



Getting an Edge with Network Analysis in Python

A Gentle Introduction

Alon Nir
(@alonnir)

PyCon UK 2018
Cardiff City Hall

My Objectives for Today

- Show you how useful network analysis can be
- Show you how easy it is to get started

Our Plan For Today

What we'll do:

1. Brief theoretical intro
2. Analyse a real network with Python
3. Explore useful and interesting examples

Our Plan For Today

What we won't do:

- In the interest of time we'll touch just a few of the many (many!) things there's to learn, just enough to get you started.
- No mathematical formulations! No greek letters!
- Dive deep into theory (Results are often intuitive but the algorithms to get them are not.)

Links to (excellent!) further resources will be provided.

What Is a Network Anyway?

And why should we care?



Why Should We Care?

Networks are everywhere, and they affect our lives daily.

As we'll see later, some examples include:

- Finding a job
- Finding a spouse
- Finding anything on Google
- Detecting bot/troll networks on Twitter
- Deciding which banks to bail out during a financial crisis
- Personalised recommendations
- Surviving a zombie outbreak

What Wikipedia Has to Say

In one very common sense of the term, a graph is an ordered pair $G = (V, E)$ comprising a set V of vertices, nodes or points together with a set E of edges, arcs or lines, which are 2-element subsets of V (i.e., an edge is associated with two vertices, and the association takes the form of the unordered pair of the vertices). To avoid ambiguity, this type of graph may be described precisely as undirected and simple.

Other senses of graph stem from different conceptions of the edge set. In one more general conception, E is a set together with a relation of incidence that associates with each edge two vertices. In another generalized notion, E is a multiset of unordered pairs of (not necessarily distinct) vertices. Many authors call these types of object multigraphs or pseudographs.

Source:

[https://en.wikipedia.org/wiki/Graph_\(discrete_mathematics\)](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics))

KISS

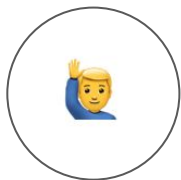
points and lines.

If you can draw a conceptual relationship between two objects, it's likely you can model the objects as a graph.

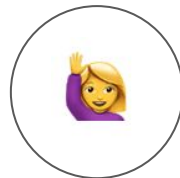
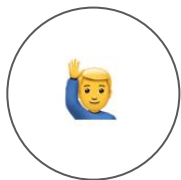
Points and Lines

Points	Lines
People	Friendships, following
Companies	Trade, acquire
Bus stops	Buses, passengers
Ubers/Lyfts/Ofo bikes	Passengers/riders
Countries	Trade, flight routes, Eurovision points
Wikipedia articles	Internal links

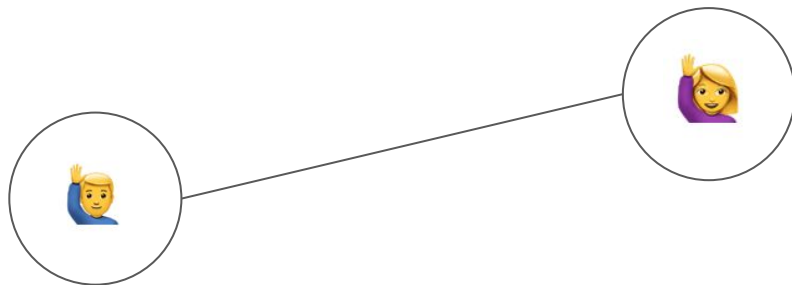
Obvious Example - Facebook



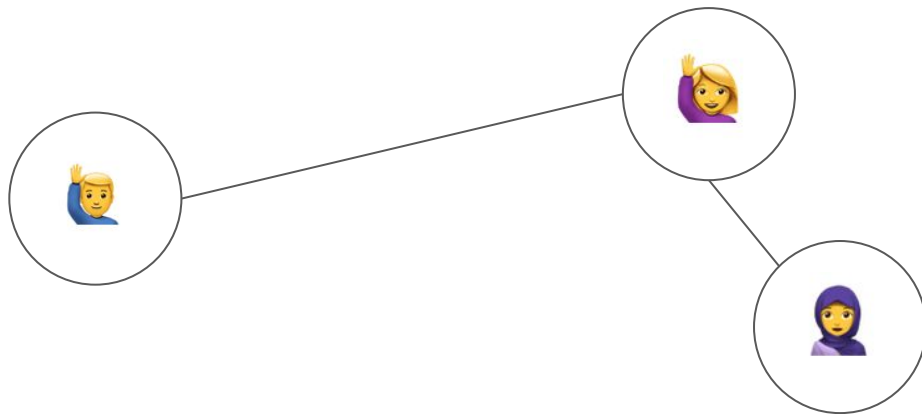
Obvious Example - Facebook



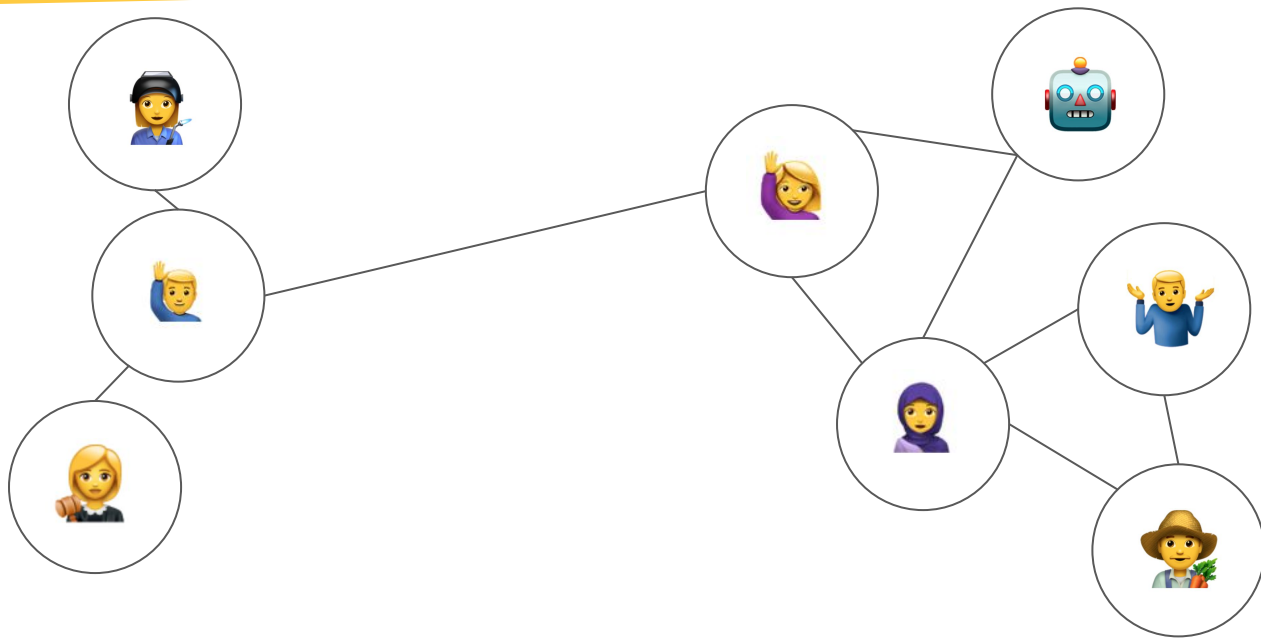
Obvious Example - Facebook



Obvious Example - Facebook



Obvious Example - Facebook



Points and Lines

Points and Lines have more formal names, which vary by discipline

Points	Lines	Discipline
Nodes	Edges, Links	Computer Science
Vertices	Edges, Arcs	Math
Sites	Bonds	Physics
Actors	Ties, Relations	Sociology


Source:

Jean Mark Gawron, Python for Social Science,

https://gawron.sdsu.edu/python_for_ss/course_core/book_draft/Social_Networks/Social_Networks.html#what-are-networks

Points and Lines

Points and Lines have more formal names, which vary by discipline



Points	Lines	Discipline
Nodes	Edges, Links	Computer Science
Vertices	Edges, Arcs	Math
Sites	Bonds	Physics
Actors	Ties, Relations	Sociology

Source:

Jean Mark Gawron, Python for Social Science,

https://gawron.sdsu.edu/python_for_ss/course_core/book_draft/Social_Networks/Social_Networks.html#what-are-networks

Main Types of Networks

UnDirected

Facebook
Linkedin

Connections are directionless
and reciprocal

Directed

Twitter
Snaphchat
Instagram
Whateverkidsusenowadays

Connections don't have to be
reciprocal - if A follows B on
Twitter, B does not
necessarily follow A

Also: MultiGraph and Multi-Directed-Graph

Main Types of Networks

Multigraph

Nodes can have more than one edge between them.

e.g. messages on whatsapp
(and multi-directed-graph)

Weighted

The connection (the edge) can be quantifiable and different from edge to edge.

e.g. sending p2p payments for varying amounts.

Network Analysis in Python

The best way to learn is to get our hands dirty with real data and work out an example, so...

```
pip install networkx
```

Network Analysis in Python

Why NetworkX?

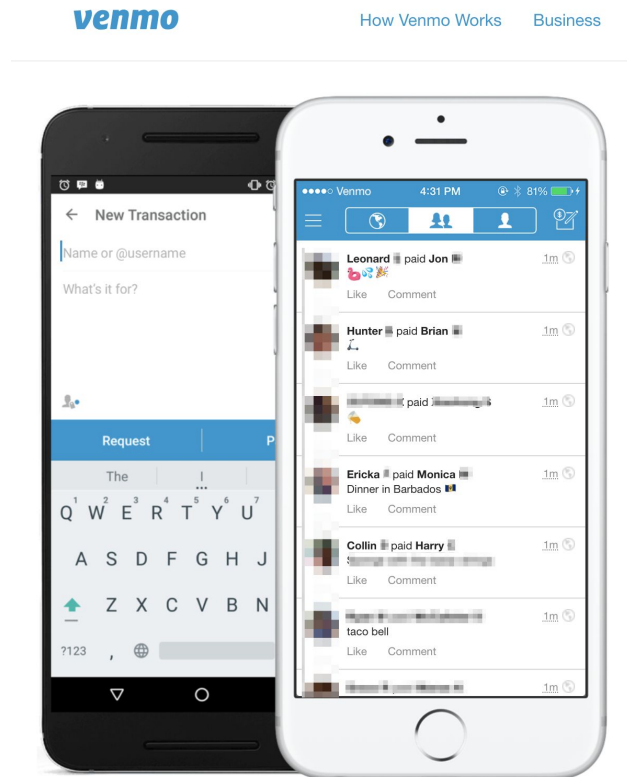
- Comprehensive (mostly)
- Friendly API
- Good documentation
- This is PyCon

Shortcomings

- Scalability to very large graphs
- Visualization of very large graphs

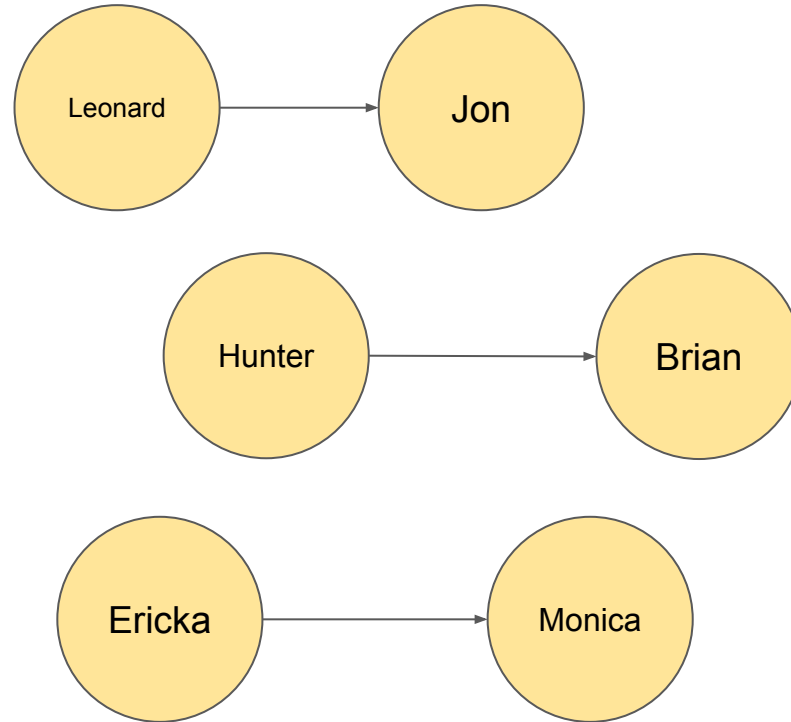
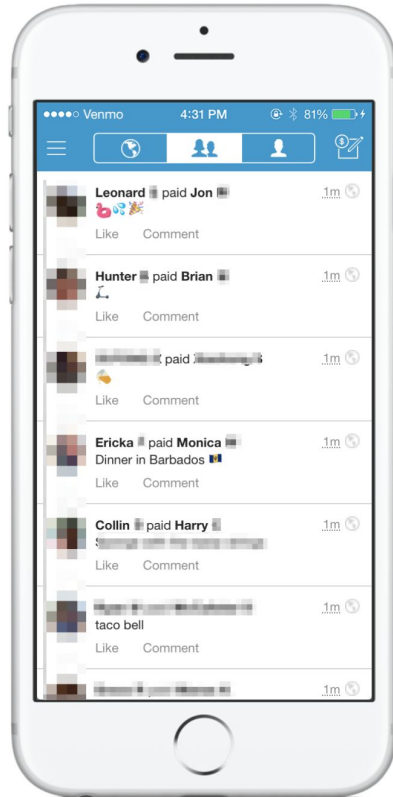
(other options: `snap.py`, `graph-tool`, `iGraph` and `Gephi`)

Example - Venmo



Link to
venmo.com

Example - Venmo



Make Python Run Now



Some Examples

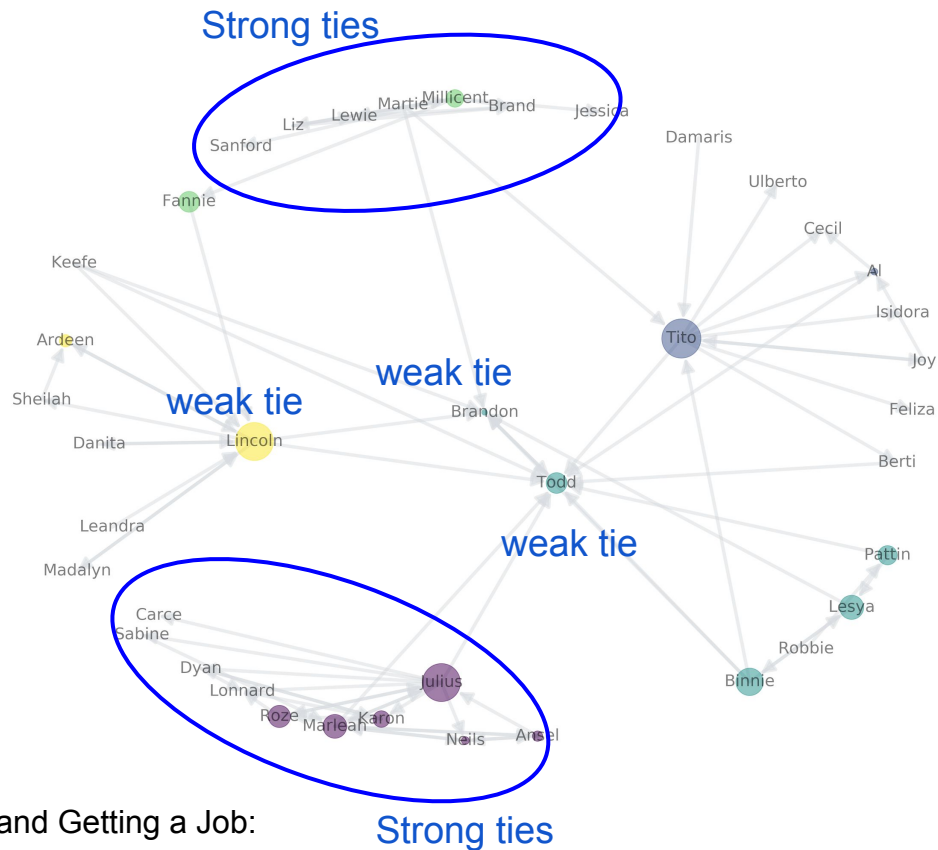


The Strength of Weak Ties

Seminal work by Mark Granovetter.

Cited 50K times!

- Jobs
- Spouses!

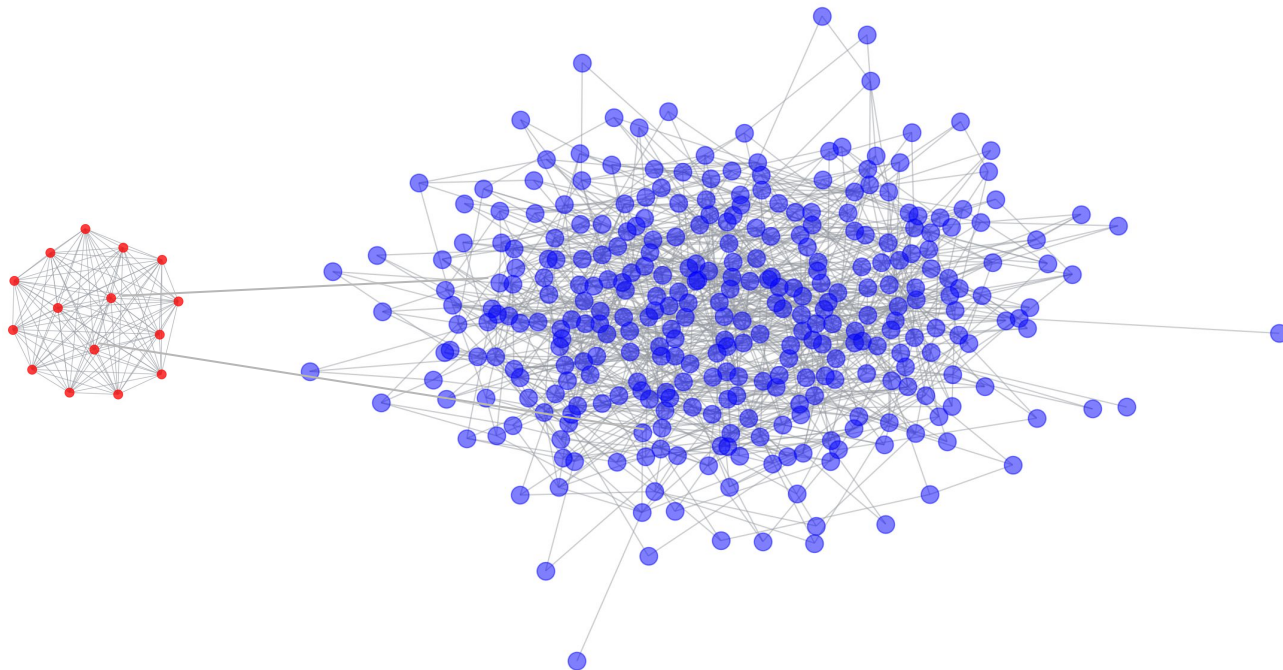


See:

Stanford Center on Poverty and Inequality: Social Networks and Getting a Job:

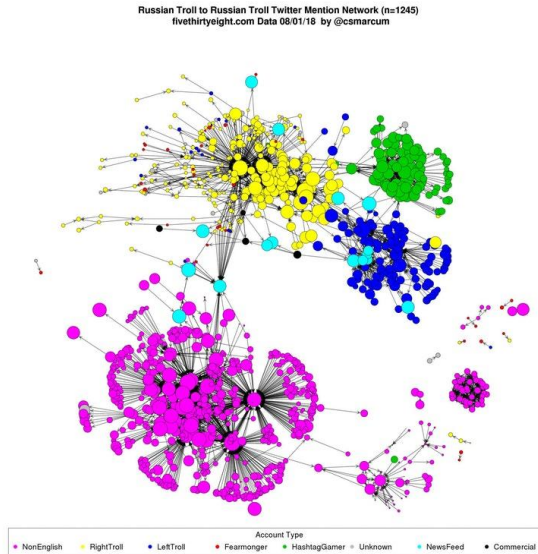
Mark Granovetter - [link](#)

Reddit Karma Cheaters



Source:
Simulated data by me.
Conceptual source: Chris Slowe, Reddit's CTO, in a SFU Data Science Talk, February 3rd 2018

Twitter Bot Networks

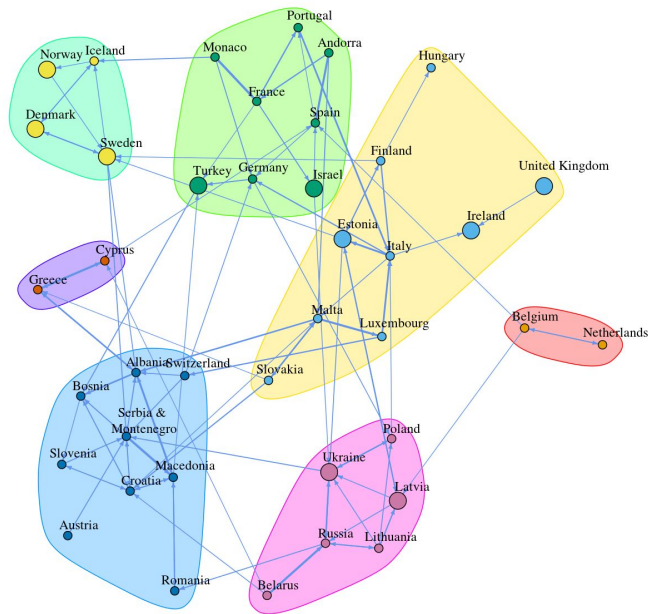


Credit:

@csmarcum https://www.reddit.com/r/dataisbeautiful/comments/9436q2/russian_troll_2_russian_troll_twitter_mention/

FiveThirtyEight data - <https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/>

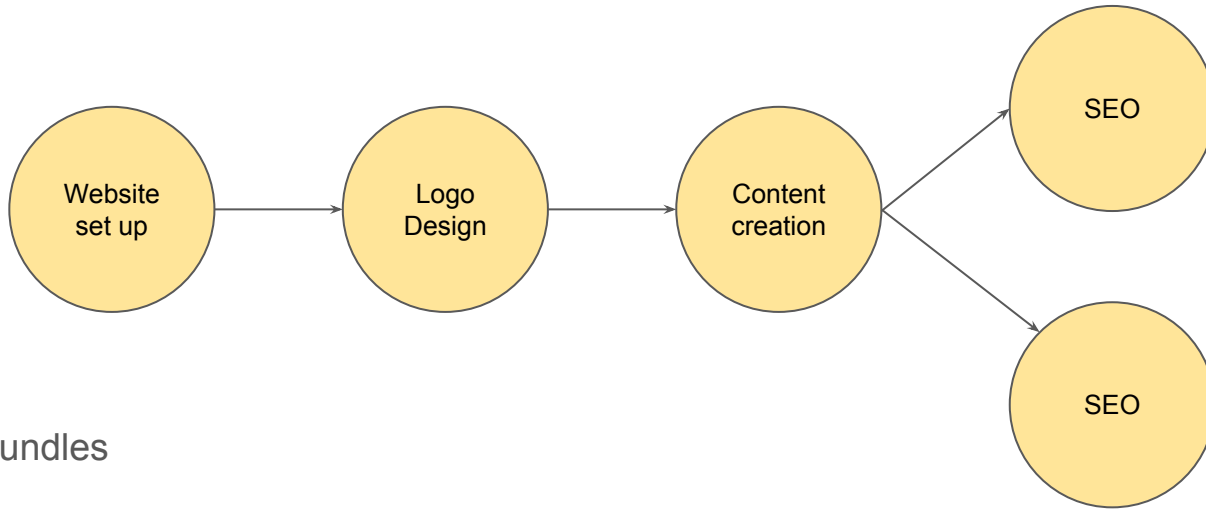
Eurovision Voting Examiner



Source:
Overthinking It - [Eurovision Voting Examiner](#)

Gig Economy Categories

Categories as nodes,
Customers that buy more than one type of service as edges.



→ Bundles

More...

Recommendations, clustering (bi-partite graphs, which we didn't cover)

Complex contagion

Closing Remarks



Conclusion

- Network analysis is a powerful, widely applicable tool.
- We barely scratched the surface.
- NetworkX is a great place to start.
- <3 Python

Next Steps

- Eric J. Ma - PyCon/PyData [Workshops](#), DataCamp course.
- Sara Guido - [NetworkX Tutorial](#) (watch out for version compatibility)
- Vaidehi Joshi - [A Gentle Introduction To Graph Theory](#)
- [Awesome](#) network analysis (resource list)
- [NetworkX Docs!](#)

Thank You!

Let's *connect*:
@alonnir



Homework Assignment

Discuss bridges,

Robustness

Disseminating info / contagion

Even paths?

לודא שכל המחברת רצה כמו שצריך אחרי שעושים ריסטארט לקרנל

! pdf גיבוי במצגות ו

לסגור מחברות אחרות ודברים אחרים כדי להקל על הזכרון

!!לשתף גם עם המחשב השני

Notes for me - do not publish

(Networks or graphs)

Source:

https://github.com/jtorrents/pydata_bcn_NetworkX/blob/master/NetworkX_SNA_workshop_with_solutions.ipynb

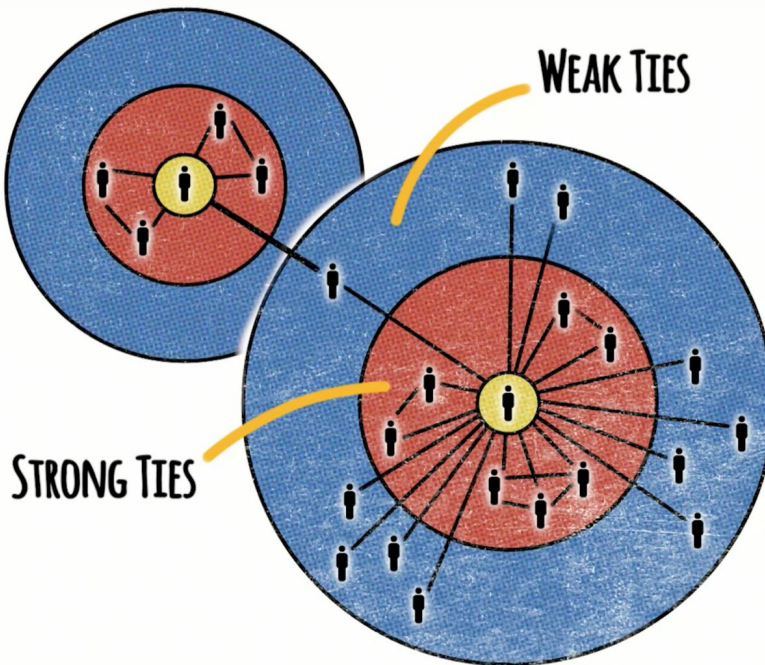
These networked structures are characterized in terms of **nodes** or **vertices** (individual actors, people, or things within the network) and the **edges** or **links** (relationships or interactions) that connect them. From a mathematical point of view, these networked structures are modeled as **Graphs**.

The Strength of Weak Ties

Seminal work by Mark Granovetter.

Cited 50K times!

- Jobs
- Spouses!



Source:

Stanford Center on Poverty and Inequality: Social Networks and
Getting a Job: Mark Granovetter - [link](#)

try to visualize the large network with

<https://stackoverflow.com/a/51535568>

also:

<https://stackoverflow.com/questions/7991138/how-do-i-visualize-social-networks-with-python>

<too big - connected components>

Communities? Something else?

CC להראות פה דוגמא גרפית טיפשית מה זה

Connected components

For undirected graphs, a **component** is a maximal connected subgraph, which means that there is at least one path between any two nodes in that subgraph.

For directed graphs, a **weakly connected component** is a subgraph that is connected if we replace all of its directed edges with undirected edges. A **strongly connected component** is a subgraph where there is a path in each direction between each pair of nodes of the subgraph.