Lecture 8

Complex Network Analysis

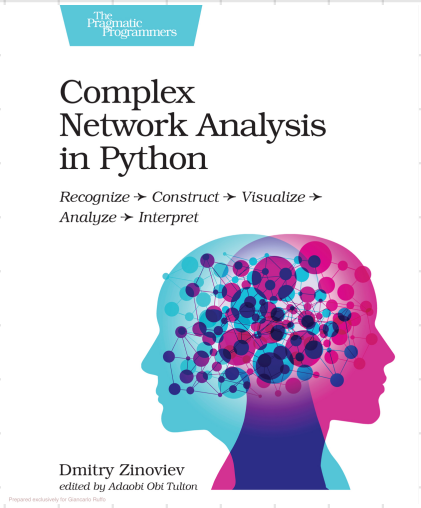
Network Construction

&

Synthetic Networks

Today's Topics

- Understanding Social Networks
- Advanced Network Construction
- Synthetic Networks
- Slice Weighted Networks



Chapters 6 & 7

Understanding Social Networks

- Ego / Socio centric Nets
- Network properties
- Acquisition of Social Networks
- Signed Networks
- Networks Collections
- Synthetic Networks

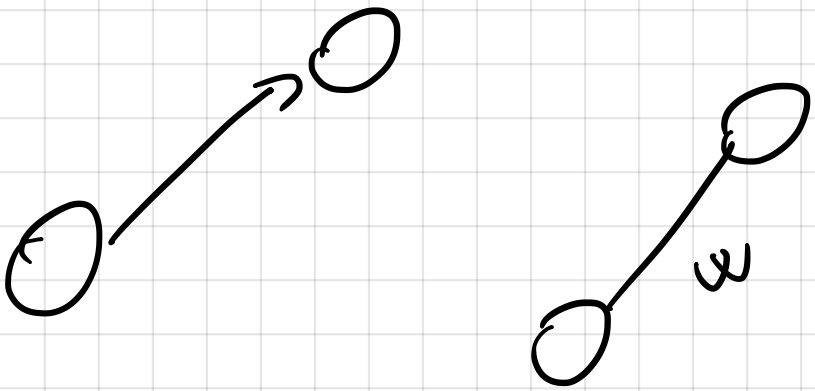
Social Networks

nodes: individuals (persons or animals)

edges: significant relationship between two individuals

directed vs undirected

weighted (weak vs strong)



Egocentric Networks

Wikipedia case study: an example of Ego-network (subject: "Complex Network")

We want to understand the structure, function and composition of connections around a single individuals

ego: central individual

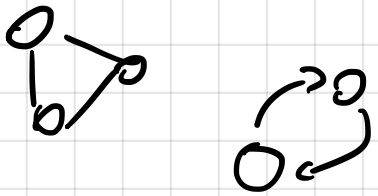
alter: all the other nodes

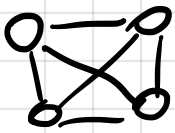
1. Start with an ego
2. Explore the alters and their contacts
   (scraping and parsing HTML files
   or
   using official APIs
   or
   copying by hand...)

Network properties

Local topology :

- structural equivalence
- triadic closure
- Balance theory





Centralities :

some actors is more important than others

Degree Distribution :

friendship paradox

small world

Network Dynamics :

emergence of hubs

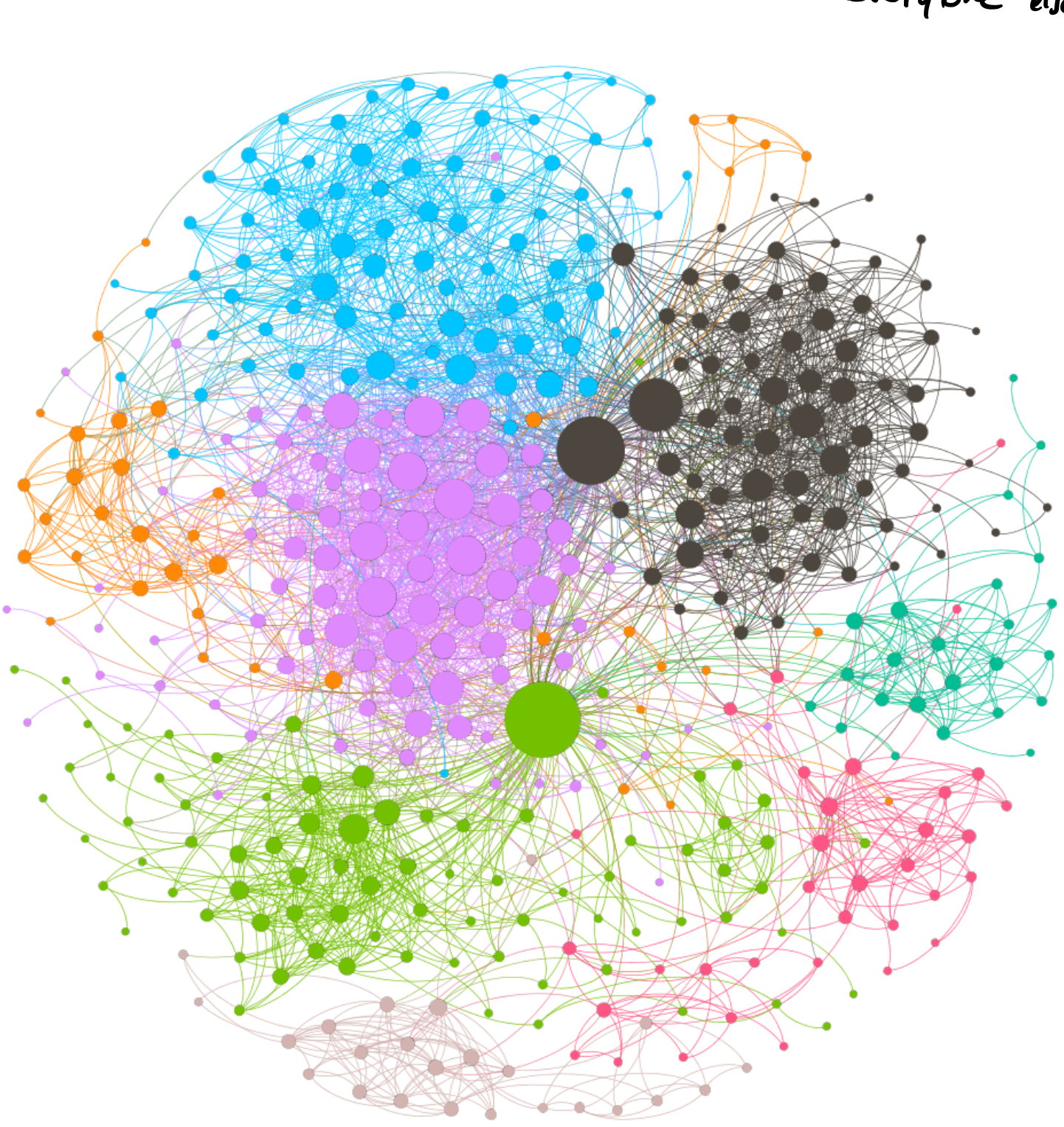
the richer gets richer

Example: Facebook

you could use GetNet until 2015

Below: my FB ego network (2012)

I'm not in the picture (I am connected to everyone else)



Sociocentric Networks

A combination of all the ego networks

It can be really large

Dunbar's number: 150 "friends"
but we have incredibly skewed distributions

A non-Trivial social network is complex

it is not a matter of size,
but the interpretation:
Which are the social theories that govern
degree distribution, centralities,
local network Topology,
community structure, network evolution ...

Acquisition of Social Networks

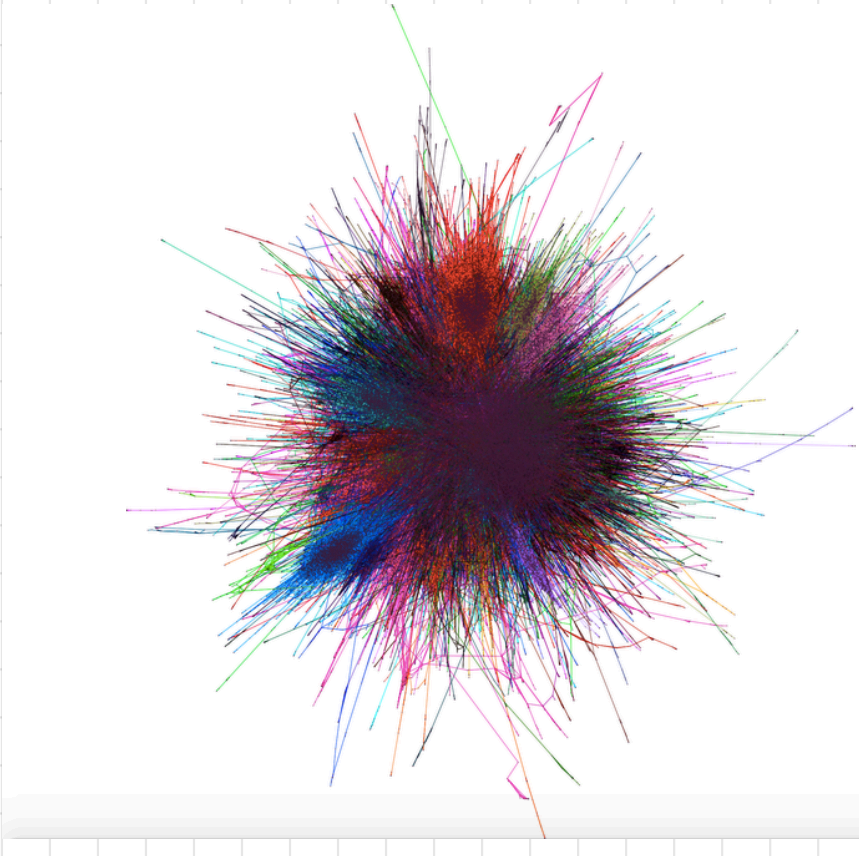
It's difficult to get a complete sn of interest

- huge
- dynamic

Where do we start?
(many snow ball processes from different seeds)

Random samples are often the key

Hoi Krug (2009)
165,795 nodes
433,118 edges



Signed Network

Weights can be negative!

You can represent
friends and enemies

Structural Balance theories

Collections of networks

Pre-boiled datasets!

SNAP: the Stanford Large
Network Datasets

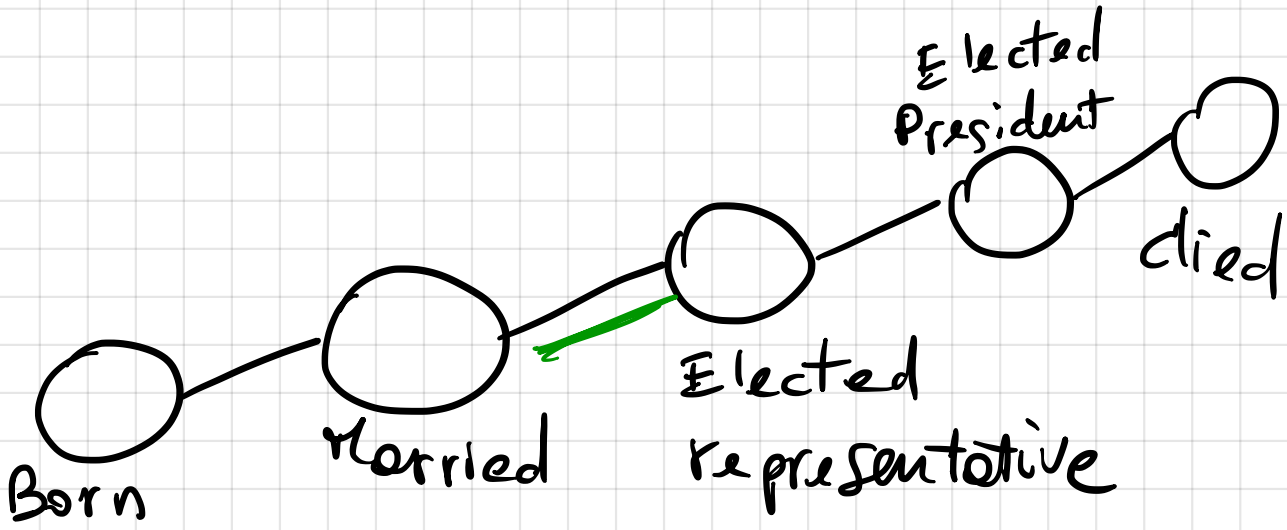
KONECT: Koblenz Network
Collection

Advanced Network Construction

- Adjacency and Incidence Matrices
- Edge Lists and Node Dictionary

Adjecency Matrix

Abraham Lincoln Life's Graph



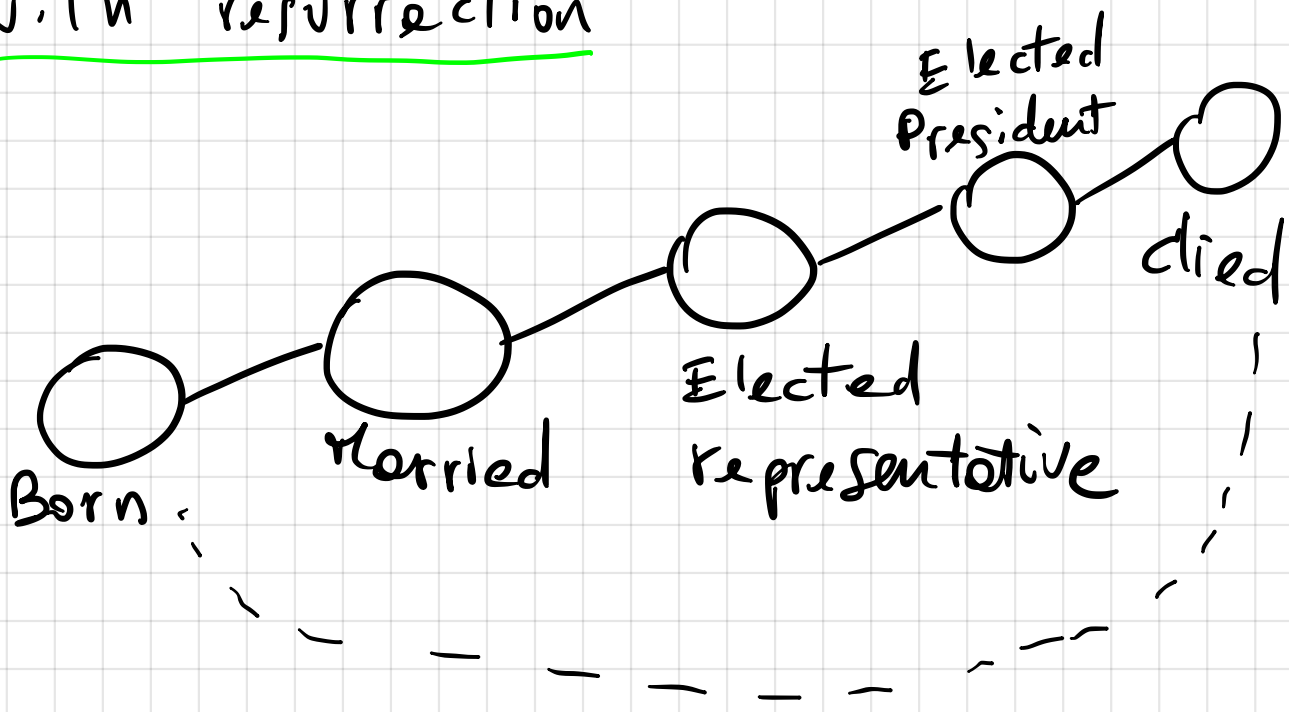
$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} B \\ M \\ ER \\ EP \\ D \end{matrix}$$

Born, M, ER, EP, D

$N \times N$

Adjecency Matrix with Weights

Abraham Lincoln Life's Graph with resurrection



$$A = \begin{pmatrix} 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 0 \\ 0 & 0 & 0 & 1. & 0 \\ 0 & 0 & 0 & 0 & 1. \\ 0.1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$N \times N$

Continue with

"04 - netConstruction"

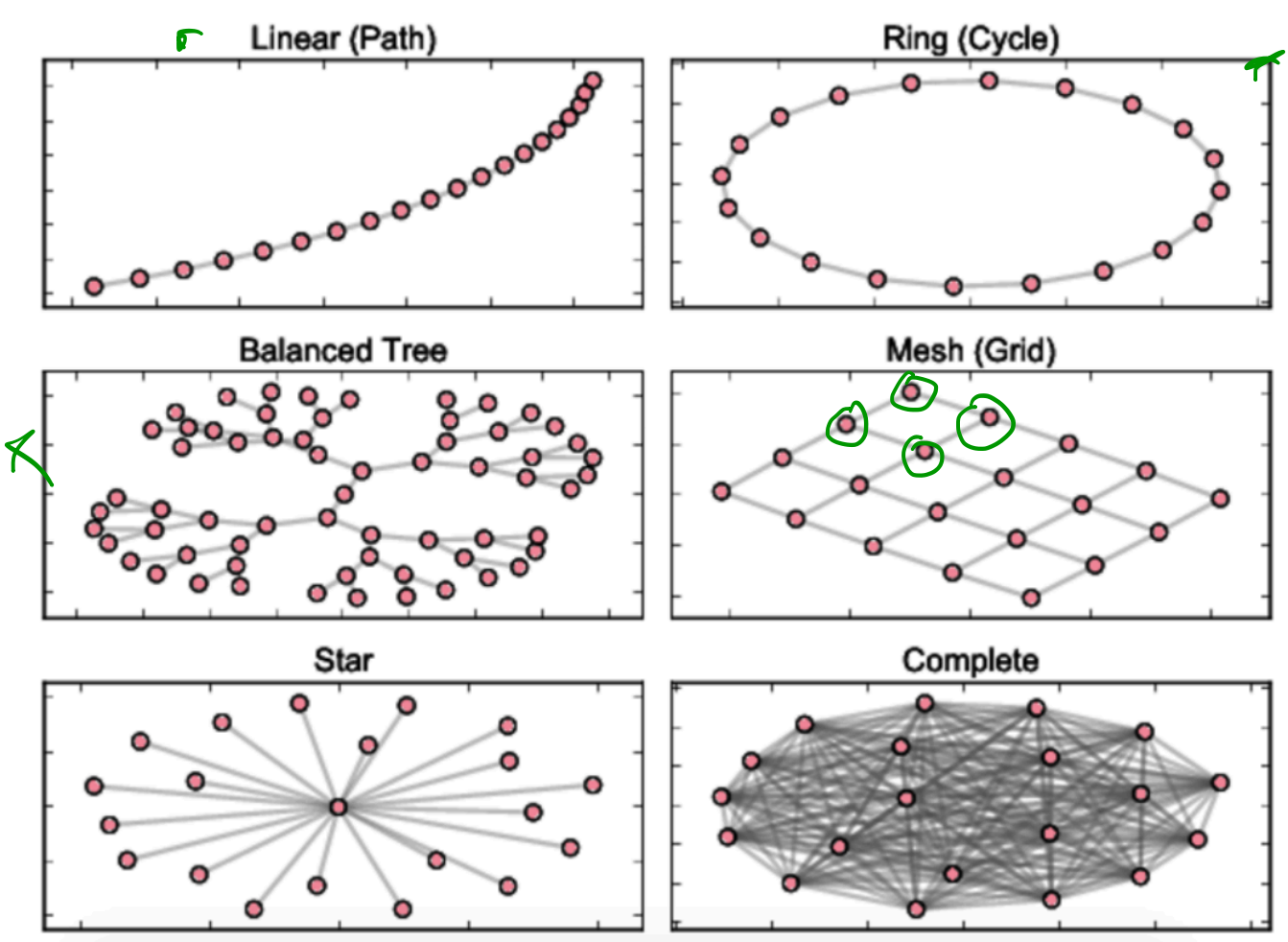
To learn:

- how to create networks from Adjacency Matrix
  - with pure python
  - with NumPy
  - with Pandas and DataFrames
- how to manipulate node attributes with Pandas
- how to create Networks with Incidence Matrices
- how to work with Edge Lists and Node Dict.

Synthetic Networks

- Alternative to real-world data
- useful to be compared with empirical networks
- they can be generated automatically (e.g., networkx) or by hand

Simple Networks:



Random Models

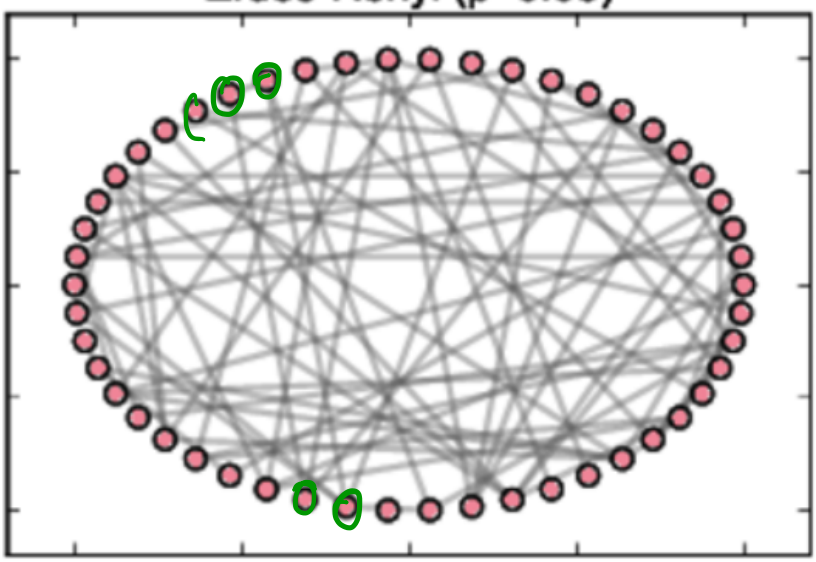
complex networks that are generated following a given hypothesis

Erdös - Rényi

- N nodes
- It can have up to $\frac{N(N-1)}{2}$ edges
- each edge is assigned with probability p

($p = 0$: no edges

$p = 1$: complete graph)



- short paths
- low clustering coefficient

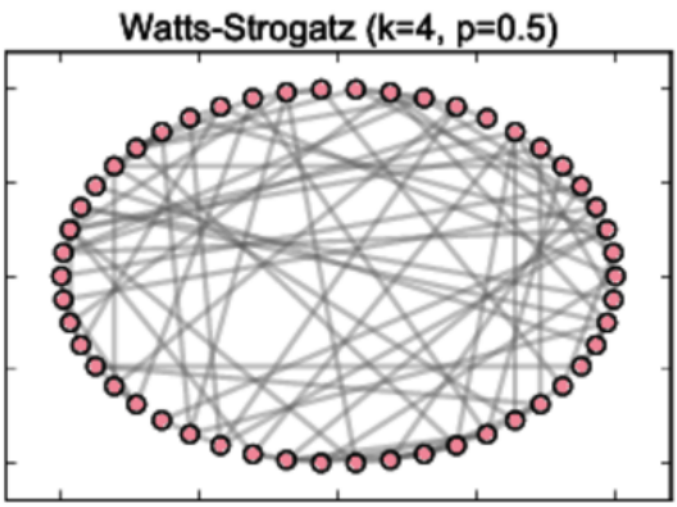
Watts - Strogatz

more realistic:

- N nodes in a ring
- each node connected with K ring neighbors
- each edge is "rewired" randomly with probability p

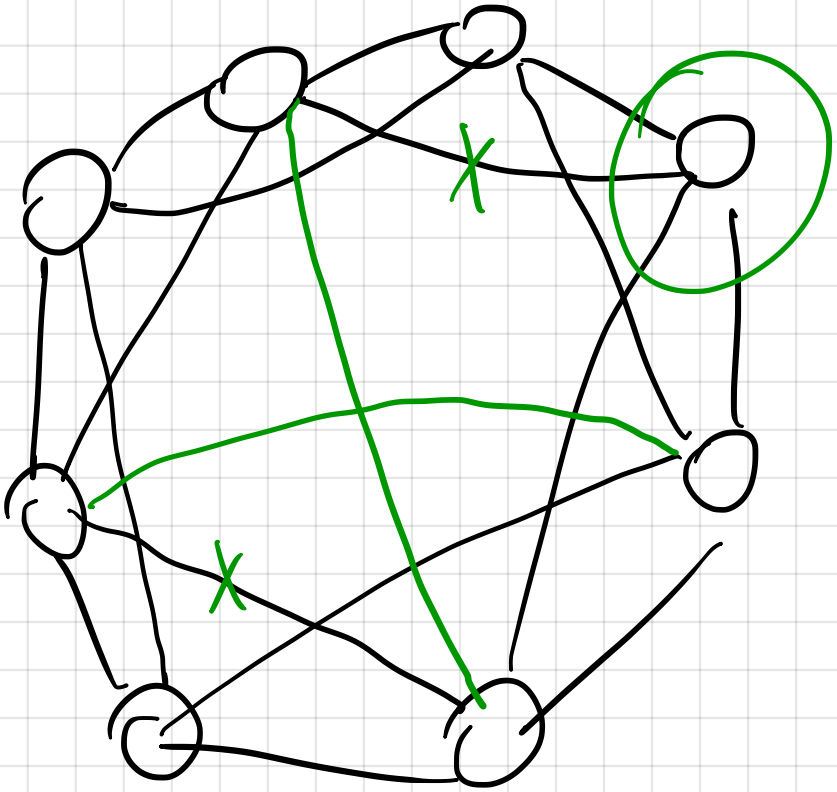
- $p = 0$: regular graph
  $p = 1$: random graph

- you can find some p where you have both high cc and low distances



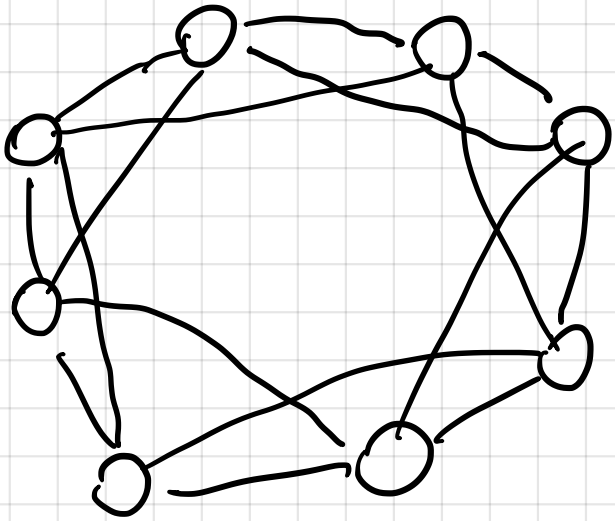
rewiring probability $p = \frac{2}{15} = 0.13$

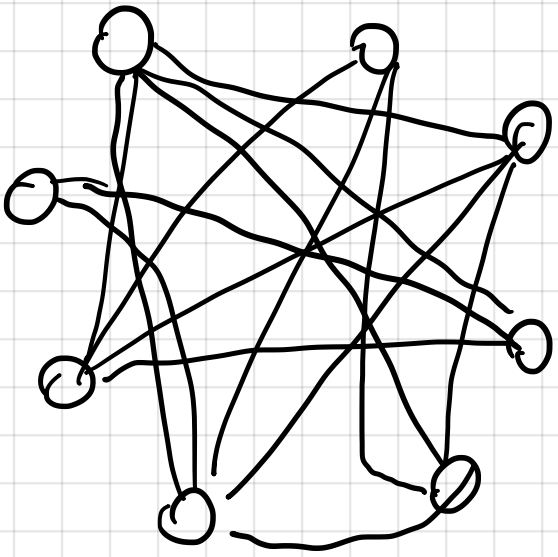
$k = 4$



distances ↓

c. coeff. ↓





$p = 0$ (regular)

high distances

high c.c. (highest!)

$\Longrightarrow$

$p = 1$ (random)

low distances

low c.c.

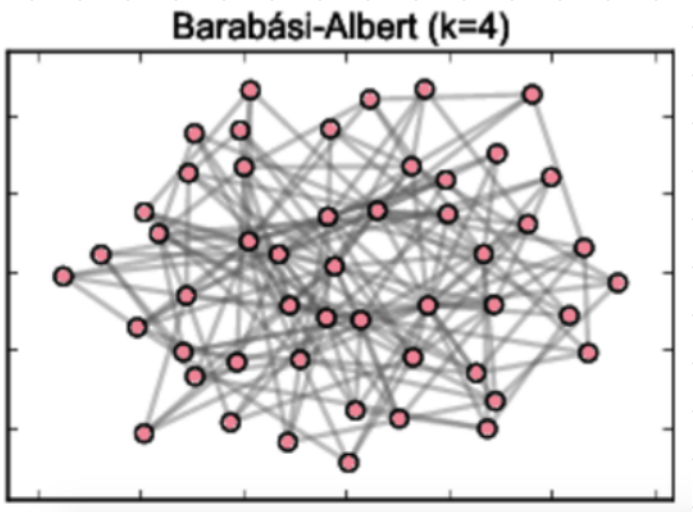
Barabasi - Albert

start generating the network with few nodes

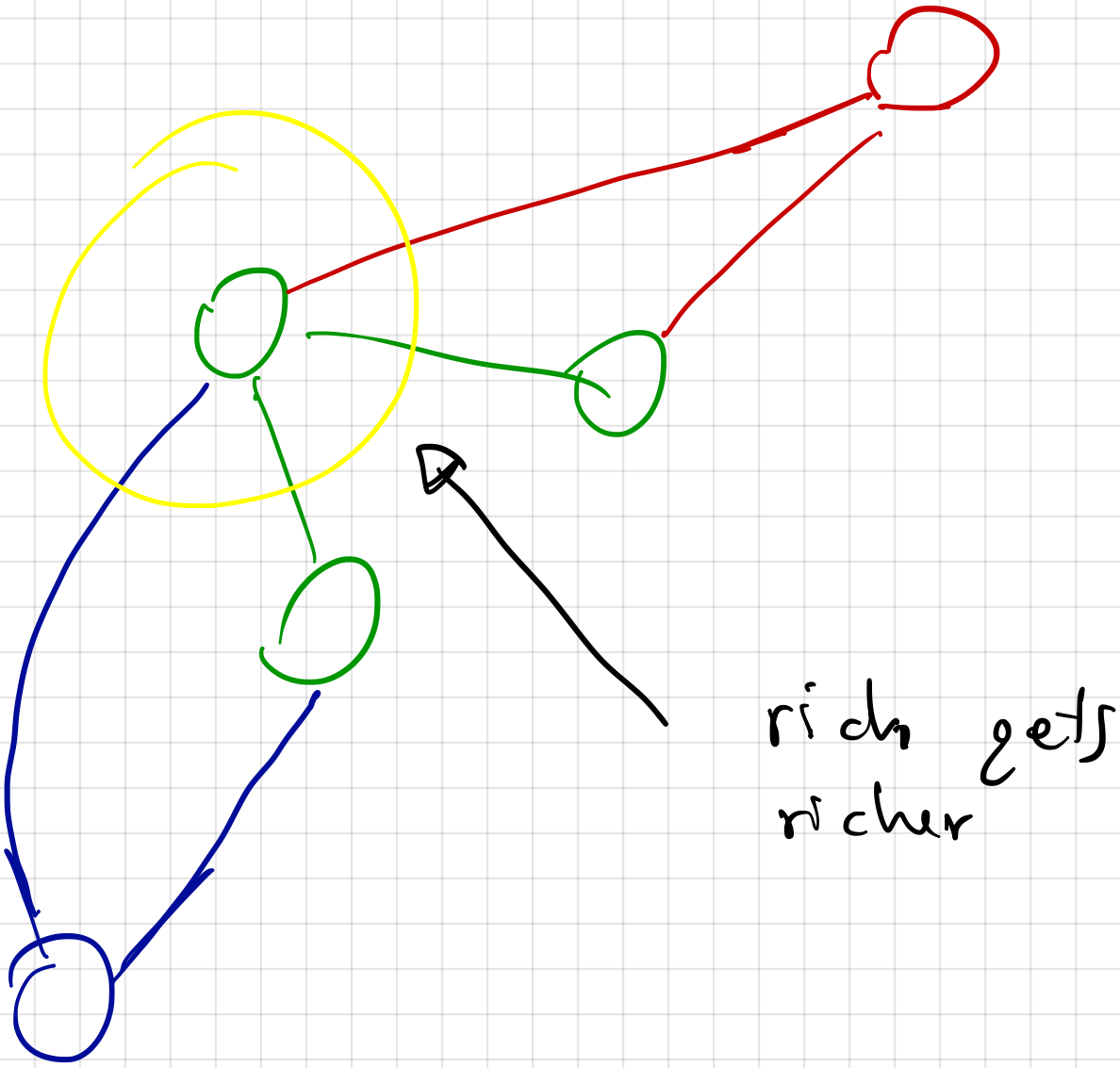
at the end of the process you have N nodes

when a new node joins, its k edges are more likely to be attached to higher degree nodes

Degree Distribution follows a power law



" preferential attachment "

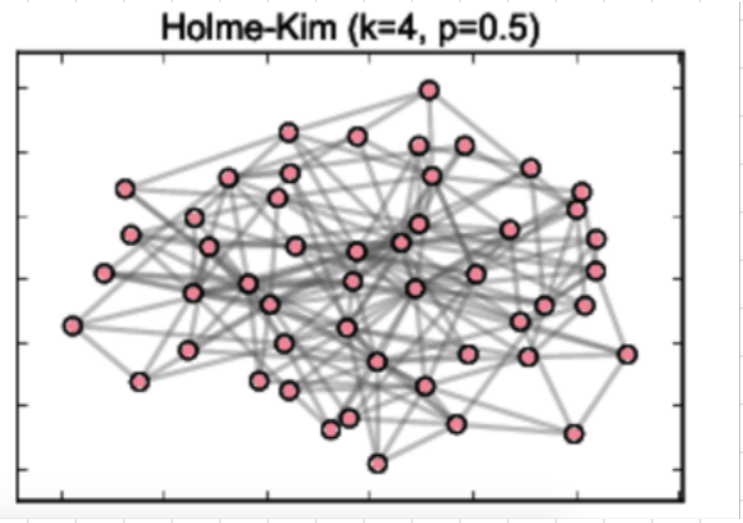

rich gets
richer

Holme - Kim

WS and BA do not form communities

HK is like BA but after adding k edges, it also adds Triads with the probability of p

Clusters are generated (more than in real life)



Exercise

Generate many random graphs with python and networkx

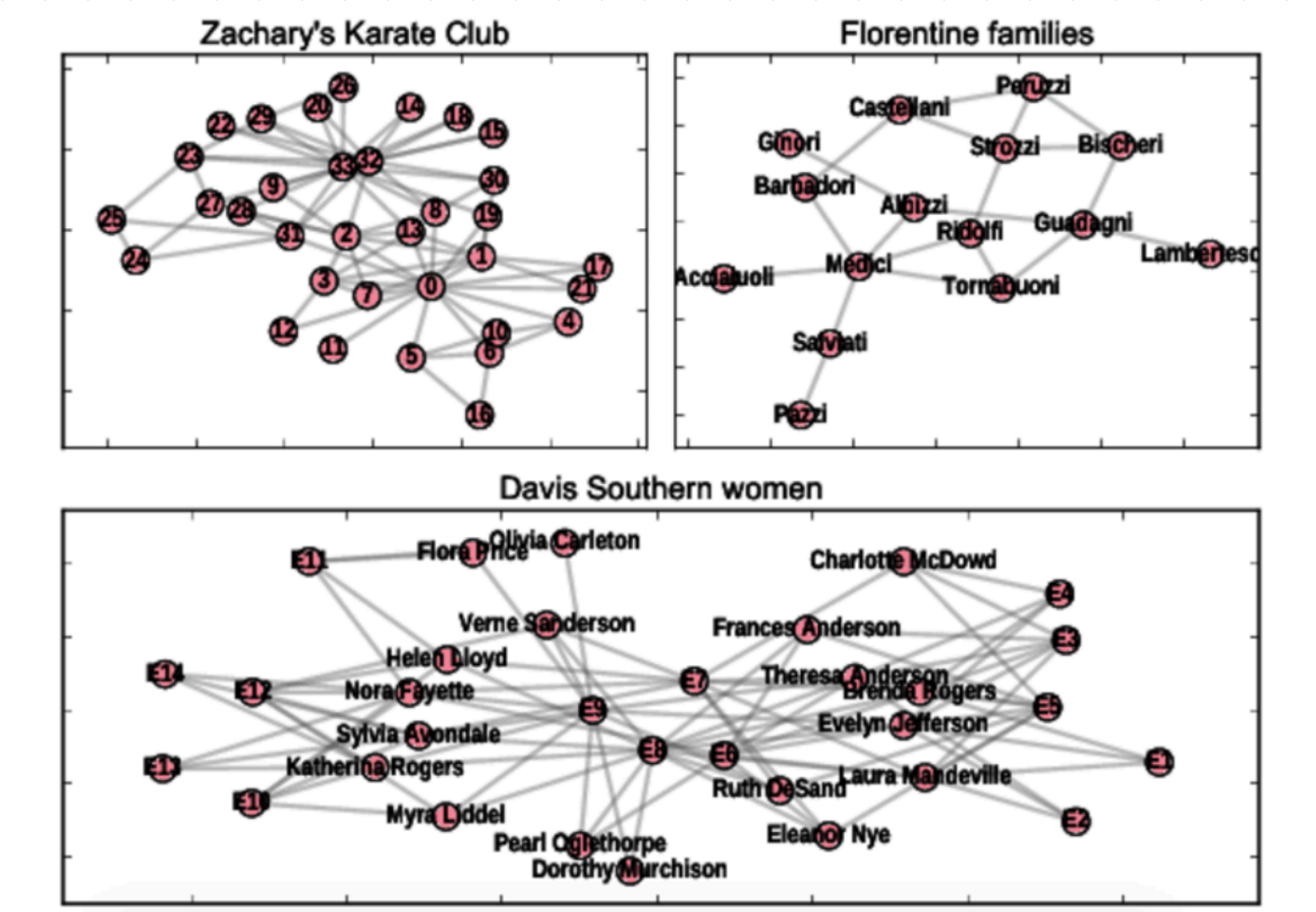
Visualize them with Gephi to highlight their characteristics

Make some math:

use python to find

- average distances
- degree distribution
- average clustering coefficient
- ...

Famous Social Networks



Your Turn:

explore notebook

"05 - generators"

Slice Weighted Networks

In weighted networks, some edges are strong, some are weak

Sometimes you cannot keep all the edges

Slicing is the process of eliminating low-strength edges

Simplest form: you choose a cut-off threshold T that controls the density of the resulting network


```
∀ edge e:
    if weight(e) < T
        remove e;
```

How to select T

1. Select a T based on edge weight distribution
2. Slice the network
3. Calculate some network properties (# components, density, ...)
4. if you are unsatisfied with results, go back to 1.

run 06_Slicing.ipynb