DESCRIPTION OF GRAPHS

DEFINITIONS

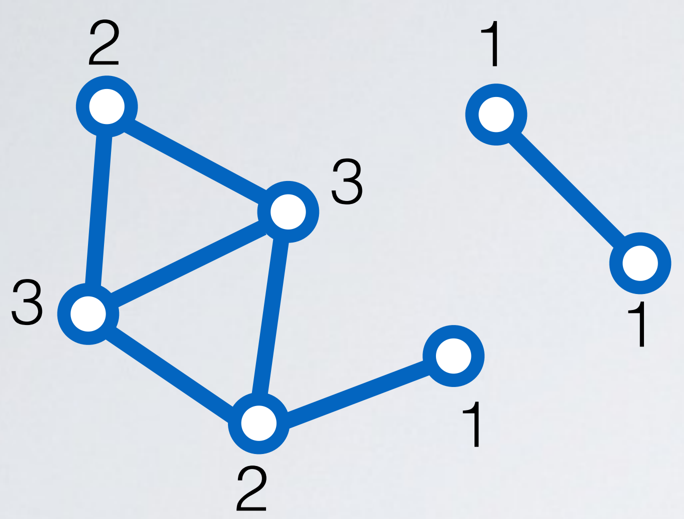
Node-Edge description

N_u	**Neighbourhood** of u, nodes sharing a link with u.
k_u	**Degree** of u, number of neighbors $\|N_u\|$.
N_u^{out}	**Successors** of u, nodes such as $(u,v) \in E$ in a directed graph
N_u^{in}	**Predecessors** of u, nodes such as $(v,u) \in E$ in a directed graph
k_u^{out}	**Out-degree** of u, number of outgoing edges $\|N_u^{out}\|$.
k_u^{in}	**In-degree** of u, number of incoming edges $\|N_u^{in}\|$
$w_{u,v}$	**Weight** of edge (u,v).
s_u	**Strength** of u, sum of weights of adjacent edges, $s_u = \sum_v w_{uv}$.

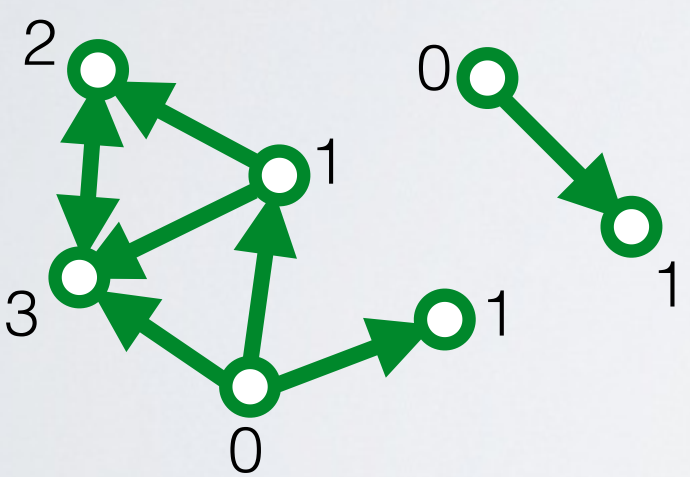
Node degree

Number of connections of a node

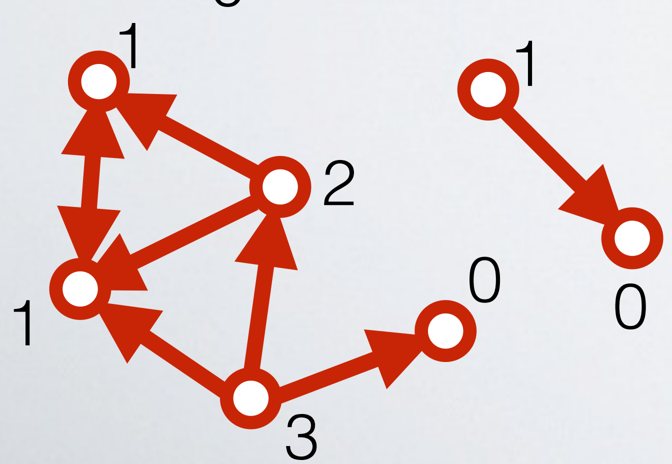
- Undirected network



- Directed network

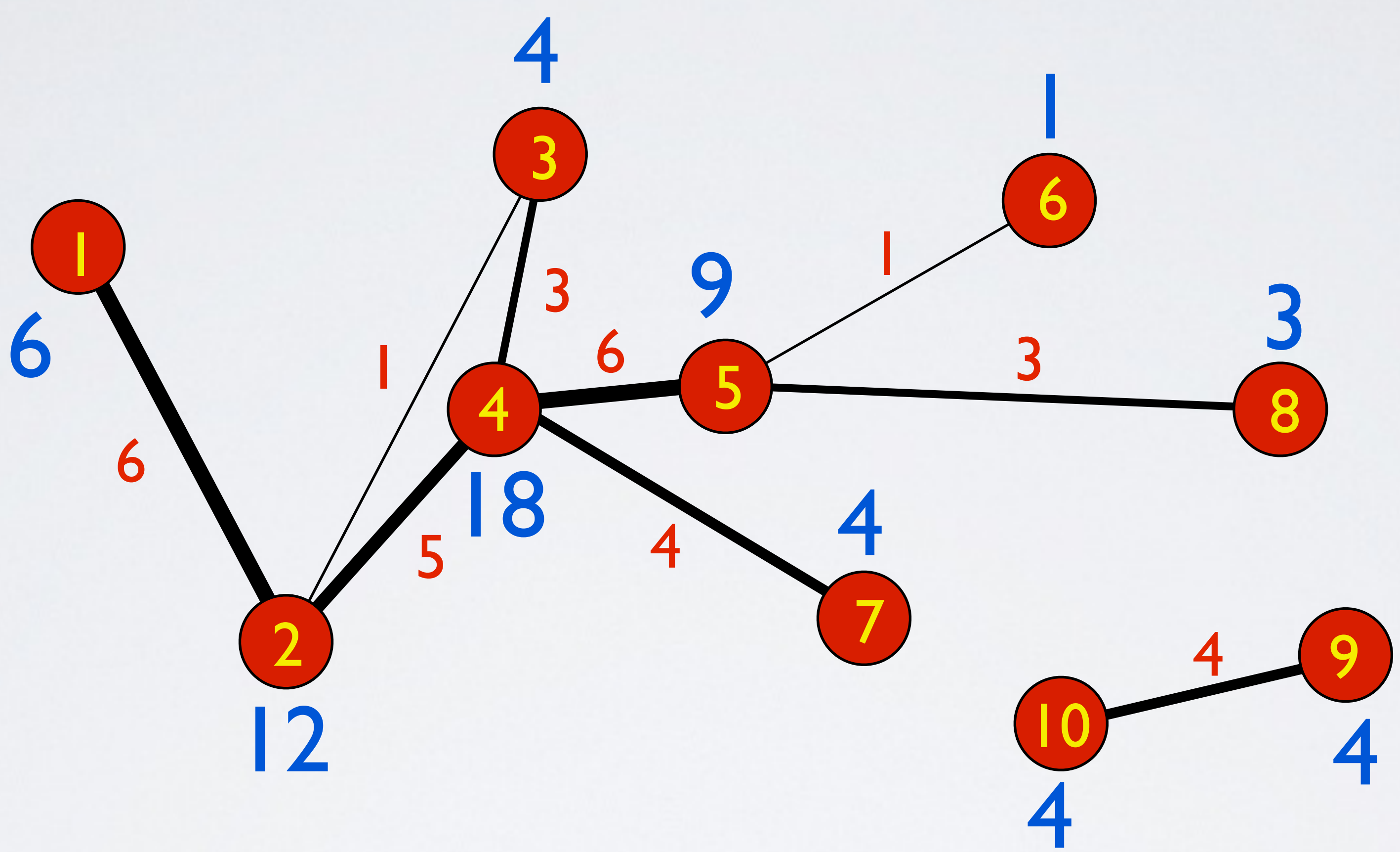


In degree



Out degree

Weighted degree: strength



USAGES OF NETWORK DESCRIPTION

- Understand an observed graph
- Compare graphs modeling related items
  - Graph of two different cryptocurrencies, two different users
  - Graphs of the same system taken at different points in time
- Compare an observed graph with a random model
  - Is my graph random?
  - Is property **p** exceptional or also observed in a similar random network?

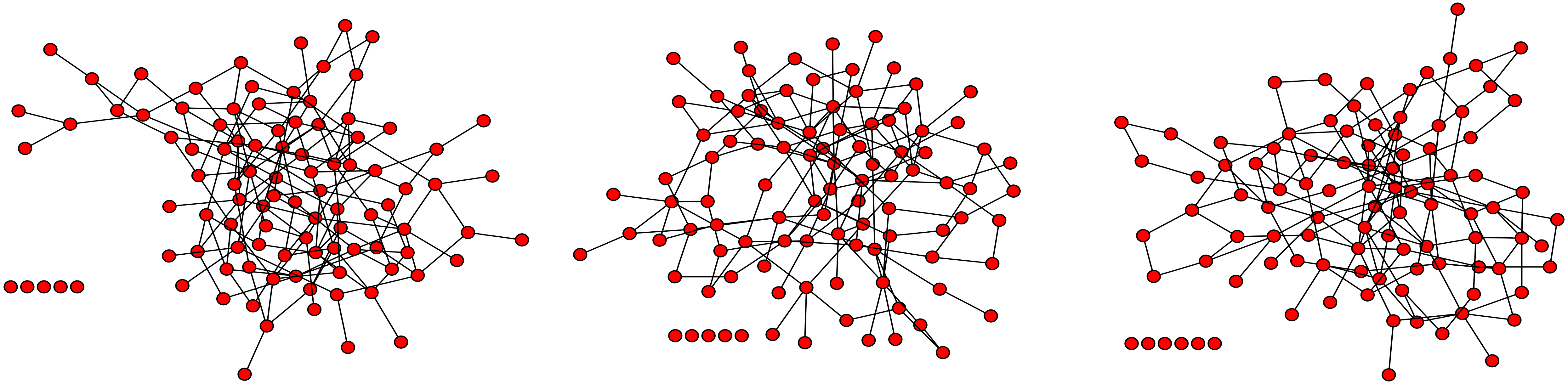
ER Random Graphs

Erdős-Rényi model: simple way to generate random graphs

- The $G(n,L)$ definition
  1. Take n disconnected nodes
  2. Add L edges uniformly at random

- The $G(n,p)$ definition
  1. Take n disconnected nodes
  2. Add an edge between any of the nodes independently with

p=0.03
N=100



SIZE

Counting nodes and edges

N/n	**size**: number of nodes $\|V\|$.
L/m	number of edges $\|E\|$
L_{max}	Maximum number of links

Undirected network: $\binom{N}{2} = N(N-1)/2$

Directed network: $\binom{N}{2} = N(N-1)$

SIZE

	#nodes (n)	#edges (m)
Wikipedia HL	2M	30M
Twitter 2015	288M	60B
Facebook 2015	1.4B	400B
Brain c. Elegans	280	6393
Roads US	2M	2.7M
Airport traffic	3k	31k

Bitcoin “transactions” ≈ 700M

Bitcoin “addresses” ≈ 800M

Non-zero balance “entities” in 2020: 20M

DENSITY

Network descriptors 1 - Nodes/Edges

$\langle k \rangle$ | **Average degree**: Real networks are sparse, i.e., typically $\langle k \rangle \ll n$. Increases slowly with network size, e.g., $d \sim \log(m)$

$$\langle k \rangle = \frac{2m}{n}$$

$d/d(G)$ | **Density**: Fraction of pairs of nodes connected by an edge in G.

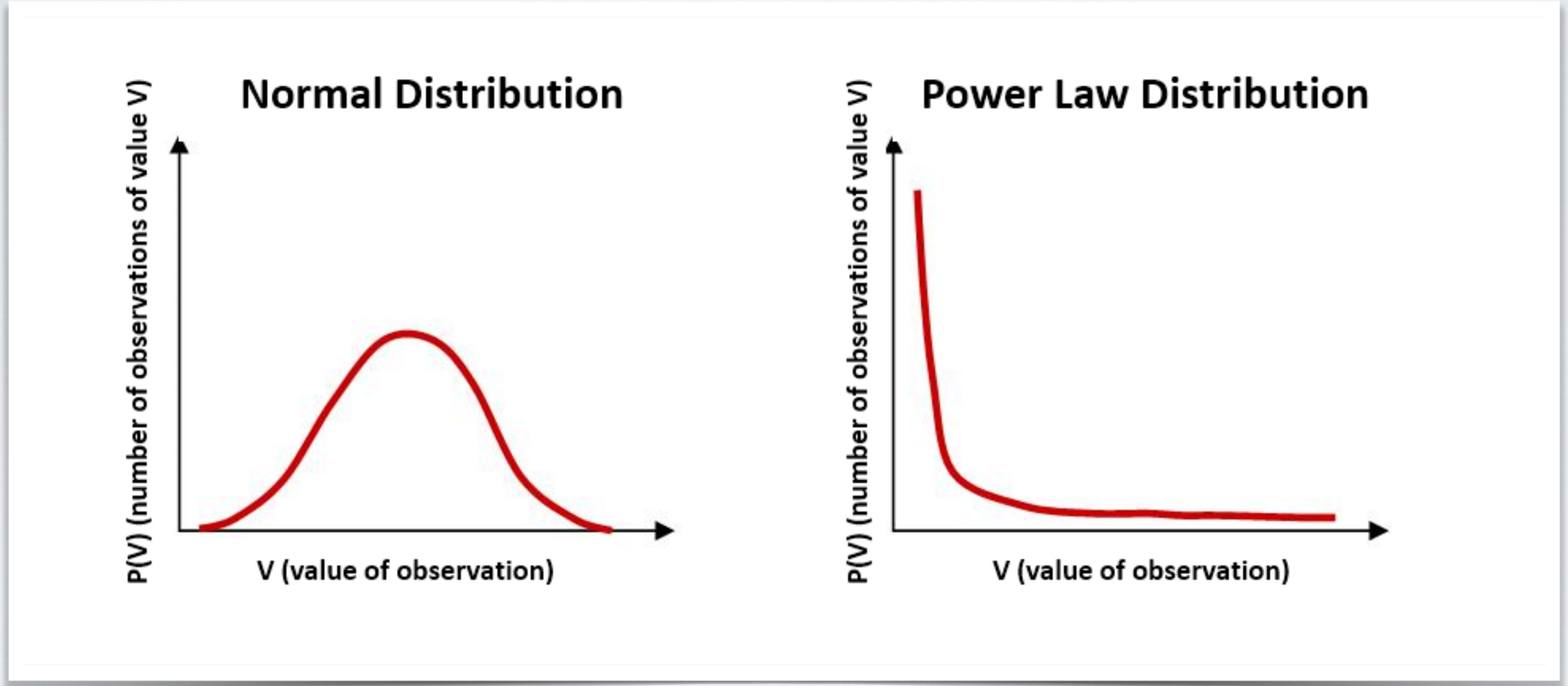
$$d = L/L_{\max}$$

DENSITY

	#nodes	#edges	Density	avg. deg
Wikipedia	2M	30M	1.5×10^{-5}	30
Twitter 2015	288M	60B	1.4×10^{-6}	416
Facebook	1.4B	400B	4×10^{-9}	570
Brain c.	280	6393	0,16	46
Roads Calif.	2M	2.7M	6×10^{-7}	2,7
Airport	3k	31k	0,007	21

Beware: density hard to compare between graphs of different sizes

DEGREE DISTRIBUTION



PDF (Probability Distribution Function)

DEGREE DISTRIBUTION

- In a fully random graph (Erdos-Renyi), degree distribution is (close to) a normal distribution centered on the average degree
- In real graphs, in general, it is not the case:
  - A high majority of small degree nodes
  - A small minority of nodes with very high degree (Hubs)
- Often modeled by a **power law**

DEGREE DISTRIBUTION

- Power law degree distribution has important implications:
  - There is no “scale” in the degree: the average degree is not representative
  - The variance is non-converging, i.e., cannot be interpreted
- How to recognize a power law?
  - Simple approximation: it is a line on a log-log plot
  - If you want to be sure, use existing packages in R or python (Maximum Likelihood Estimation)

Proper definition

$$P(k) \sim Ck^{-\alpha}$$

The distribution is controlled by the exponent α

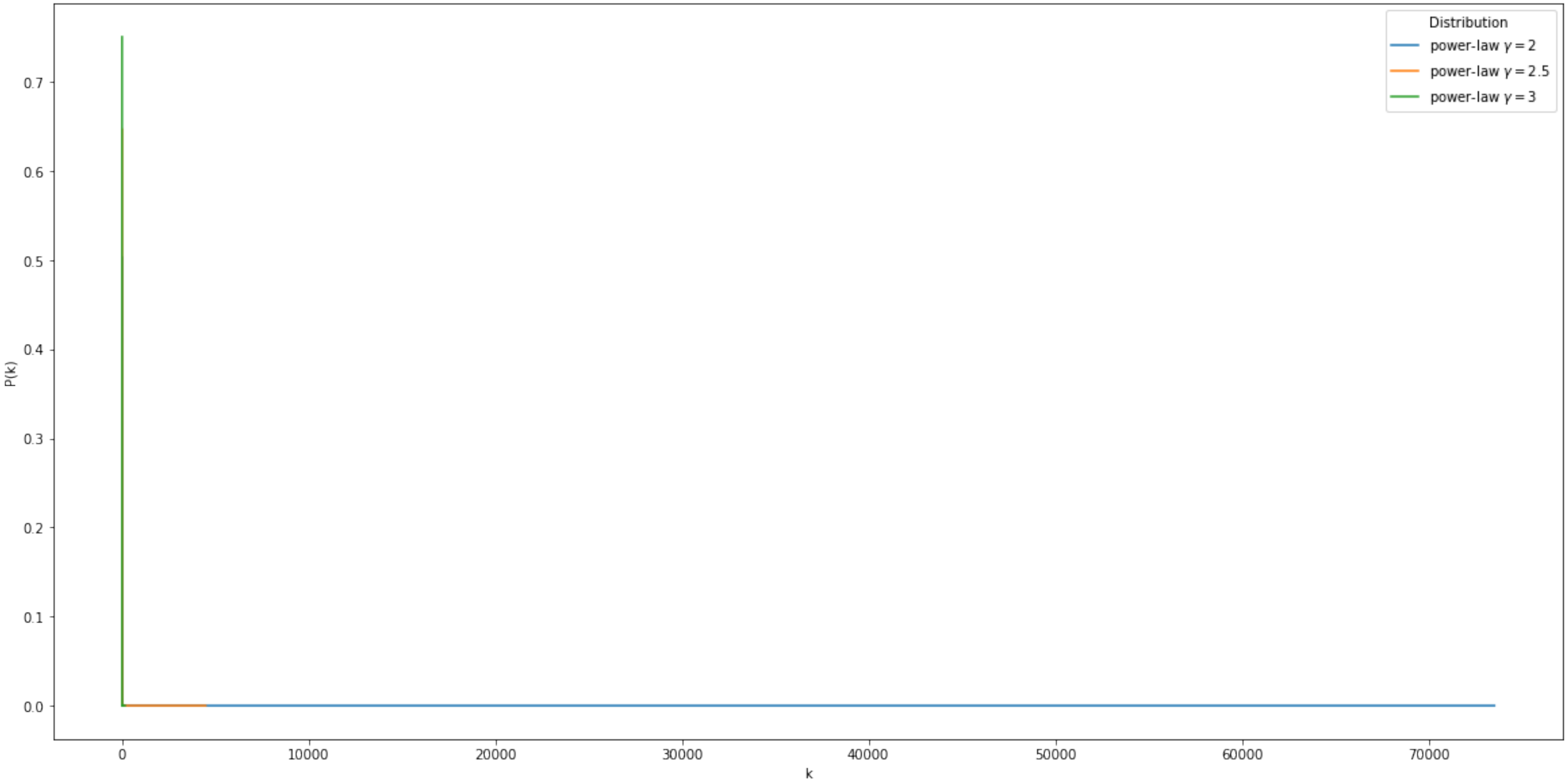
$$P(k) = (\alpha - 1)k_{\min}^{\alpha-1}k^{-\alpha}$$

$$P(k) = \frac{\alpha - 1}{k_{\min}}\left(\frac{k}{k_{\min}}\right)^{-\alpha}$$

Scale-free networks

Power law plotted with a linear scale, for k<100000

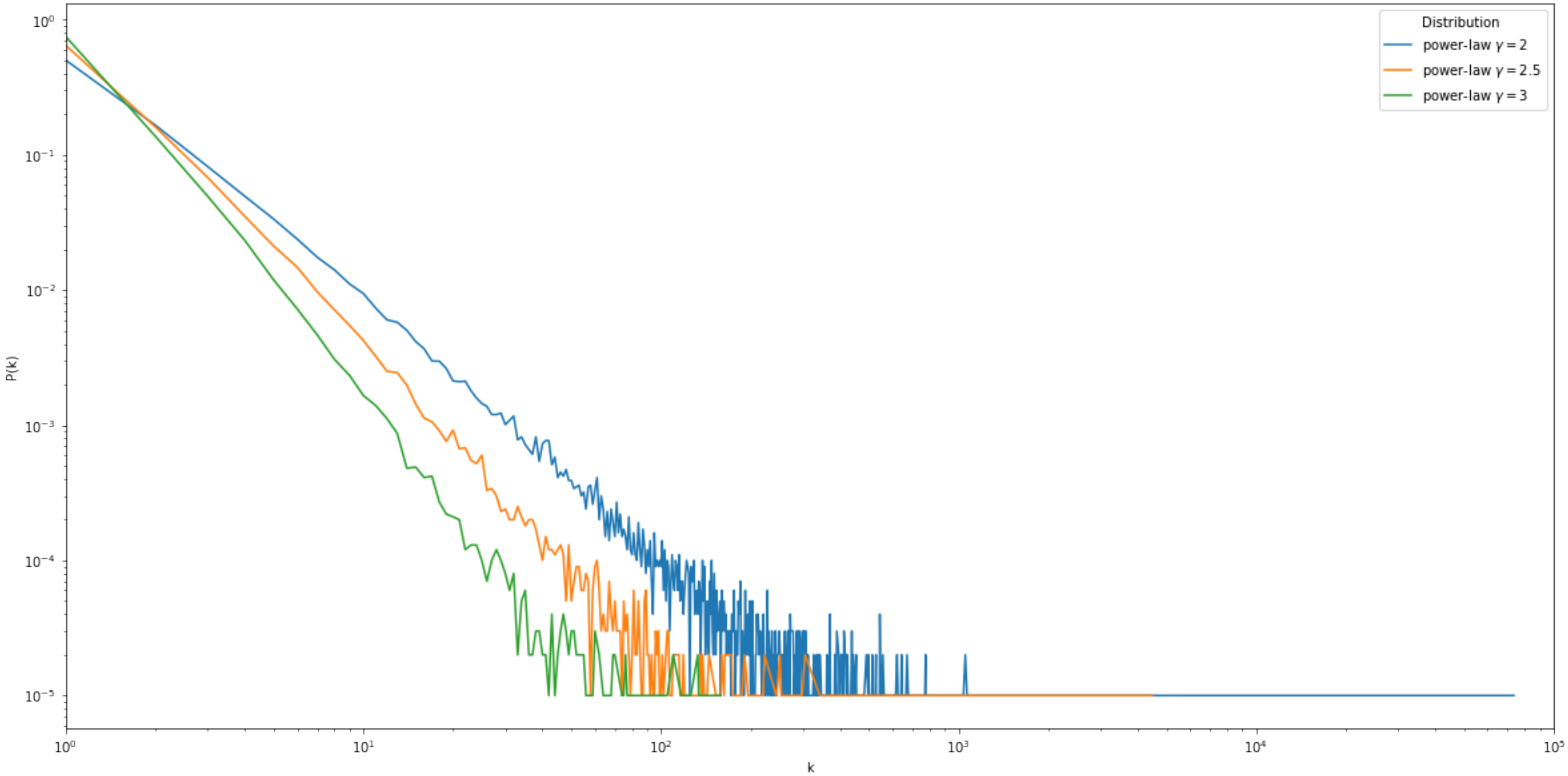
(100 000 samples)



Scale-free networks

Power law plotted with a log-log scale, for k<100000

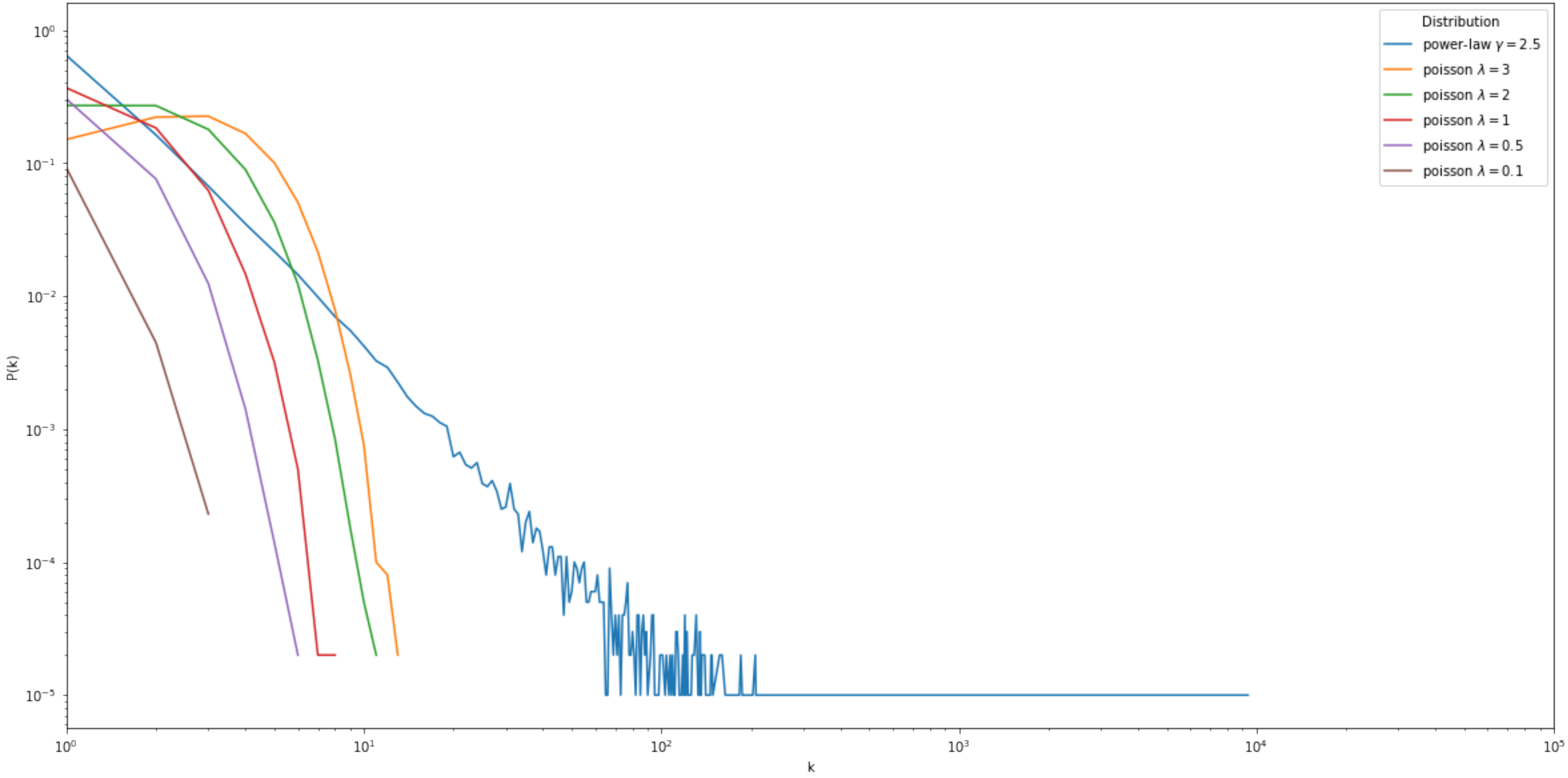
(100 000 samples)



Scale-free networks

Comparing a poisson distribution and a power law

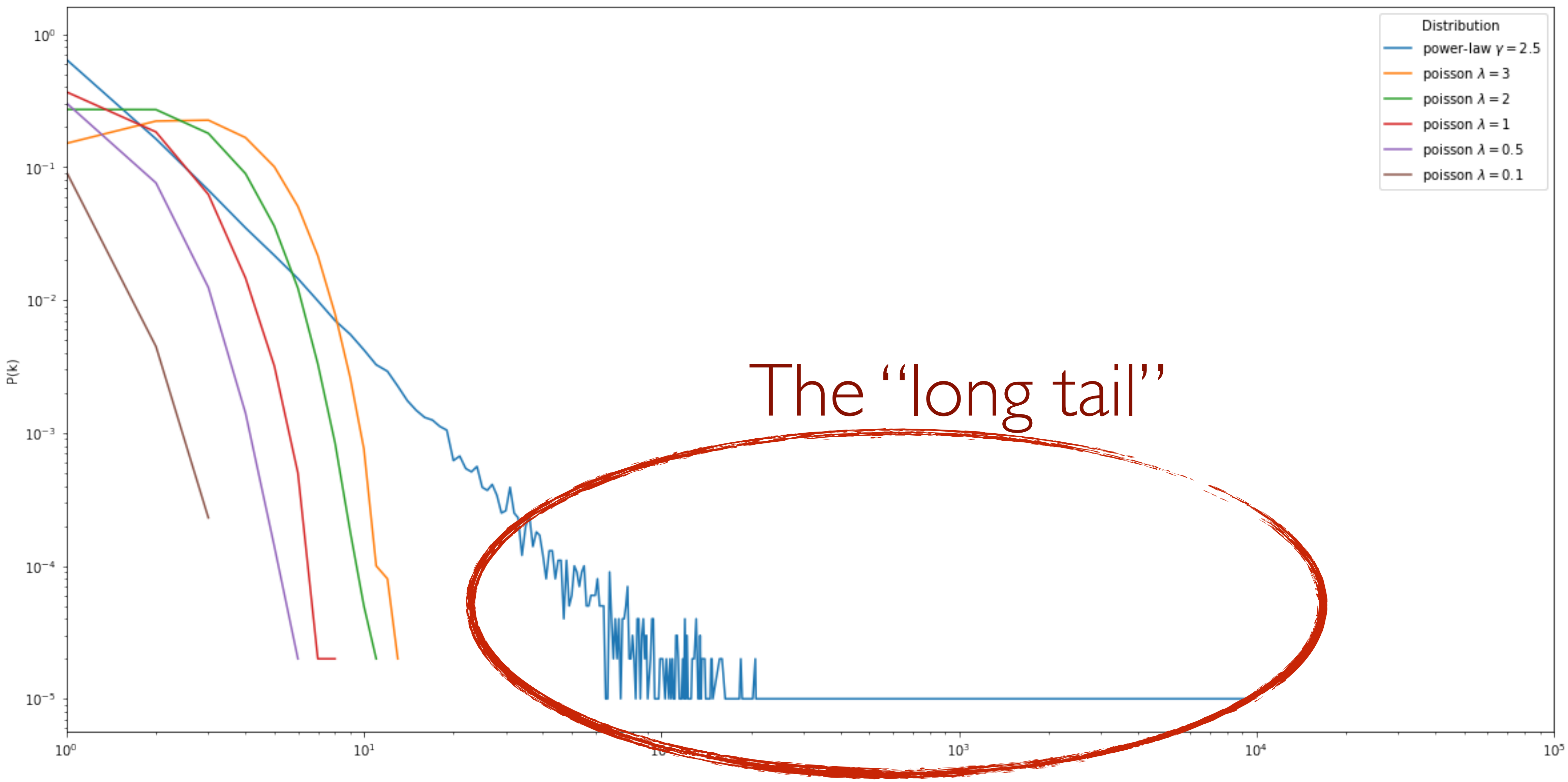
$$\frac{\lambda^k e^{-\lambda}}{k!}$$



Scale-free networks

Comparing a poisson distribution and a power law

$$\frac{\lambda^k e^{-\lambda}}{k!}$$



SUBGRAPHS

Subgraphs

Subgraph $H(W)$ (induced subgraph): subset of nodes W of a graph $G = (V, E)$ and edges connecting them in G, i.e., subgraph $H(W) = (W, E'), W \subset V, (u, v) \in E' \iff u, v \in W \land (u, v) \in E$
Clique: subgraph with $d = 1$
Triangle: clique of size 3
Connected component: a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph
Strongly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths
Weakly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths if we disregard directions

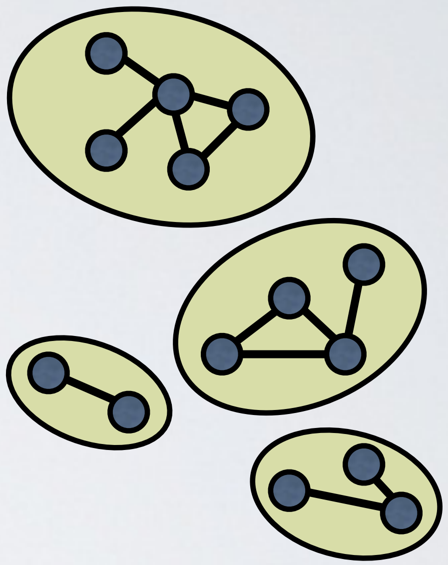
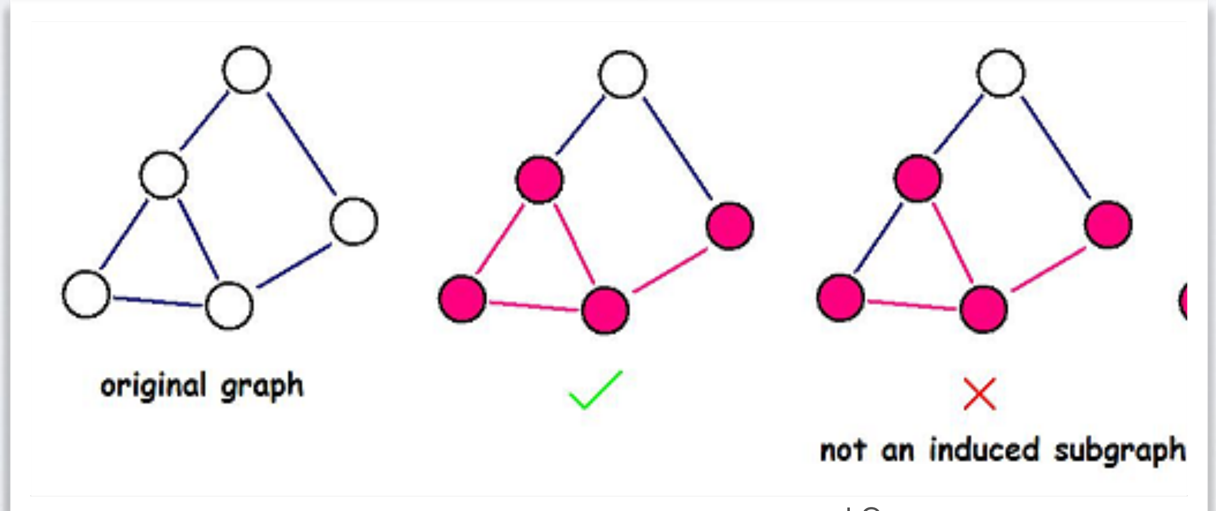


Figure after Newman, 2010



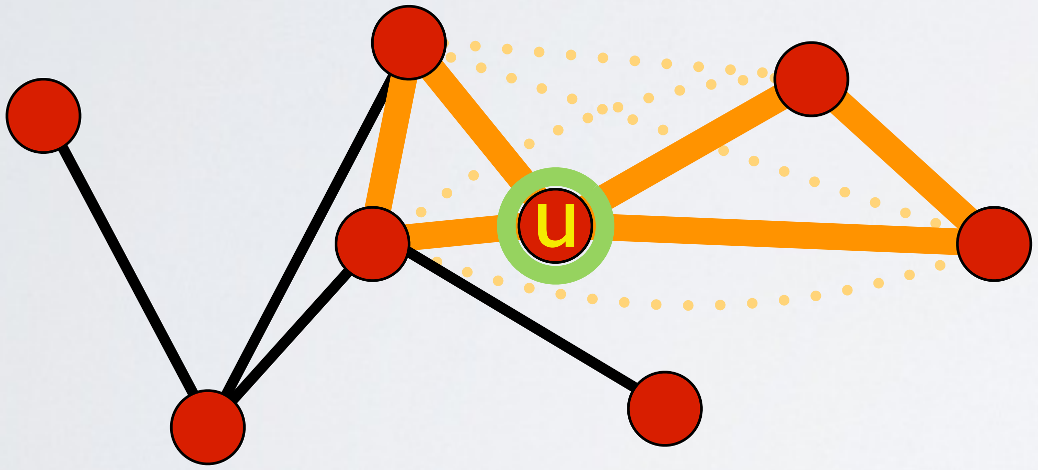
After "A. DZY Loves Physics"

CLUSTERING COEFFICIENT

- **Clustering coefficient** or **triadic closure**
- Triangles are considered important in real networks
  - Think of social networks: *friends of friends are my friends*
  - # triangles is a big difference between real and random networks

CLUSTERING COEFFICIENT

C_u - **Node clustering coefficient:** density of the subgraph induced by the neighborhood of u, $C_u = d(H(N_u))$. Also interpreted as the fraction of all possible triangles in N_u that exist, $\frac{\delta_u}{\delta_u^{\max}}$



Edges: 2
Max edges: 4*3/2=6
C_u=2/6=1/3

CLUSTERING COEFFICIENT

$\langle C \rangle$ - **Average clustering coefficient:** Average clustering coefficient of all nodes in the graph, $\bar{C} = \frac{1}{N} \sum_{u \in V} C_u$.

C^g - **Global clustering coefficient:** Fraction of all possible triangles in the graph that do exist, $C^g = \frac{3\Delta}{\Delta^{\max}}$

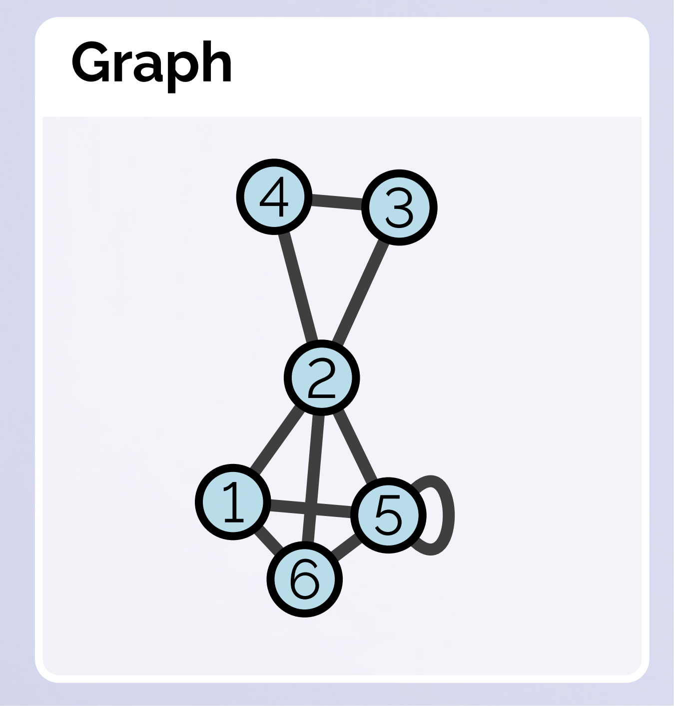
CLUSTERING COEFFICIENT

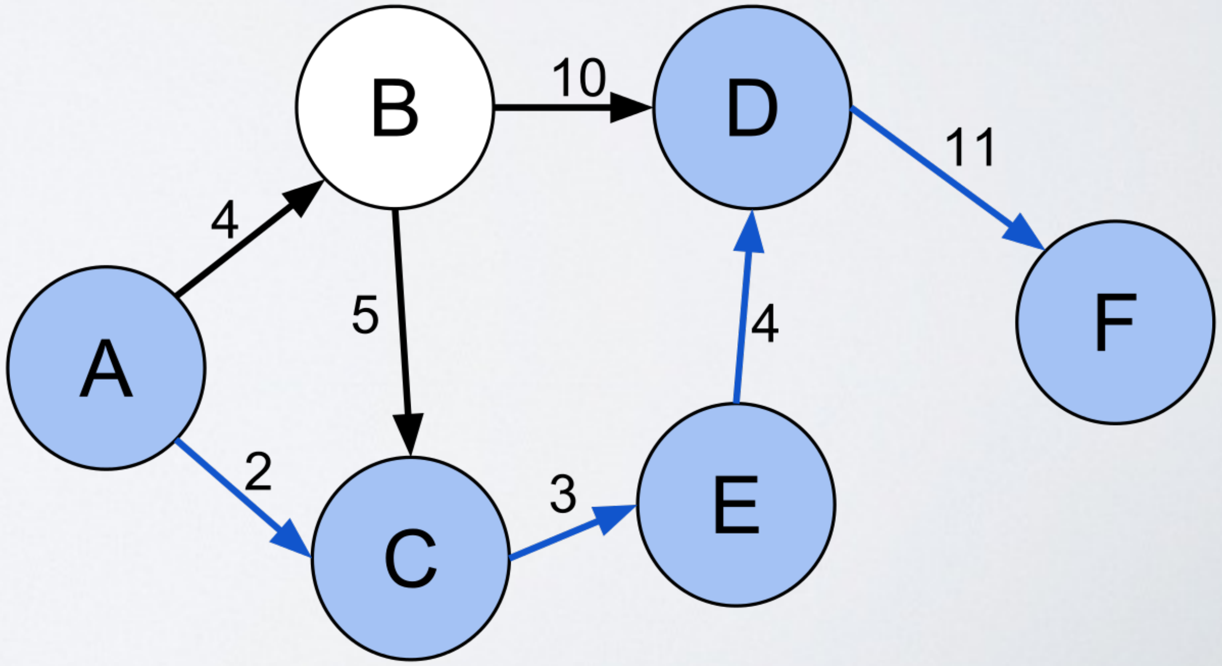
- Global CC:
  - In random networks, GCC = density
    - =>very small for large graphs
  - Facebook ego-networks: 0.6
  - Twitter lists: 0.56
  - California Road networks: 0.04

PATH RELATED SCORES

Paths - Walks - Distance

Walk: Sequences of adjacent edges or nodes (e.g., **1.2.1.6.5** is a valid walk)
Path: a walk in which each node is distinct.
Path length: number of edges encountered in a path
Weighted Path length: Sum of the weights of edges on a path
Shortest path: The shortest path between nodes u, v is a path of minimal *path length*. Often it is not unique.
Weighted Shortest path: path of minimal *weighted path length*.
$\ell_{u,v}$**: Distance**: The distance between nodes u, v is the length of the shortest path





PATH RELATED SCORES

Network descriptors 2 - Paths

$\ell_{\max}$ **Diameter**: maximum *distance* between any pair of nodes.

$\langle \ell \rangle$ **Average distance**:

$$\langle \ell \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} d_{ij}$$

AVERAGE PATH LENGTH

- The famous 6 degrees of separation (Milgram experiment)
  - (More on that next slide)
- Not too sensible to noise
- Tells you if the network is “stretched” or “hairball” like

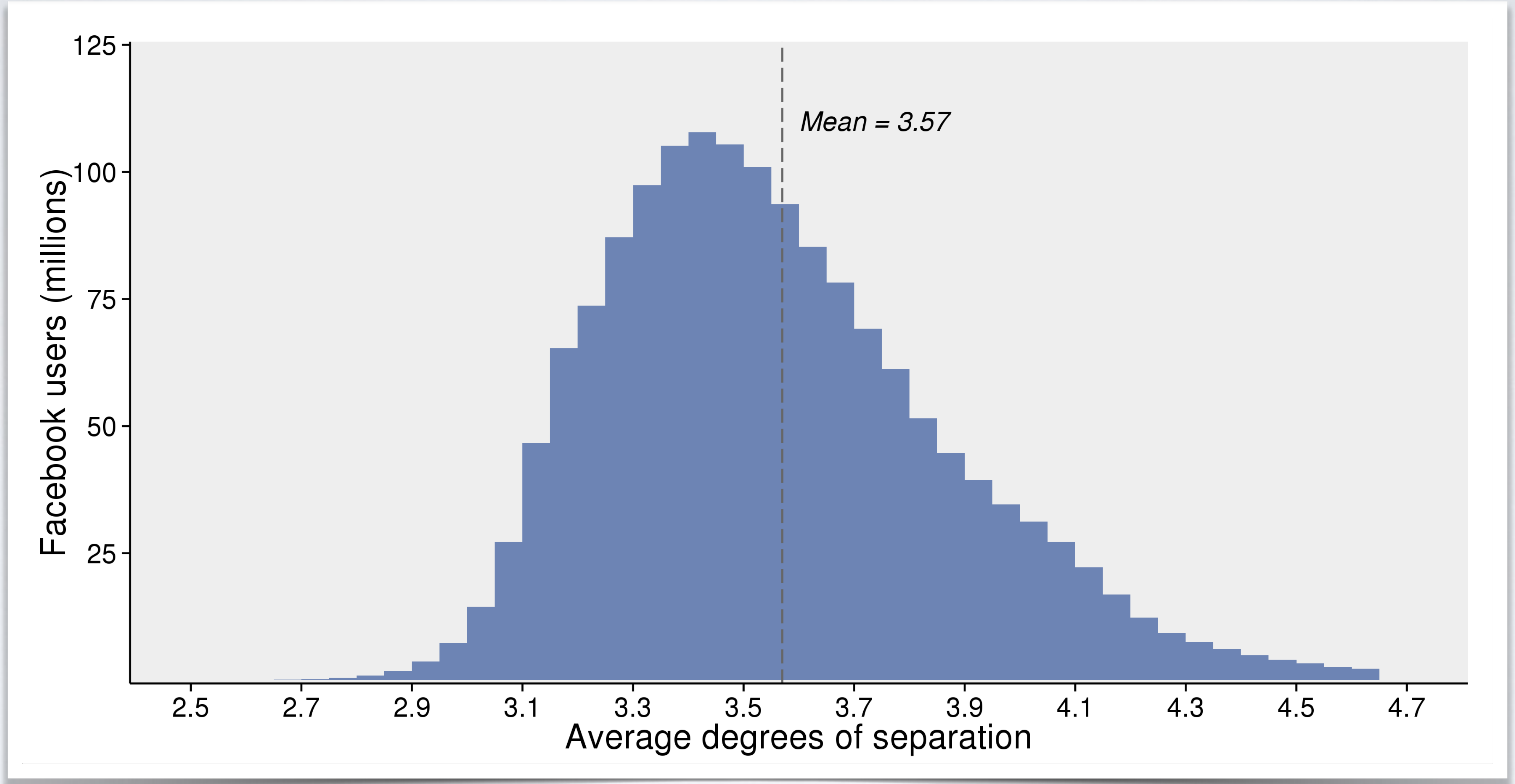
SIDE-STORY: MILGRAM EXPERIMENT

- Small world experiment (60's)
  - Give a (physical) mail to random people
  - Ask them to send to someone they don't know
    - They know his city, job
  - They send to their most relevant contact
- Results: In average, 6 hops to arrive

SIDE-STORY: MILGRAM EXPERIMENT

- Many criticism on the experiment itself:
  - Some mails did not arrive
  - Small sample
  - …
- Checked on “real” complete graphs (giant component):
  - MSN messenger
  - Facebook
  - The world wide web
  - …

SIDE-STORY: MILGRAM EXPERIMENT



Facebook

SMALL WORLD

Small World Network

A network is said to have the **small world** property when it has some structural properties. The notion is not quantitatively defined, but two properties are required:

- Average distance must be short, i.e., $\langle \ell \rangle \approx \log(N)$
- Clustering coefficient must be high, i.e., much larger than in a random network , e.g., $C^g \gg d$, with d the network density

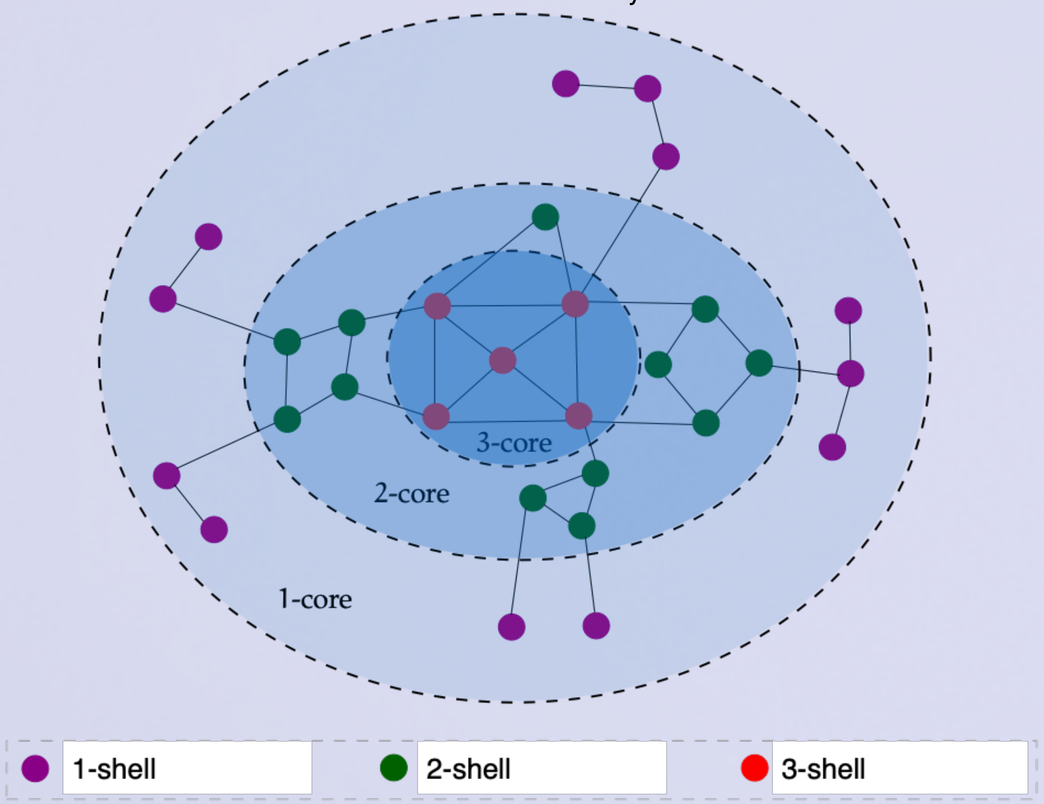
CORE-PERIPHERY : CORENESS

Goal: To identify dense cores of high degree nodes in networks

Cores and Shells

Many real networks are known to have a **core-periphery** structure, i.e., there is a densely connected core at its center and a more peripheral zone in which nodes are loosely connected between them and to the core.

k-core: The k-core (core of order k) of $G(V, E)$ is the largest subgraph $H(C)$ such as all nodes have at least a degree k, i.e., $\forall u \in C, k_u^H \leq k$, with$k_u^H$ the degree of node u in subgraph H.
coreness: A vertex u has coreness k if it belongs to the k-core but not to the $k+1$-core.
c-shell: all vertices whose coreness is exactly c.



- A k-core of G can be obtained by recursively removing all the vertices of degree less than k, until all vertices in the remaining graph have at least degree k.

GRAPHS AS MATRICES

Matrices in short

Matrices are mathematical objects that can be thought as *tables* of numbers. The size of a matrix is expressed as $m \times n$, for a matrix with m rows and n columns. **The order (row/column) is important**.
M_{ij} is a notation representing the element on **row** m and **column** j.

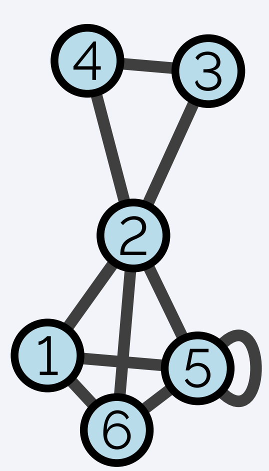
ADJACENCY MATRIX

A - Adjacency matrix

The most natural way to represent a graph as a matrix is called the Adjacency matrix A. It is defined as a square matrix, such as the number of rows (and the number of columns) is equal to the number of nodes N in the graph. Nodes of the graph are numbered from 1 to N, and there is an edge between nodes i and j if the corresponding position of the matrix A_{ij} is not 0.

- A value on the diagonal means that the corresponding node has a **self-loop**
- the graph is **undirected**, the matrix is **symmetric**: $A_{ij} = A_{ji}$ for any i, j.
- In an **unweighted** network, and edge is represented by the value 1.
- In a **weighted** network, the value A_{ij} represents the **weight** of the edge (i, j)

Graph



A - Adjacency Mat.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

ADJACENCY MATRIX

Typical operations on A

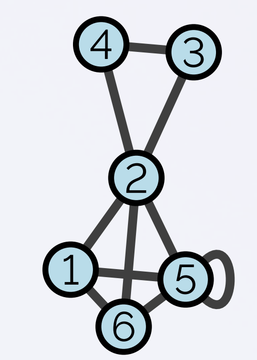
Some operations on Adjacency matrices have straightforward interpretations and are frequently used

---

Multiplying A by **itself** allows to know the number of walks of a given length that exist between any pair of nodes: A^2_{ij} corresponds to the number of walks of length 2 from node i to node j, A^3_{ij} to the number of walks of length 3, etc.

Multiplying A by a **column vector** W of length $1 \times N$ can be thought as setting the i th value of the vector to the ith node, and each node *sending* its value to its neighbors (for undirected graphs). The result is a column vector with N elements, the ith element corresponding to the sum of the values of its neighbors in W. This is convenient when working with **random walks** or **diffusion** phenomenon.

Graph



A - Adjacency Mat.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

A^2

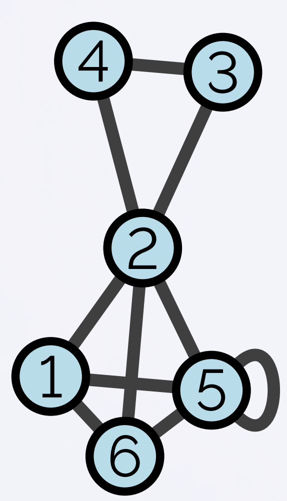
$$\begin{pmatrix} 3 & 2 & 1 & 1 & 3 & 2 \\ 2 & 5 & 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 \\ 3 & 3 & 1 & 1 & 4 & 3 \\ 2 & 2 & 1 & 1 & 3 & 3 \end{pmatrix}$$

RANDOM WALK MATRIX

Random Walk matrix

Another useful matrix of a graph is the **Random Walk Transition Matrix** R. It is the column normalized version of the adjacency matrix. R_{ij} can be understood as the probability for a random walker located on node i to move to j.

Graph



A - Adjacency Mat.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Random W. mat.

$$\begin{pmatrix} 0 & \frac{1}{5} & 0 & 0 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{3} \\ 0 & \frac{1}{5} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{5} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{5} & 0 & 0 & \frac{1}{4} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{5} & 0 & 0 & \frac{1}{4} & 0 \end{pmatrix}$$

EXEMPLE OF GRAPH ANALYSIS

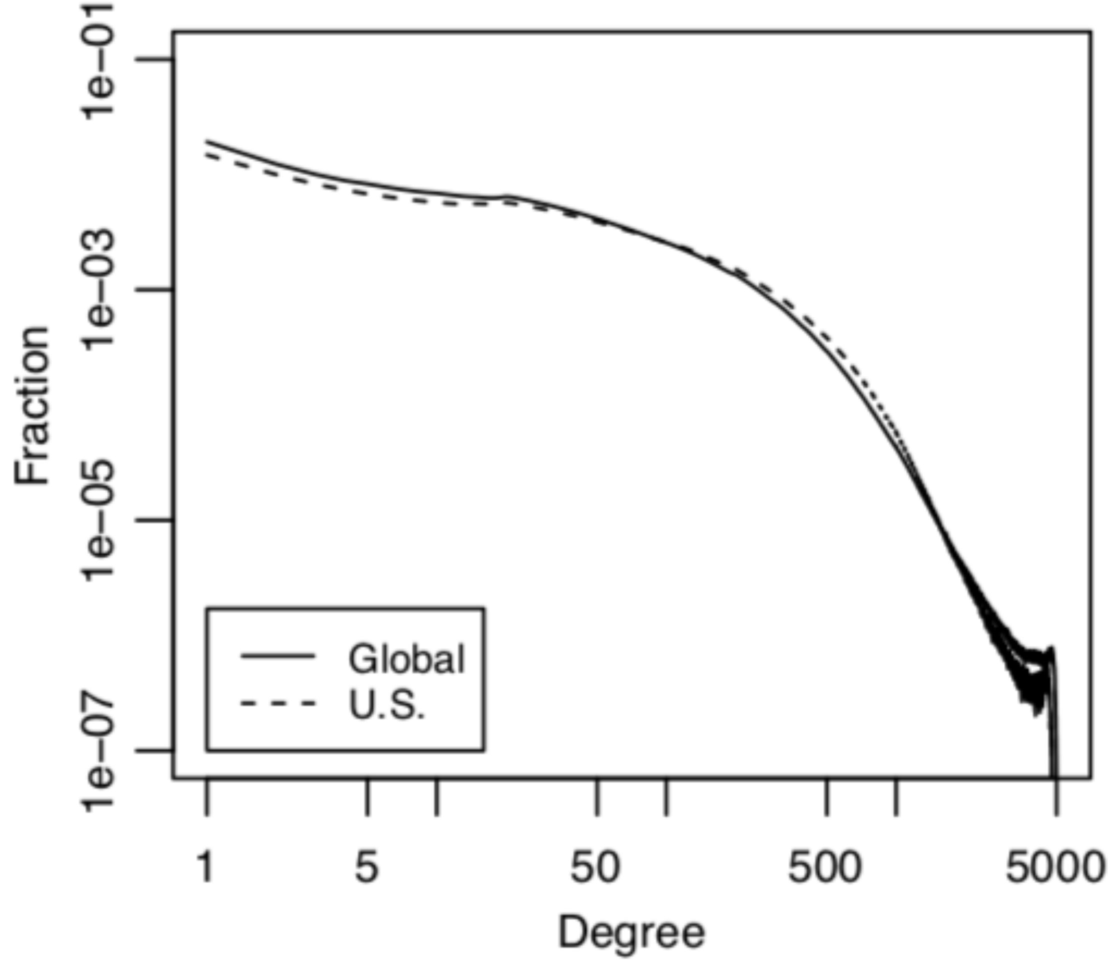
- Source: [The Anatomy of the Facebook Social Graph, Ugander et al. 2011]

- The Facebook friendship network in 2011

EXEMPLE OF GRAPH ANALYSIS

- 721M users (nodes) (active in the last 28 days)
- 68B edges
- Average degree: 190 (average # friends)
- Median degree: 99
- Connected component: 99.91%

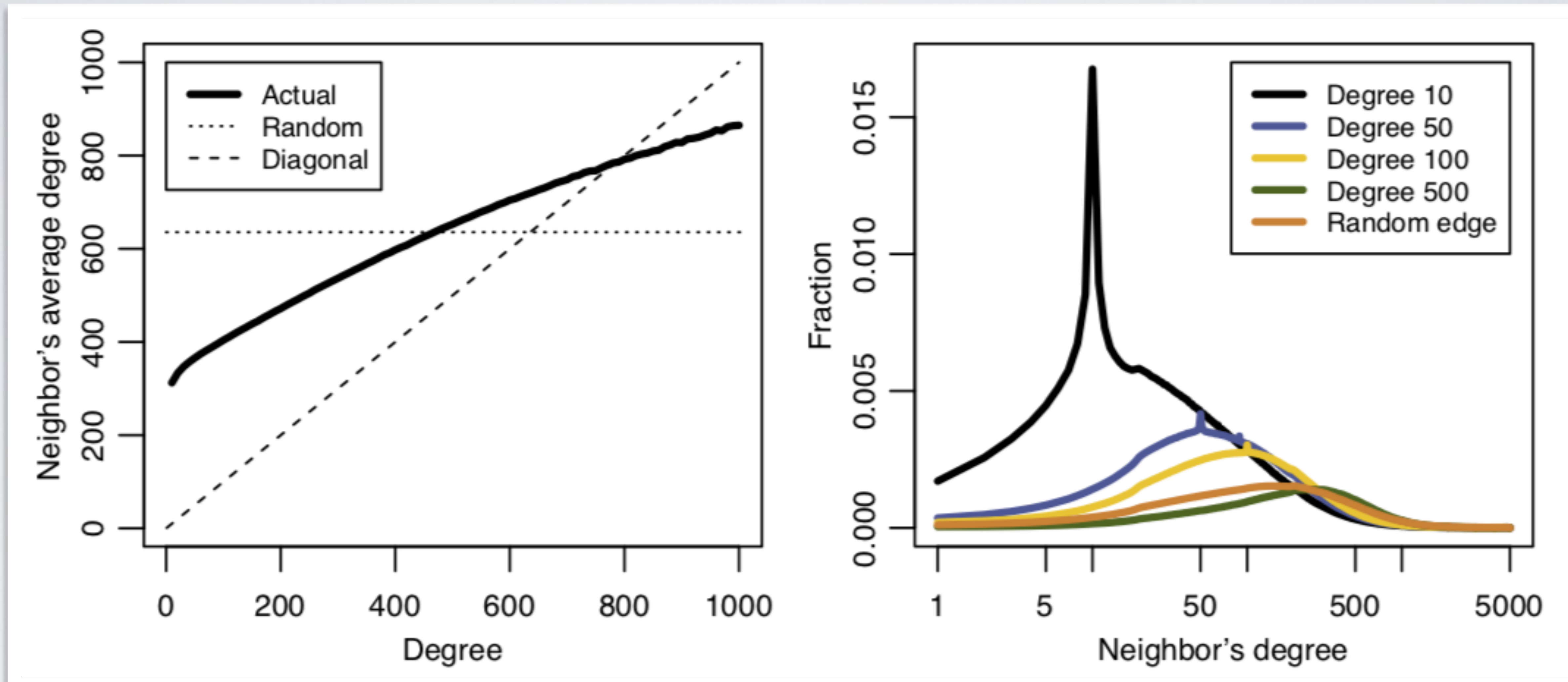
EXEMPLE OF GRAPH ANALYSIS



(a)

Degree distribution

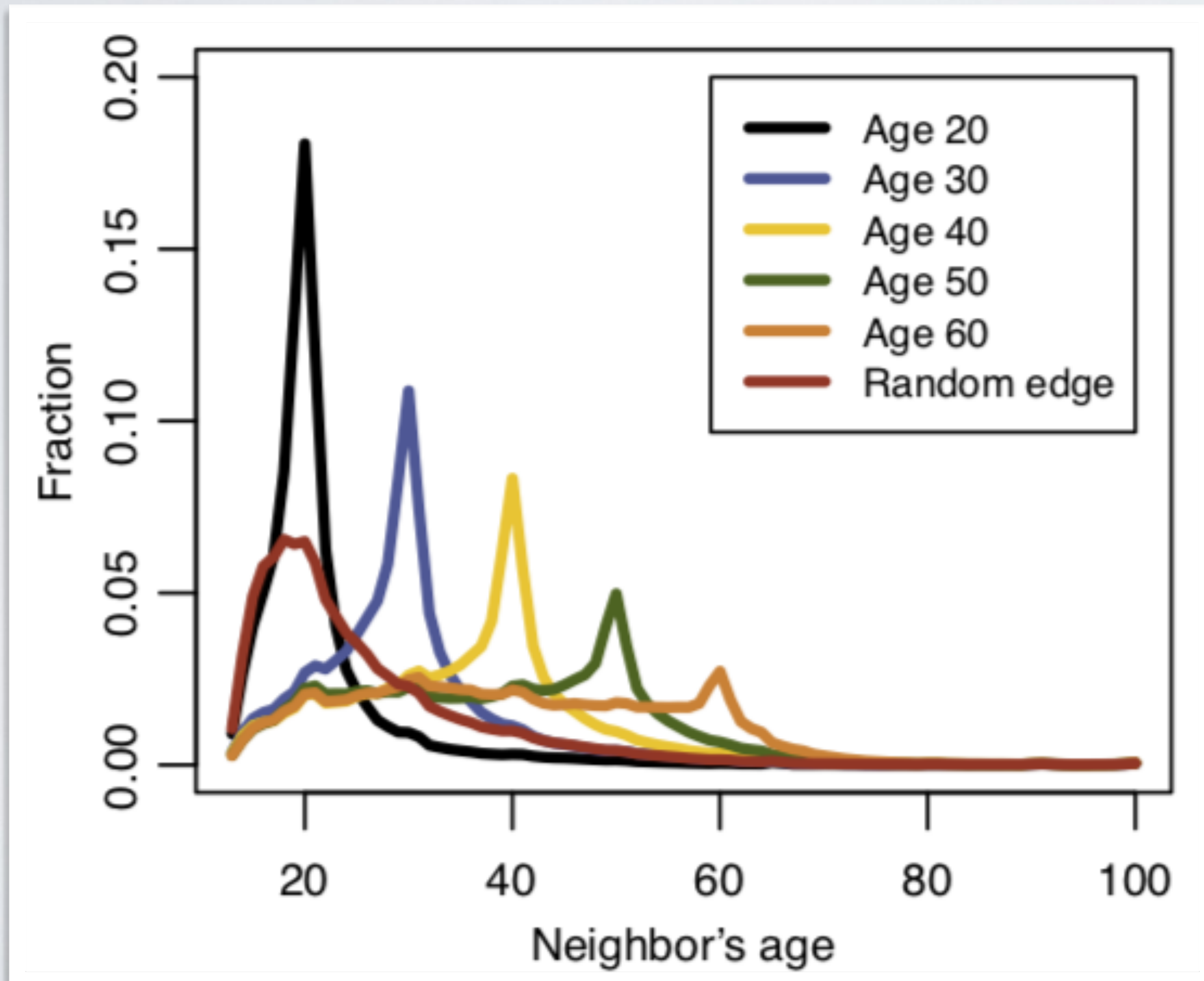
EXEMPLE OF GRAPH ANALYSIS



My friends have more
Friends than me!

Many of my friends have the
Same # of friends than me!

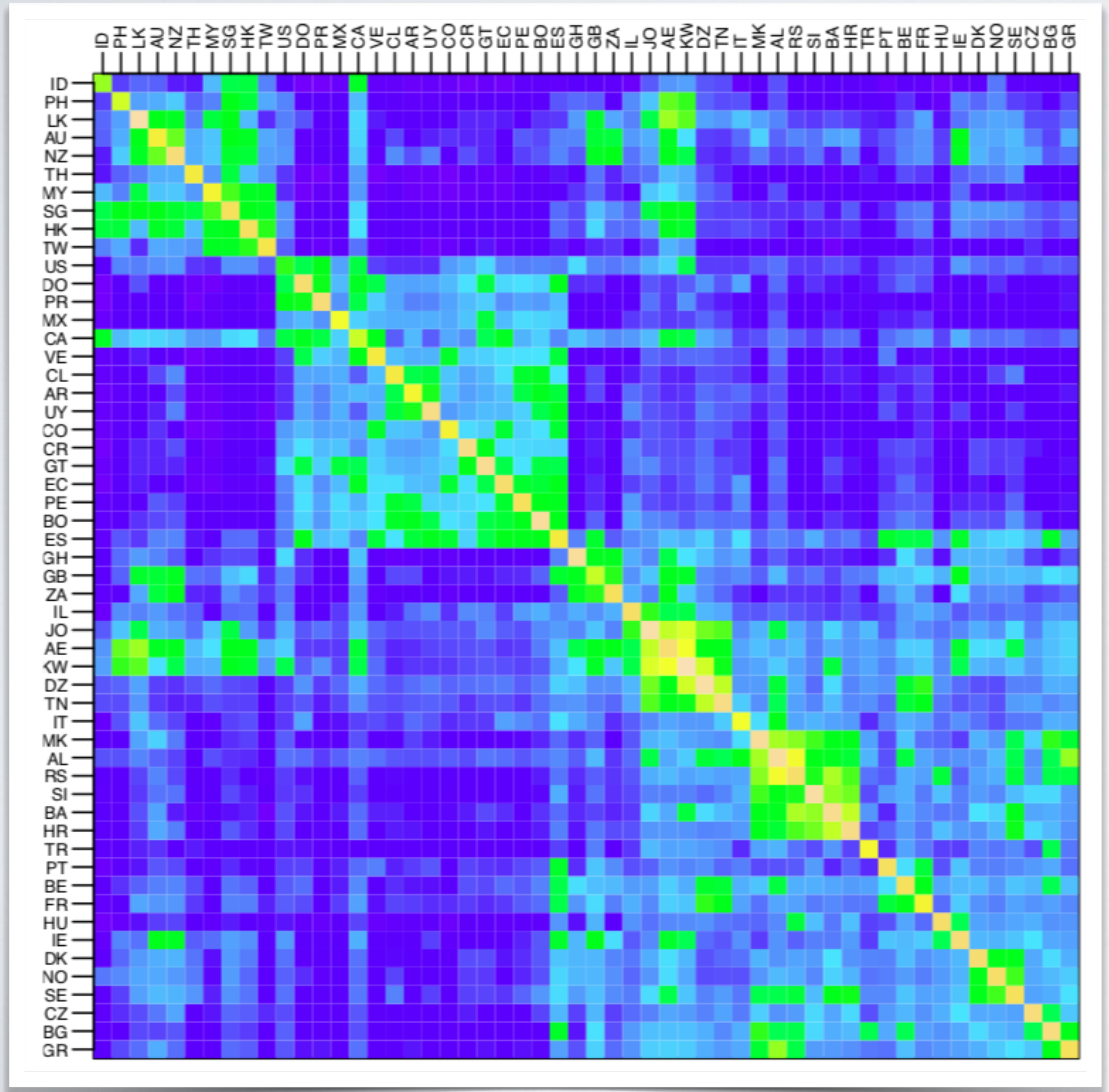
EXEMPLE OF GRAPH ANALYSIS



Age homophily

(More next class)

EXEMPLE OF GRAPH ANALYSIS



Country similarity

84.2% percent of edges are within countries

(More in the community detection class)