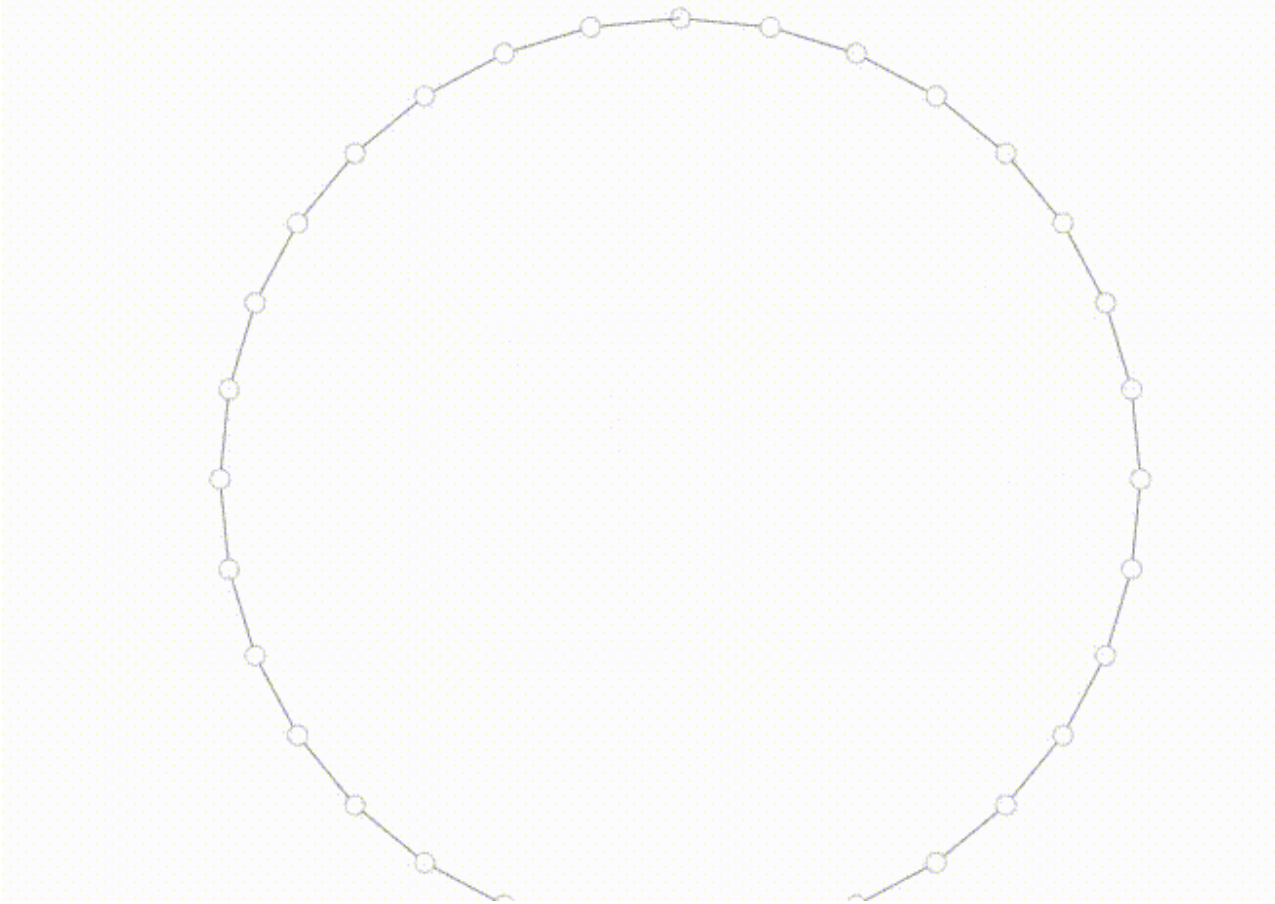
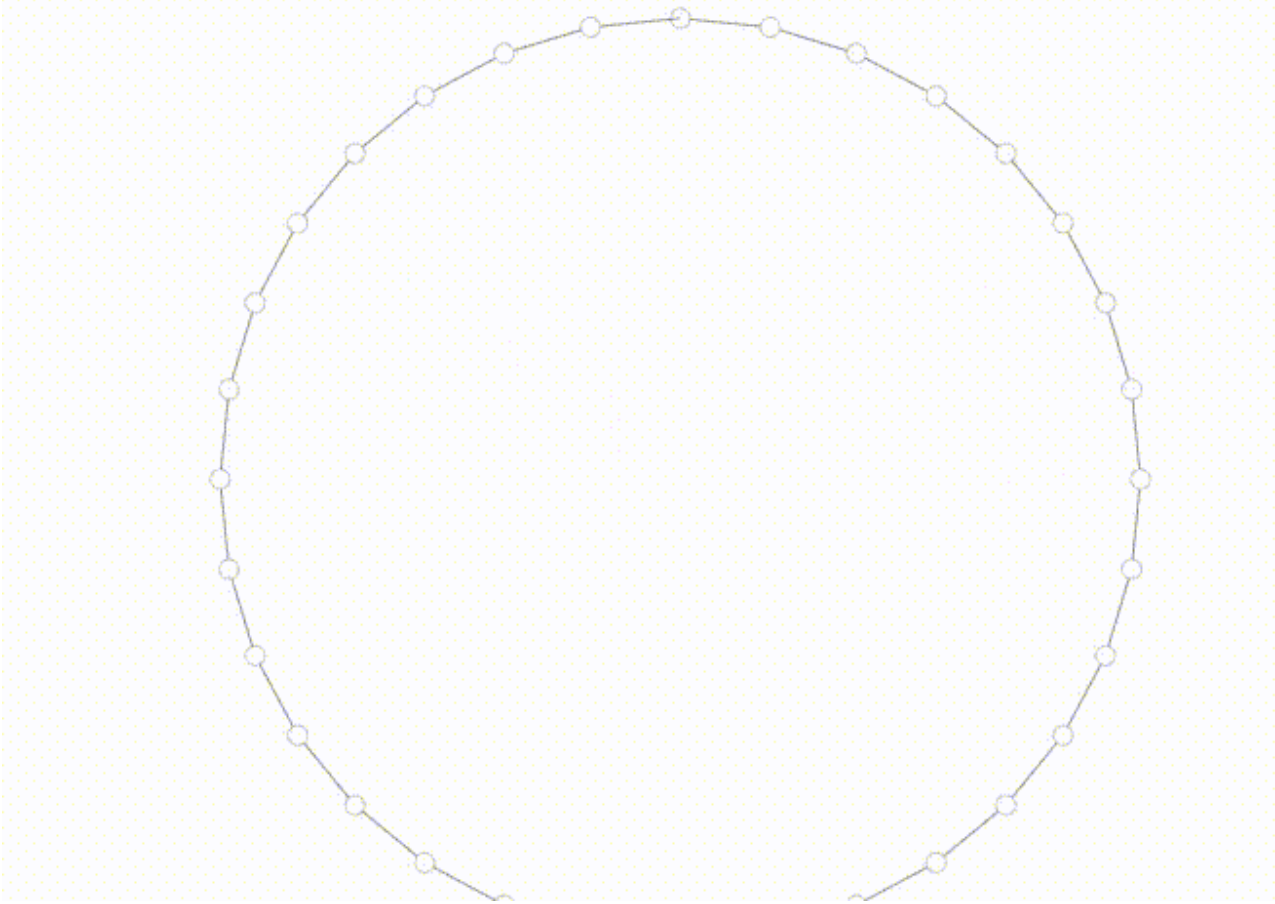
Principles of Network Analysis with NetworkX



PLAY ALONG AT:
KAGGLE.COM/STKBAILEY/NASHVILLE-MEETUP/KERNELS

Principles of Network Analysis with NetworkX



FEBRUARY 11, 2018

STEPHEN BAILEY

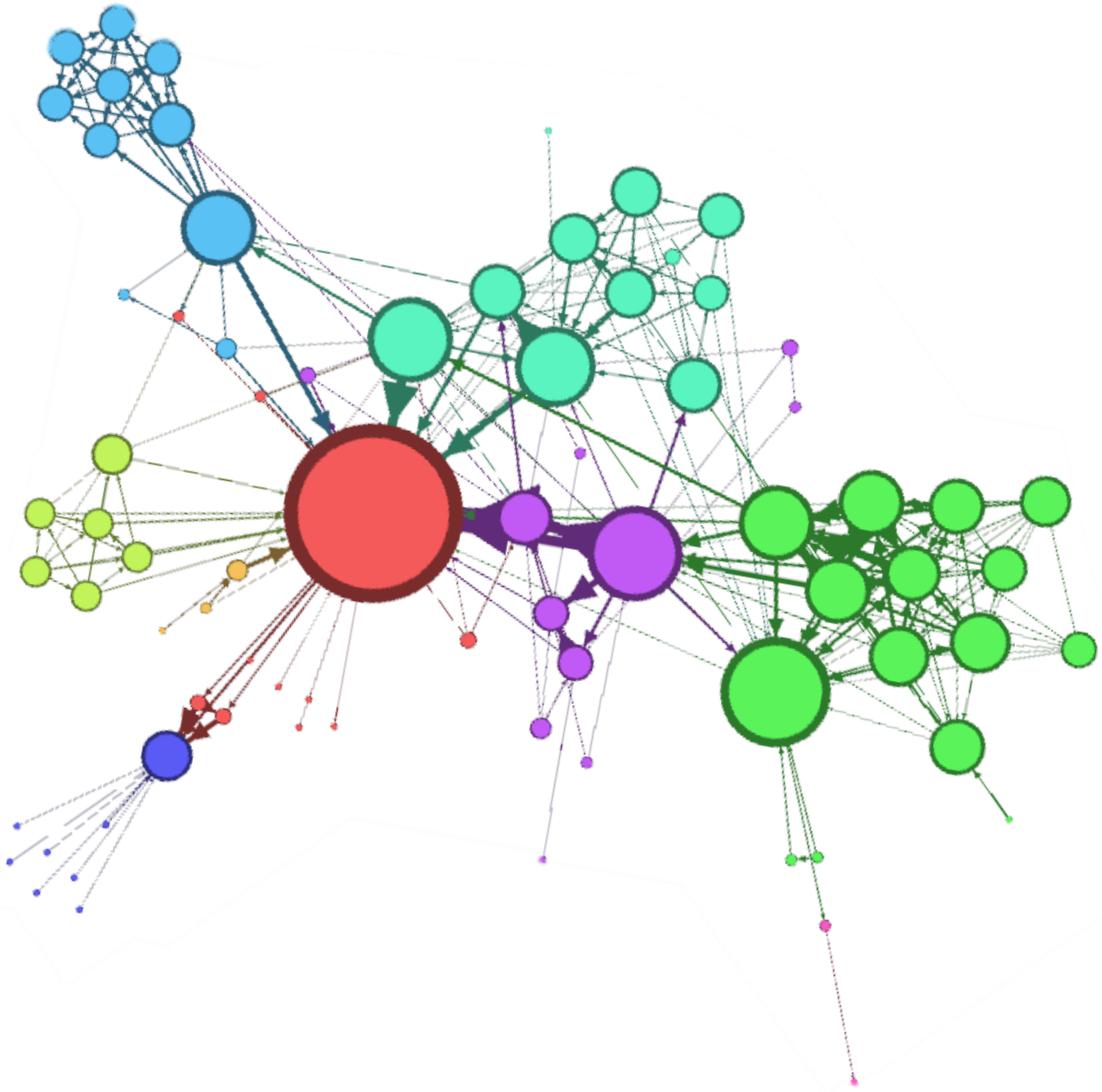
KAGGLE.COM/STKBAILEY/NASHVILLE-MEETUP

Who's this for?

I want to know what a graph is.

I want to do graph analysis in Python but don't know how.

I have some data that may have interesting relationships but don't know how to analyze.

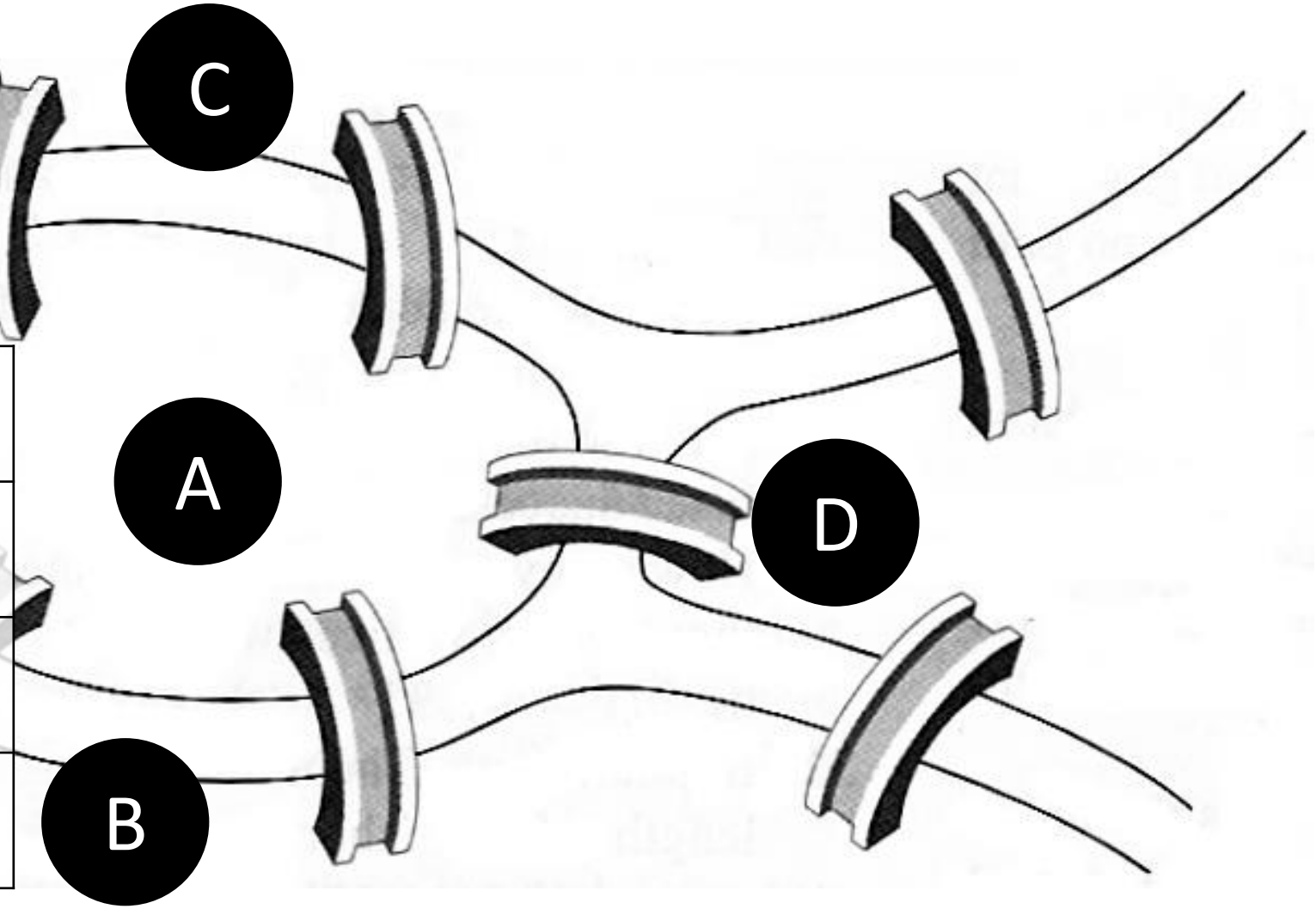


Principles of Network Analysis with NetworkX

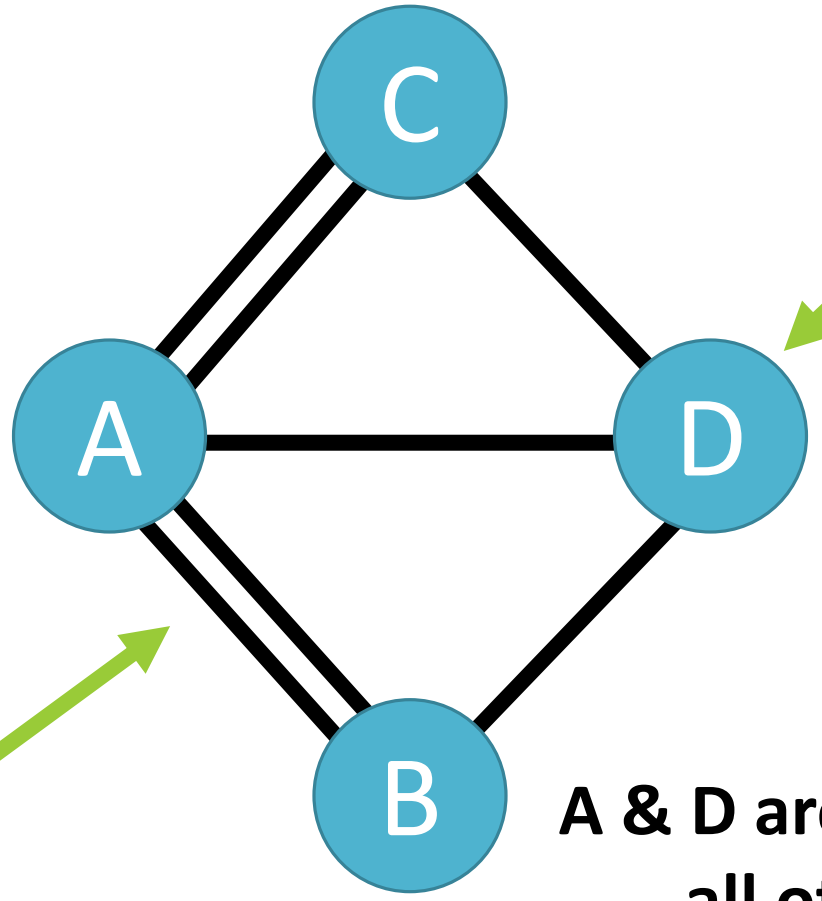
1. What is a network?
2. Build a network
3. Measure a network
4. Plot a network

Can you traverse every bridge just once?

	A	B	C	D
A	-	2	2	1
B	2	-	0	1
C	2	0	-	1
D	1	1	1	-



Graph



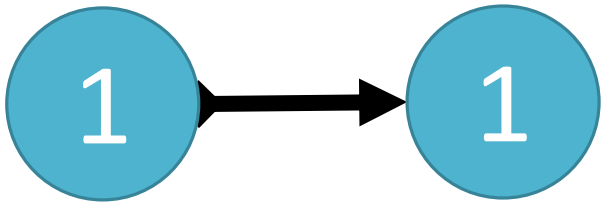
Node

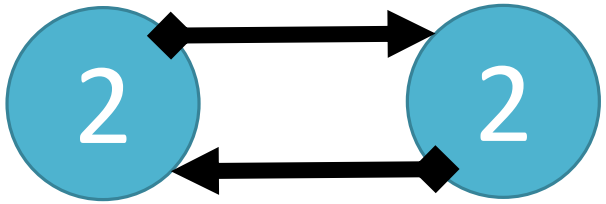
A has the most connections

Edge

A & D are connected to all other nodes

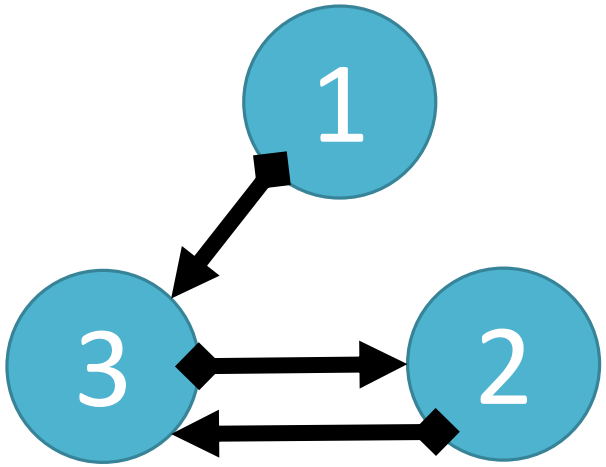
To travel into AND out of a node, requires 2 edges.



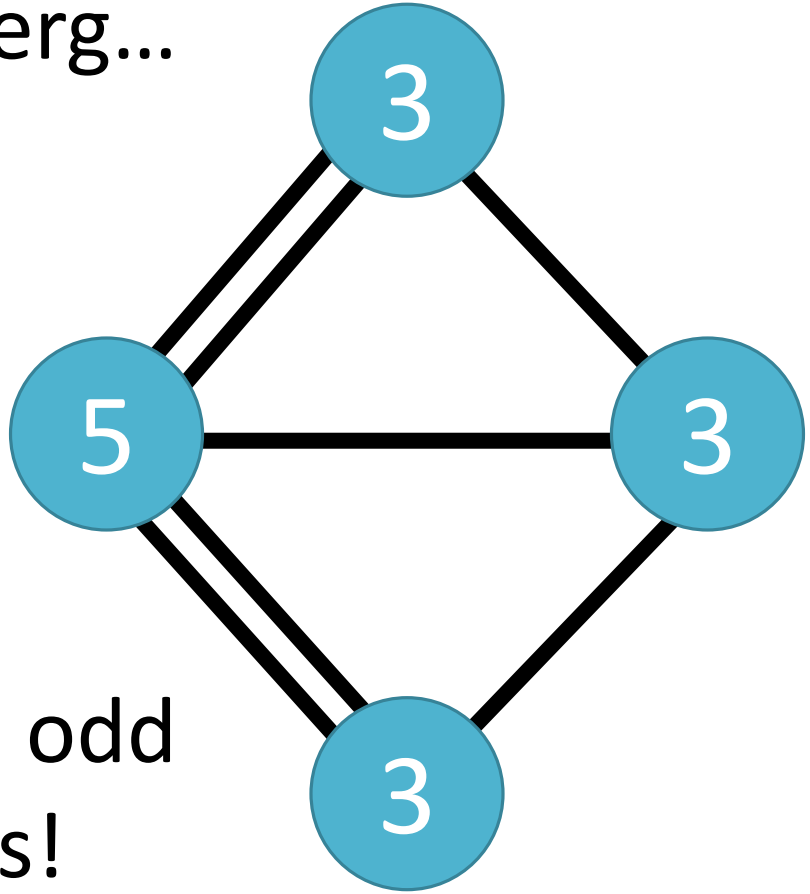


Therefore… either:

- All nodes have an even number of edges.
- Exactly two nodes have an odd number of edges.

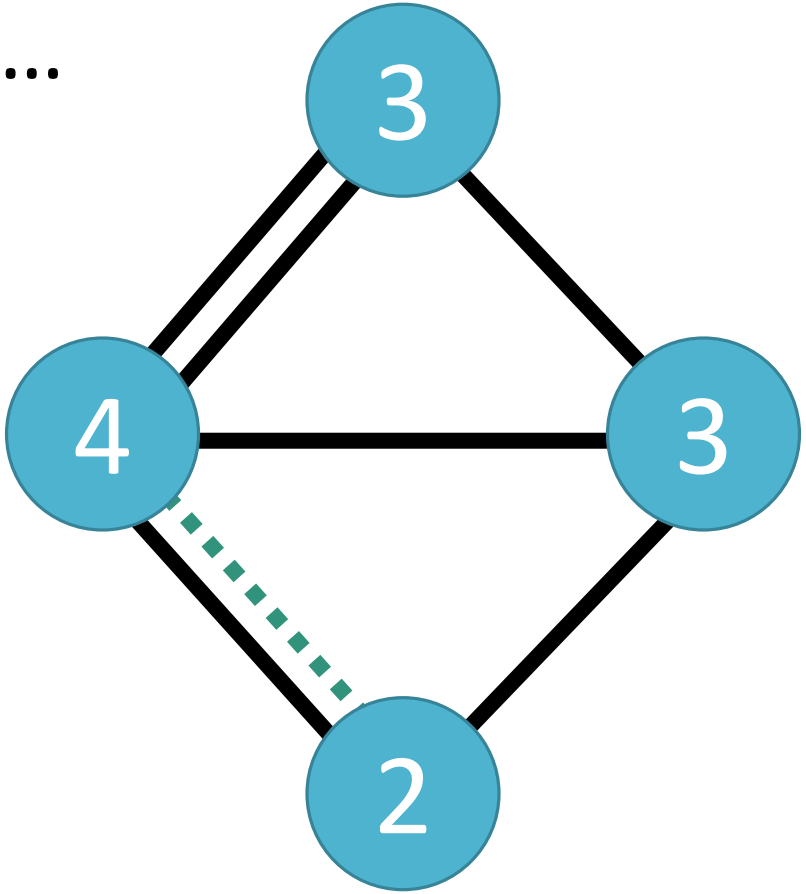


Back to Konigsberg…



Four nodes with odd number of edges!

That’s the ticket…

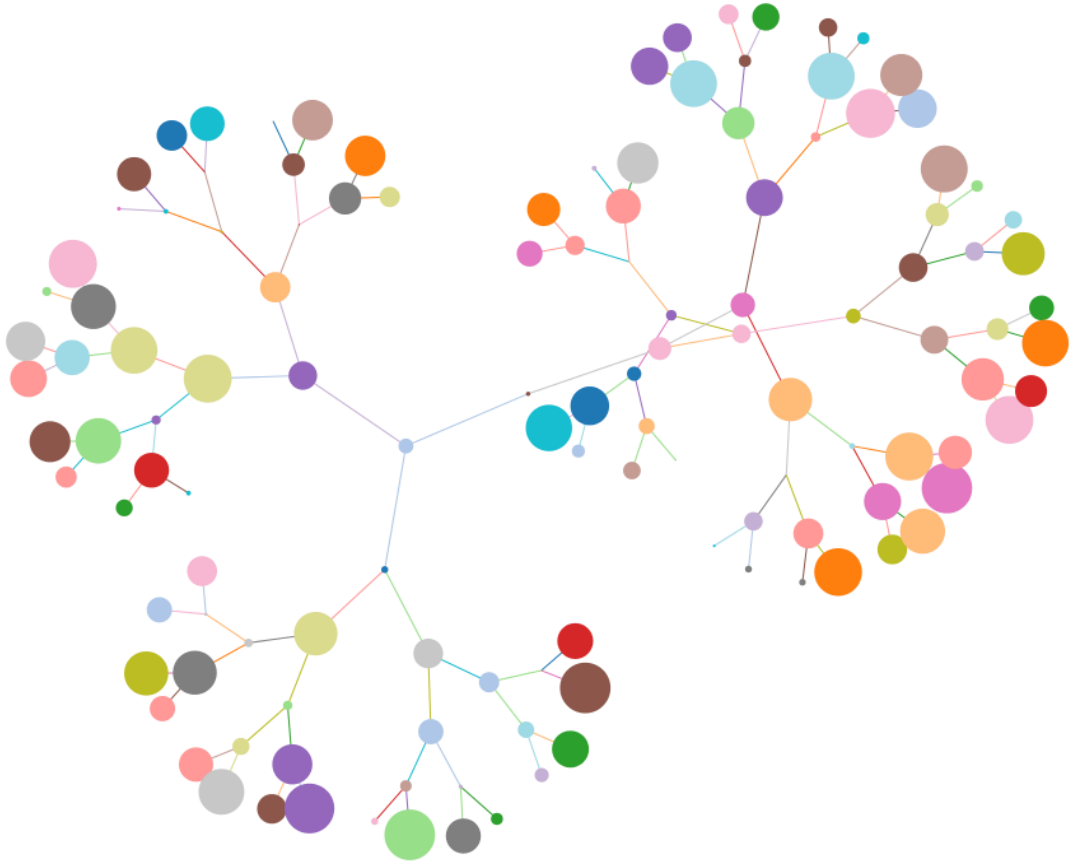


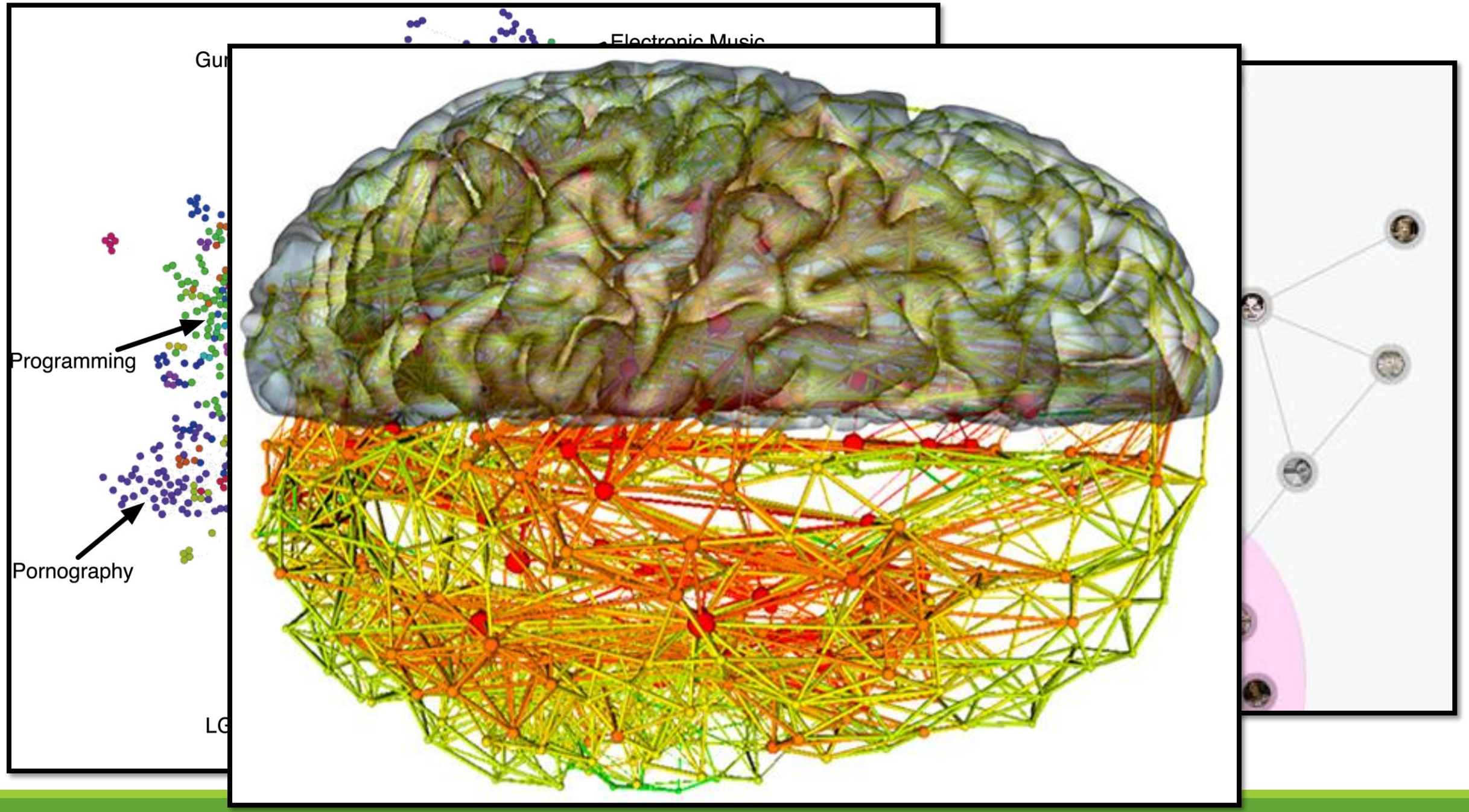
Graphs

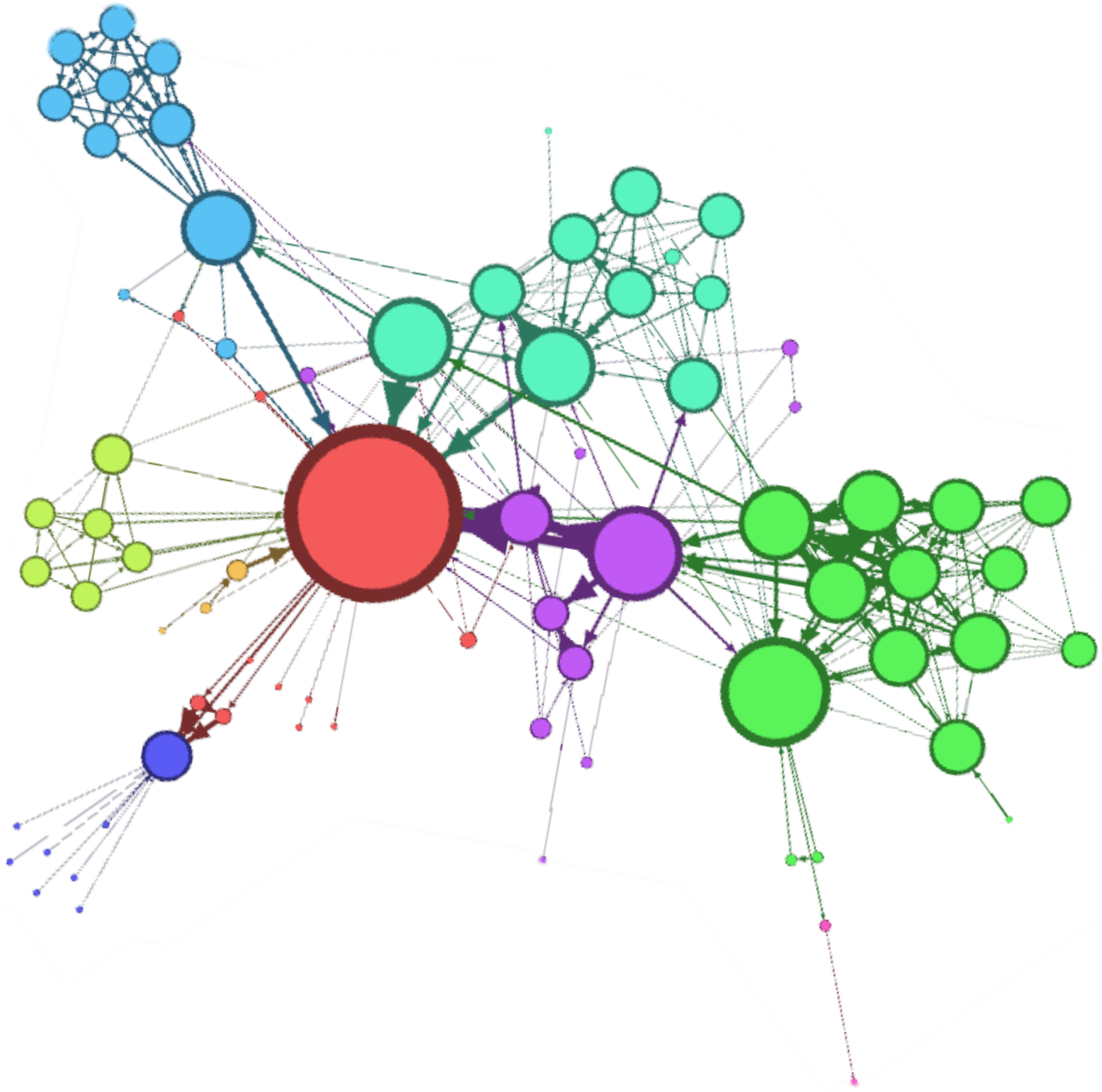
Useful for understanding **relationships**.

Describe **topology** (arrangement) of networks.

Many derivative measurements.




Programming
Pornography



Principles of Network Analysis with NetworkX

1. What is a network?
2. Build a network
3. Measure a network
4. Plot a network

NetworkX

Software for complex networks

Stable (notes)

2.0 – September 2017

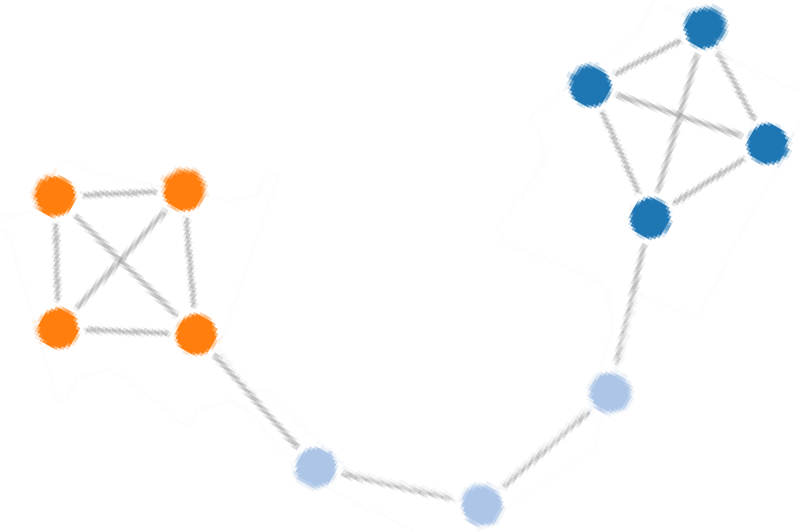
download | doc | pdf

Latest (notes)

2.1 development

github | doc | pdf

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.



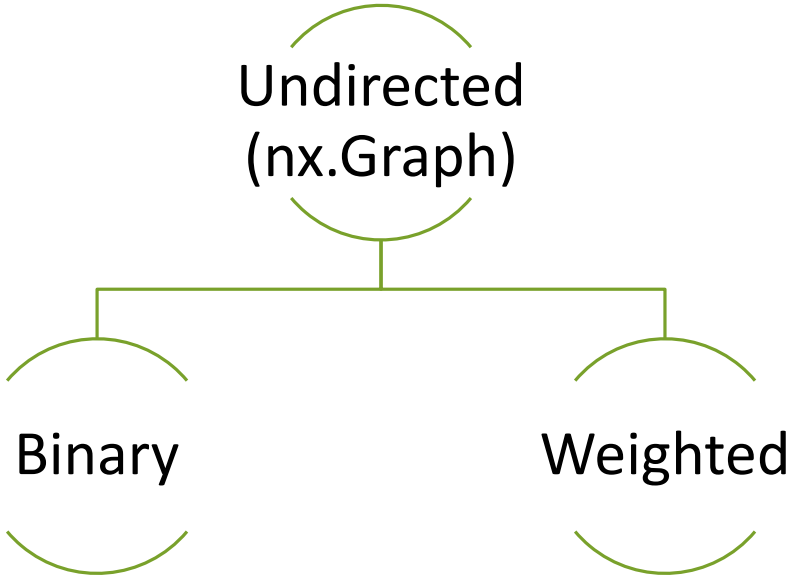
Initializing a graph

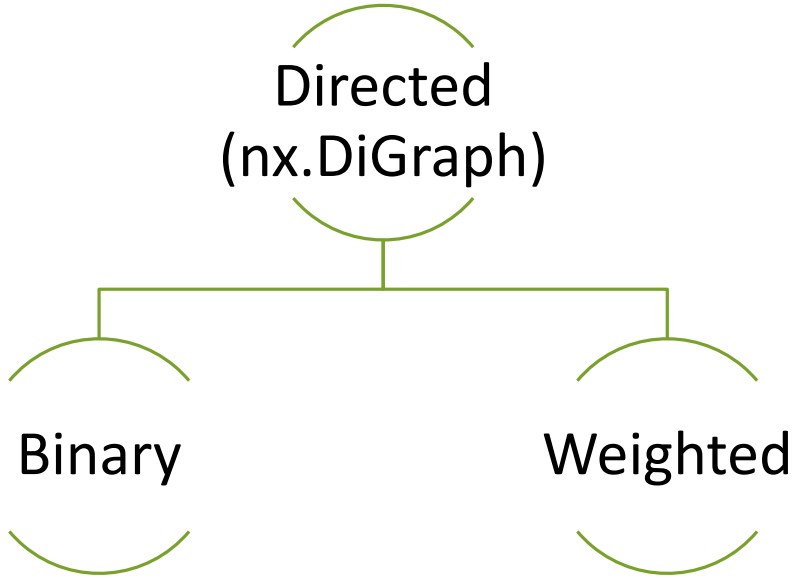
```
import networkx as nx

g = nx.Graph()
g
```

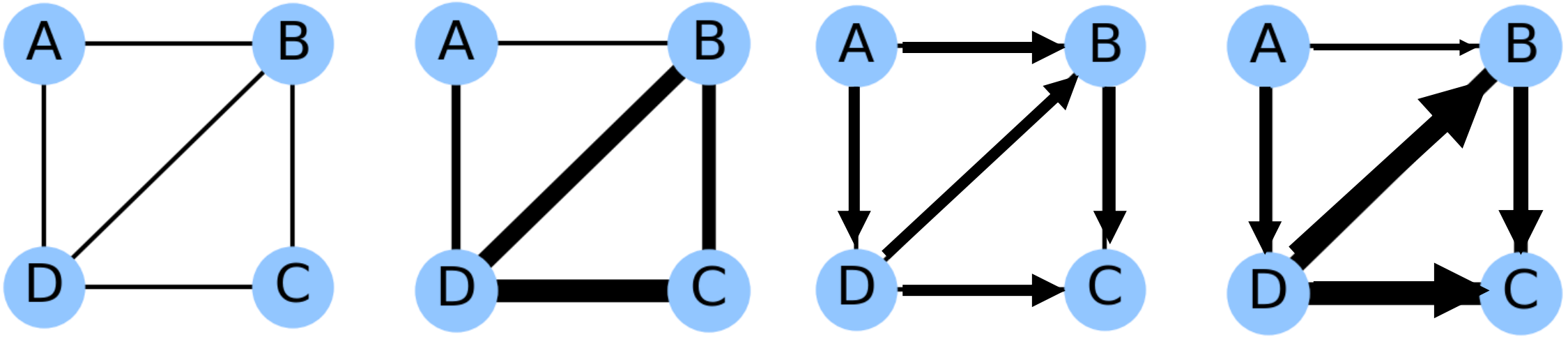
<networkx.classes.graph.Graph>

Other graph classes:

- DiGraph
- MultiGraph
- MultiDiGraph


Undirected
(nx.Graph)
Binary
Weighted

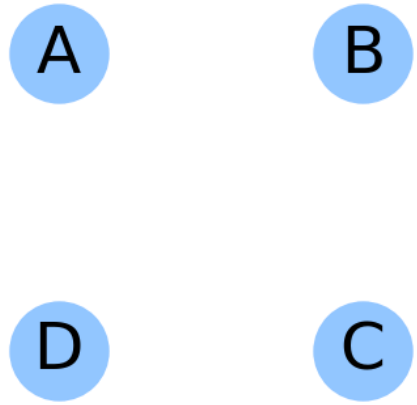

Directed
(nx.DiGraph)
Binary
Weighted


A
B
D
C
A
B
D
C
A
B
D
C
A
B
D
C

Adding nodes manually

```
g.add_node('A')
g.add_node('B')
g.add_node('C')
g.add_node('D')

g.nodes
```

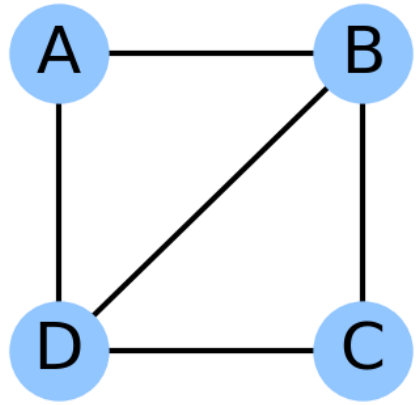
['A', 'B', 'C', 'D']



Adding edges manually

```
g.add_edge(u='A', v='B')
g.add_edge('A', 'D')
g.add_edge('B', 'C’)
g.add_edge('B', 'D')
g.add_edge('C', 'D')

g.edges
```

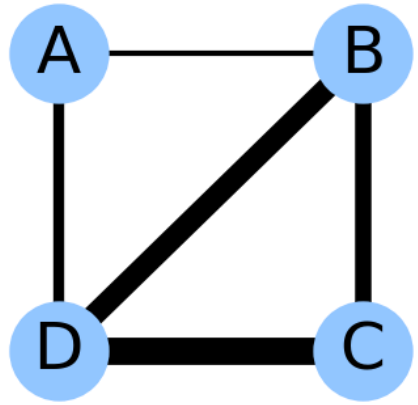
```
[('A', 'B'), ('A', 'D'),
 ('B', 'C'), ('B', 'D'),
 ('C', 'D')]
```



Setting attributes

```
edge_weights = {
    ('A', 'B'): 1,
    ('A', 'D'): 2, …}
nx.set_edge_attributes(
    G=g,
    values=edge_weights,
    name='weight’)

g.edges(data=True)
```

```
[('A', 'B', {'weight': 1}),
 ('A', 'D', {'weight': 2}),
 ('B', 'D', {'weight': 3})…]
```



Graph objects are (basically) dicts-of-dicts-of-dicts

```
g['A']
```

```
AtlasView({
      'B': {'weight': 1},
      'D': {'weight': 2}
 })
```

```
g['A']['B']
```

```
 {'weight': 1}
```

```
g['A']['B']['weight']
```

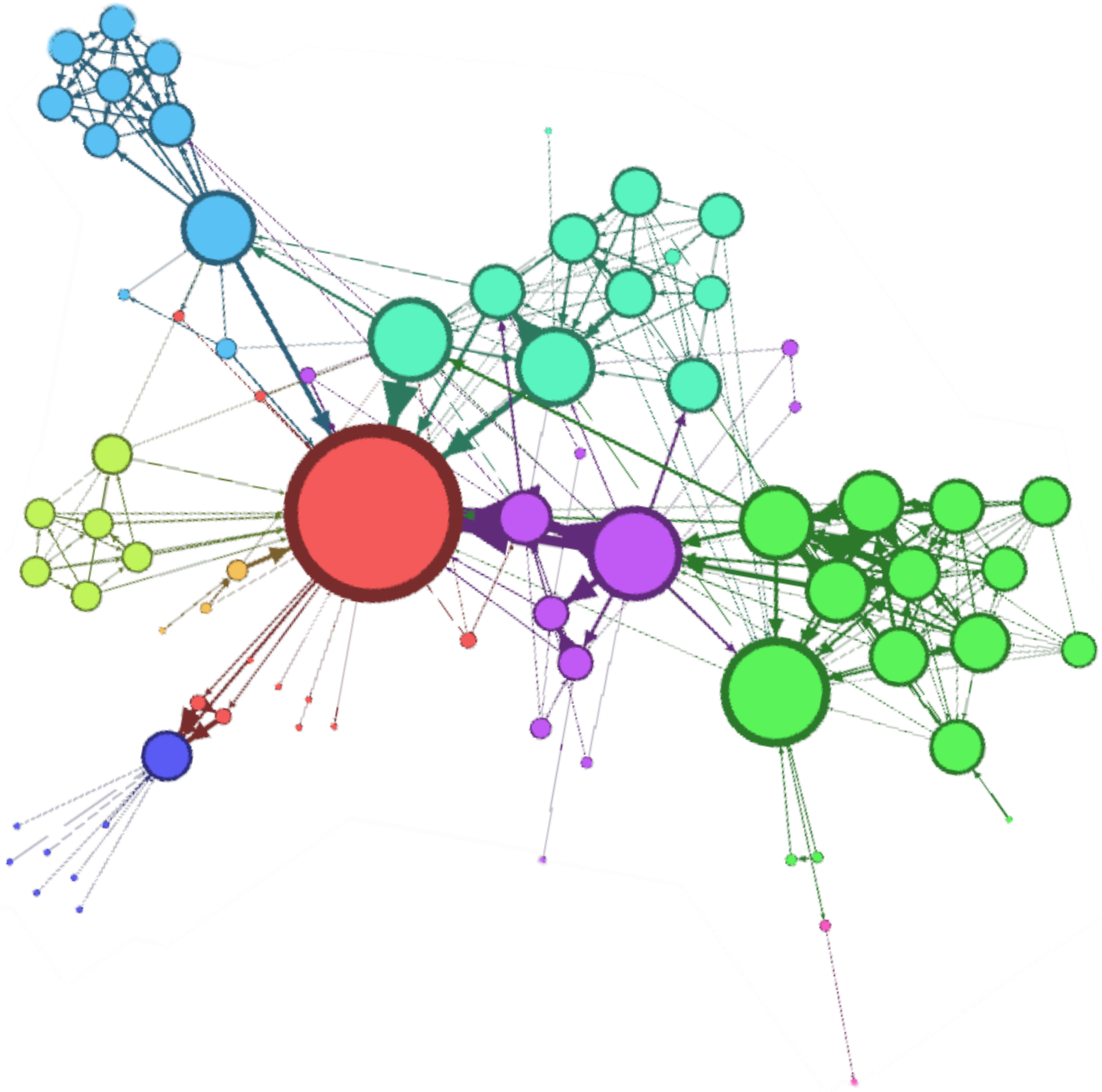
```
 1
```

Edge List: The heart of a graph

If you can put your data into an iterable of edges, then you can quickly construct a graph.

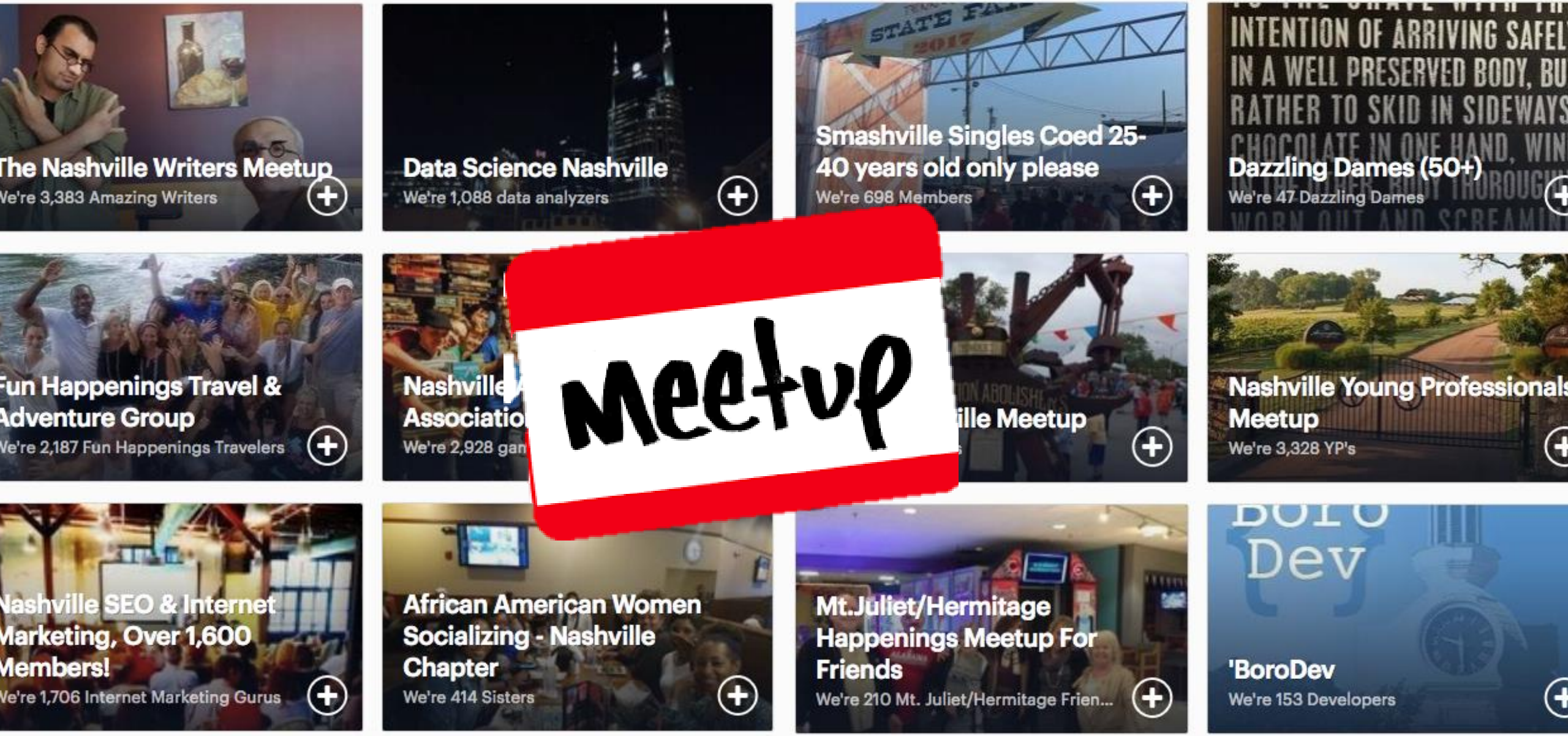
```
g = nx.from_edgelist(el)
```

source	target	weight
A	B	1
A	D	2
B	D	3
B	C	4
C	D	5



Principles of Network Analysis with NetworkX

1. What is a network?
2. Build a network
3. Measure a network
4. Plot a network


The Nashville Writers Meetup
We're 3,383 Amazing Writers
Data Science Nashville
We're 1,088 data analyzers
Smashville Singles Coed 25-40 years old only please
We're 698 Members
Dazzling Dames (50+)
We're 47 Dazzling Dames
Fun Happenings Travel & Adventure Group
We're 2,187 Fun Happenings Travelers
Meetup
Nashville Young Professionals Meetup
We're 3,328 YP's
Nashville SEO & Internet Marketing, Over 1,600 Members!
We're 1,706 Internet Marketing Gurus
African American Women Socializing - Nashville Chapter
We're 414 Sisters
Mt.Juliet/Hermitage Happenings Meetup For Friends
We're 210 Mt. Juliet/Hermitage Frien...
'BoroDev
We're 153 Developers

Graphing Nashville MeetUps

Data available via REST API.

1. Members join Groups online and RSVP to events.
   - We only count a member-group mapping as valid if they have RSVPed to 2 events in the past year.

Create two networks:

- Relationships between groups
  - i.e., # of shared members
- Relationships between members
  - i.e, # of shared groups

Graphing Nashville MeetUps

PyNash

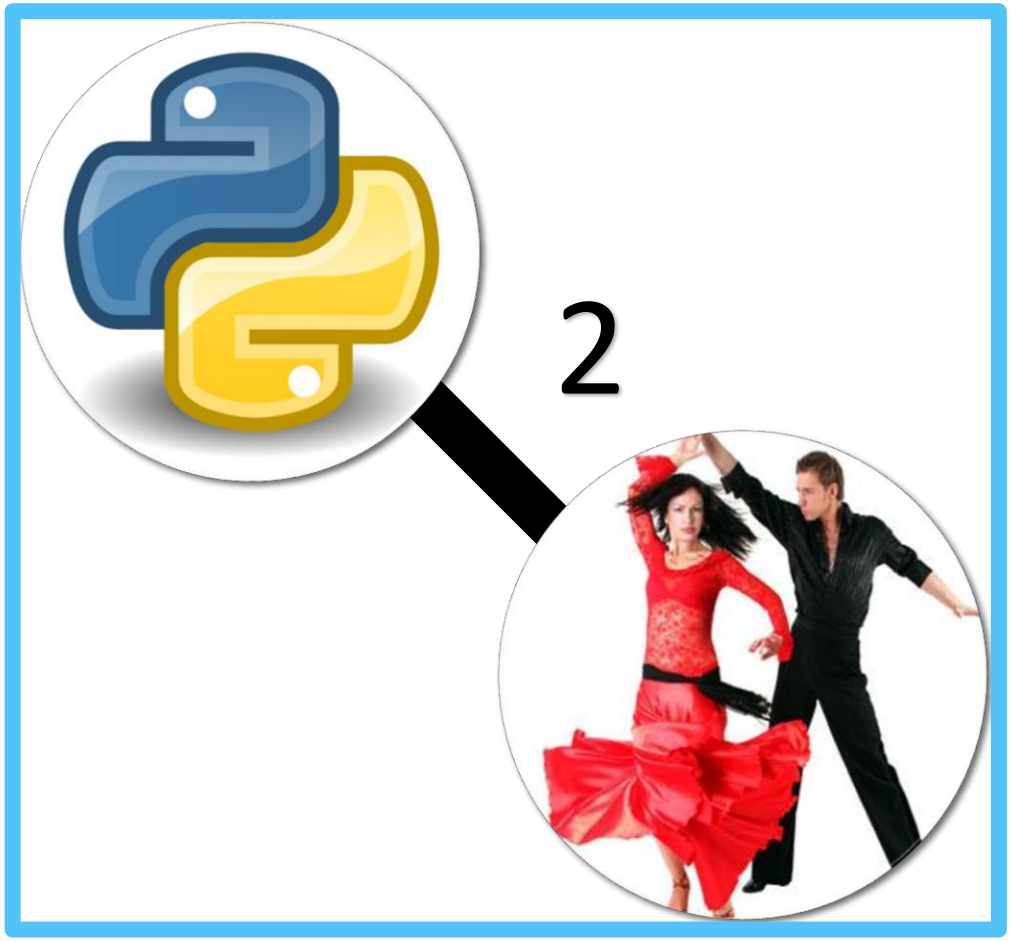
Chevy

Steve

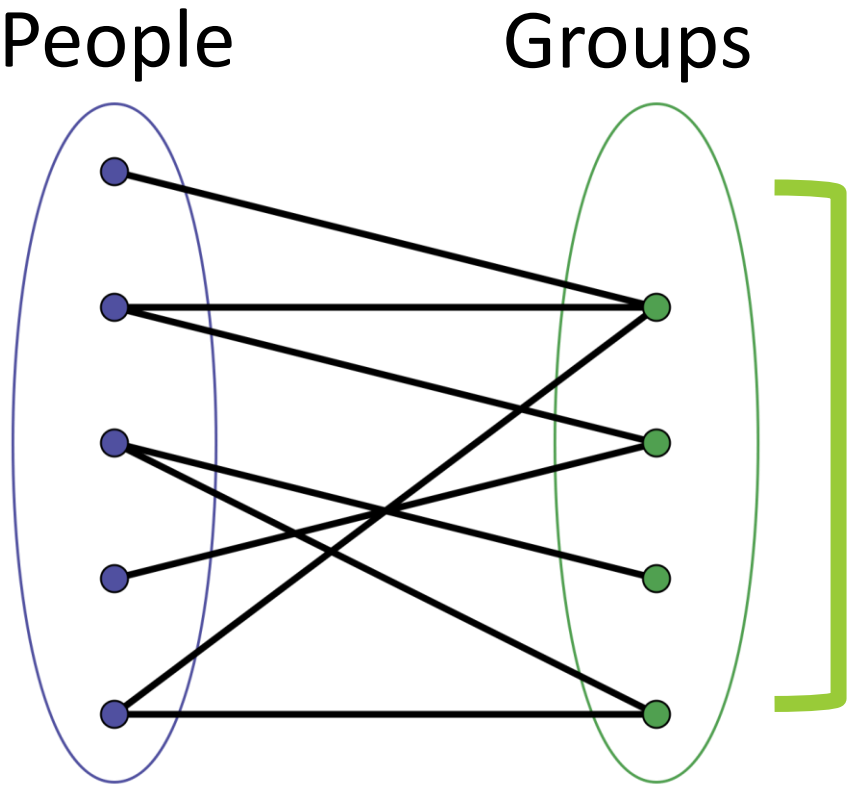
Martin

Nashville Salsa Congress

Member & Group Graphs



Sidebar: Bipartite graphs



Two classes,
No within-class connections

Other examples:
- Actor → Movie
- Student → School
- Person → Hobby

```
g = weighted_projected_graph(
        bipartite_graph,
        list_of_class_nodes)
```

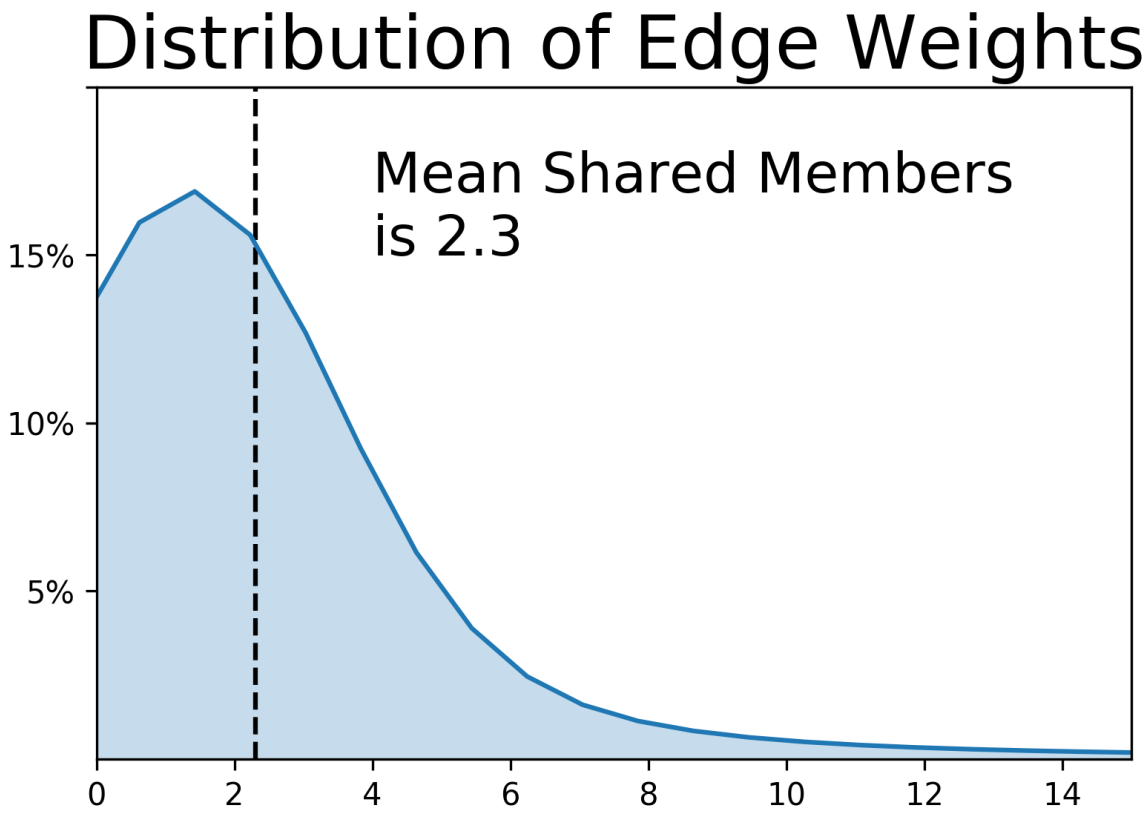
Let’s do it!

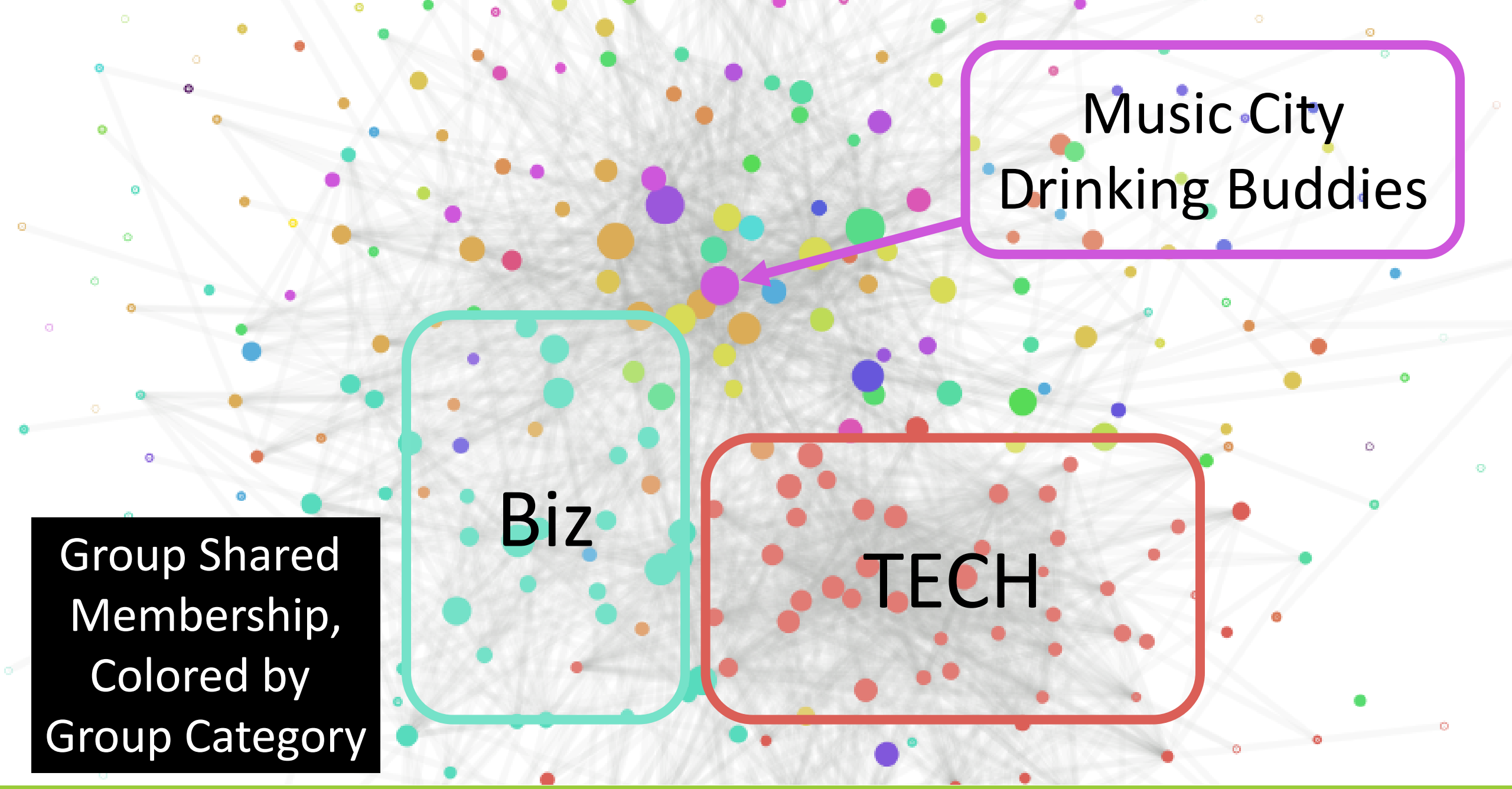

```
group_edges =
    pd.read_csv('group-edges.csv')
g = nx.from_pandas_edgelist(
        df = group_edges,
        source = 'group1',
        target = 'group2',
        edge_attr = 'weight')
```

group1	group2	weight
19292162	535553	2
19292162	19194894	1
19292162	19728145	1
19292162	18850080	2
19292162	1728035	1
19292162	22817838	2
19292162	19997487	2
19292162	18855476	2
19292162	18955830	1
19292162	11294262	1

Group Graph Breakdown

Metric	Command	Value
Num. Groups	`len(g.nodes)`	446
Num. Cnxns	`len(g.edges)`	6692




Music City
Drinking Buddies
Biz
TECH
Group Shared Membership, Colored by Group Category

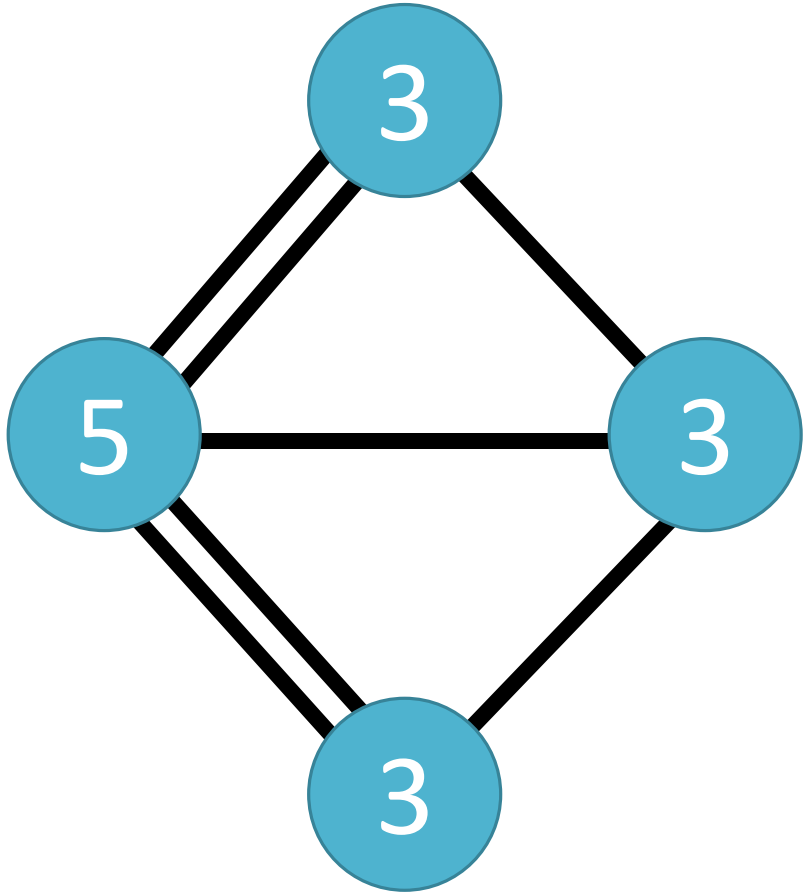
Graph Measures: General principles

- Typically accessed via *nx.some_algorithm(g)*
- Often returns a dictionary of *{node: value, …}*
- Interpretations depend on data!

```
# Example measures
nx.degree(g)
nx.clustering(g)
nx.shortest_path(g)
nx.betweenness_
    centrality(g)
```

Graph Measures: Degree

Total number of edges at a node

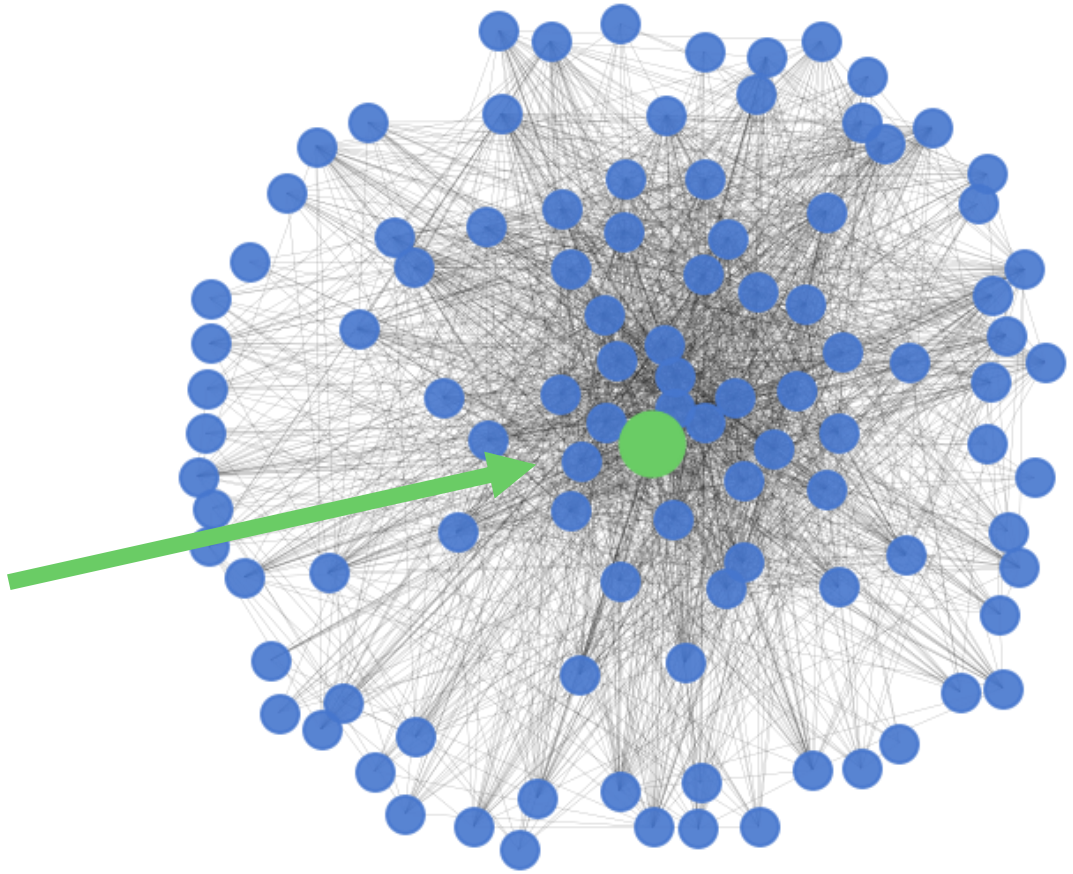


Graph Measures: Degree

Total number of edges at a node

```
deg = nx.degree(g)
deg['PyNash']
```

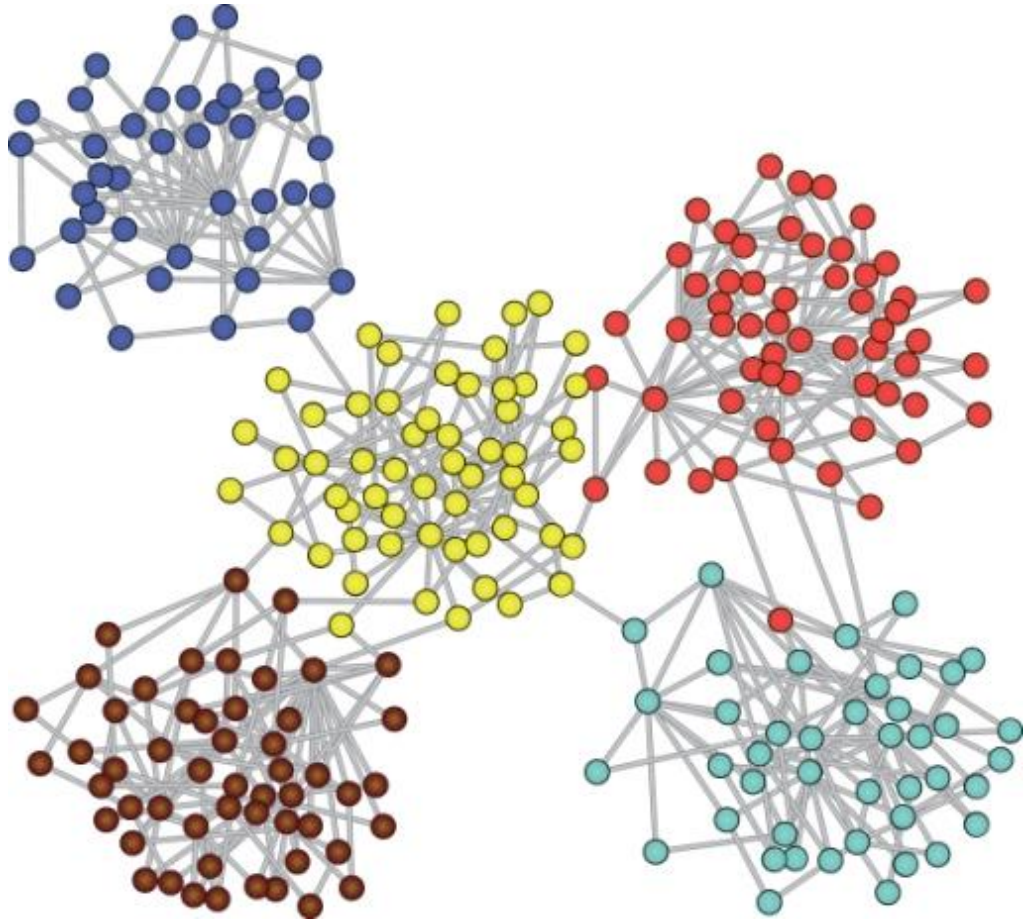
```
101
```



Graph Measures: Clustering

Likelihood of a node's connections also being connected

```
nx.clustering(g)
```

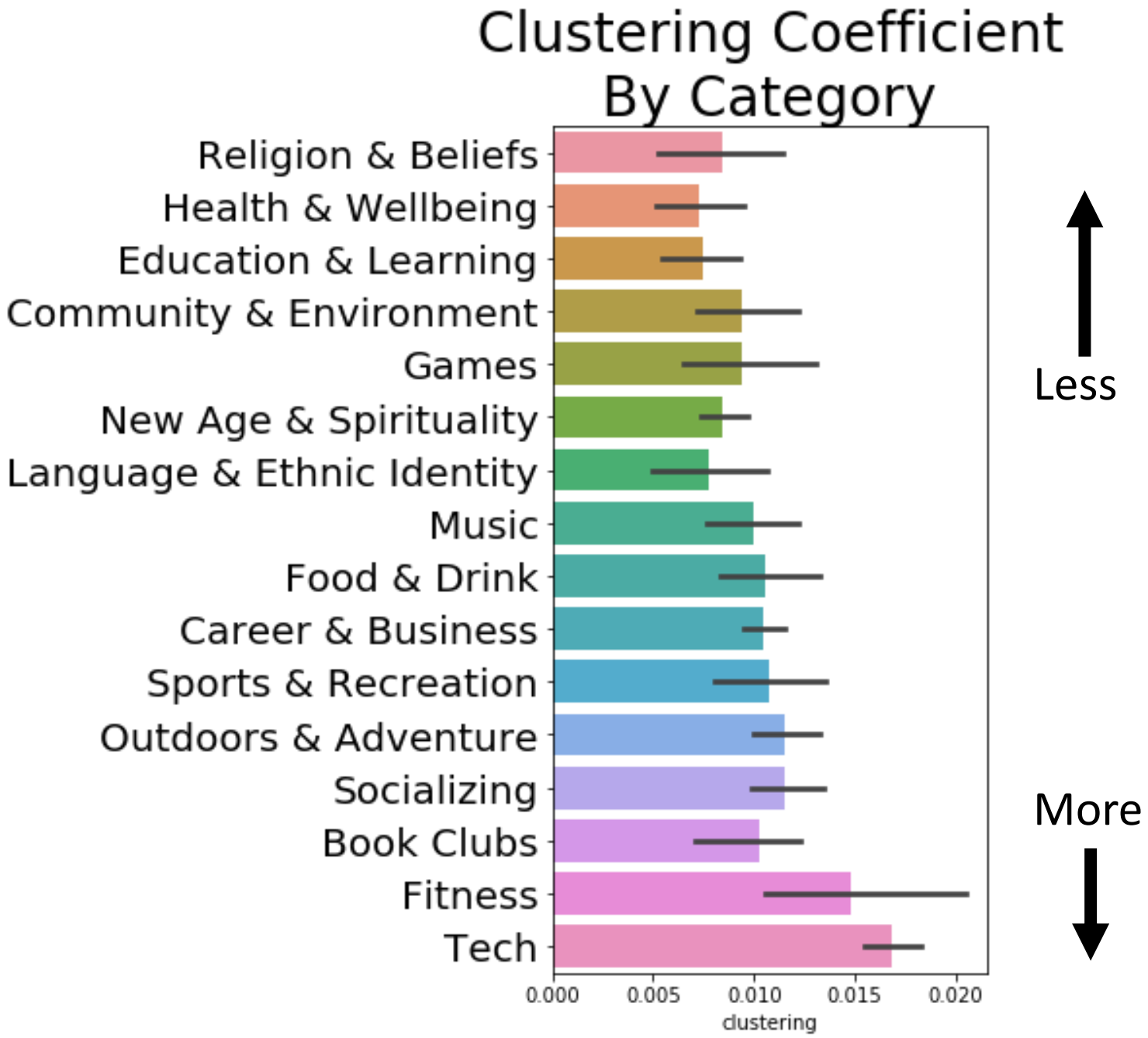


Low Clustering

- Drinking Buddies
- Movie Lovers
- Free & Cheap Events
- Nash. Musicians

High Clustering

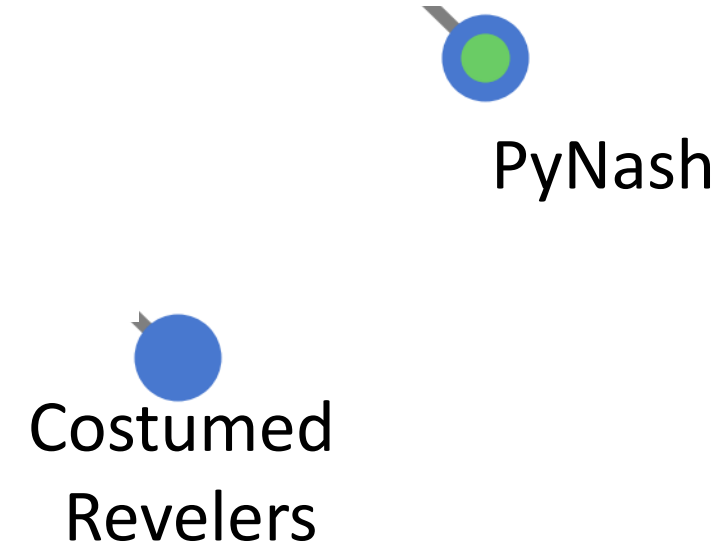
- Nash.JS
- Nashville D&D
- Data Science Nash.
- Mid TN 40+ Singles

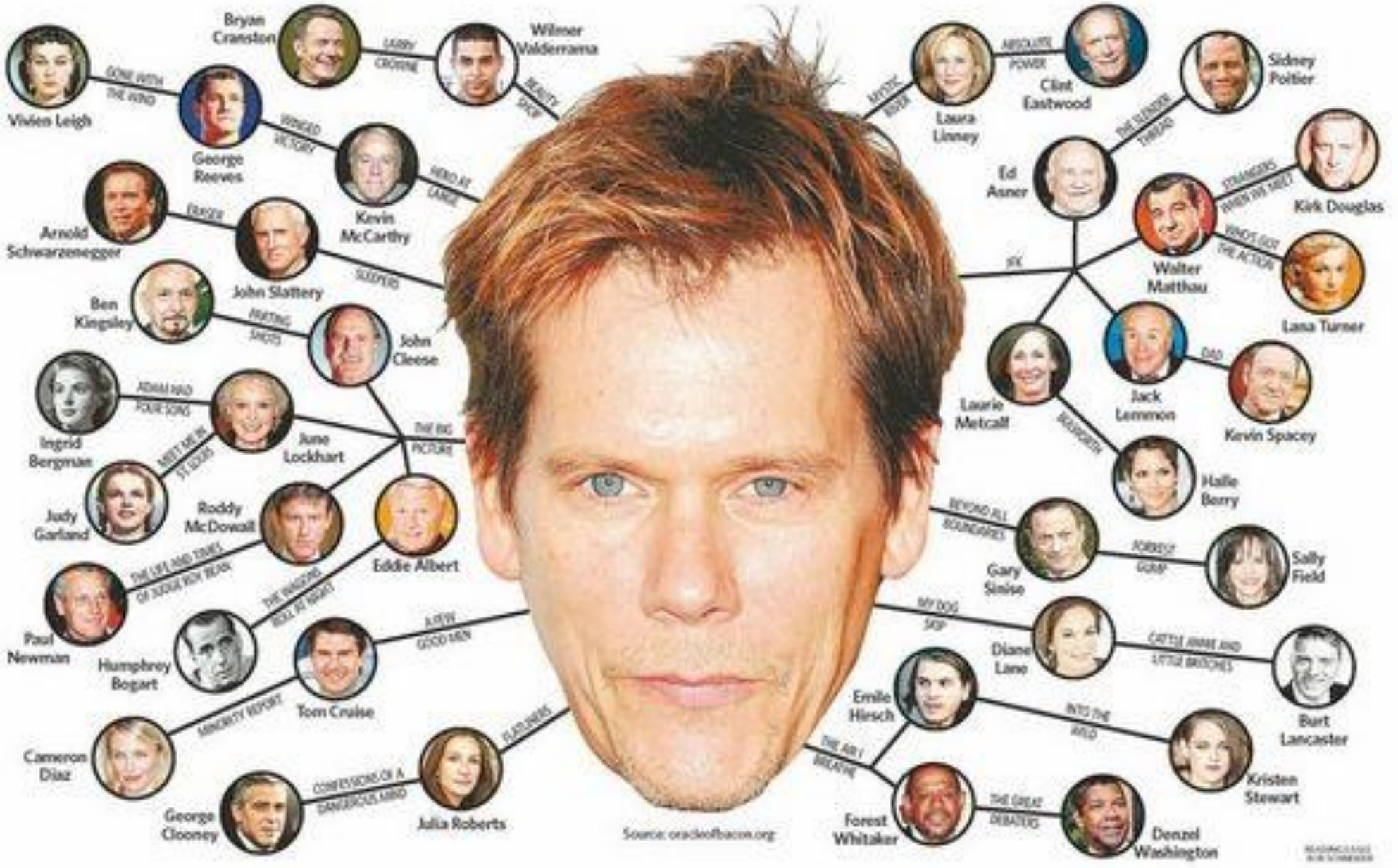


Graph Measures: Path Length

Number of “steps” to get from one node to another

```
nx.shortest_path_length(
  G=g,
  source='PyNash',
  target='CostumedRevelers')
```



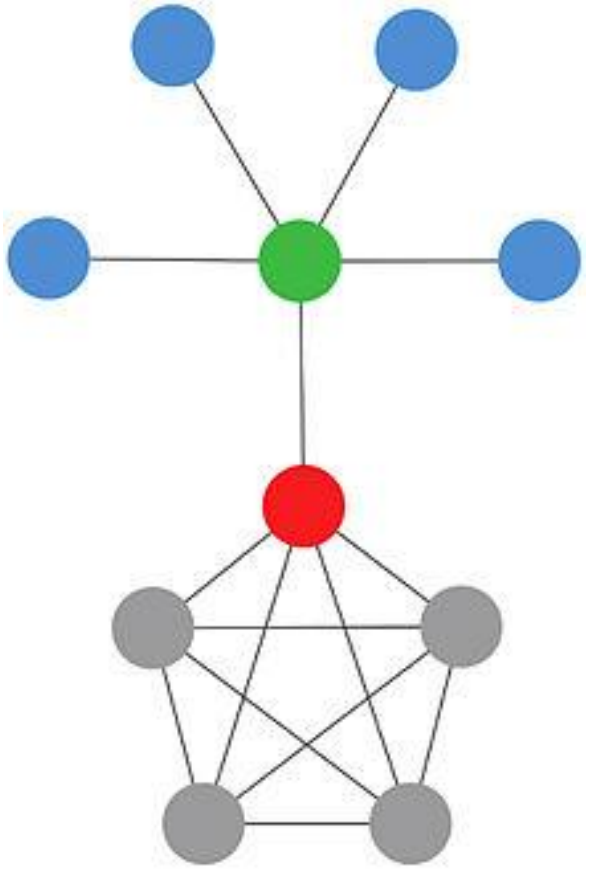

Bryan Cranston
Wilmer Valderrama
Vivien Leigh
George Reeves
Kevin McCarthy
Arnold Schwarzenegger
John Slattery
Ben Kingsley
John Cleese
Ingrid Bergman
June Lockhart
Judy Garland
Roddy McDowall
Eddie Albert
Paul Newman
Humphrey Bogart
Tom Cruise
Cameron Diaz
George Clooney
Julia Roberts
Source: oracleofbacon.org
Laura Linney
Clint Eastwood
Sidney Poitier
Ed Asner
Kirk Douglas
Walter Matthau
Lana Turner
Laurie Metcalf
Jack Lemmon
Kevin Spacey
Halle Berry
Gary Sinise
Sally Field
Diane Lane
Burt Lancaster
Emile Hirsch
Kristen Stewart
Forest Whitaker
Denzel Washington

Graph Measures: Centrality

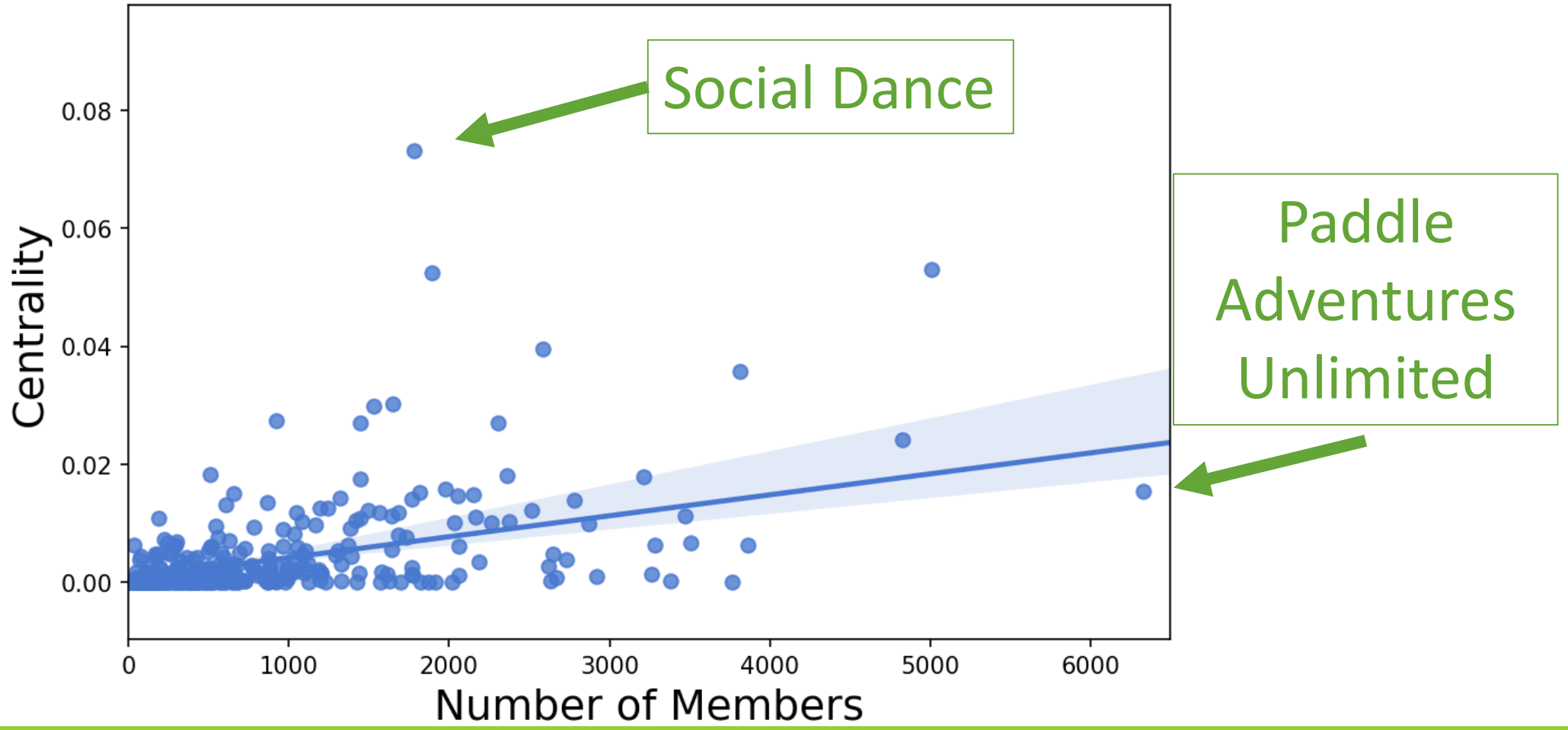
How important a node is to connecting the whole network

```
nx.betweenness_centrality(g)
```

Important metric – often used to assess "vulnerability"



Centrality is not simply the “size” – it is also who they connect.



The groups that tie Nashville together

Group Name	Category	Centrality
Stepping Out Social Dance	Dancing	0.0731
Eat Love Nash	Socializing	0.0531
What the Pho!	Food & Drink	0.0525
Middle TN 40+ singles	Singles	0.0394
Nashville Hiking Meetup	Outdoors & Adventure	0.0362
20s in Nashville	Socializing	0.0357
Nashville SEO	Career & Business	0.0301

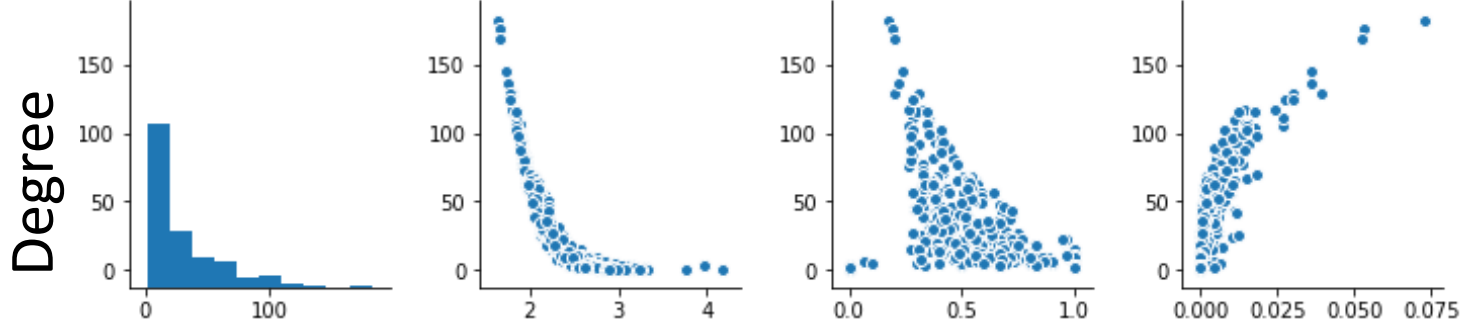
Adding measures to a DataFrame

Since measures are output as dictionaries, you can add them to node-indexed DataFrames by converting them into Series.

```
from pandas import DataFrame
df = DataFrame(index=g.nodes)


clust = nx.clustering(g)
df['clust'] = pd.Series(clust)
```

Once in a DataFrame, traditional plots are possible.

Measures are often correlated with each other, especially in simple graphs.



Degree Path Length Clustering Centrality

What group to join?

To infiltrate the network…

- Attend groups with high centrality and degree

To go deep in a community…

- Attend MeetUps with high clustering, and go to several connected ones

… But who do I talk to???

Member & Group Graphs

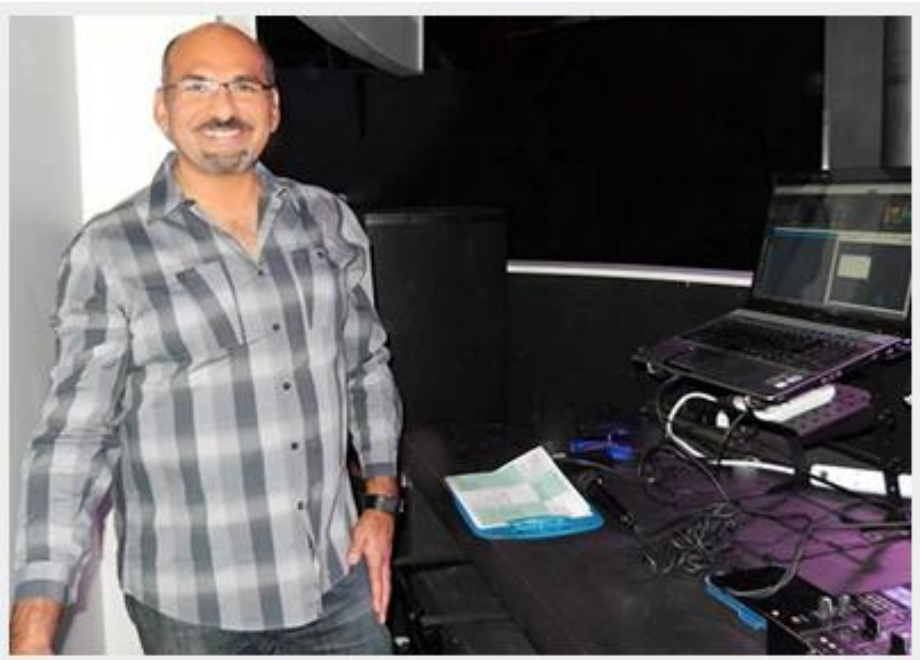
Nashville’s Elite Meeter-Uppers

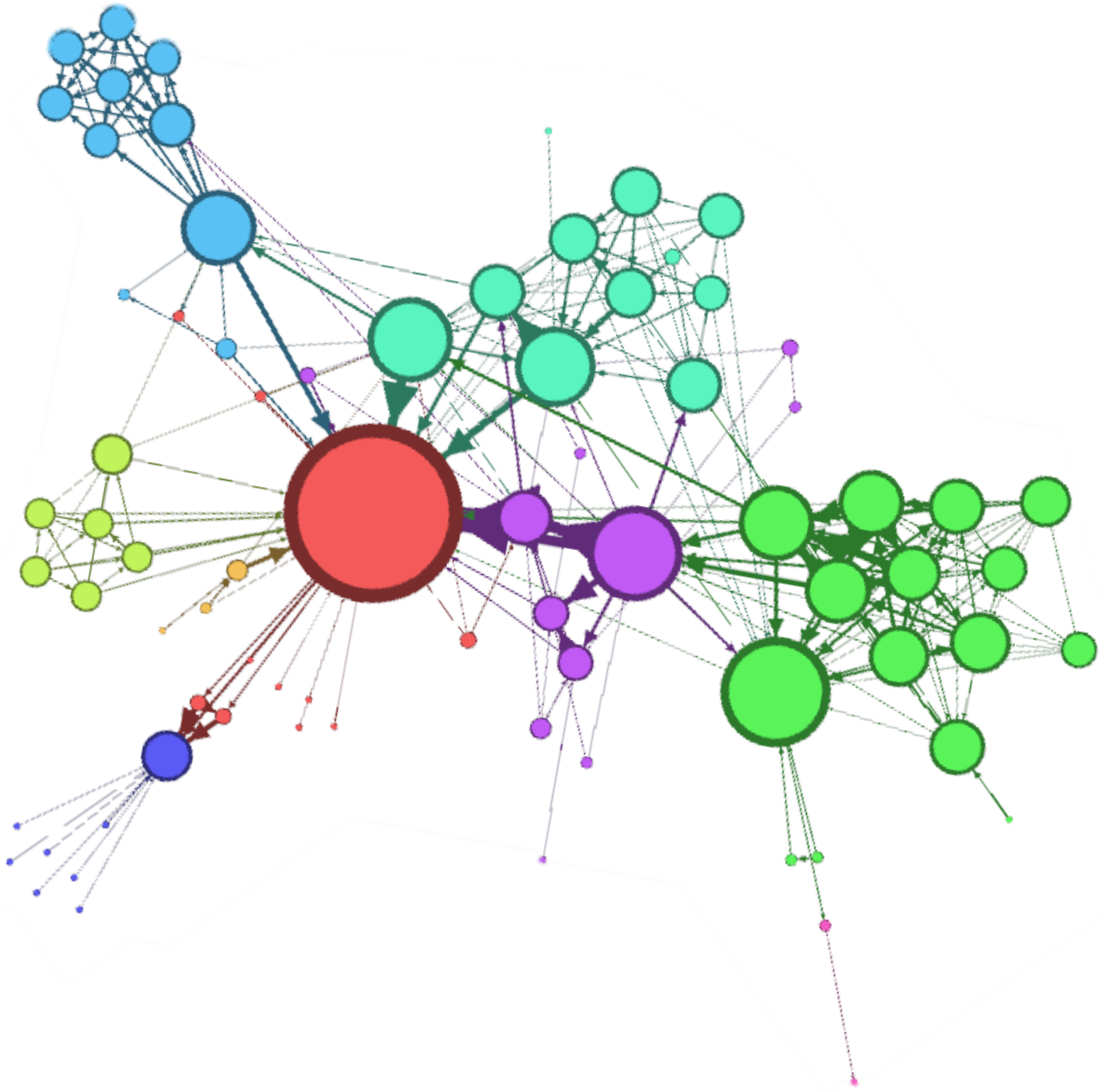
Member of 185 other Meetups

Matt Kenigson

President

Makerspace





Principles of Network Analysis with NetworkX

1. What is a network?
2. Build a network
3. Measure a network
4. Plot a network

Plotting with NetworkX and Matplotlib

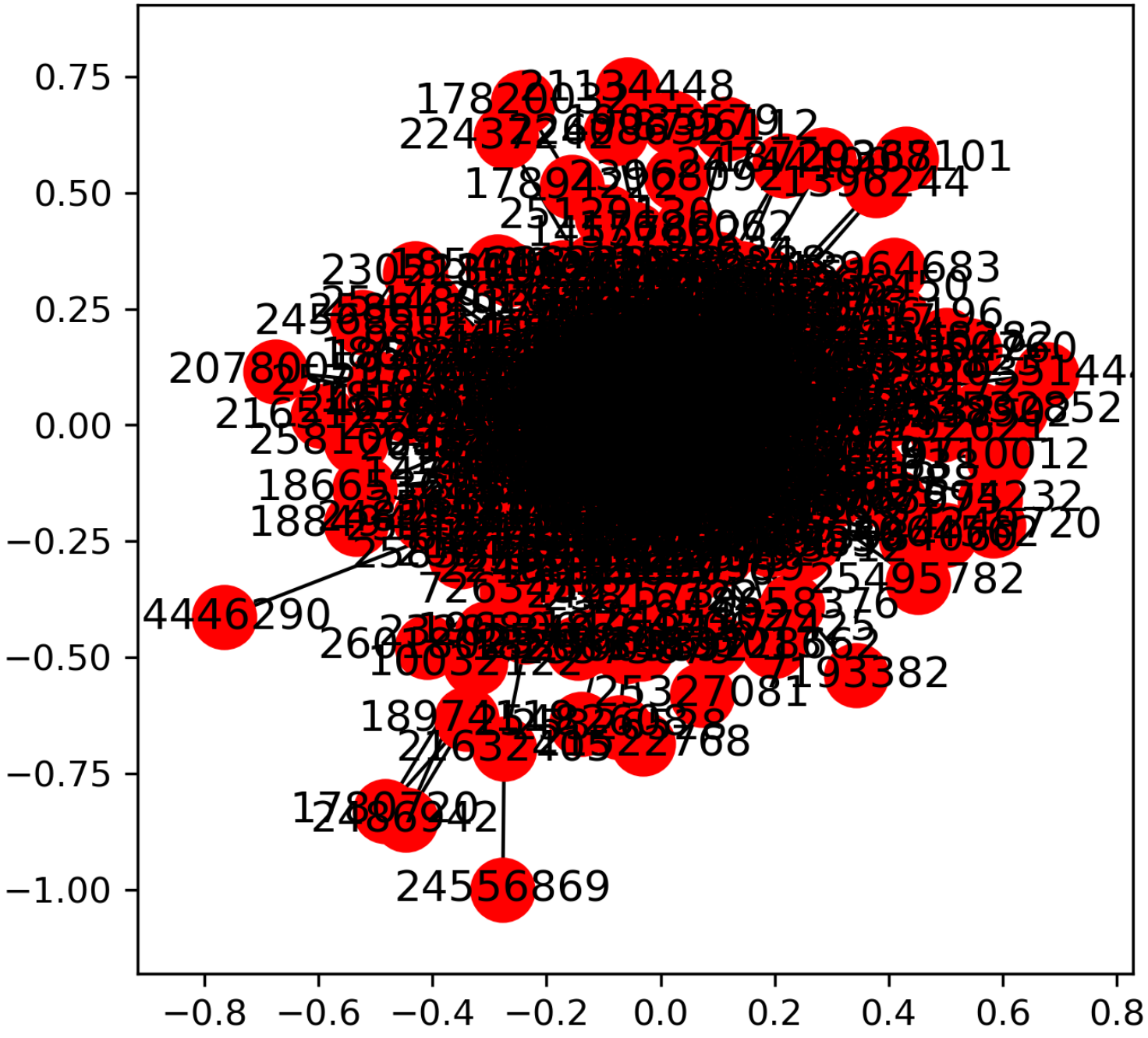
Generally, three steps:

1. Arrange nodes
2. Plot graph
3. Adjust figure

```
pos = nx.spring_layout(g)
nx.draw_networkx(g, pos)

plt.show()
```

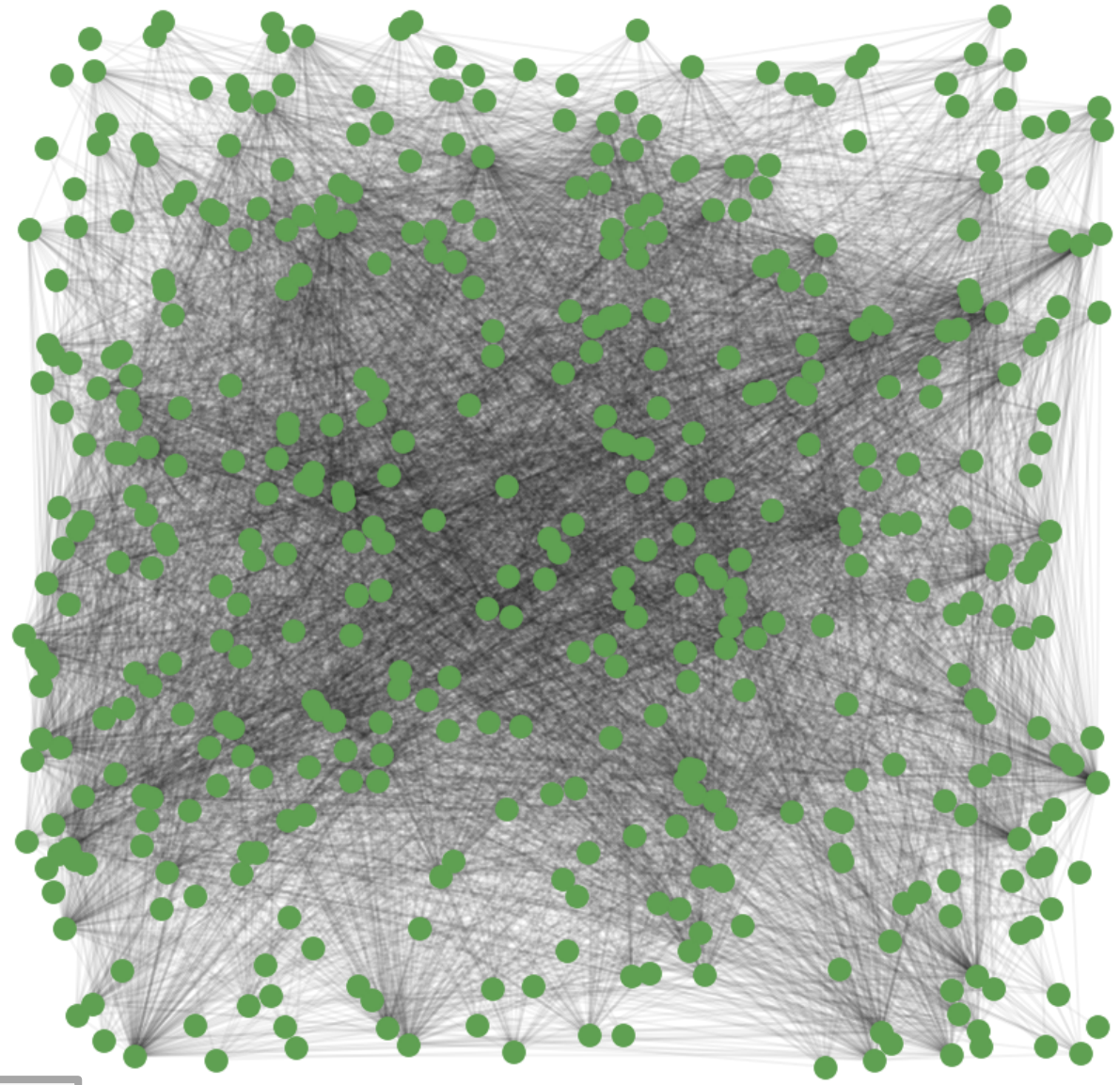
Full Defaults

The essence of beauty



The Random Layout

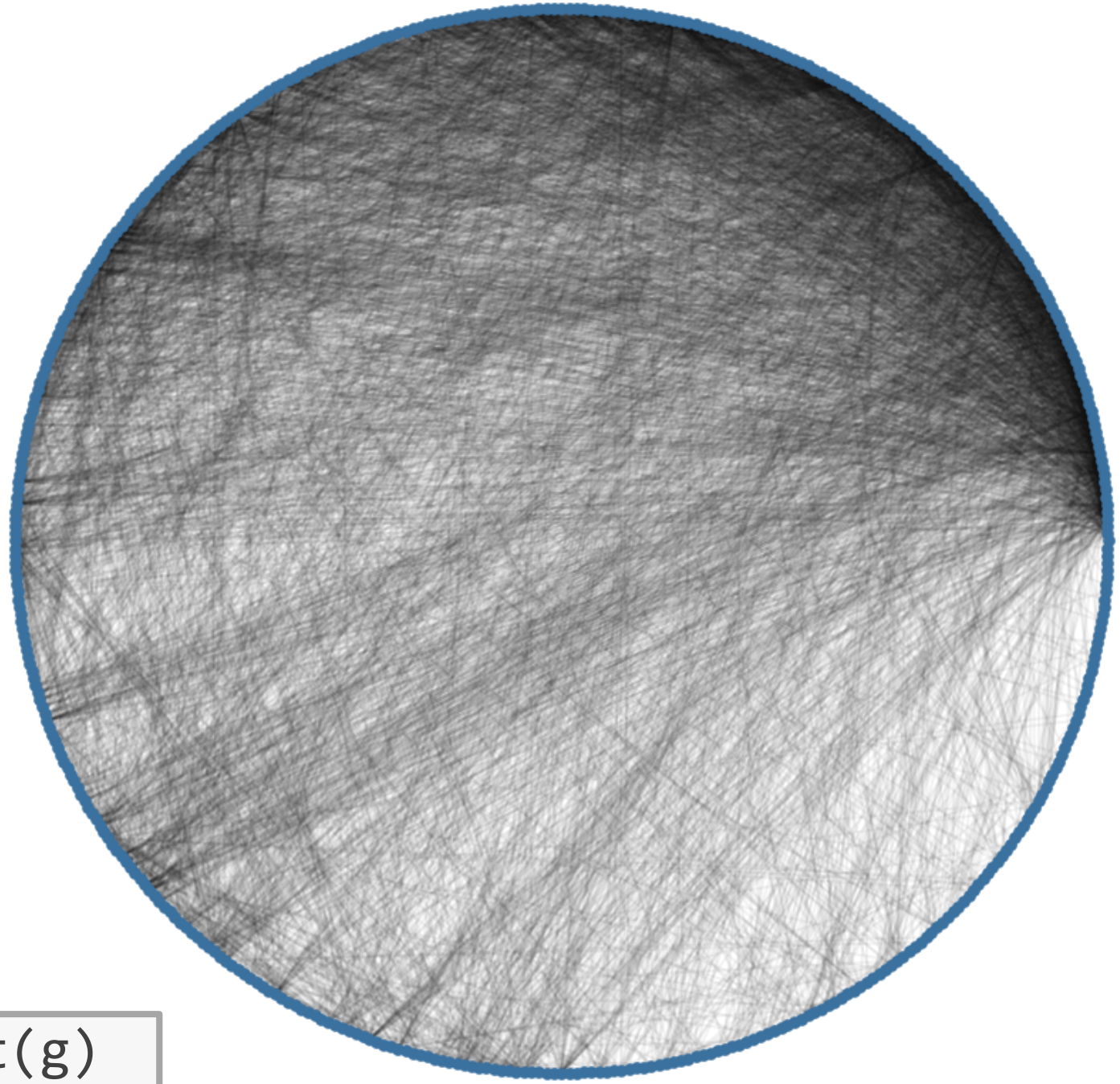
Nodes are placed randomly throughout the graph.



```
pos = nx.random_layout(g)
```

The Circular Layout

Nodes aligned on outside, all edges “contained”

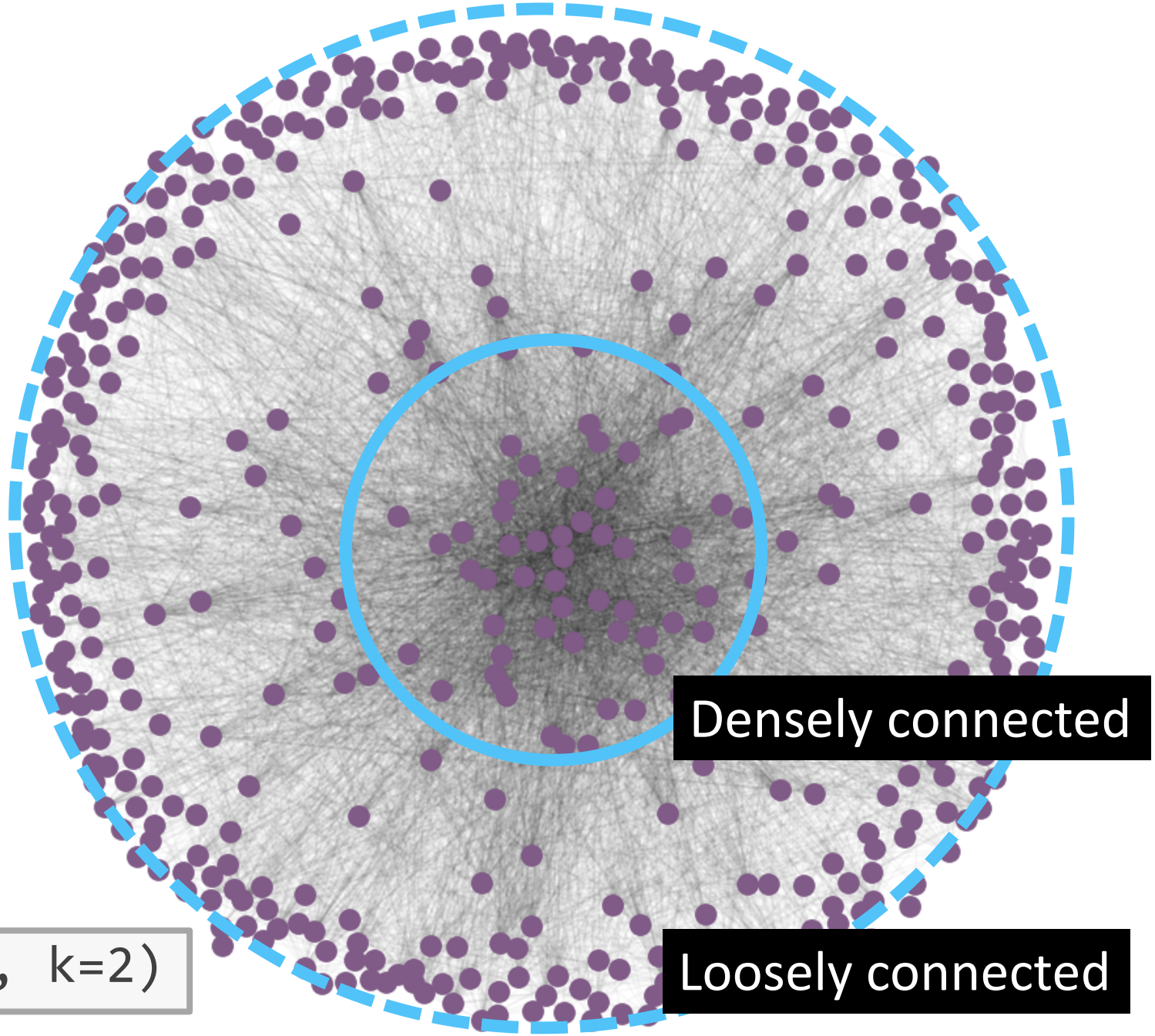


```
pos = nx.circular_layout(g)
```

The Spring Layout

Nodes ”spring” from the center, getting pushed further if they are less densely connected.

```
pos = nx.spring_layout(g, k=2)
```



Plotting with NetworkX and Matplotlib

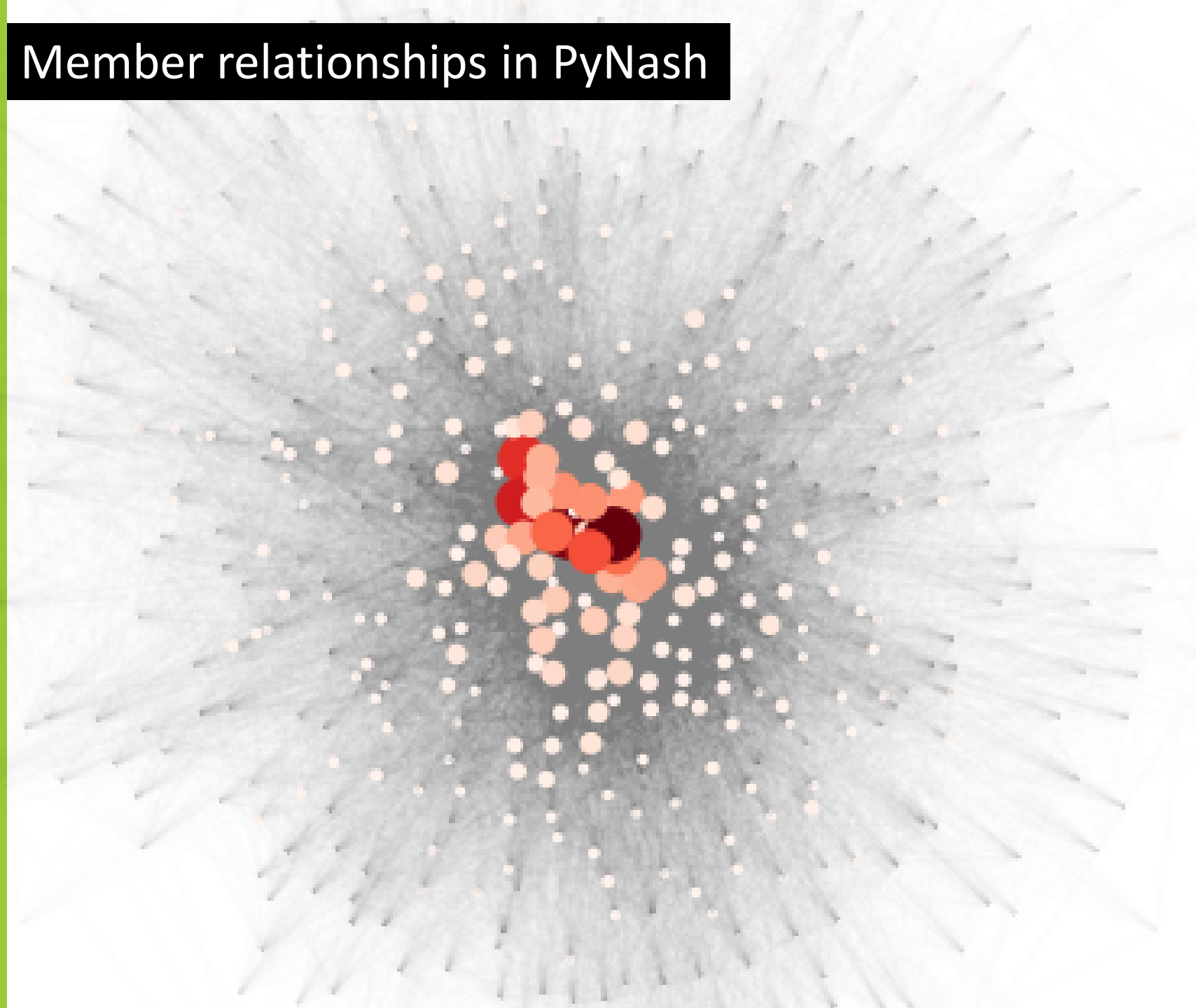
Can break drawing step into stages for more precise handling.

```
nx.draw_networkx_nodes
nx.draw_networkx_edges
nx.draw_networkx_labels
```

```
# Common Plot Attributes


with_labels=False
node_color=None
node_size=100
width=1   # edge width
alpha=1   # transparency
```

The Spring Layout: Emphasizing centrality

More central nodes are brighter red and larger.

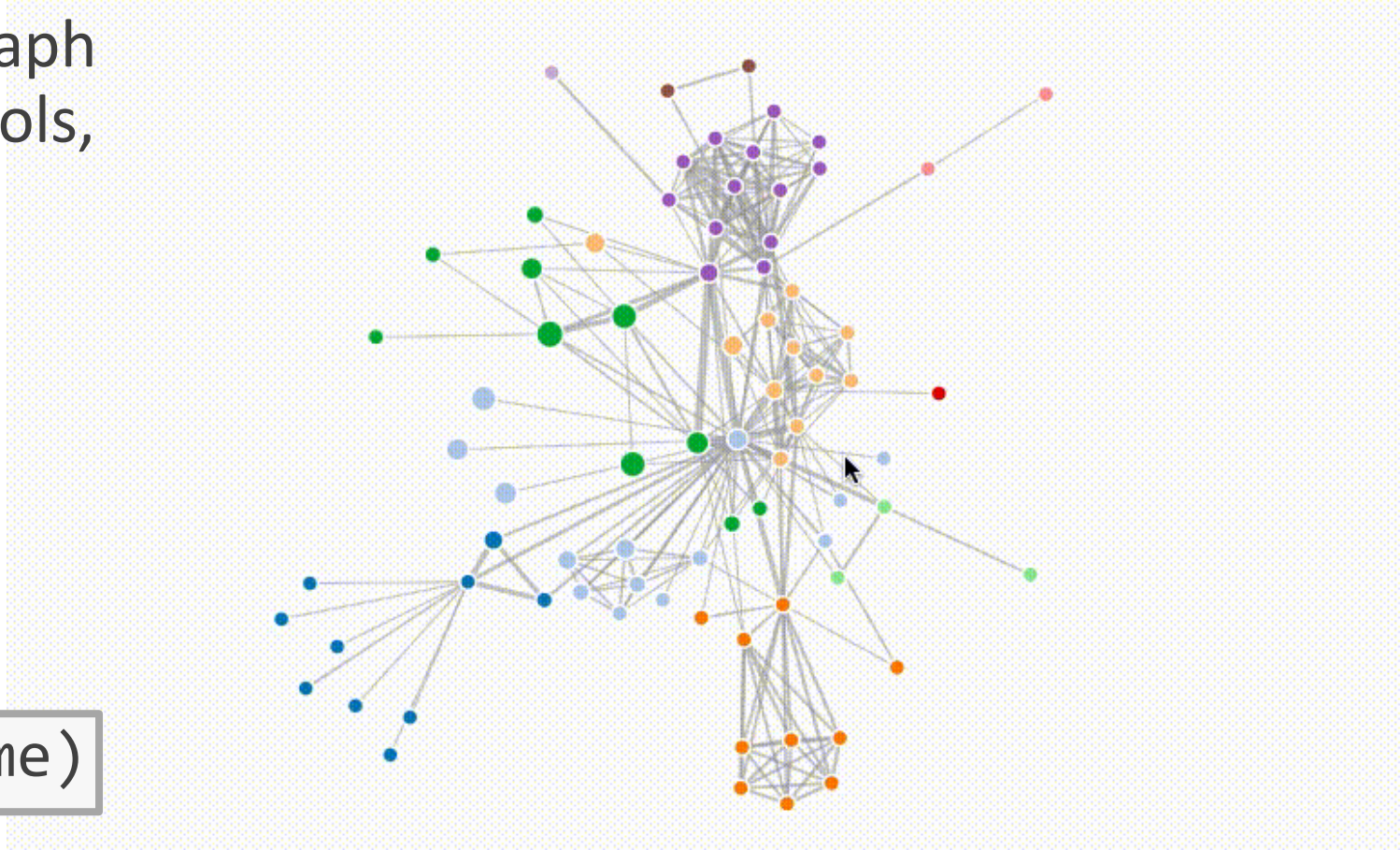


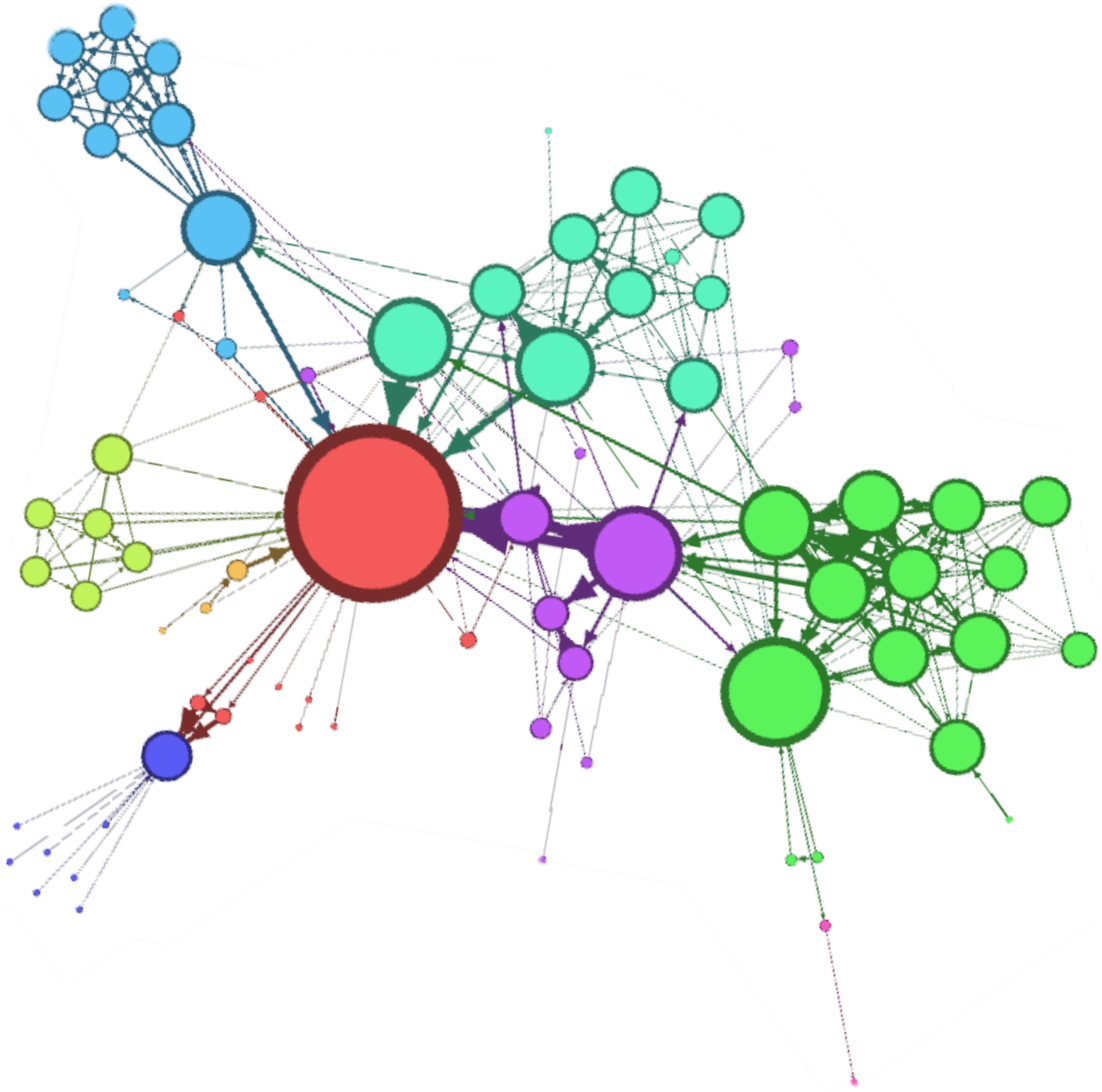
Exporting graph data

There are many other graph analysis packages and tools, especially for plotting

- d3.js
- Plotly
- GraphViz
- Neo4j

```
nx.write_pajek(g, fname)
```





Principles of Network Analysis with NetworkX

1. What is a network?
2. Build a network
3. Measure a network
4. Plot a network

Take-Aways

PRINCIPLES

Garbage in, garbage out

Spend time preprocessing edges BEFORE creating the graph

Measures

- Degree – first stop
- Clustering – “clumping”
- Centrality – “influence”

Use color & transparency for more comprehensible plots

NETWORKX

NetworkX is excellent for Pythonic exploration of data:

- I/O
- Analysis
- Visualization

For very large graphs, some operations can take a long time.