Week 7

Social Network Analysis

Dr Anagi Gamachchi

Discipline of Information Systems and Business Analytics,
Deakin Business School

Outline

- **Motivation for SNA**
- **Graph Theory**
- **Graph Measures**
- **SNA Case Study**

What is Social Network Analysis?

This dialogue occurs often in the life of any SNA researcher

"So…what do you do?"

"I'm a social network analysis researcher."

"Oh, so you play on Facebook and get paid for it?"

The answer is "Yes", but SNA is much more than Facebook.

- In fact, SNA is an extremely versatile methodology that predates Twitter and Facebook by at least 30 years
- SNA can be described as a "**study of relationships by means of graph theory**." consisting of *nodes* and *ties* (also called *edges*, *links* or *connections*).

Tsvetovat, M., & Kouznetsov, A. (2011). Social Network Analysis for Startups, O'Reilly Media.

Analysing Relationship

The science of SNA boils down to one central concept — the **relationships**

- E.g. In an interpersonal context, it can be *friendship*, *influence, affection*, *trust*—or conversely, *dislike*, *conflict*, or many other things.

Does social network exist without social media?

Social Media: interactive computer-mediated <u>technologies</u> that facilitate the creation or sharing of information, ideas, career interests and other forms of expression via virtual communities and networks

Social Network: <u>relationship</u> between actors.

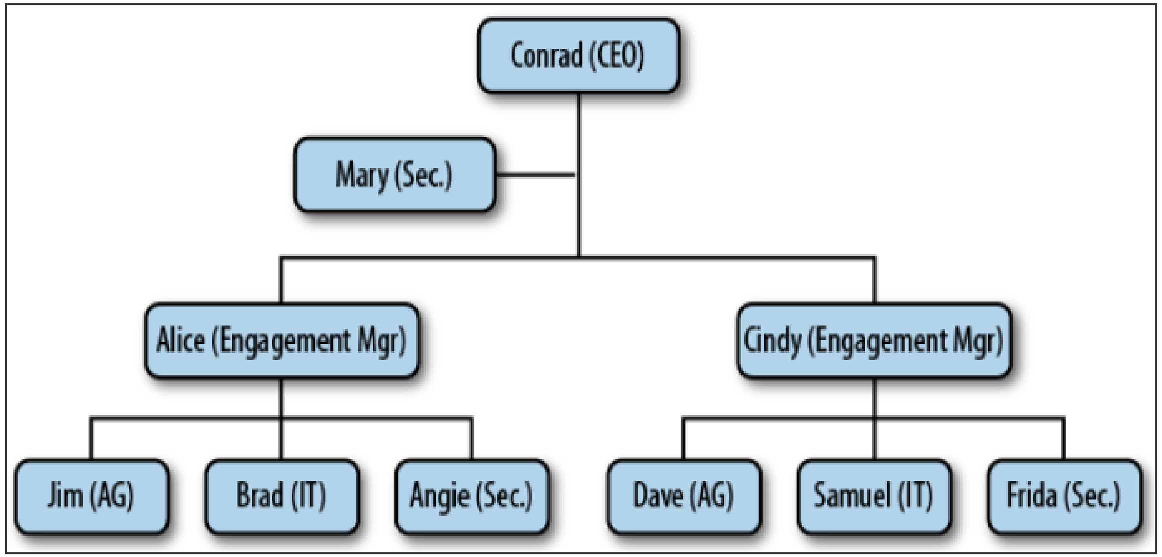
- **Formal SN**: relationships are <u>explicitly captured</u> and <u>presented</u> on various media platforms - social media sites (e.g. facebok, twitter, etc.), organizational charts,
- **Informal SN**: relationships that <u>exist but NOT explicitly captured</u> and <u>represented</u> on any media platform.

Tsvetovat, M., & Kouznetsov, A. (2011). Social Network Analysis for Startups, O'Reilly Media.

The Power of Informal Networks – Case study

ACME Consulting is a family-owned business in accounting and auditing. After 25 years of operation, ACME was restructured by a new CEO to meet new demand in the current time.

Organization Chart After Restructured



On the surface, the change was good:

- The project teams got to know clients;
- finance and IT people talked to each other
- routine problems got resolved quickly
- Customers were happier.

But **under the surface**, trouble was brewing:

- project teams spent more and more time “fire fighting”—fixing previous mistakes

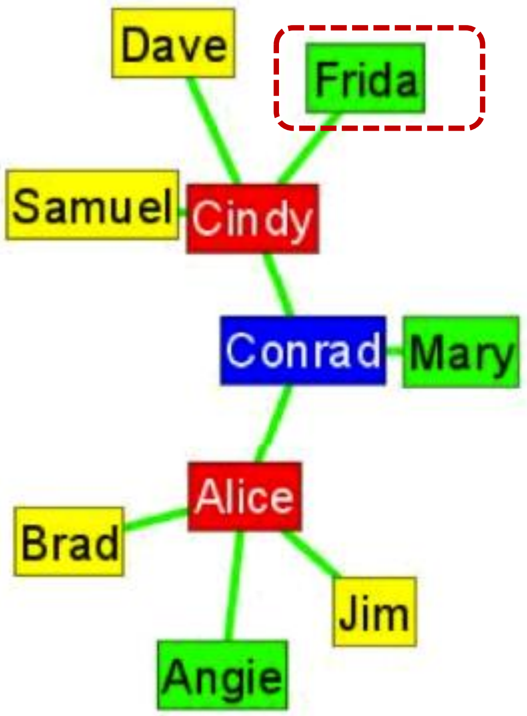
Tsvetovat, M., & Kouznetsov, A. (2011). Social Network Analysis for Startups, O’Reilly Media.

The Power of Informal Networks (cont.)

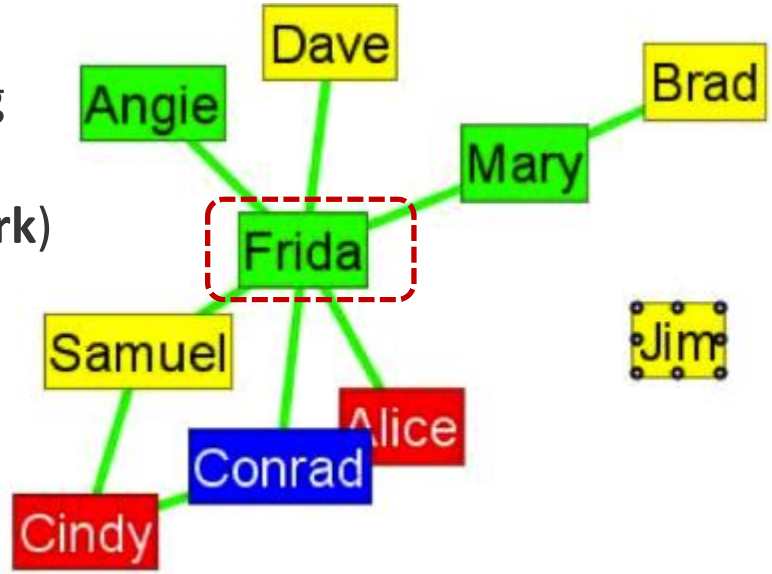
The CEO realized that many mistakes had to do with improper forms being filled with the government, misplaced documents, and other concerns that we would call "routine paperwork". A consultant was hired to help fixing the company 's problem.

SNA approach was adopted.

ACME Consulting org chart (**formal network**)



ACME Consulting org chart (**informal network**)



Tsvetovat, M., & Kouznetsov, A. (2011). Social Network Analysis for Startups, O'Reilly Media.

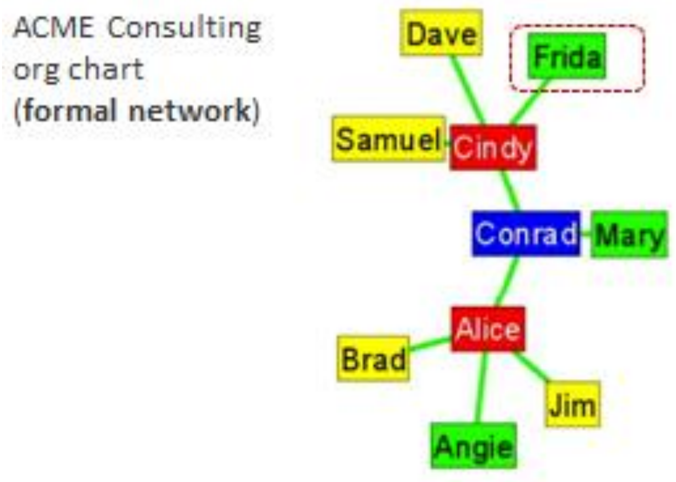
The Power of Informal Networks (cont.)

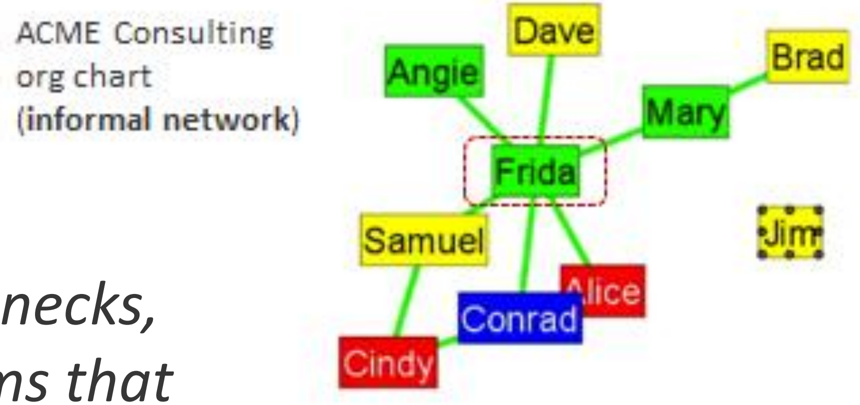
Identified Problem:

- <u>Frida</u> was one of the oldest employees of the firm, regarded as a <u>mentor</u> by <u>almost everyone else</u>.
- However, in Conrad's reorganization, she had been <u>moved to an offsite client team</u>, and her advice became unavailable on a day-to-day

Solution:

- Frida was <u>promoted and stayed at the headquarters</u> as a trusted advisor, training the younger employees, and the company rapidly recovered





***Lesson Learned:* Every organization has information bottlenecks, rumor mills, competing cliques and other potential problems that only *Social Network Analysis can detect*.**

Tsvetovat, M., & Kouznetsov, A. (2011). Social Network Analysis for Startups, O'Reilly Media.

Outline

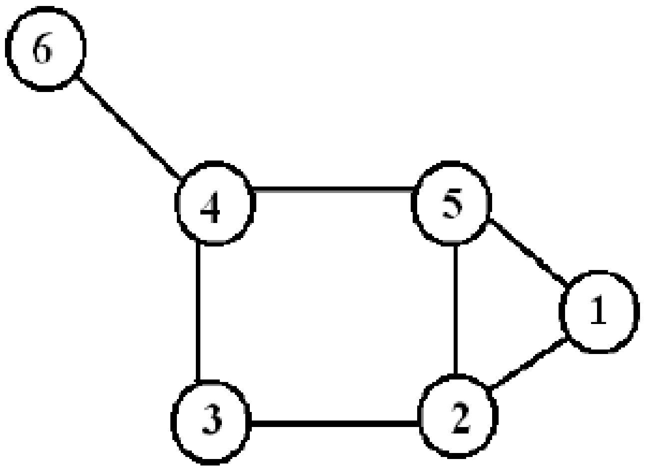
- Motivation for SNA
- **Graph Theory**
- Graph Measures
- SNA Case Study

Graph Definition

Mathematical foundation of SNA is based on **Graph Theory**

- **Vertex**:
  - Basic Element
  - Drawn as a *node* or a *dot*

- **Edge**:
  - A set of two elements
  - Drawn as a line connecting two vertices



- V:={1,2,3,4,5,6}
- E:={{1,2},{1,5},{2,3},{2,5},{3,4},{4,5},{4,6}}

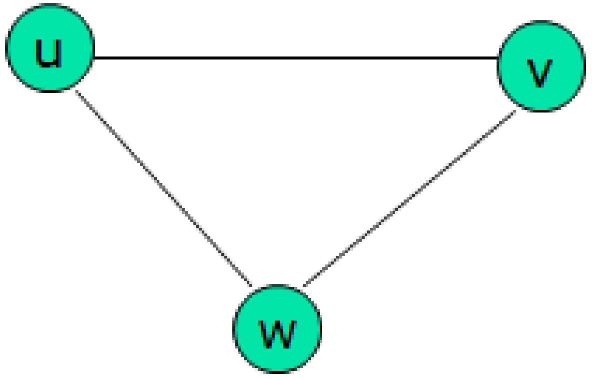
Graph Definition (cont.)

Two main types of Graphs:

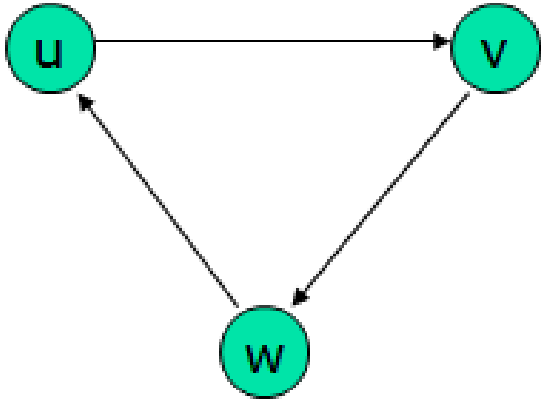
- **Undirected Graph**: Symmetric Relationship, e.g., Facebook friends and LinkedIn connections require mutual confirmation.

- **Directed Graph:** Asymmetric Relationship, e.g., Person A is the boss of Person B.

Undirected



Directed

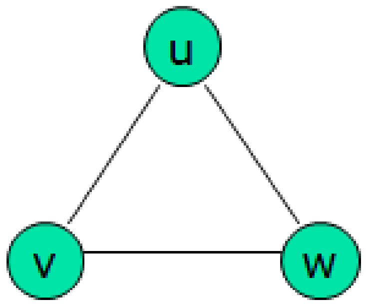


Newman, Mark, *Networks*, 2nd edn (Oxford, 2018; online edn, Oxford Academic, 18 Oct. 2018),

Representation- Adjacency Matrix

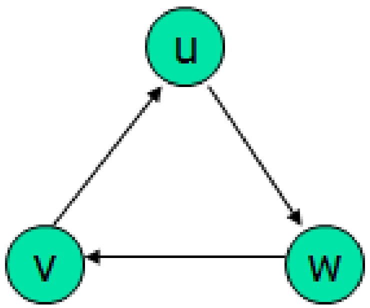
- The basic way to represent a social network mathematically is a **matrix**:

Undirected Graph G (V, E)



	v	u	w
v	0	1	1
u	1	0	1
w	1	1	0

Directed Graph G (V, E)



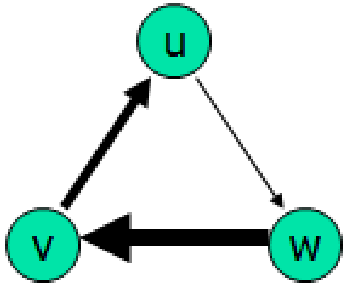
	v	u	w
v	0	1	0
u	0	0	1
w	1	0	0

A "1" in cell means that there's a relationship (edge) between nodes

Newman, Mark, *Networks*, 2nd edn (Oxford, 2018; online edn, Oxford Academic, 18 Oct. 2018),

Representation- Adjacency Matrix (cont.)

- In case of a valued graph:



	v	u	w
v	0	2	0
u	0	0	1
w	3	0	0

- The major downside of adjacency matrices is that zeros consume the same amount of memory as the other cells.
- In real social networks, there are a lot of zero cells—in fact, over 90% of cells

from	to	value
v	**u**	2
u	**w**	1
w	**v**	3

The solution is to represent networks as **edge-lists**

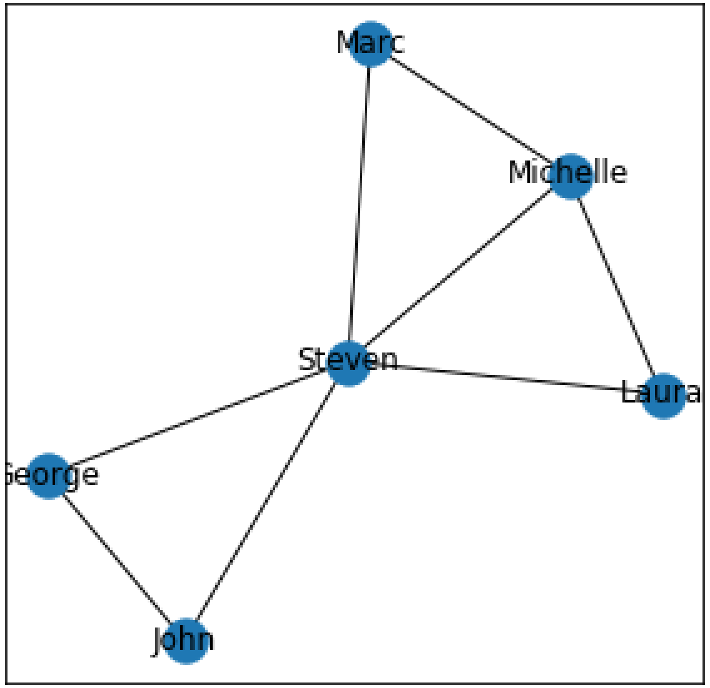
Generating Graphs in Python

- **Symmetric network** of people "working together"

```
import networkx as nx
import matplotlib.pyplot as plt

G_symmetric = nx.Graph()

G_symmetric.add_edge('Steven',  'Laura')
G_symmetric.add_edge('Steven',  'Marc')
G_symmetric.add_edge('Steven',  'John')
G_symmetric.add_edge('Steven',  'Michelle')
G_symmetric.add_edge('Laura',   'Michelle')
G_symmetric.add_edge('Michelle','Marc')
G_symmetric.add_edge('George',  'John')
G_symmetric.add_edge('George',  'Steven')
print(nx.info(G_symmetric))
```

```
Name:
Type: Graph
Number of nodes: 6
Number of edges: 8
```

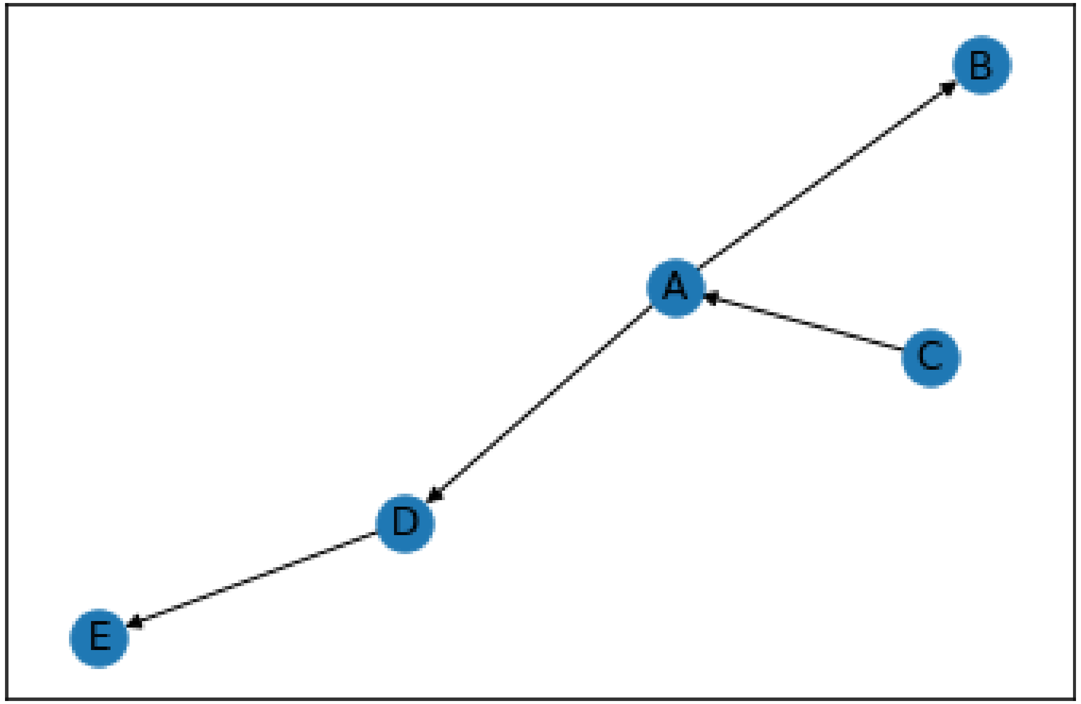
```
plt.figure(figsize=(5,5))
nx.draw_networkx(G_symmetric);
```



Generating Graphs in Python (cont.)

- **Asymmetric network** of entities

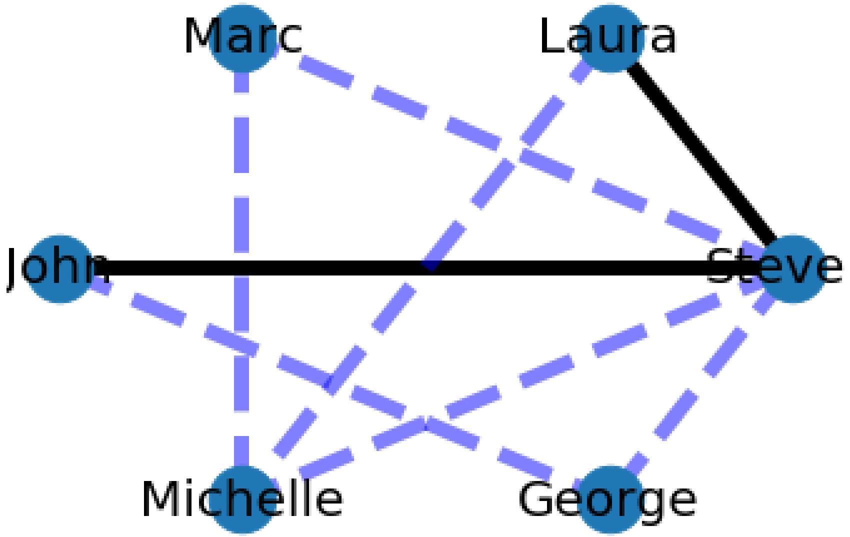
```
G_asymmetric = nx.DiGraph()
G_asymmetric.add_edge('A','B')
G_asymmetric.add_edge('A','D')
G_asymmetric.add_edge('C','A')
G_asymmetric.add_edge('D','E')
nx.spring_layout(G_asymmetric)
nx.draw_networkx(G_asymmetric)
```



Generating Graphs in Python (cont.)

- **Weighted network**

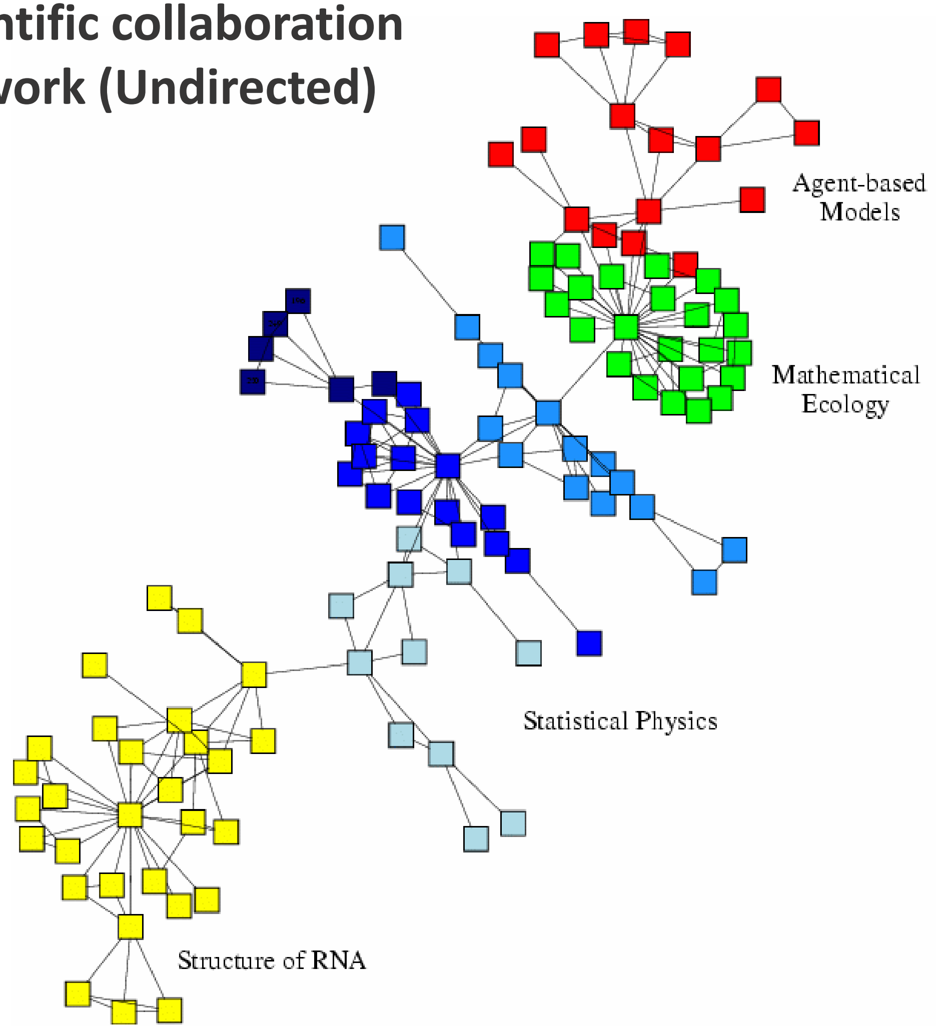
```
G_weighted = nx.Graph()

#Weights are provided to edges
G_weighted.add_edge('Steven',  'Laura',   weight=25)
G_weighted.add_edge('Steven',  'Marc',    weight=8)
G_weighted.add_edge('Steven',  'John',    weight=11)
G_weighted.add_edge('Steven',  'Michelle',weight=1)
G_weighted.add_edge('Laura',   'Michelle',weight=1)
G_weighted.add_edge('Michelle','Marc',    weight=1)
G_weighted.add_edge('George',  'John',    weight=8)
G_weighted.add_edge('George',  'Steven',  weight=4)
```



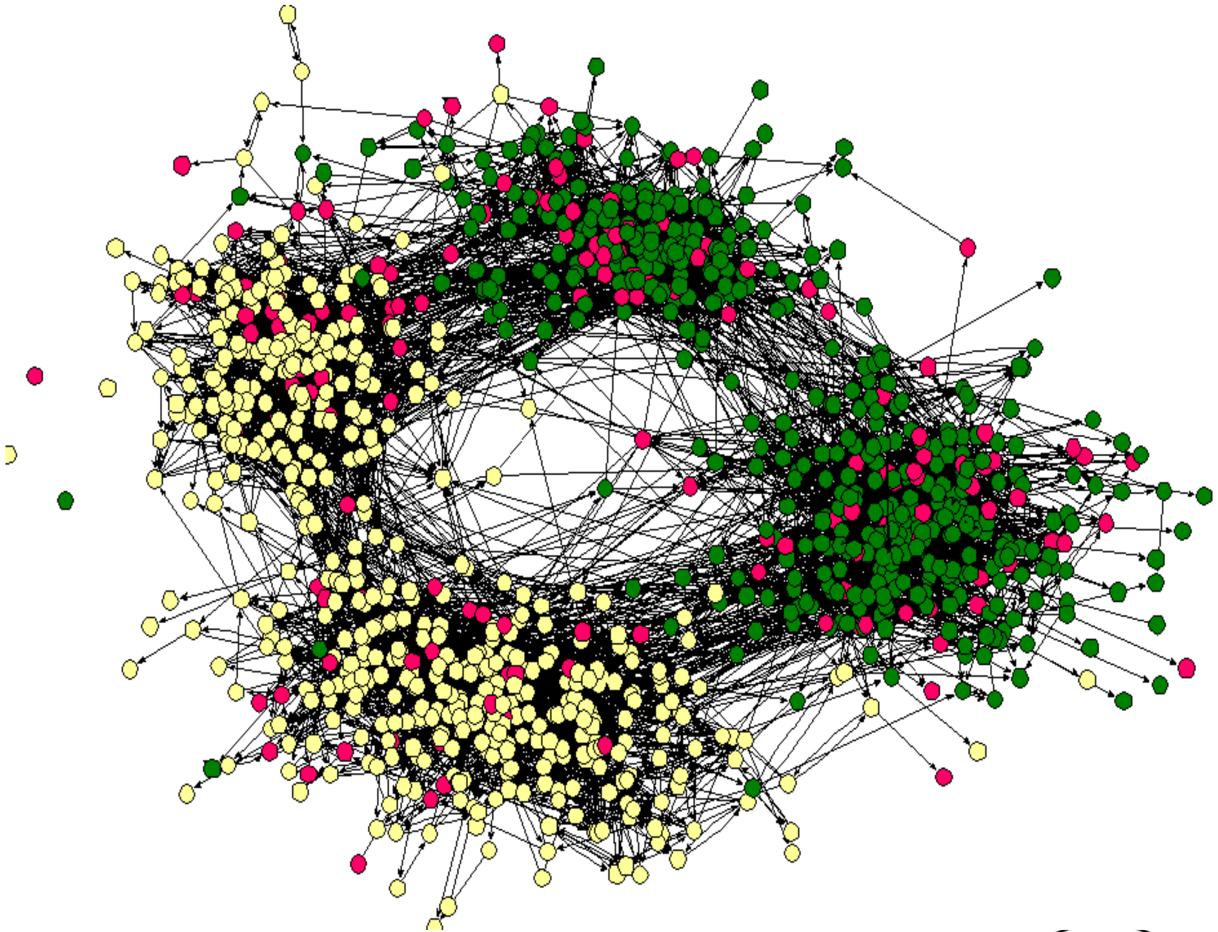
The weights indicate how strong/close the relationships are.

Real-Life Examples

Scientific collaboration network (Undirected)

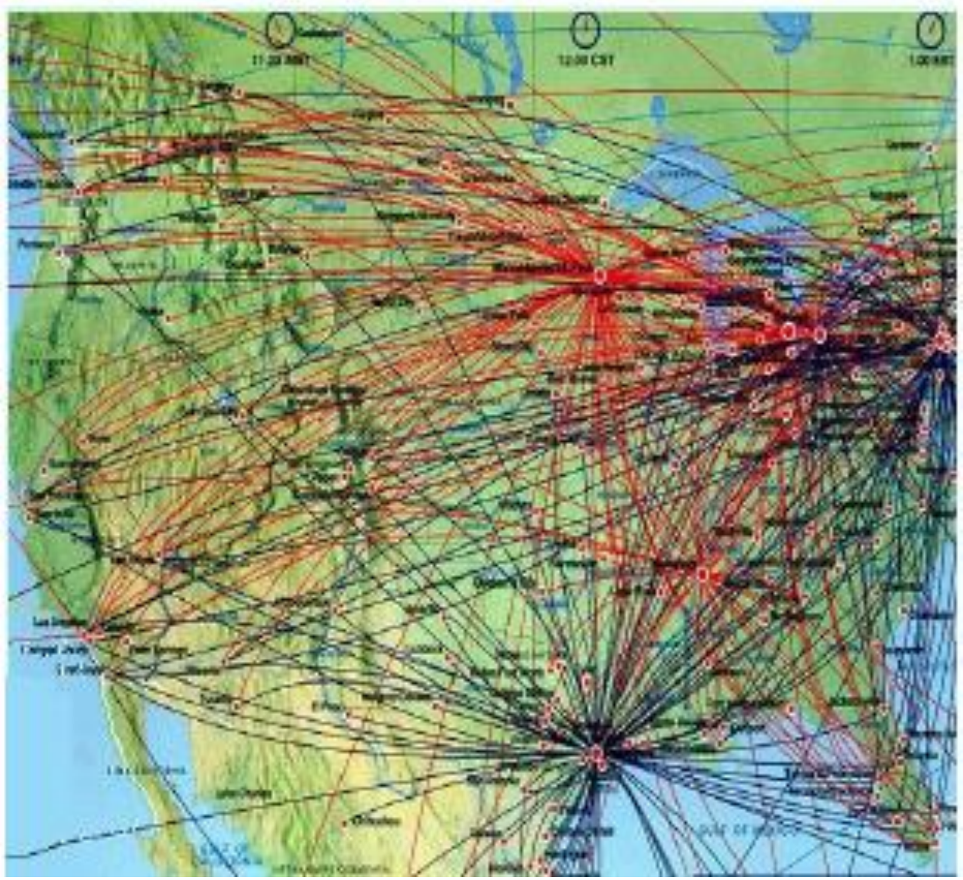


Friendship Network (Directed)



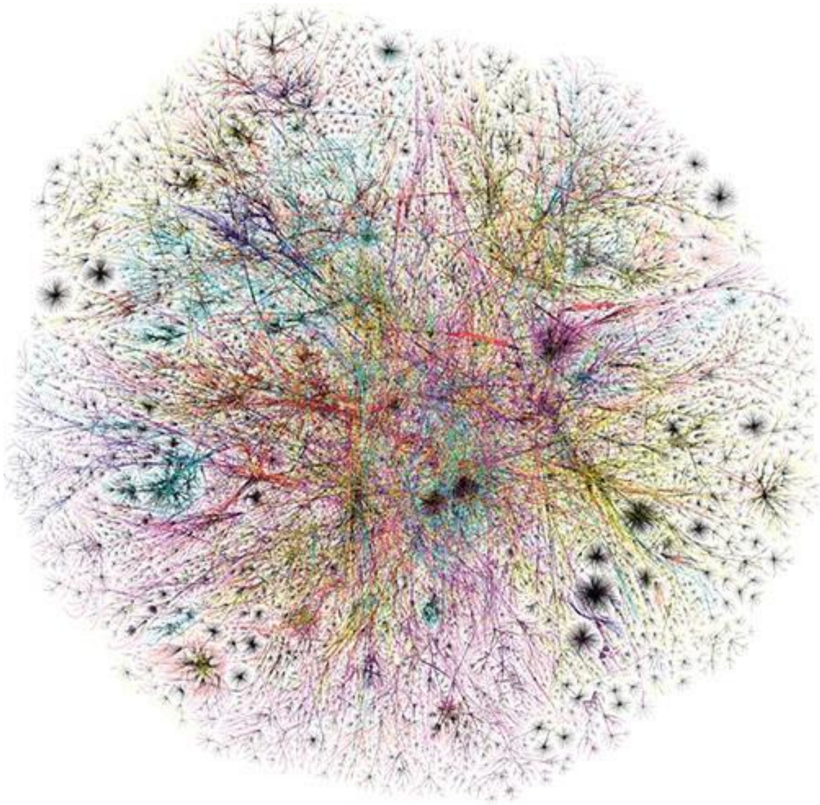
Real-Life Examples

Transportation Network



Yang Mu CS105 Introduction to Social Network UMass Boston

The Internet



Newman, Mark, *Networks*, 2nd edn (Oxford, 2018; online edn, Oxford Academic, 18 Oct. 2018),

Outline

- Motivation for SNA
- Graph Theory
- **Graph Measures**
- SNA Case Study

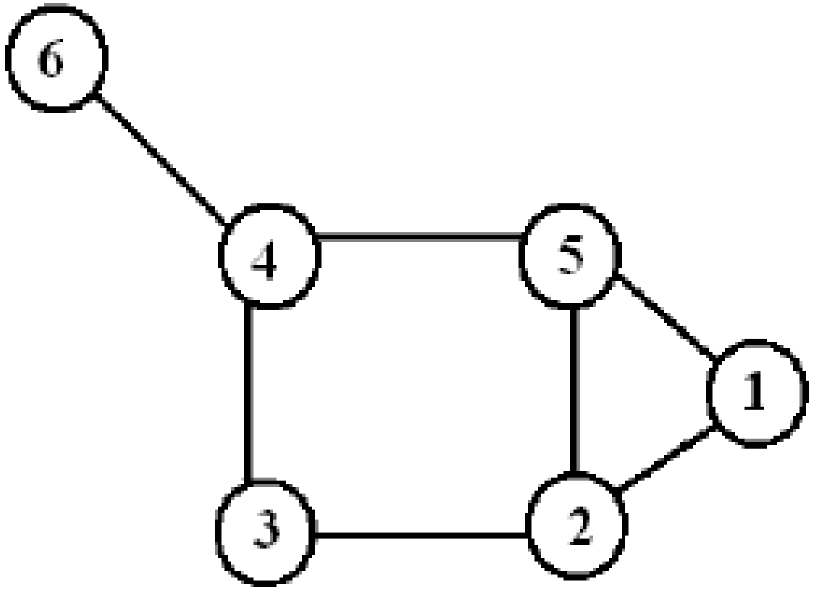
Graph Measures

- Graph characteristics can be evaluated quantitatively using various metrics:
  - **Topological Distance**
  - **Diameter of graph**

- Some measures are used to evaluate node characteristics.
  - **Centrality Analysis** (Degree, Betweeness, Closeness**)**

Topological Distance

- The shortest path between two nodes in network



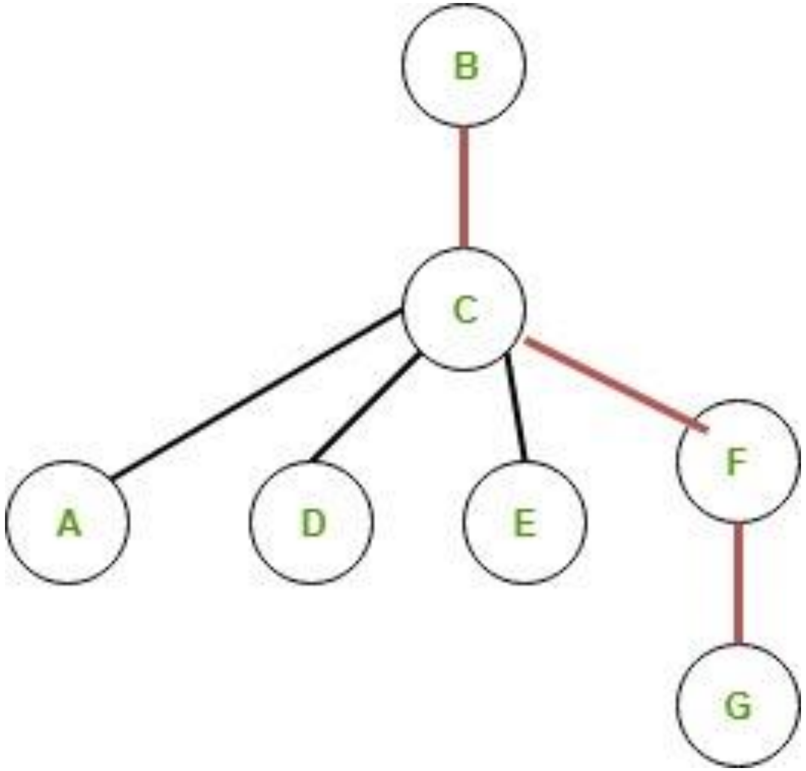
Distance Matrix

$$
\begin{array}{c c}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} &
\begin{pmatrix}
0 & 1 & 2 & 2 & 1 & 3 \\
1 & 0 & 1 & 2 & 1 & 3 \\
2 & 1 & 0 & 1 & 2 & 2 \\
2 & 2 & 1 & 0 & 1 & 1 \\
1 & 1 & 2 & 1 & 0 & 2 \\
3 & 3 & 2 & 1 & 2 & 0
\end{pmatrix}
\end{array}
$$

- the (shortest) distance of a node from every other node in the network using **breadth-first search algorithm** (implemented in python)

Diameter of graph

- defined as the maximal distance between the pair of vertices. Way to solve it is to find all the paths and then find the maximum of all



Diameter: **3**

BC → CF → FG

Graph Theory

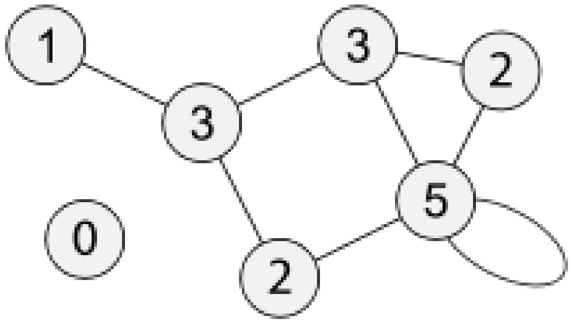
Centrality Analysis

- In graph theory and network analysis, indicators of **centrality** identify the most important vertices within a graph.
- Applications include:
  - identifying the most influential person(s) in a social network,
  - key infrastructure nodes on the Internet or urban networks,
  - super-spreaders of disease.
- Popular Types of centralities:
  - **Degree Centrality**
  - **Betweenness Centrality**
  - **Closeness Centrality**

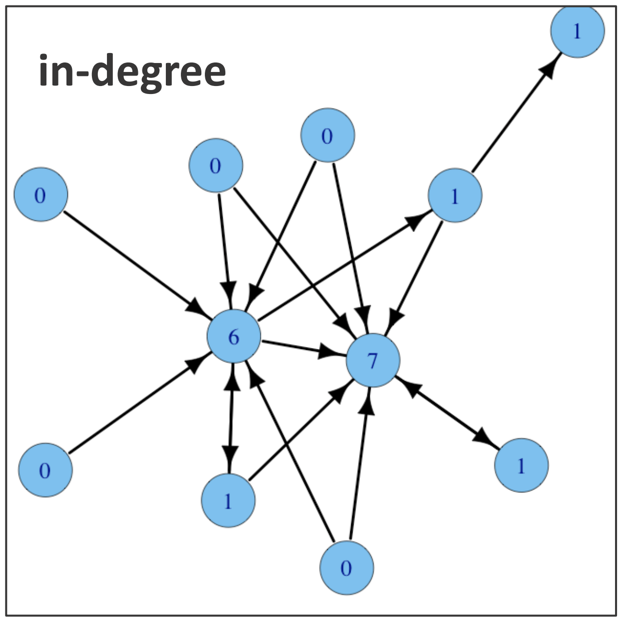
Newman, Mark, 'Measures and metrics', *Networks*, 2nd edn (Oxford, 2018; online edn, Oxford Academic, 18 Oct. 2018)

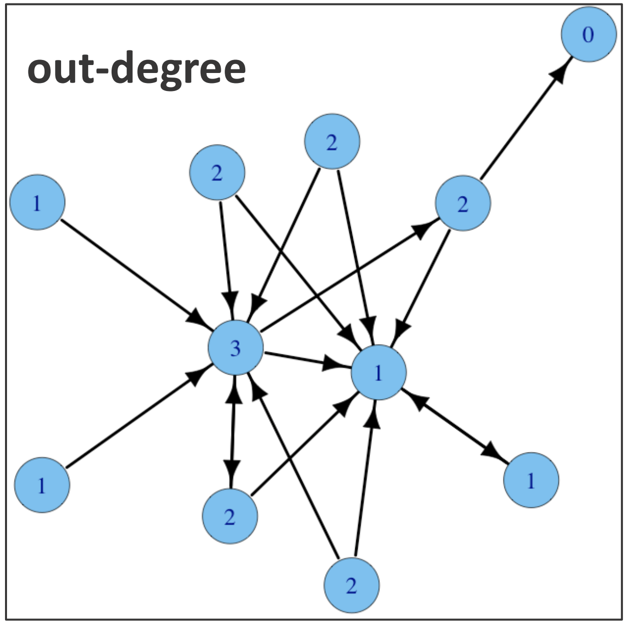
Degree Centrality

- **Degree Centrality** of a node defines the <u>number of connections</u> a node has
- If the network is <u>directed</u>, we have two versions of the measure:
  - **in-degree** is the number of in-coming links, or the number of predecessor nodes;
  - **out-degree** is the number of out-going links, or the number of successor nodes.



Note the value difference in the nodes



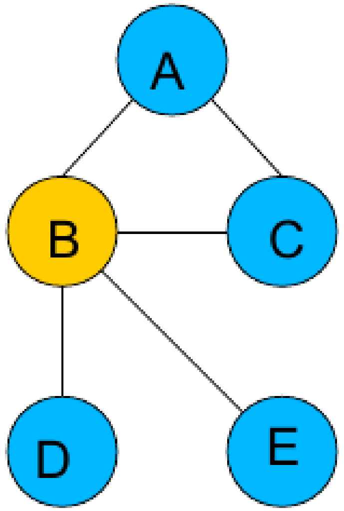


The degree can be interpreted in terms of the immediate opportunity/risk of a node for catching whatever is flowing through the network (such as some information or virus).

Newman, Mark, 'Measures and metrics', *Networks*, 2nd edn (Oxford, 2018; online edn, Oxford Academic, 18 Oct. 2018)

Betweenness centrality

- Betweenness is measured by the number of shortest paths in the graph that pass through the node divided by the total number of shortest path



Shortest paths are:
AB, AC, ABD, ABE, BC, BD BE, CBD, CBE, DBE

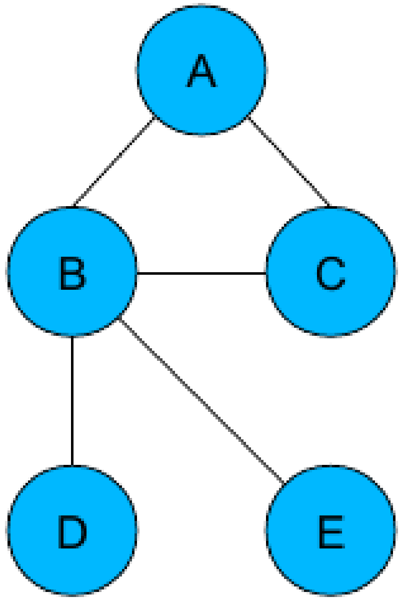
B has a betweenness Centrality score of : 9/10 = **0.9**

- Nodes with a high betweenness centrality are interesting because they:
  - Control information flow in a network
  - May be required to carry more information

Nabati, R. Introduction to Network And SNA Theory: Basic Concepts, University of Strathclyde, Glasgow

Closeness centrality

- Closeness is measured by the normalized inverse of the sum of topological distance in the graph counting from the current node **(Number of vertex – 1)/**



	A	B	C	D	E	Total Distance		Closeness
A	0	1	1	2	2	6	(5-1)/6 =	0.67
B	1	0	1	1	1	4		1.00
C	1	1	0	2	2	6		0.67
D	2	1	2	0	2	7		0.57
E	2	1	2	2	0	7		0.57

- **Node B** is the most central one in spreading information from it to the other nodes in the network

Nabati, R. Introduction to Network And SNA Theory: Basic Concepts, University of Strathclyde, Glasgow

Outline

- Motivation for SNA
- Graph Theory
- Graph Measures
- **SNA Case Study**

Society of Friends Case Study

- Before there were Facebook friends, there was the Society of Friends, known as the **Quakers**, founded in England in the mid-seventeenth century. The data used in this case study is a list of names and relationships among the earliest seventeenth-century Quakers.

Node name — Node Attributes

quakers_nodelist.csv

1	Name	Historical Significance	Gender	Birthdate	Deathdate	ID
2	Joseph Wyeth	religious writer	male	1663	1731	10013191
3	Alexander Skene of Newtyle	local politician and author	male	1621	1694	10011149
4	James Logan	colonial official and scholar	male	1674	1751	10007567
5	Dorcas Erbery	Quaker preacher	female	1656	1659	10003983
6	Lilias Skene	Quaker preacher and poet	male	1626	1697	10011152
7	William Mucklow	religious writer	male	1630	1713	10008595
8	Thomas Salthouse	Quaker preacher and writer	male	1630	1691	10010643

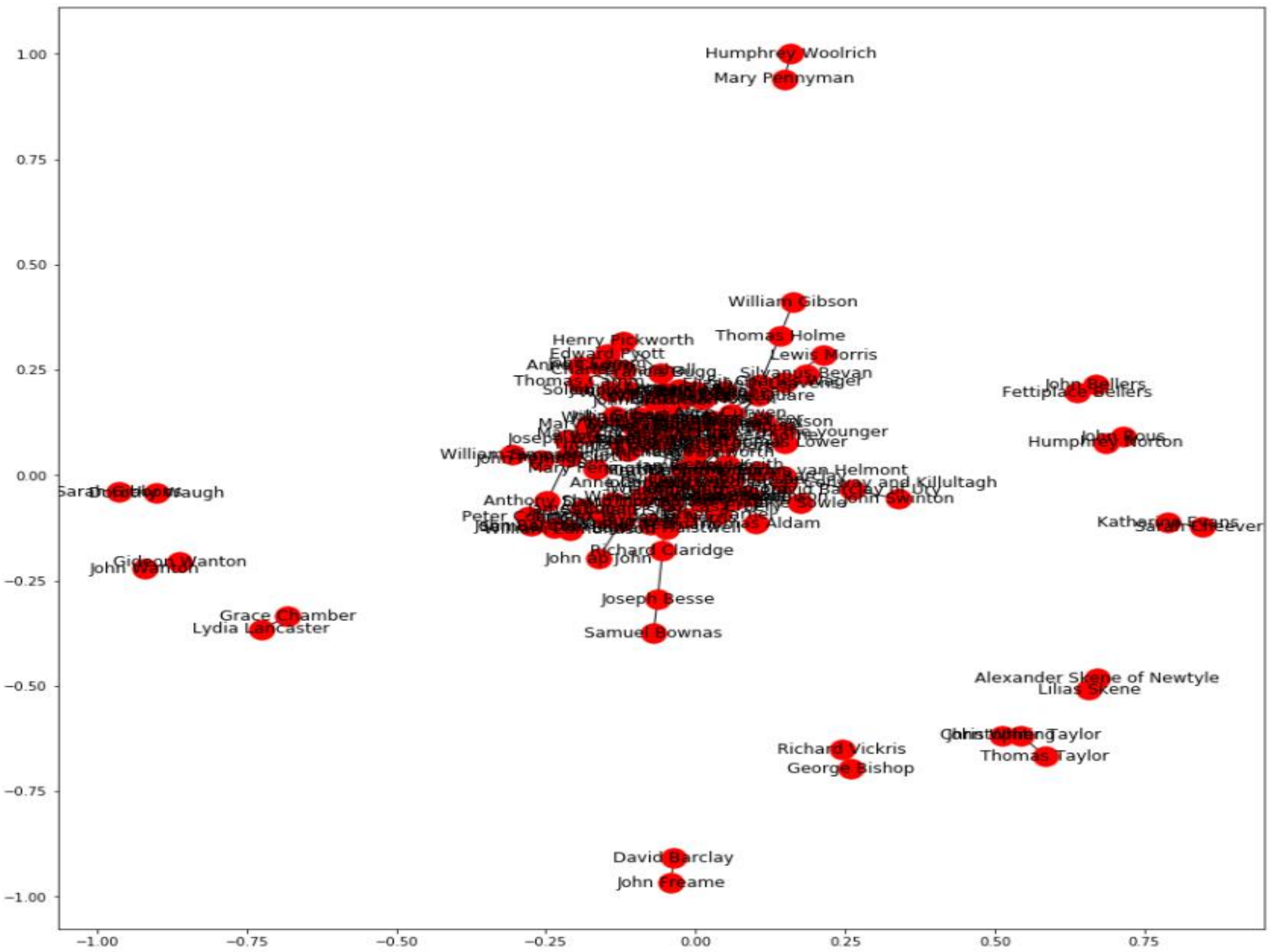
quakers_edgelist.csv

1	Source	Target
2	George Keith	Robert Barclay
3	George Keith	Benjamin Furly
4	George Keith	Anne Conway Viscountess Conway and Killultagh
5	George Keith	Franciscus Mercurius van Helmont
6	George Keith	William Penn
7	George Keith	George Fox
8	George Keith	George Whitehead
9	George Keith	William Bradford

Problem:
Identify the most important/influential people in this network.

Solution:
SNA

Network Density Analysis



Network Visualization

```
plt.figure(figsize=(15,15))
nx.draw_networkx(G);
```

Density Analysis

```
density = nx.density(G)
print("Network density:", round(density,3))
```

```
Network density: 0.025
```

The network has relatively low **density**, suggesting loose associations

Degree Centrality Analysis

Calculating **degree centrality** and adding it as an attribute to the network.

```
degree_dict = dict(G.degree(G.nodes()))
nx.set_node_attributes(G, degree_dict, 'degree')
#Access all attribute for one node
#Note the "degree" attribute is available
print(G.node['William Penn'],'\n')
```

```
{'historical_significance': 'Quaker leader and founder of Pennsylvania', 'gender': 'male', 'birth_year': '1644', 'death_year': '1718', 'sdfb_id': '10009531', 'degree': 18}
```

The community is organized around several disproportionately large **hubs**, among them founders of the denomination like **Margaret Fell** and **George Fox**, as well as important political and religious leaders like **William Penn**

Sort and Display top 20 nodes by degree

```
sorted_degree = sorted(degree_dict.items(),
                       key=itemgetter(1), reverse=True)
print("Top 20 nodes by degree:")
for d in sorted_degree[:20]:
    print(d)
```

```
Top 20 nodes by degree:
('George Fox', 22)
('William Penn', 18)
('James Nayler', 16)
('George Whitehead', 13)
('Margaret Fell', 13)
('Benjamin Furly', 10)
('Edward Burrough', 9)
('George Keith', 8)
('Thomas Ellwood', 8)
('Francis Howgill', 7)
```

Betweenness Centrality Analysis

Calculate **betweenness centrality** for all nodes.

```python
# Calculate betweenness centrality
betweenness_dict = nx.betweenness_centrality(G)

# Assign to attributes in your network
nx.set_node_attributes(G, betweenness_dict, 'betweenness')

#You can sort betweenness  centrality
sorted_betweenness = sorted(betweenness_dict.items(),
                            key=itemgetter(1), reverse=True)

print("Top 20 nodes by betweenness centrality:")
for b in sorted_betweenness[:20]:
    print(b[0],":",round(b[1],3))
```

```
Top 20 nodes by betweenness centrality:
William Penn : 0.24
George Fox : 0.237
George Whitehead : 0.126
Margaret Fell : 0.121
James Nayler : 0.104
Benjamin Furly : 0.064
Thomas Ellwood : 0.046
George Keith : 0.045
John Audland : 0.042
Alexander Parker : 0.039
John Story : 0.029
John Burnyeat : 0.029
John Perrot : 0.028
James Logan : 0.027
```

William Penn and **George Fox** are among those with high betweenness, who are at the important position to receive and influence information in the network

Analysis with Combined Measures

*Which of the high **betweenness** centrality nodes had low **degree**?*

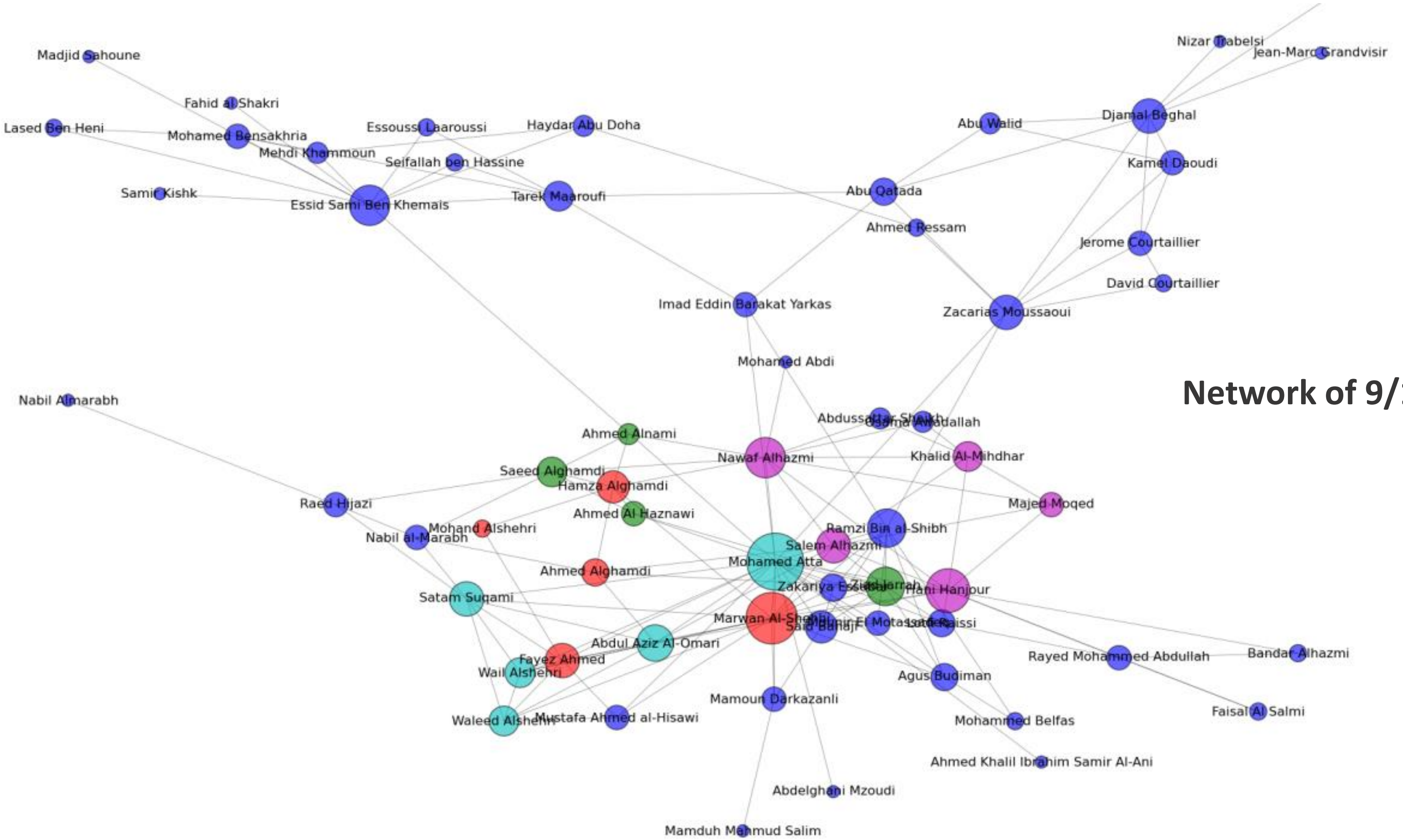
```
#First get the top 20 nodes by betweenness as a list
top_betweenness = sorted_betweenness[:20]

#Then find and print their degree
for tb in top_betweenness: # Loop through top_betweenness
    degree = degree_dict[tb[0]] # Use degree_dict to access a node's degree, see footnote 2
    print("Name:", tb[0], "| Betweenness Centrality:", round(tb[1],2), "| Degree:", degree)
```

```
Name: William Penn | Betweenness Centrality: 0.24 | Degree: 18
Name: George Fox | Betweenness Centrality: 0.24 | Degree: 22
Name: George Whitehead | Betweenness Centrality: 0.13 | Degree: 13
Name: Margaret Fell | Betweenness Centrality: 0.12 | Degree: 13
Name: James Nayler | Betweenness Centrality: 0.1 | Degree: 16
Name: Benjamin Furly | Betweenness Centrality: 0.06 | Degree: 10
Name: Thomas Ellwood | Betweenness Centrality: 0.05 | Degree: 8
Name: George Keith | Betweenness Centrality: 0.05 | Degree: 8
Name: John Audland | Betweenness Centrality: 0.04 | Degree: 6
Name: Alexander Parker | Betweenness Centrality: 0.04 | Degree: 6
Name: John Story | Betweenness Centrality: 0.03 | Degree: 6
Name: John Burnyeat | Betweenness Centrality: 0.03 | Degree: 4
Name: John Perrot | Betweenness Centrality: 0.03 | Degree: 7
Name: James Logan | Betweenness Centrality: 0.03 | Degree: 4
Name: Richard Claridge | Betweenness Centrality: 0.03 | Degree: 2
Name: Robert Barclay | Betweenness Centrality: 0.03 | Degree: 3
Name: Elizabeth Leavens | Betweenness Centrality: 0.03 | Degree: 2
Name: Thomas Curtis | Betweenness Centrality: 0.03 | Degree: 5
Name: John Stubbs | Betweenness Centrality: 0.02 | Degree: 5
Name: Mary Penington | Betweenness Centrality: 0.02 | Degree: 4
```

Members with **high betweenness centrality** but relatively **low degree centrality**, like Elizabeth Leavens and Mary Penington, who may have acted as **brokers**, connecting multiple groups

Other Applications of SNA

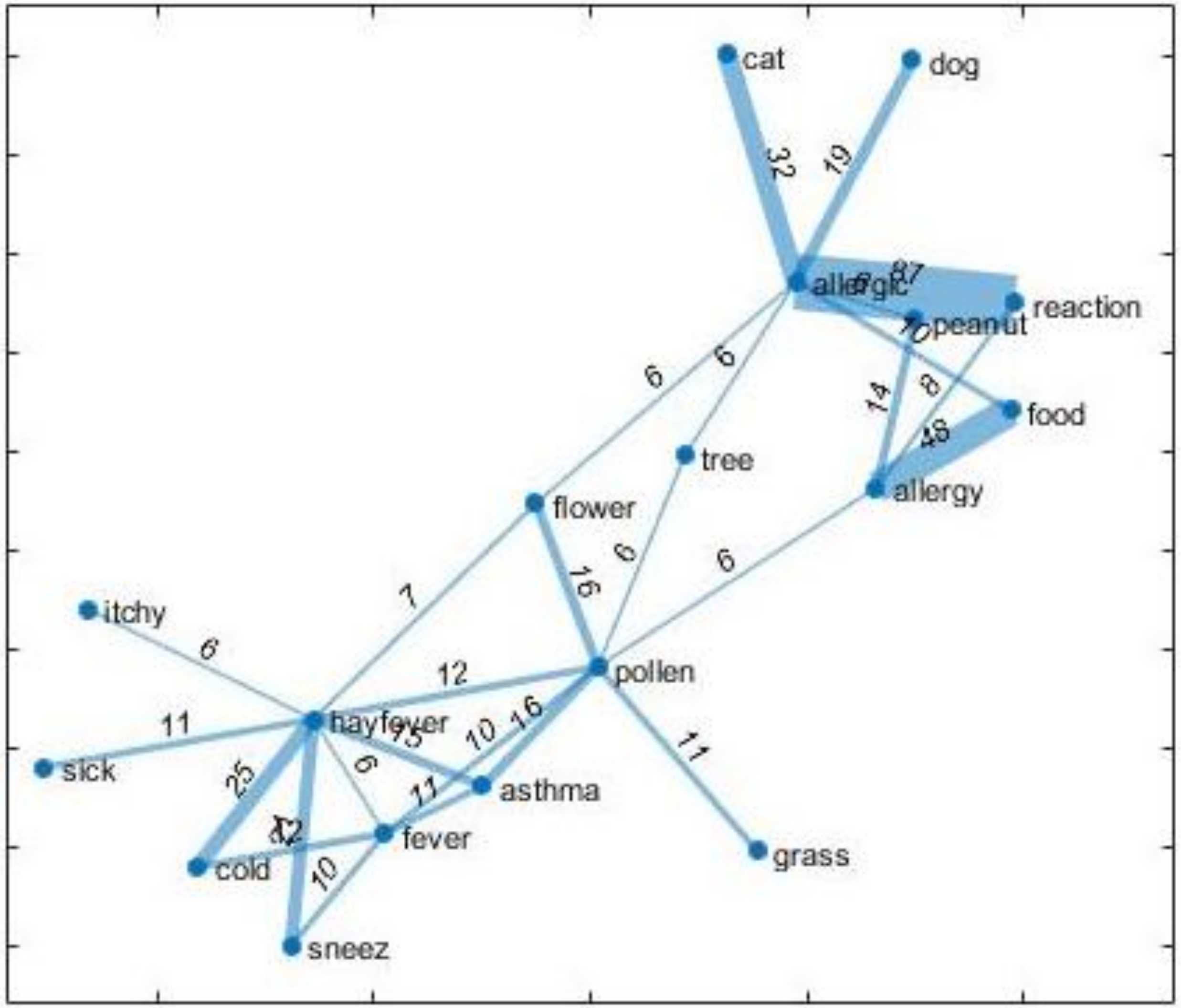
SNA in terrorism forensics



Network of 9/11 hijackers

Tsvetovat, M., & Kouznetsov, A. (2011). Social Network Analysis for Startups, O'Reilly Media.

Other Applications of SNA



SNA in health analytics

Analysis of **Allergic Conditions and Causes** based on Twitter Data

Nodes: Words
Edges: Co-occurrence in Sentences

SJ Miah, HQ Vu (2020). Towards developing a Healthcare Situation Monitoring Method for Smart City Initiatives: A Citizen Safety Perspective. *Australasian Journal of Information Systems,* https://journal.acs.org.au/index.php/ajis/article/view/2551

In this lecture, we have covered:

- Introduction to the concepts of social network analysis
- Foundational frameworks of SNA based on Graph Theory.
- Quantitative Measures of Graph
- Case study of important nodes identification.

Summary