandas

```
In [7]: degree_data = pandas.Series(dict(nx.degree(g)))
In [8]: degree_data
Out[8]:
Alice      3
Bob        1
Charlie    1
David      1
```

- And we know from earlier in semester how to plot graphs from pandas series: pandas series has method .plot()

Plotting degree data

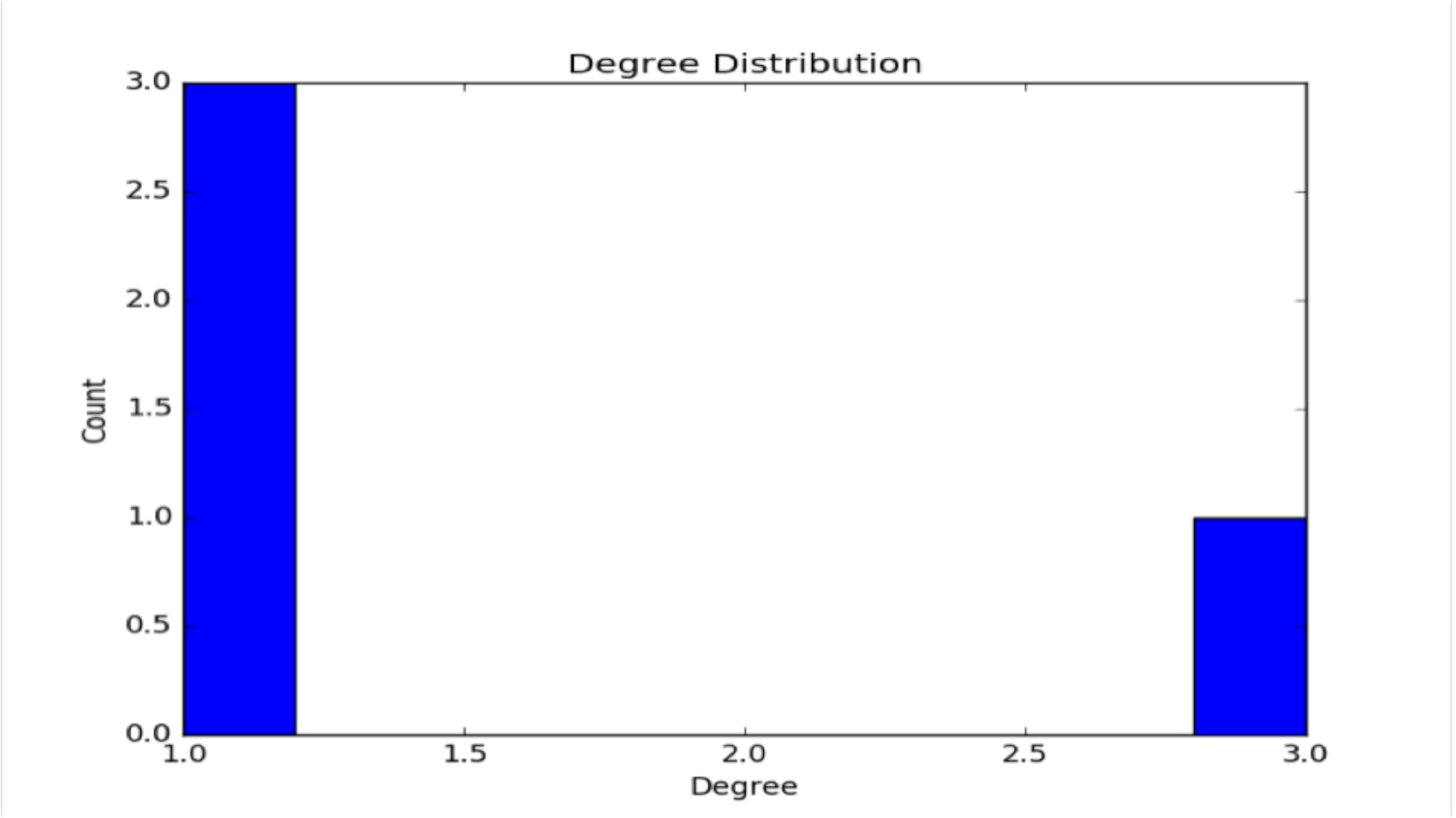
- We want a *histogram:* Bar plot where the things on the x-axis have a specific meaningful order (e.g., numbers), as opposed to being categorical (e.g., names of justices)
- degree_data.plot(kind='hist')
- (Unimportant side note: There is an abbreviation for that. Can write .hist() instead of .plot(kind='hist'))

To make plot of series look nice

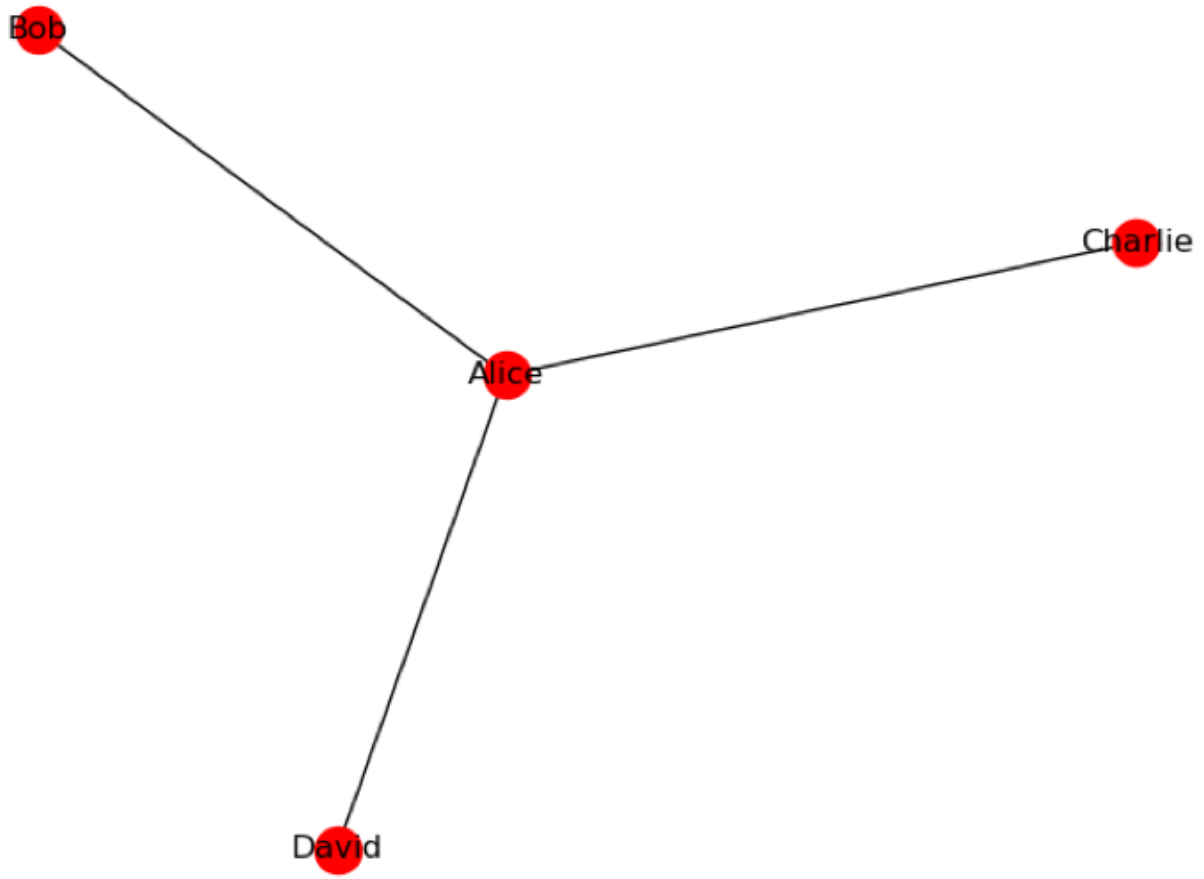
- pandas put in stuff automatically for some of the plots we did before from dataframes, but it doesn't always. If need be:

- plt.xlabel('string I want to see below x axis')
- plt.ylabel('Similarly for y')
- plt.title('String I want up top in title position')

Plot with some appropriate labels



Remember what our graph looks like



Centrality

- Alice is connected to everybody else; Bob, Charlie, and David are connected only to one node each (Alice)
- Alice is obviously the most central
- Various *centrality measures* to tell which nodes are most central
- (Prof. Philip Yu of UIC CS found to be "most central" computer science author by one such measure)

What is maximum degree of node in graph with n nodes?

A. n
B. $n - 1$
C. $n - 2$
D. $n(n - 1) / 2$
E. 42

One simple measure of centrality of a node

- degree of node / maximum possible degree of any node in that node's graph
- For *n*-node graph:
  - degree of node / (*n* - 1)
- networkx will give us (a dictionary of) the centrality of every node in graph g:

```
cent = nx.degree_centrality(g)
```

For our little graph

```
In [11]: cent = nx.degree_centrality(g)


In [12]: cent
Out[12]:
{'Alice': 1.0, 'Bob': 0.3333333333333333,
 'Charlie': 0.3333333333333333,
 'David': 0.3333333333333333}
```

Path lengths

- How many edges do we need to walk over to get from one node to another?
- 0 to get from node to itself
- 1 to get to immediate neighbor
- > 1 to get to all other nodes

Getting all path lengths

- neworkx has operator for this; gives somewhat complex data structure back
- pandas to the rescue: It knows how to handle that data structure and turn it into a dataframe, which we already know about:

```
pandas.DataFrame(dict(nx.all_pairs_shortest_path_length
(g)))
```

All path lengths in our graph

```
p = pandas.DataFrame(nx.all_pairs_shortest_path_length(g))
```

```
>>> p
         Alice  Bob  Charlie  David
Alice        0    1        1      1
Bob          1    0        2      2
Charlie      1    2        0      2
David        1    2        2      0
```

Another stat: Average shortest path length

- networkx will calculate the average over all shortest path lengths for you:

```
# Get the average path length
print(networkx.average_shortest_path_length(g))
1.5
```

Our 4 node graph is kinda dull

- Point is to apply these sorts of techniques to e.g., graphs of various types of social networks with thousands to 1 billion+ nodes
- Our example data (real data):
  - nodes = twitter users
  - edge = follows relationship (could be directed; could ignore direction)
  - ~40,000 pairs of follower, followee
  - (This particular bit of twitter formed by technique called snowball sampling starting at Computational Legal Analytics)

Large networks

- Stored as text files
- One line for each link with line containing names (string or number) of nodes
  - Notice that if we know all the links then we know what the nodes are
  - Both comma and space are common delimiters for between the two nodes of an edge in large network work
  - Both are, broadly speaking, CSV
- We'll use pandas to read these in

Recall: Plotting degree data (in next lab!)

- Plotting *histogram:* Bar plot where the things on the x-axis have a specific meaningful order (e.g., numbers), as opposed to being categorical (e.g., names of justices)
- degree_data.plot(kind='hist') or can use
  - Abbreviation: Can write .hist() instead of .plot(kind='hist'))

matplotlib.pyplot and labeling your plot

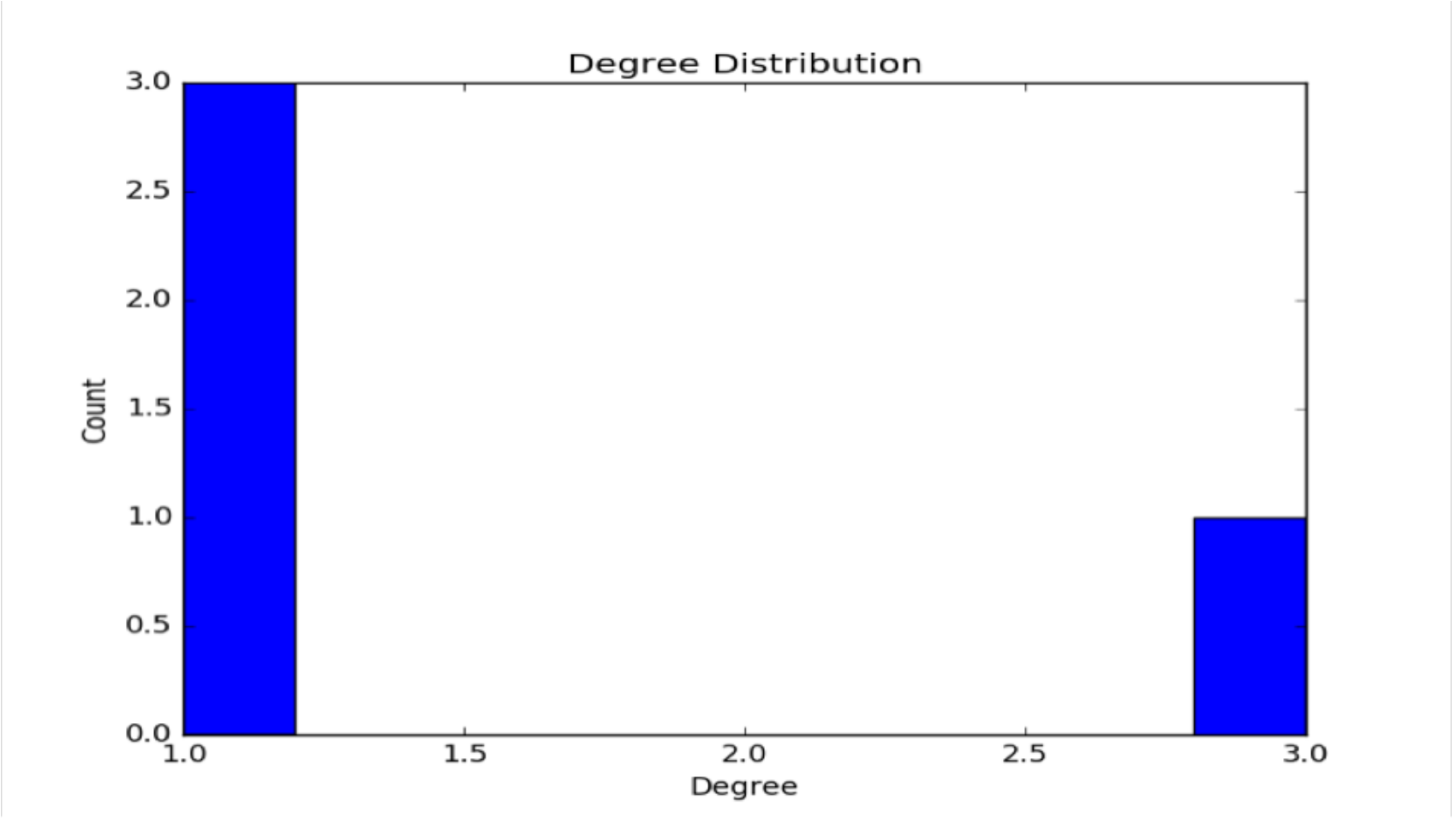
- pandas can put up plots *without* importing matplotlib.pyplot
- But *do* need to import matplotlib.plyplot if we want to be able to alter plot's labels, etc.
  - Gives us a "handle" for the plot

To make plot of series look nice

- pandas put in stuff automatically for some of the plots we did before from *dataframes*, but it doesn't always especially for series. If need be:

- import matplotlib.pyplot as plt
- plt.xlabel('string I want to see below x axis')
- plt.ylabel('Similarly for y')
- plt.title('String I want up top in title position')

Plot with some appropriate labels



Large networks

- Stored as text files
- One line for each link with line containing names (string or number) of nodes
  - Notice: if we know all links then we know the nodes
  - Both comma and space are *common* delimiters for between two nodes of edge in large network work
  - Both are, broadly speaking, CSV
- We'll use pandas to read in such files

Reading graphs from files pandas

- (The joys of working with real data! ☺)
- Can still have encoding issues for, e.g., Chinese node name
- *CSV* files and csv_read *default*: *rows separated by newlines* and *items in rows separated by commas*, but can specify item separator either
  - Comma: pandas.read_csv(<filename>)
  - Space: pandas.read_csv(<filename>, sep = ' ') # Project!

DEGREE DISTRIBUTIONS

How many close (real world) neighbors

- Estimate number of other people living within 100 feet of where you sleep at night:

A. 0–7
B. 8–15
C. 16–32
D. 32–64
E. 65+

How many close neighbors

- Estimate highest number of other people living within 100 feet of bed of anyone in Chicago area not in dorm, prison, military, or hospital

A. 0–32
B. 32–64
C. 66–125
D. 128–250
E. 250+

How many FB friends?

- Estimate your number of FB friends (or followers if larger)

A. 0–200
B. 200–400
C. 400–600
D. 600–999
E. 1000+

Maximum number FB followers?

- Estimate maximum number of followers of most followed person on FB?

A. 5000
B. 50,000
C. 500,000
D. 5,000,000
E. 50,000,000

Ronaldo, Shakira

- Vin Diesel has about 100 million followers
- Shakira has about 103.5 million followers
- Cristiano Ronaldo has about 122 million followers

Power law degree distributions

- # of followers does *not* approximate classic bell curve distribution of
  - Heights of Homo sapiens
  - Times of runners
  - Number of real-world neighbors
  - Perhaps: Number of FB friends of people in this class?

- Compare Ronaldo's FB followers to human height: No 50-foot tall (much less 50 mile tall) outliers!

PROFILING: ALIEN INTELLIGENCE

What Is Profiling?

- Analyzing people's characteristics in order to
  - Classify
  - Predict
- Digital technologies allow the creation of profiles that incorporate patterns discovered in large quantities of data

We All Profile. So Why Object?

- New factors
  - A cultural change
  - New techniques
  - Both leading to increased power.
- The cultural change has a long history.

Cultural Change

- Many people
  - Place vast amounts of information in the hands of third parties (e.g., use of gmail)
  - Detail events in their lives on social networks
  - Provide details about their friends lives on those networks,
  - Create (on social networks) networks of friends, acquaintances and organizations.

More and More Accessible Information

- The US Supreme Court recognizes this fact: “It is “a totally different thing to search a man’s pockets and use against him what they contain, from ransacking his house for everything which may incriminate him.” . . . If his pockets contain a cell phone, however, that is no longer true. Indeed, a cell phone search would typically expose to the government far *more* than the most exhaustive search of a house: A phone not only contains in digital form many sensitive records previously found in the home; it also contains a broad array of private information never found in a home in any form—unless the phone is.”
  - *US v. Riley*.

In 1840

Commercial credit bureaus compiled “information about the personal life, habits, property, and financial reputation of all American merchants and entrepreneurs. . . . During the late 1850s, some of these firms began to publish rating books that displayed the creditworthiness of each individual in cryptic alphanumeric codes. In essence, what early commercial reporting firms sought to do was to convert an individual’s local reputation into an easily readable, centralized summary of creditworthiness for remote lenders. In the process they did something more profound: they created the modern concept of financial identity.”

- Lauer, *Creditworthy: A History of Consumer Surveillance and Financial Identity in America*

Informational Privacy

- Informational privacy is the ability to control what others do with information about you.

Privacy In Public

- The family holiday dinner example.
- Your control over the collection and use of information consists in others voluntarily refraining from collecting and using that information.

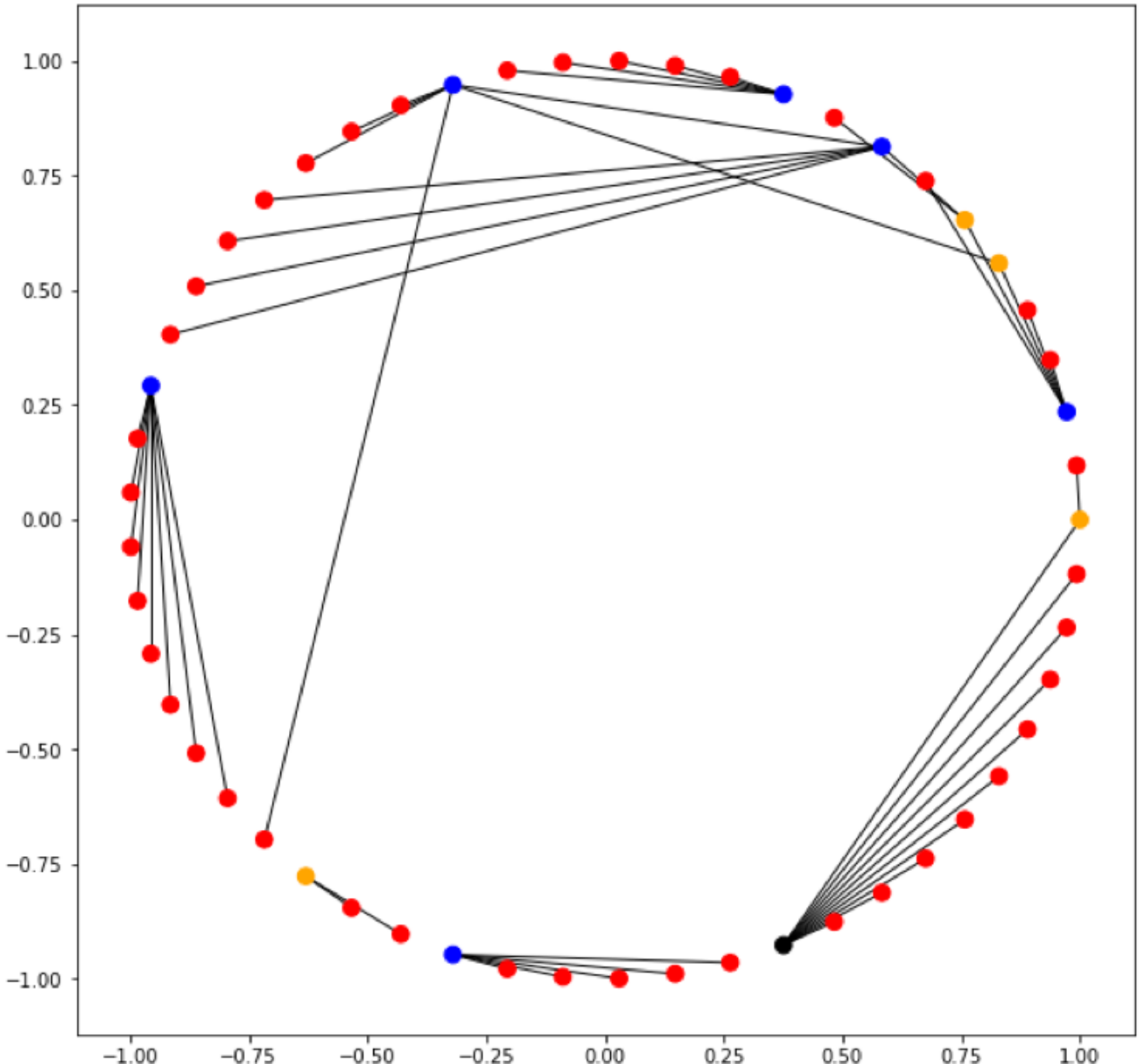
Why Privacy in Public Matters

- Vast amounts of personal information is in the hands of others.
- Adequate privacy requires control over those others.
- That is what privacy in public gives.
- Are new techniques of data collection and analysis consistent with privacy in public?

New Techiques

- We look at two:
  - Predictive analytics (Supreme Court database)
  - Network analysis.

Cell Phone Calls



Machine Learning/Artificial Intelligence

- The aliens land. Imagine beneficent aliens who come in peace.
- One of their first acts is to provide us with an analytics/artificial intelligence program that predicts future thought and action.
  - Call the program AI, for *alien intelligence*.
- We have no explanation or understanding of why AI predicts what it does.

Alien Intelligence

- Even the best human computer science experts find large parts of the AI program completely unintelligible.
  - It appears to involve programming and statistical techniques unknown to us.
- Its predictions are more accurate than ours but, like ours, still have a fairly high error rate.

Should We Use AI?

- Humans—businesses, governments, and individuals—embrace the program, and
- Many (humans) propose using AI systematically in the widest possible range of contexts as a basis for prediction and action.
- Would this be a good idea?

What AI Would Do

- AI creates winners and losers—
  - a very large number of winners and losers since AI governs the widest possible range of contexts.
- Losers will face great difficulty in escaping the categorizations that condemn them to that role.
- The high error rate means that many of the categorizations are wrong, and
- The lack of feedback ensures that AI will not correct its errors.

Unjust and Destabilizing

- AI's unintelligibility means that there is no way to explain to losers why such treatment is not capricious and arbitrary.
- Such a predictive system is both profoundly unjust and a serious threat to social stability.