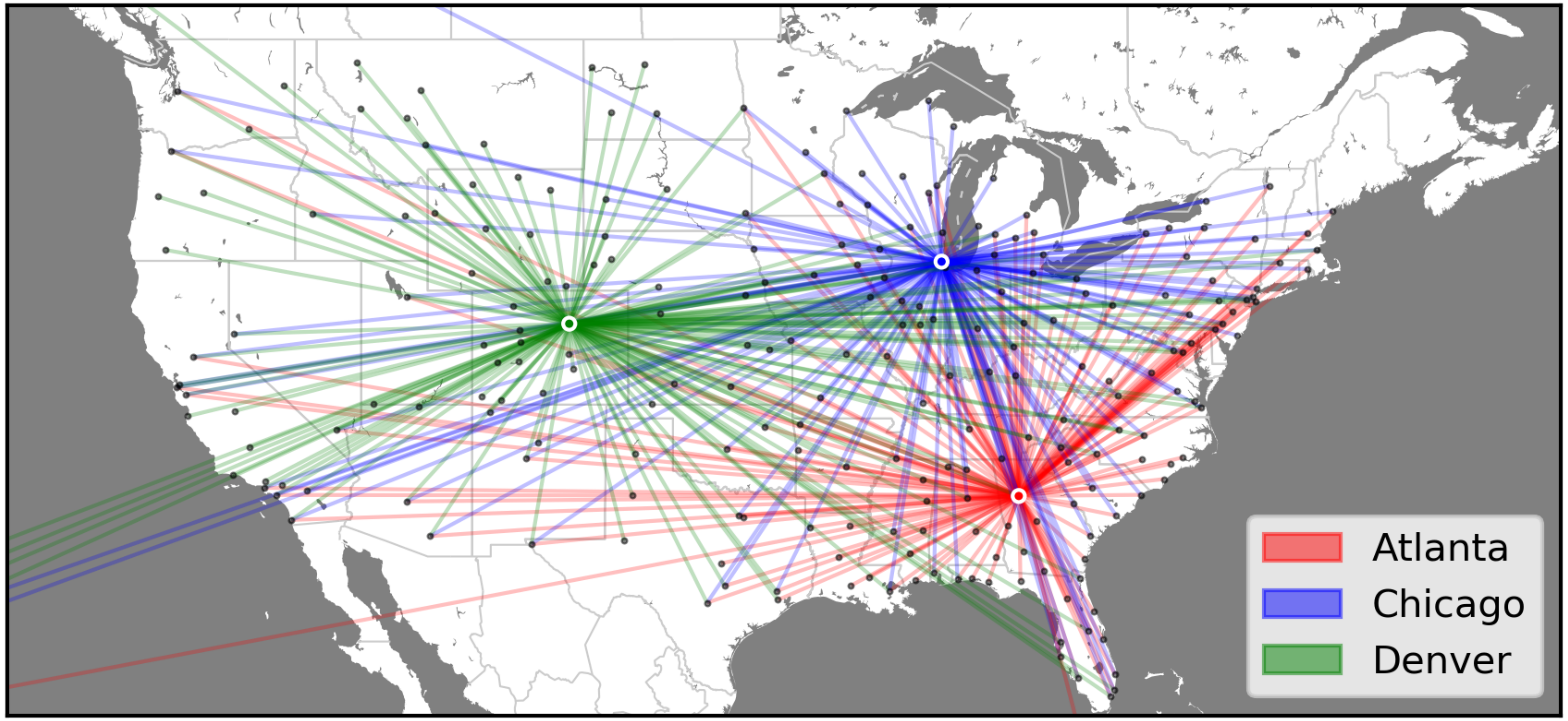
Chapter 3:
Hubs

A First Course in Network Science

Outline

- Centrality measures
- Centrality distributions
- The friendship paradox
- Ultra-small worlds
- Robustness
- Core decomposition

Real networks are heterogeneous



Some nodes (and links) are much more important (**central**) than others!

Centrality measures

- **Centrality**: measure of importance of a node
- **Measures**:

1. Degree
2. Closeness
3. Betweenness

Degree

- **Degree of a node:** number of neighbors of the node

$$k_i = \text{number of neighbors of node } i$$

- High-degree nodes are called **hubs**

- **Average degree of the network:**

$$\langle k \rangle = \frac{\sum_i k_i}{N} = \frac{2L}{N}$$

```
G.degree(2) # returns the degree of node 2
G.degree()  # dict with the degree of all nodes of G
```

Closeness

Idea: a node is the more central the *closer* it is to the other nodes, on average

$$g_i = \frac{1}{\sum_{j \neq i} \ell_{ij}}$$

where ℓ_{ij} is the distance between nodes *i* and *j*

```
nx.closeness_centrality(G, node) # closeness centrality
                                 # of node
```

Betweenness

Idea: a node is the more central the *more often it is crossed by paths*

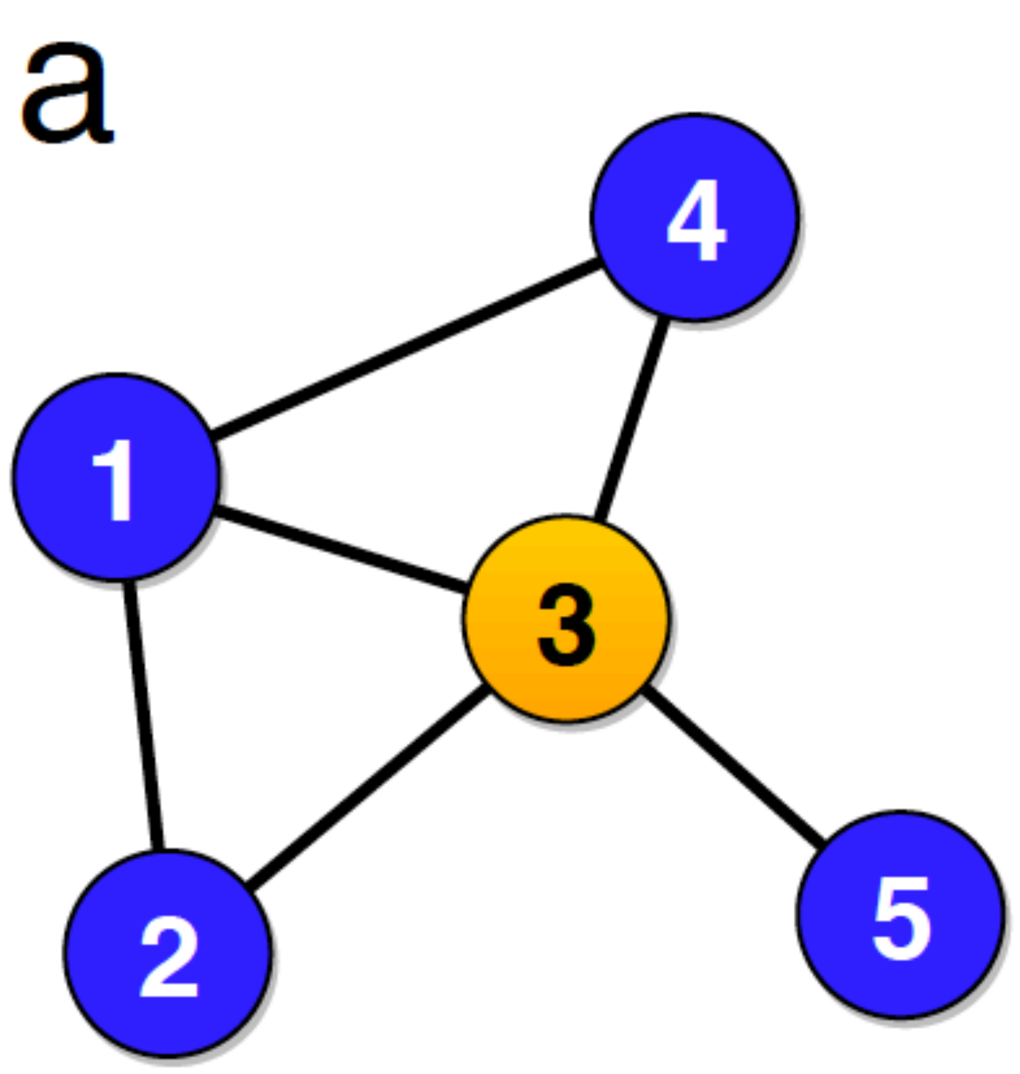
$$b_i = \sum_{h \neq j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}}$$

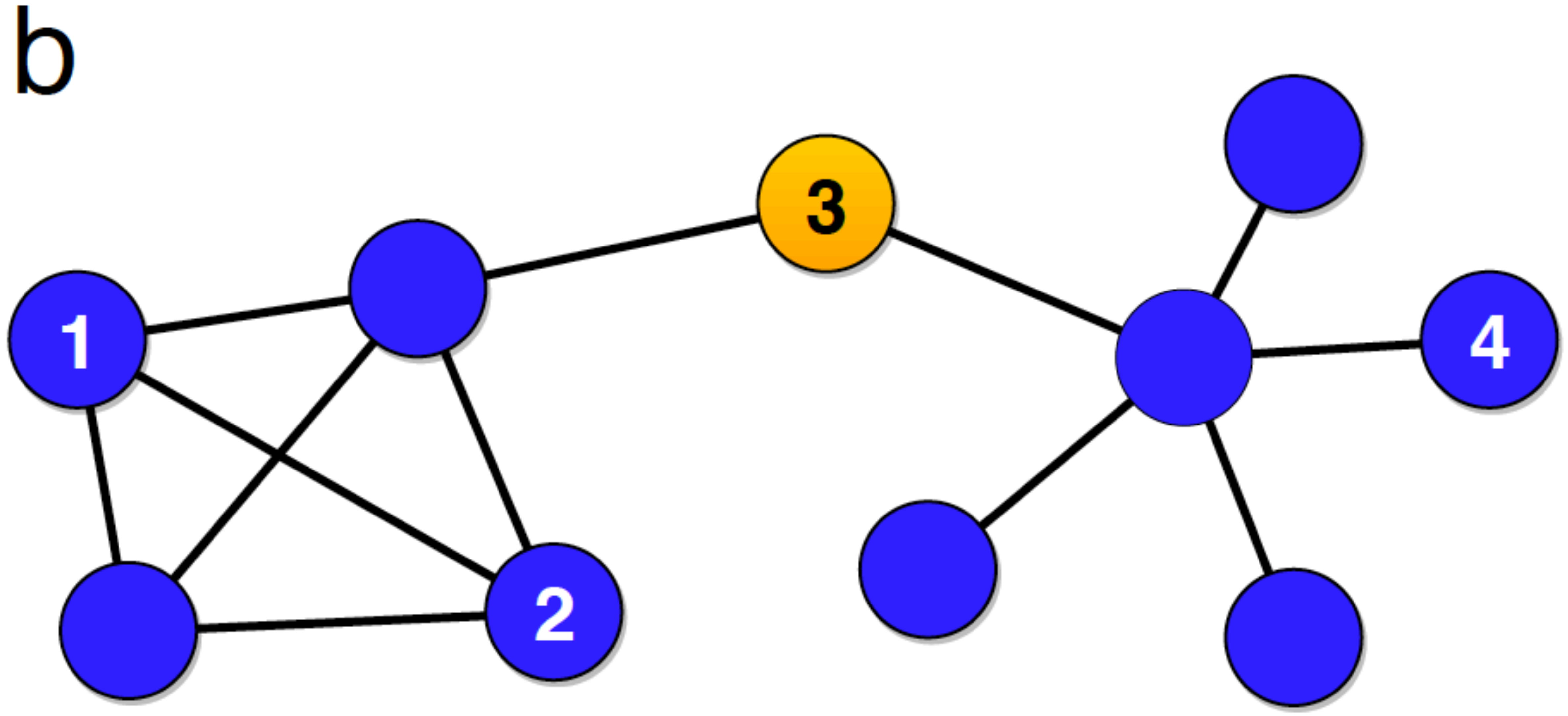
σ_{hj} = number of shortest paths from h to j

$\sigma_{hj}(i)$ = number of shortest paths from h to j running through i

Betweenness

Hubs usually have high betweenness, but there can be nodes with high betweenness **that are not hubs**





Betweenness

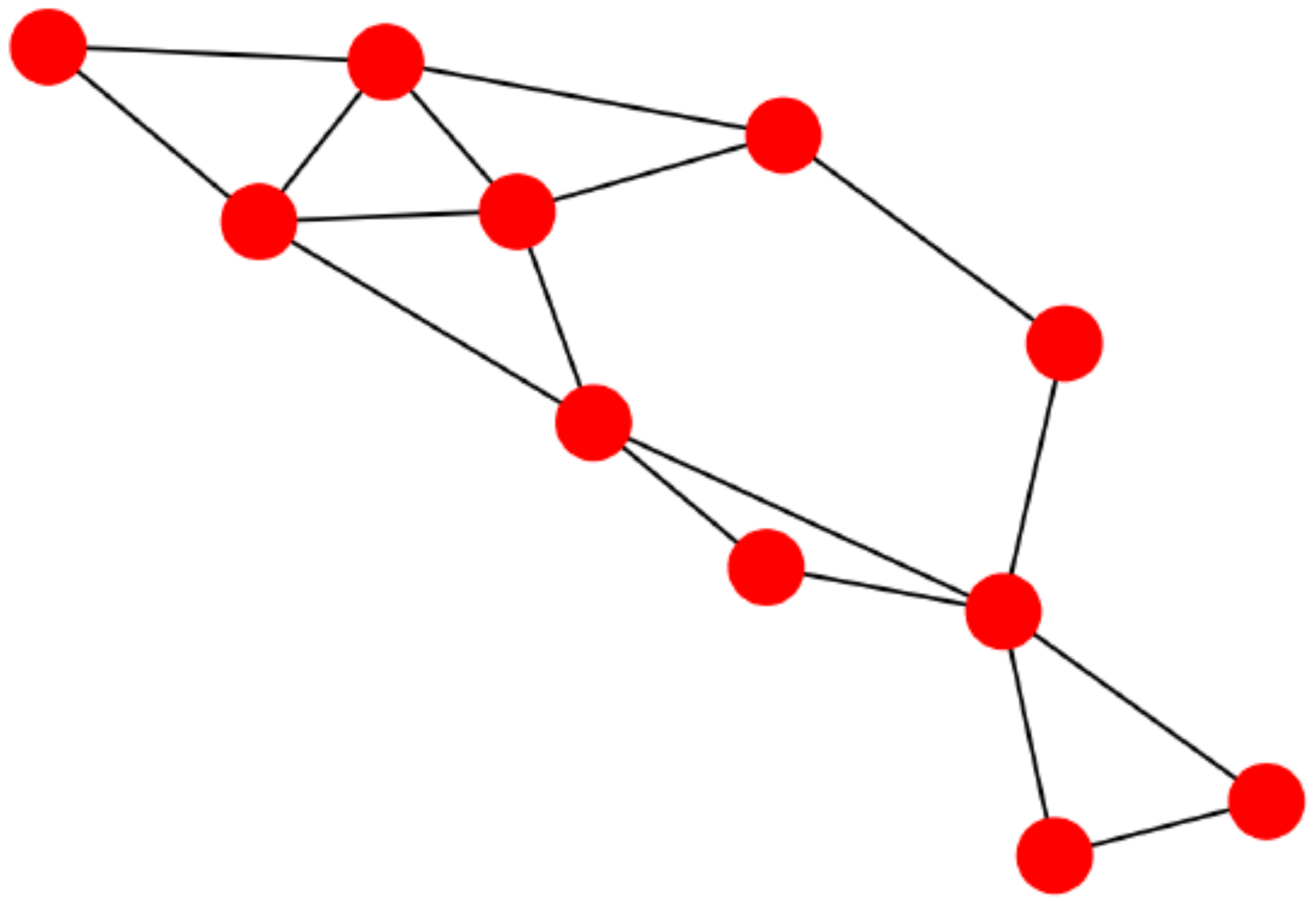
- Betweenness can be easily extended to links
- **Link betweenness**: fraction of shortest paths among all possible node pairs that pass through the link

```
nx.betweenness_centrality(G)       # dict nodes ->
                                   # betweenness centrality
nx.edge_betweenness_centrality(G)  # dict links ->
                                   # betweenness centrality
```

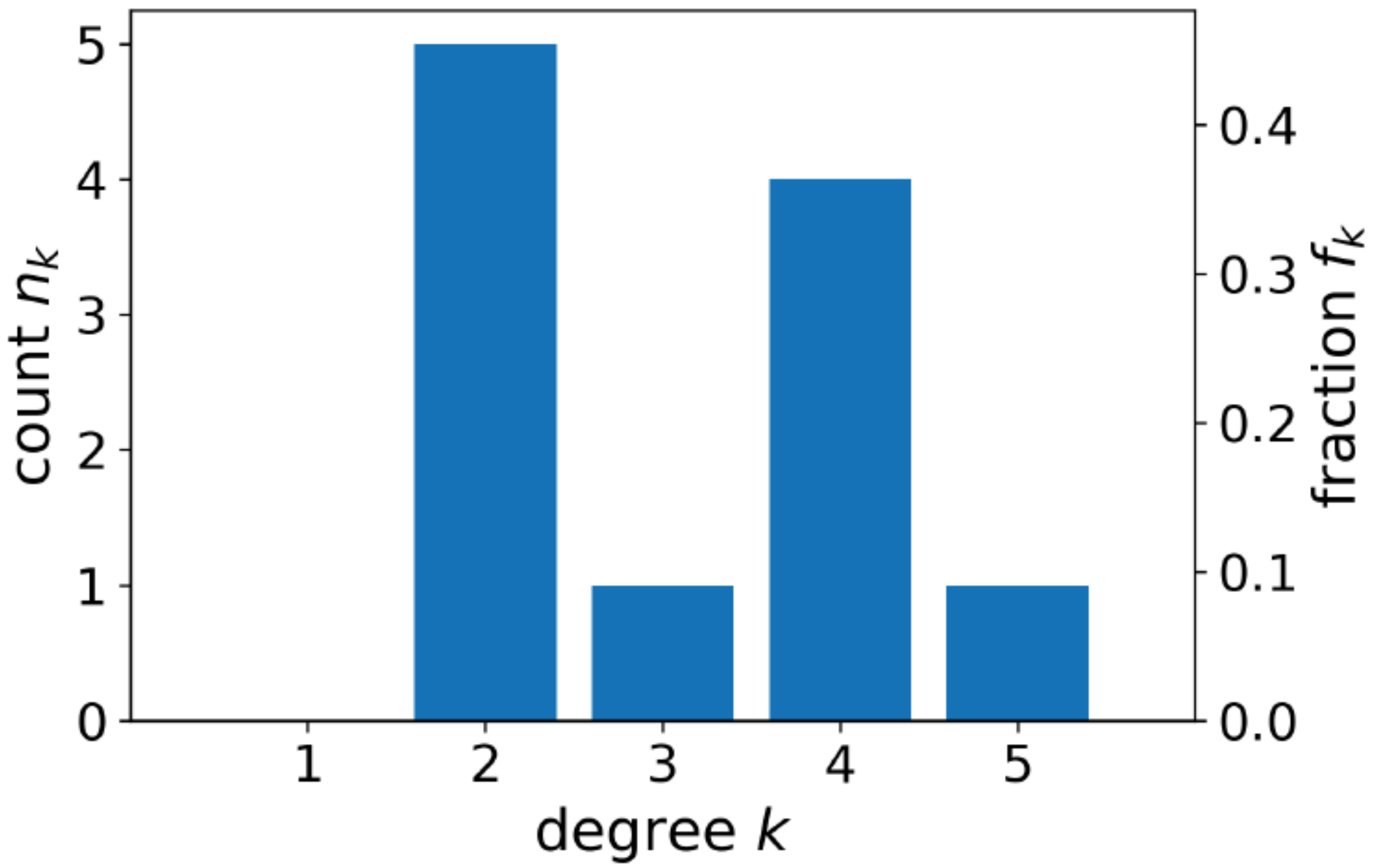
Centrality distributions

- On small networks it makes sense to ask which nodes or links are most important
- On large networks **it does not**
- **Solution:** statistical approach
- Instead of focusing on individual nodes and links, we consider **classes** of nodes and links with similar properties

Centrality distributions

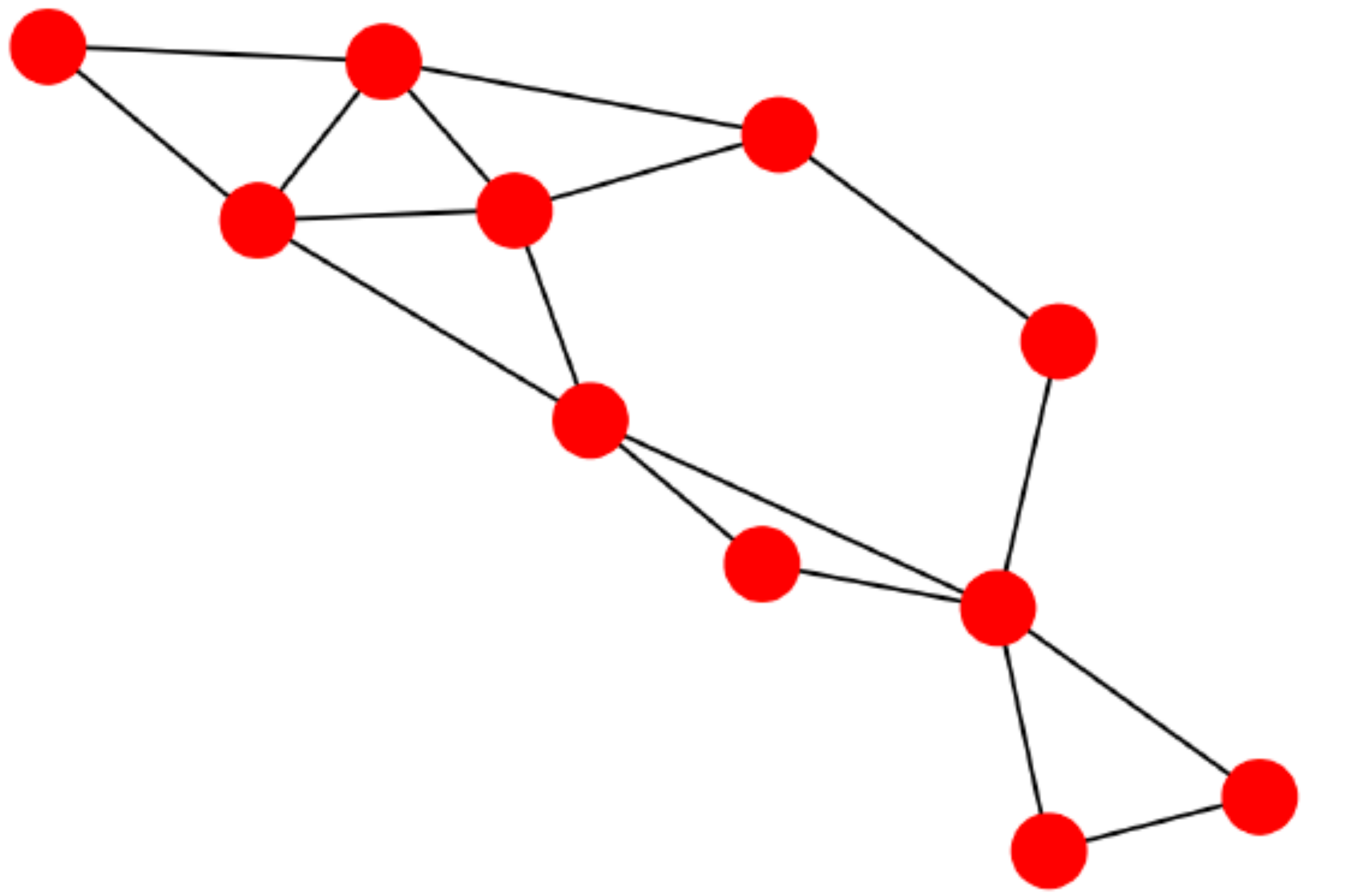


Histogram

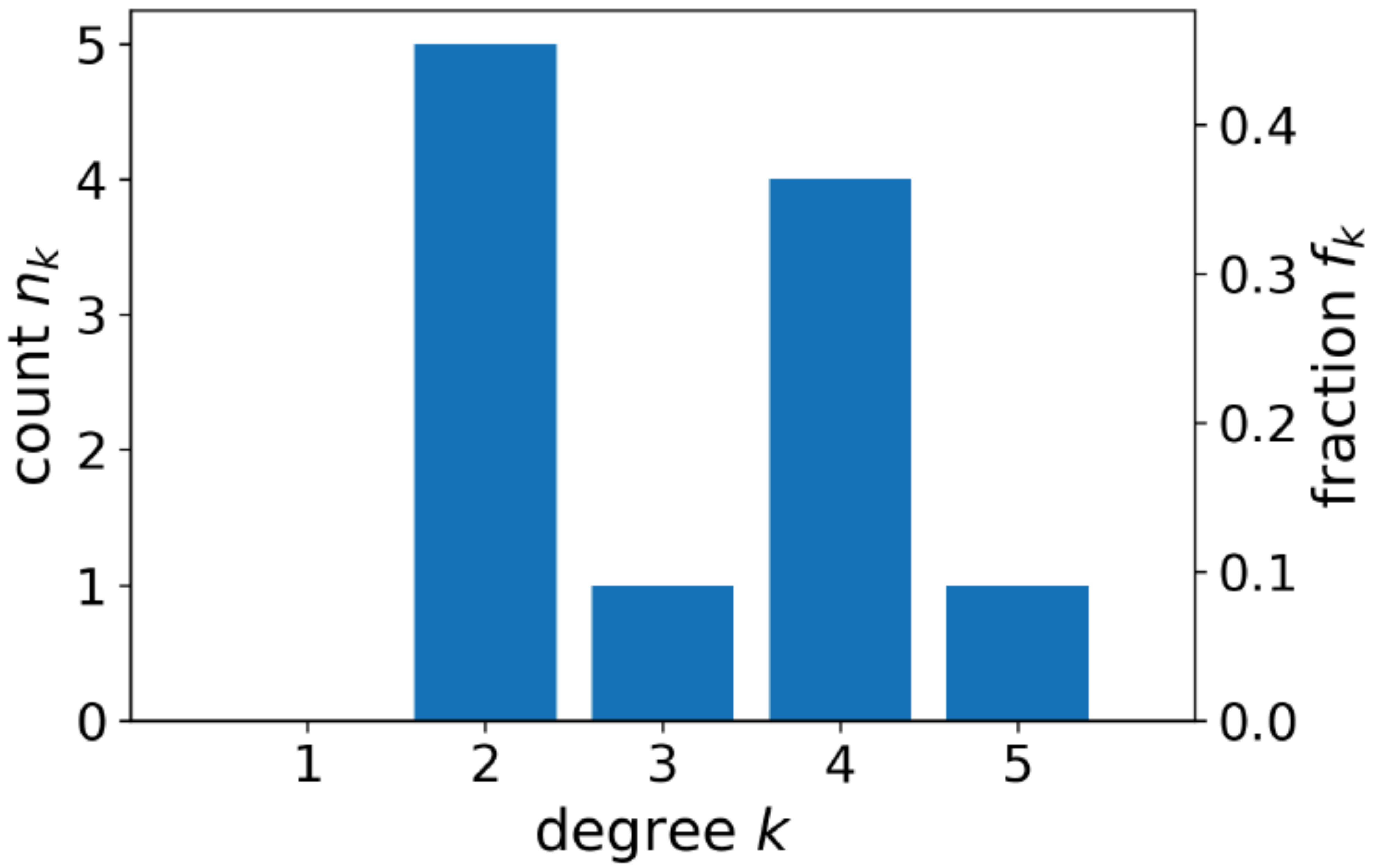


- n_k = number of nodes with degree k
- $f_k = \frac{n_k}{N}$ = frequency of degree k

Centrality distributions



Histogram



- When $N \to \infty$, f_k becomes the **probability** p_k of having degree k
- p_k versus k is the **probability distribution** of node degree

Cumulative distributions

- If the variable is *not integer* (e.g., betweenness), the range of the variable is divided into intervals (bins) and we count how many values fall in each interval

- **Cumulative distribution *P(x)*:** probability that the variable takes values *larger* than *x* as a function of *x*

- **How to compute it:** by summing the frequencies of the variable inside the intervals to the right of *x*

$$P(x) = \sum_{v \geq x} f_v$$

Logarithmic scale

- **Question**: how to plot a probability distribution if the variable spans a large range of values, from small to (very) large?

- **Answer**: use the **logarithmic** scale

- **How to do it**: report the logarithms of the values on the x- and y-axes

$$\log_{10} 10 = 1$$

$$\log_{10} 1{,}000 = \log_{10} 10^3 = 3$$

$$\log_{10} 1{,}000{,}000 = \log_{10} 10^6 = 6$$

Probability distributions

Discrete distributions

Integer variable X, population n

n_k = number of events with X=k

Probability that variable X takes value k

$$P(X = k) = \frac{n_k}{n} \longrightarrow \sum_k P(X = k) = 1$$

Probability distributions

Continuous distributions

Continuous variable X

Probability that variable takes value in the range [a, b]

$$Pr[a \leq X \leq b] = \int_a^b P(x)dx$$

Probability density

$$Pr[x_{min} \leq X \leq x_{max}] = \int_{x_{min}}^{x_{max}} P(x)dx = 1$$

Probability distributions

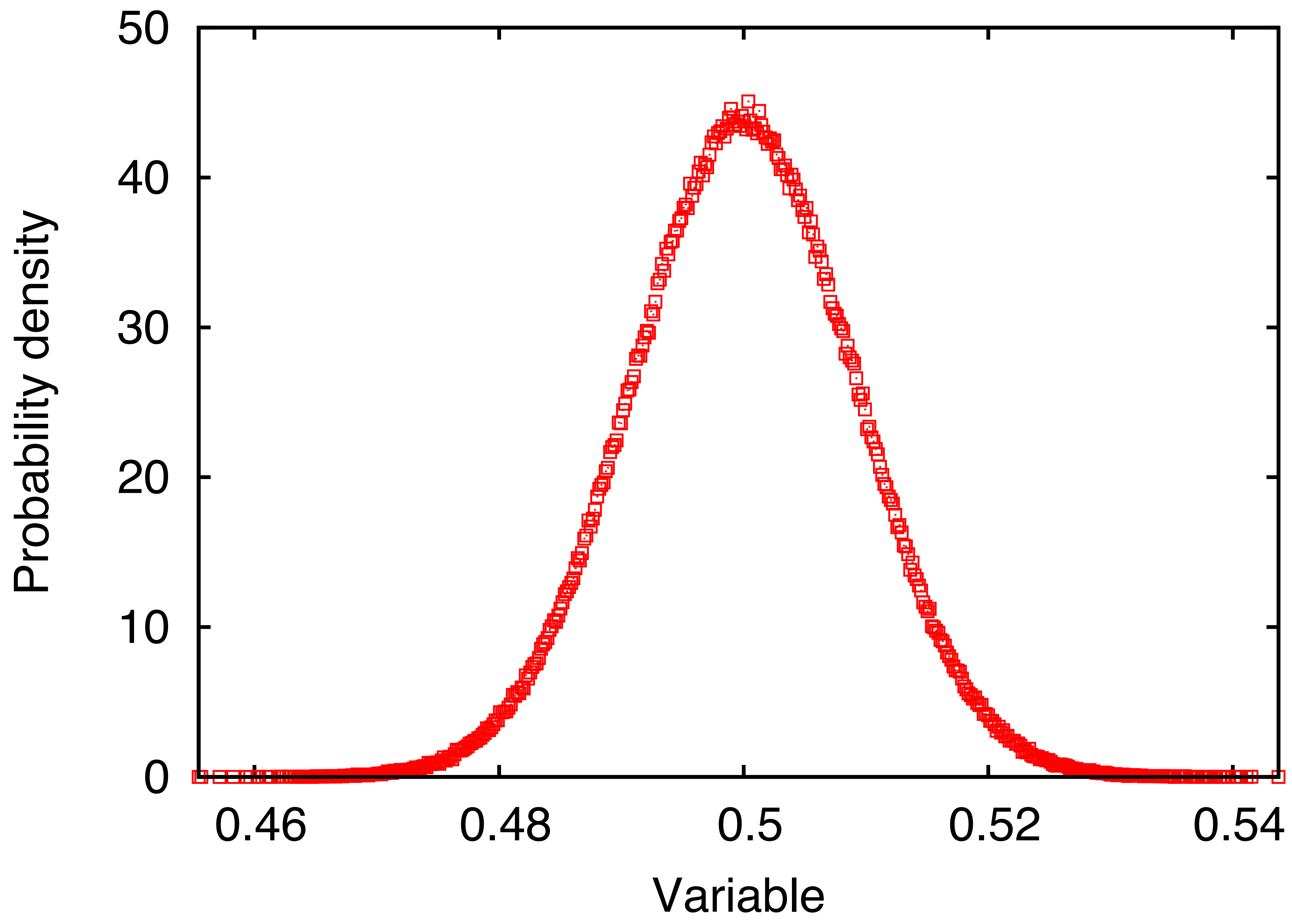
Gaussian (or **normal**) distribution: $P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

$\mu =$ mean, expectation

$\sigma =$ standard deviation

Applications: infinite!

1) Statistics of errors
2) Central limit theorem
3) Diffusion
4) (some) social statistics
5) Etc.



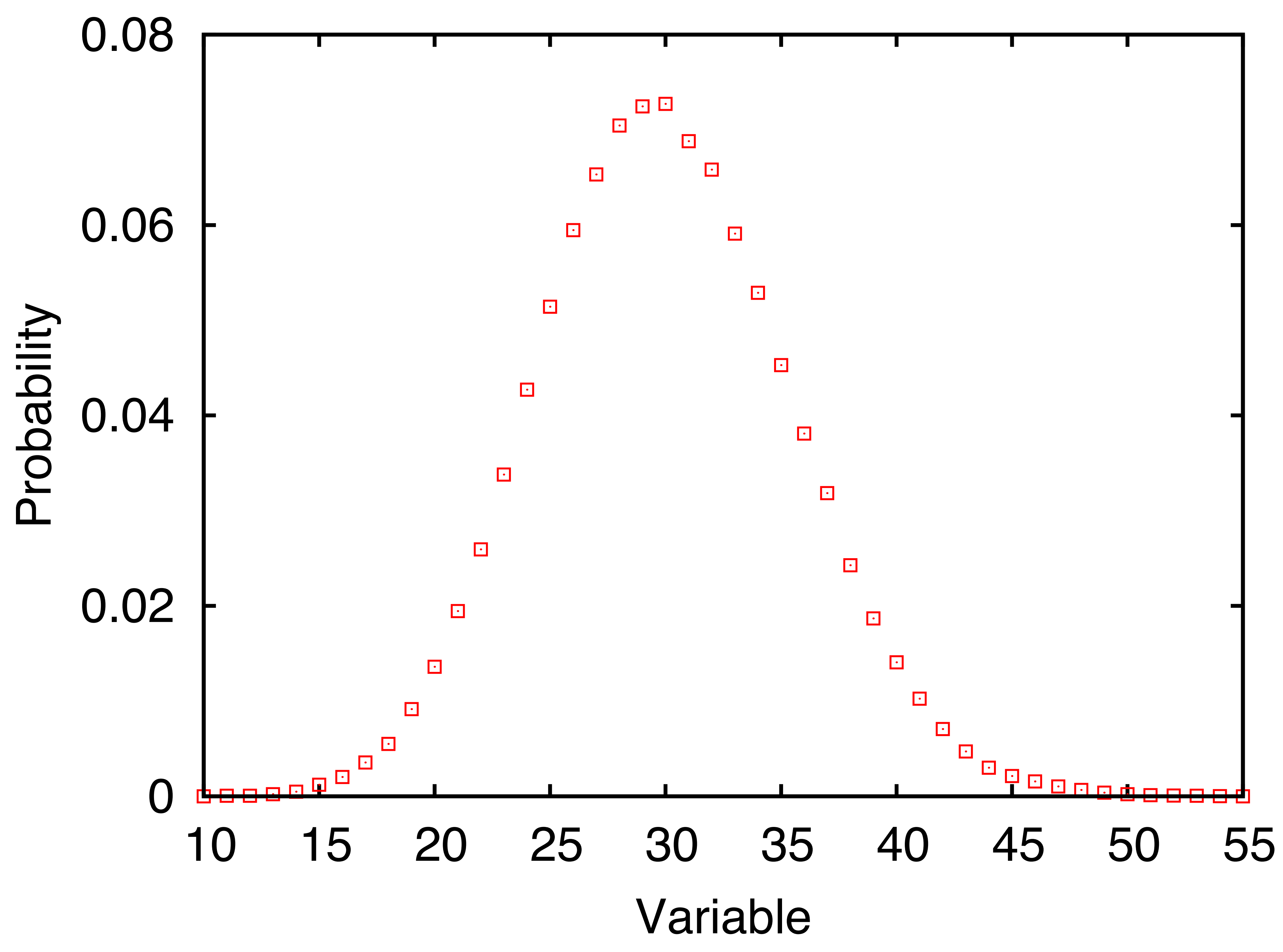
Probability distributions

Poissonian distribution: $P(X=k) = \dfrac{\lambda^k \, e^{-\lambda}}{k!}$

$\lambda =$ mean, expectation

Applications:

1) Poisson process
2) Radioactive decay (number of events)
3) DNA mutations
4) Birth-death processes
5) Number of goals in soccer games
6) Etc.



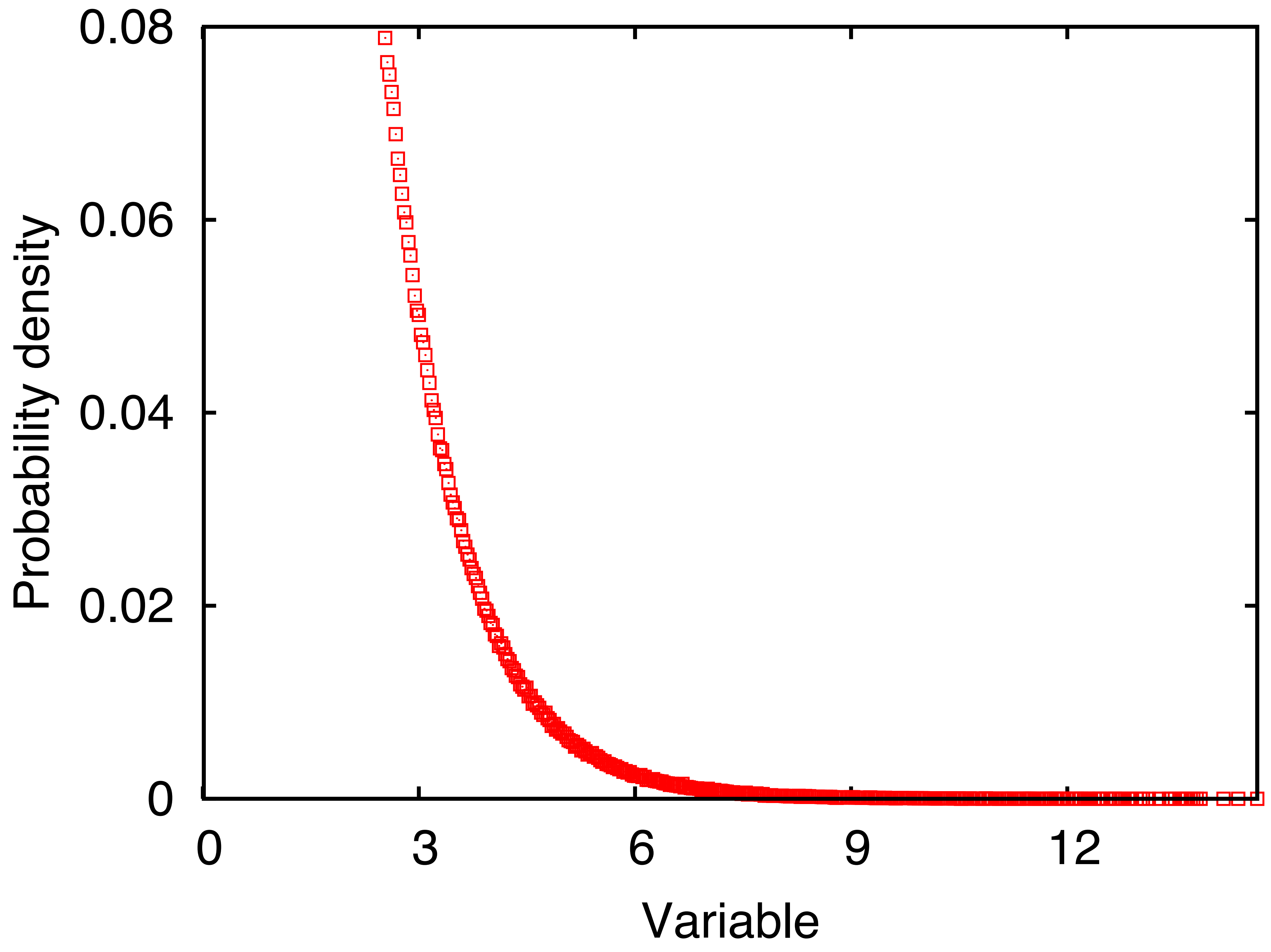
Probability distributions

Exponential distribution: $P(x) = \lambda\, e^{-\lambda x} \qquad (x \geq 0)$

$\lambda =$ rate parameter

Applications:

1) Inter-event times of Poisson process
2) Radioactive decay (time until decay)
3) Queuing theory
4) Physics: gas in uniform gravitational field
5) Etc.

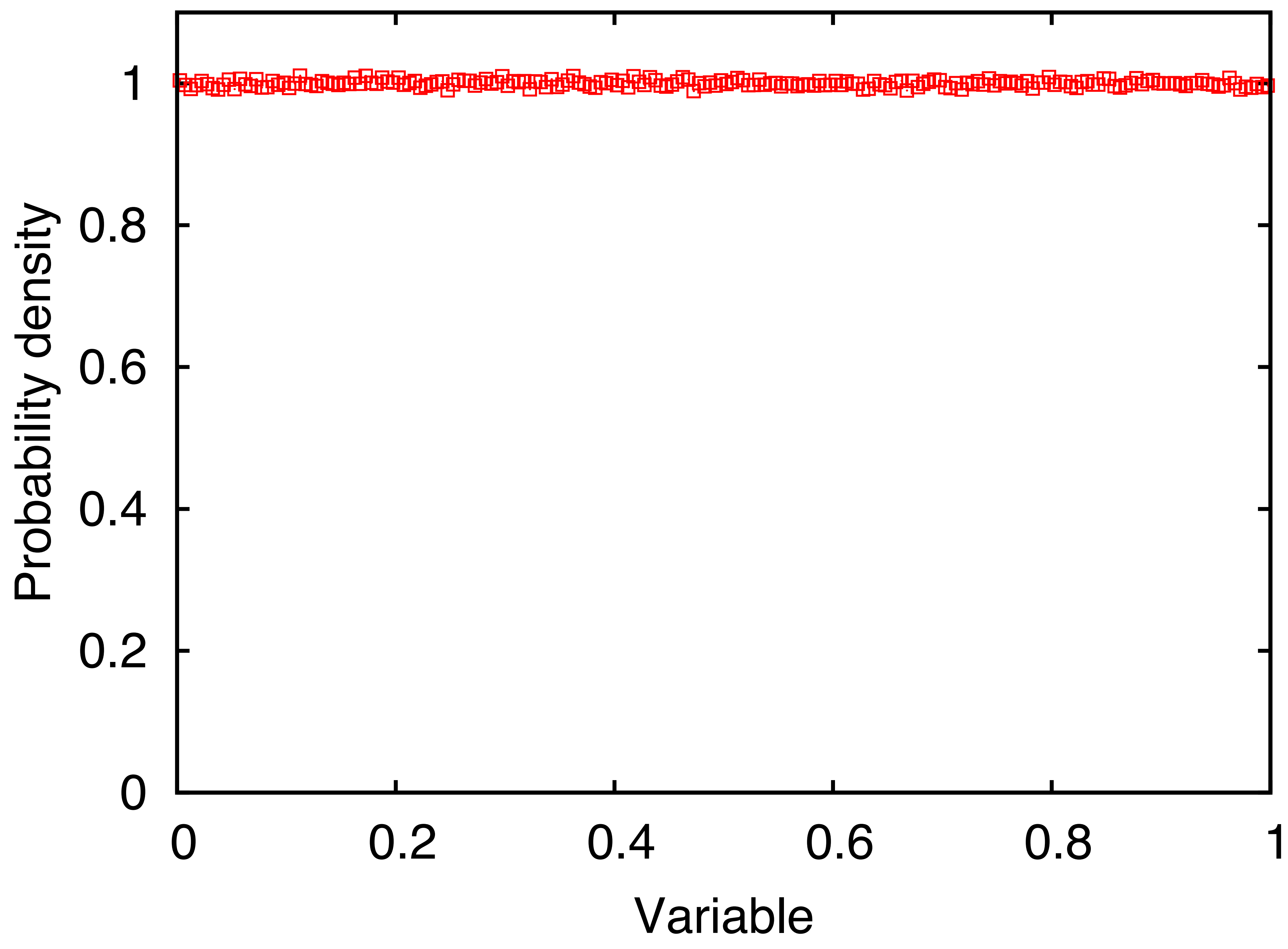


Probability distributions

Uniform distribution: $P(x) = \dfrac{1}{b-a} \qquad (x \in [a, b])$

Applications:

1) P-value distribution in statistics
2) Random number generators
3) Etc.



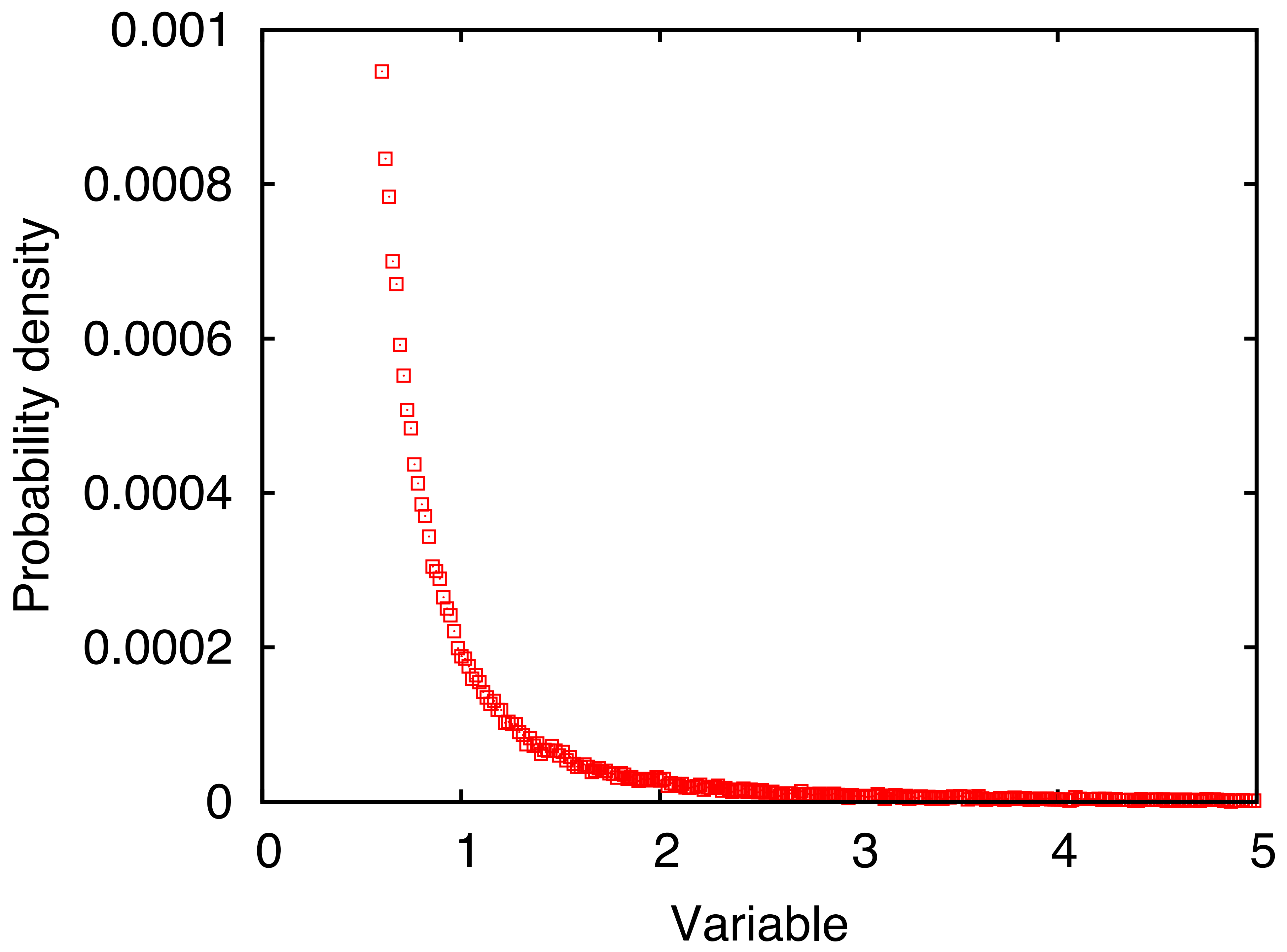
Probability distributions

Power law distribution: $P(x) = \frac{\alpha - 1}{x_{min}} \left(\frac{x}{x_{min}} \right)^{-\alpha} \quad (x \geq x_{min})$

$\alpha =$ exponent

Applications:

1) Critical phenomena
2) Wealth distribution
3) Earthquakes
4) Forest fires
5) Human dynamics
6) Network degree distributions
7) Hydrology: rainfalls
8) Etc.



Probability distributions

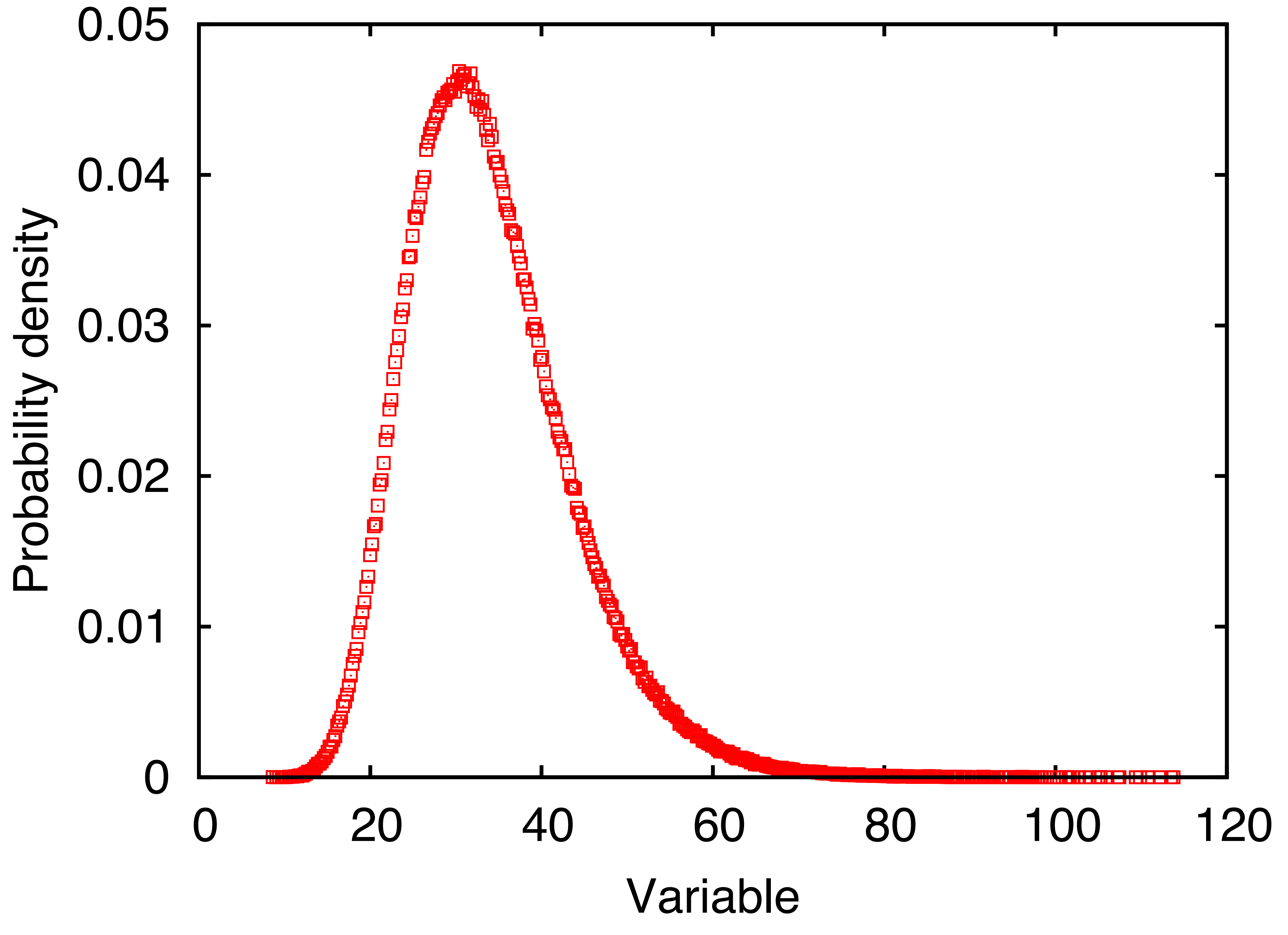
Lognormal distribution: $P(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\log x - \mu)^2}{2\sigma^2}} \qquad (x > 0)$

$\mu =$ location parameter

$\sigma =$ scale parameter

Applications:

1) Multiplicative processes
2) City size
3) Paper citations
4) Language size
5) Blood pressure
6) Reliability analysis
7) Etc.



Using the log-scale

The log-scale on the x-axis, the y-axis, or both, may facilitate the identification of a distribution

Gaussian

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

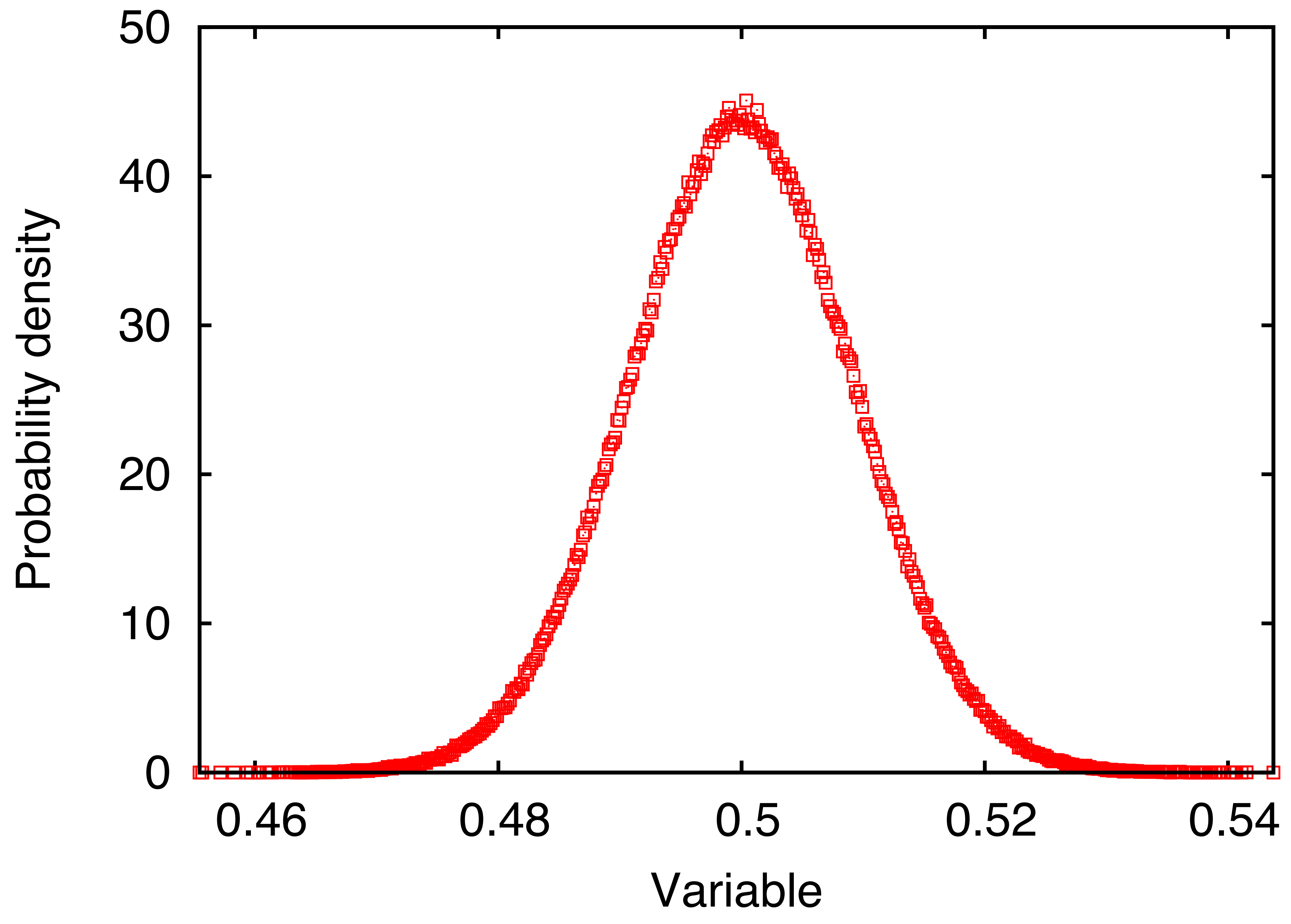
$$\color{red}{\log P(x)} = -\log(\sigma\sqrt{2\pi}) - \frac{(\color{red}{x} - \mu)^2}{2\sigma^2}$$

On a log-lin diagram a Gaussian looks like a **parabola**!

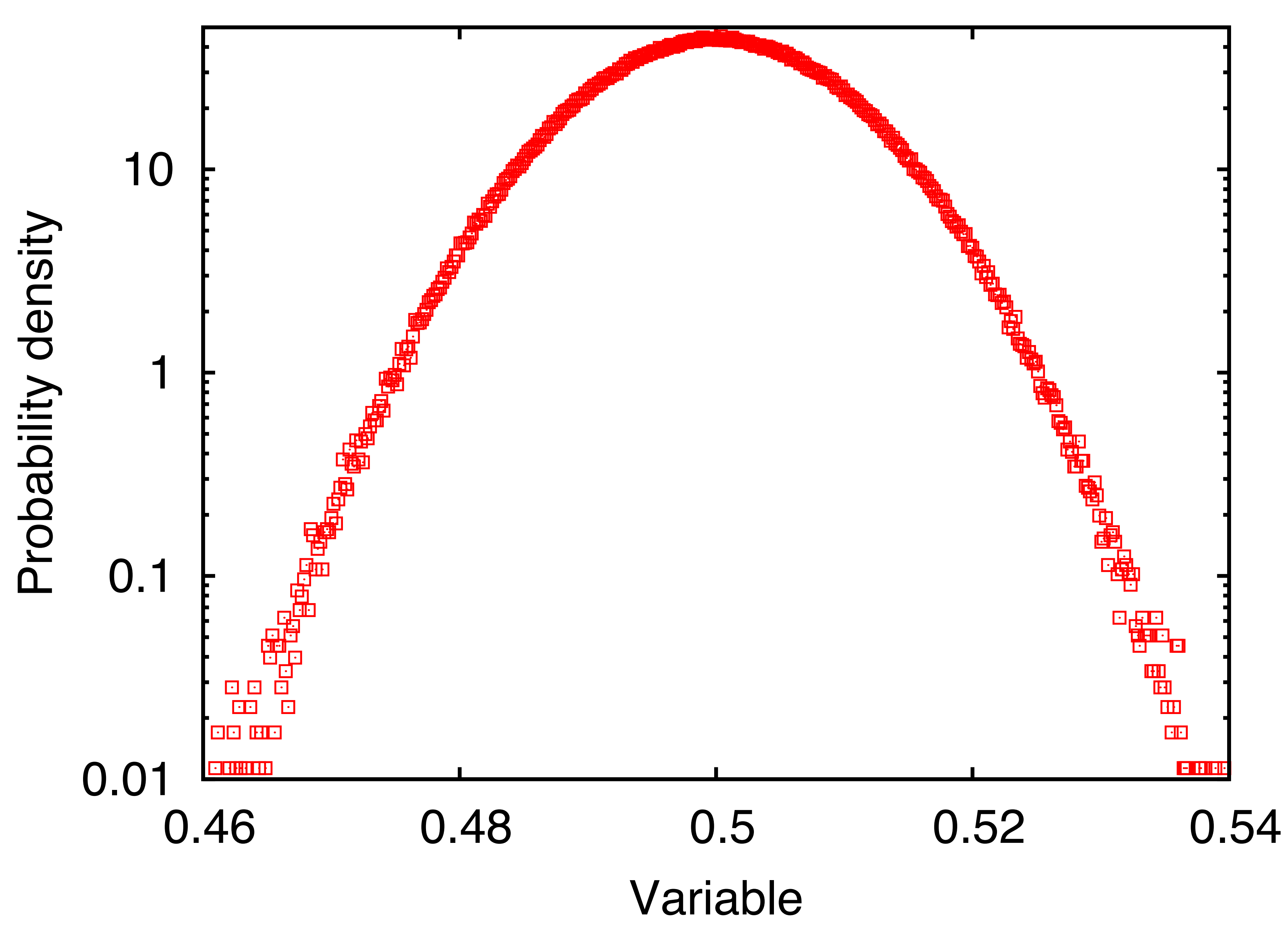
Using the log-scale

Gaussian distribution

Standard (lin-lin) scale



Log-lin scale



Using the log-scale

Exponential

$$P(x) = \lambda e^{-\lambda x}$$

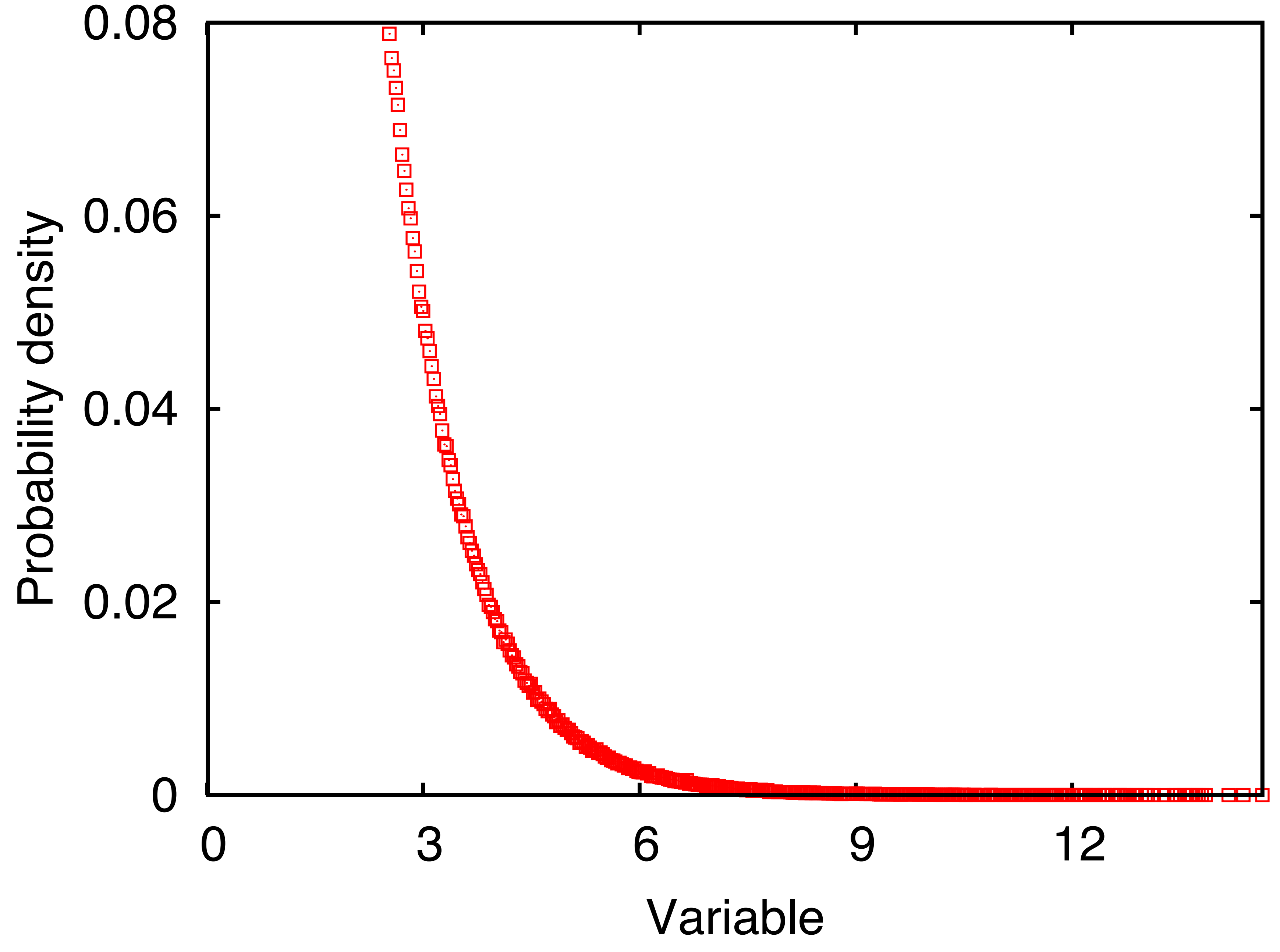
$$\log P(x) = \log \lambda - \lambda x$$

On a log-lin diagram an exponential looks like a **straight line**!

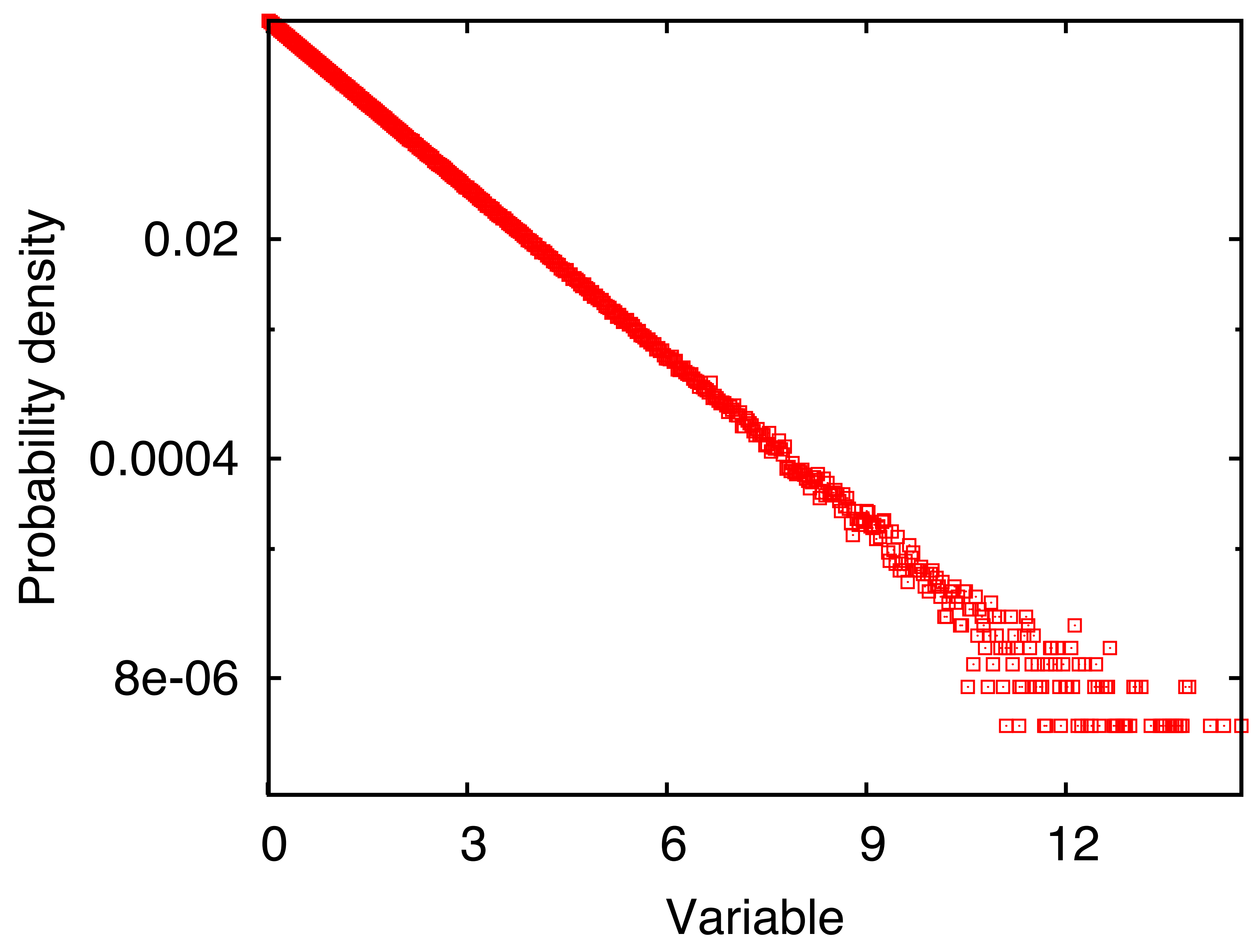
Using the log-scale

Exponential distribution

Standard (lin-lin) scale



Log-lin scale



Using the log-scale

Power law

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

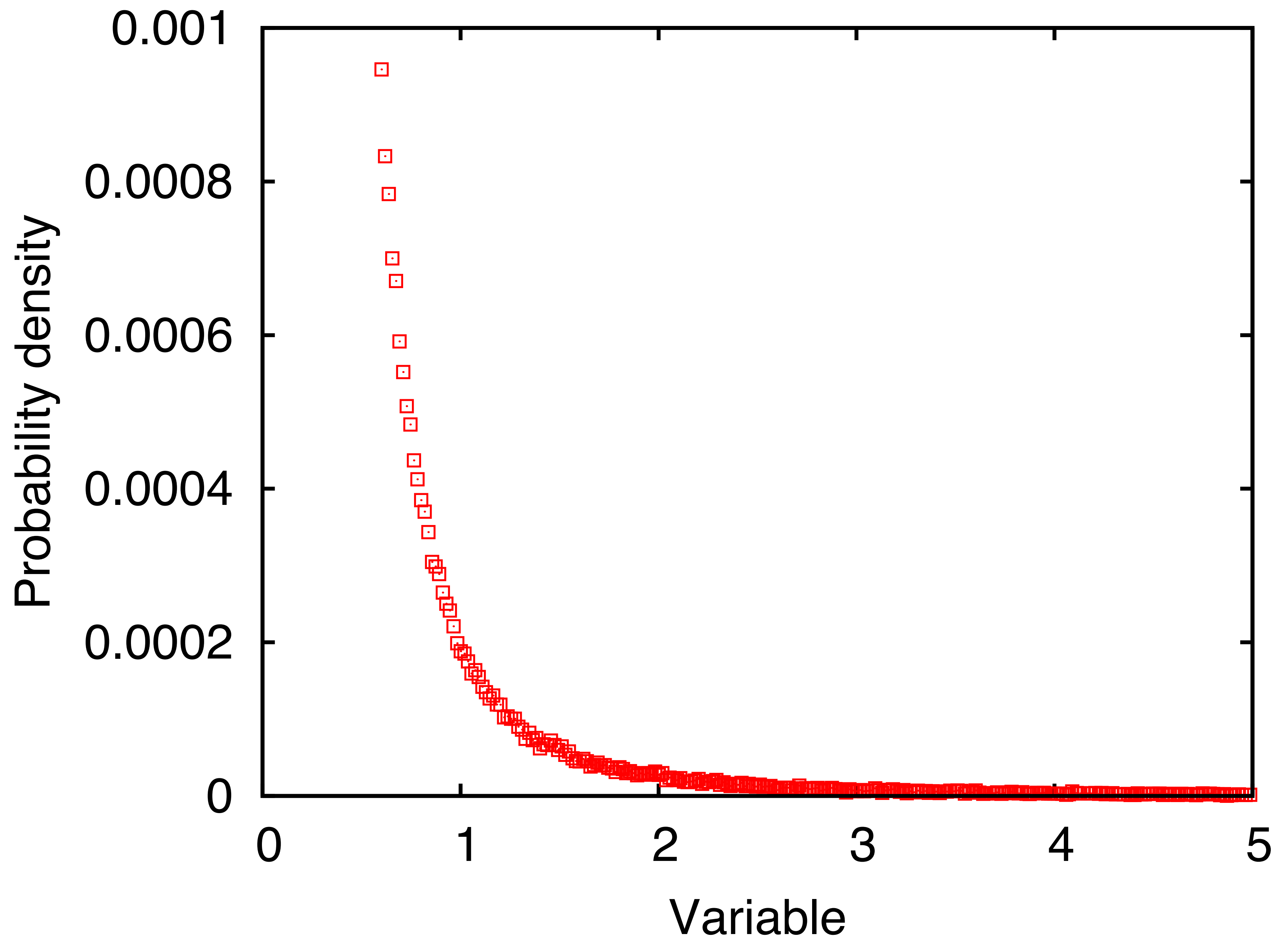
$$\log P(x) = \log \left(\frac{\alpha - 1}{x_{min}} \right) - \alpha \log x + \alpha \log x_{min}$$

On a log-log diagram a power law looks like a **straight line**!

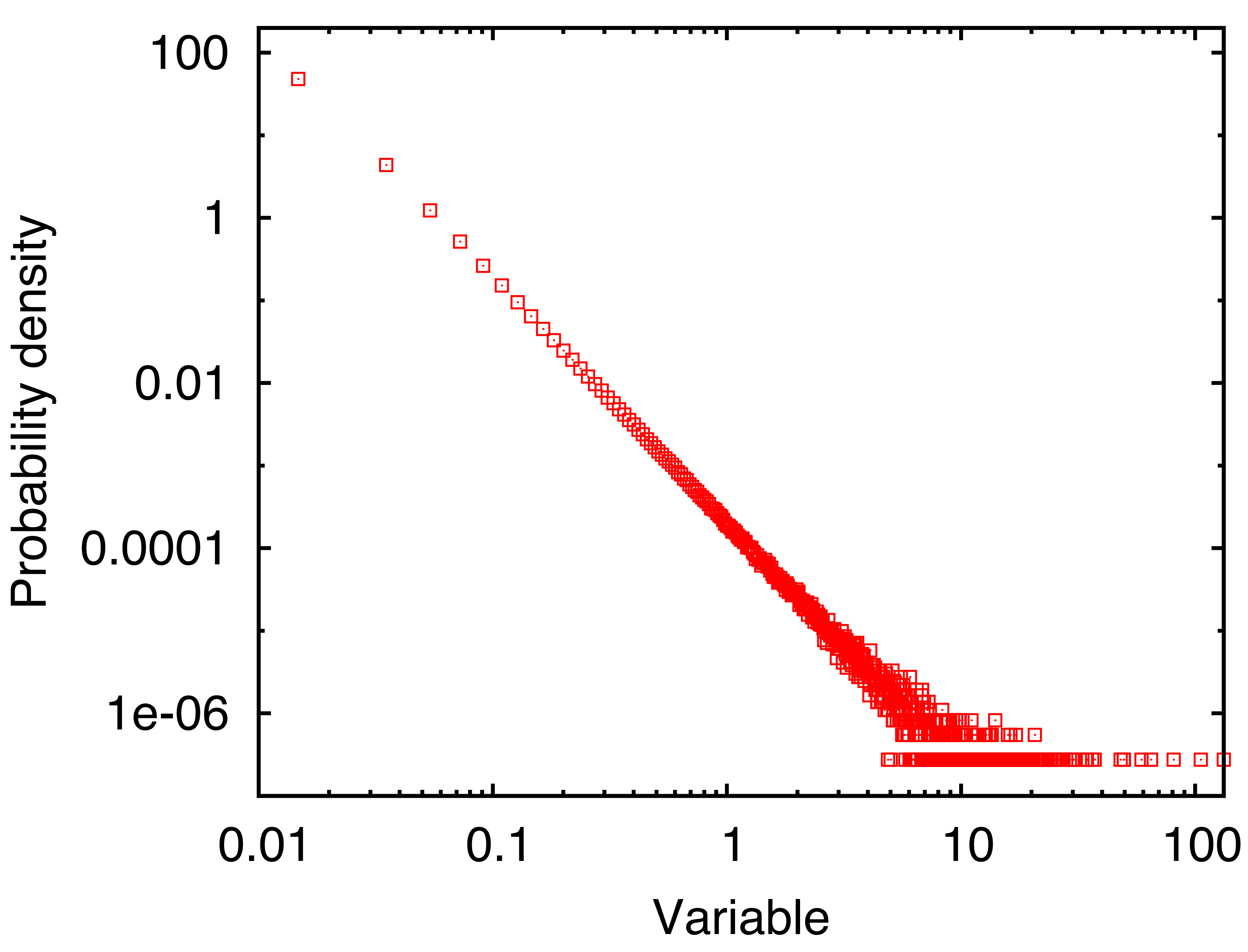
Using the log-scale

Power law distribution

Standard (lin-lin) scale



Log-log scale



Using the log-scale

Lognormal

$$P(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\log x - \mu)^2}{2\sigma^2}}$$

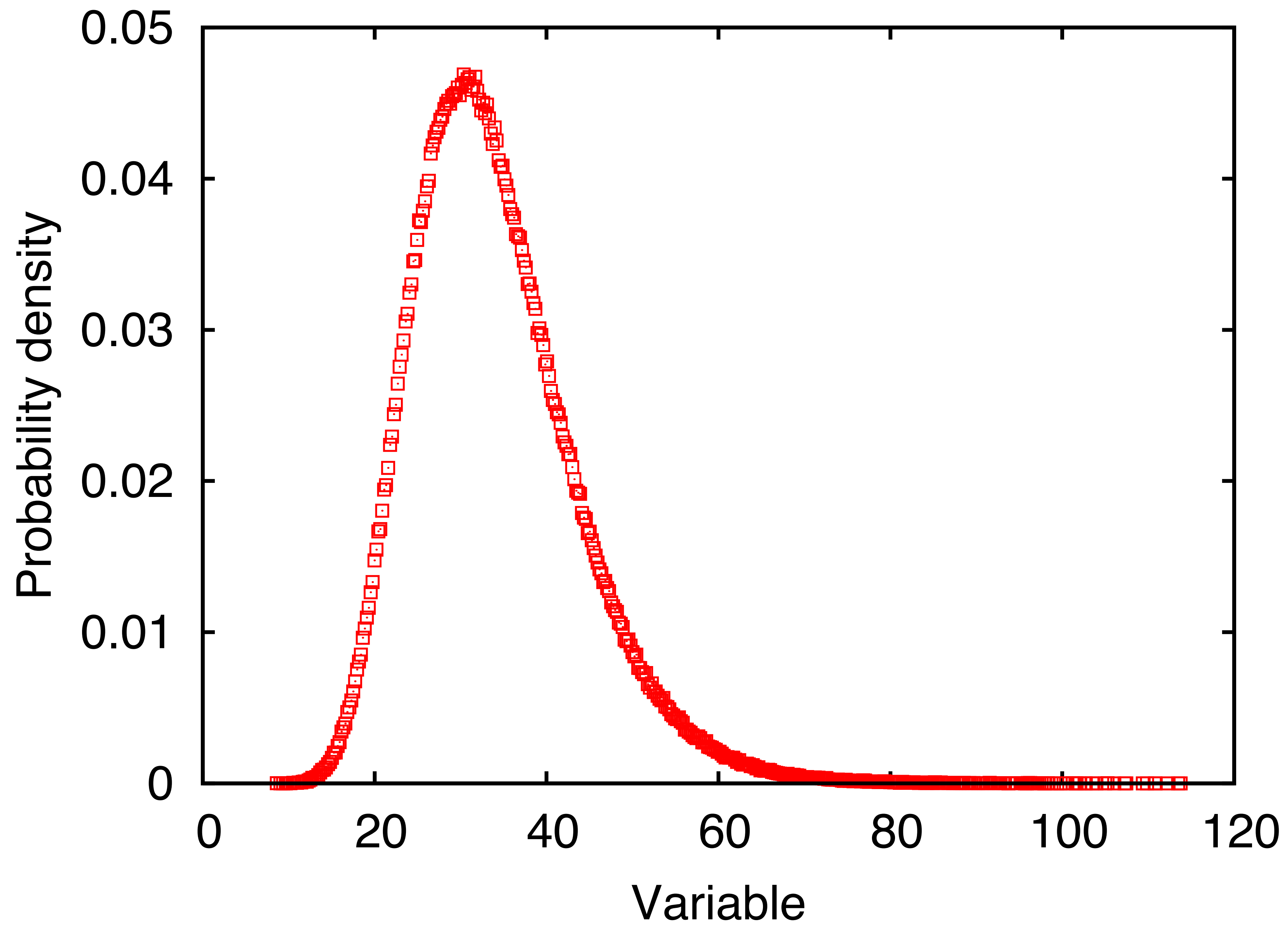
$$\log P(x) = -\log x - \log(\sigma\sqrt{2\pi}) - \frac{(\log x - \mu)^2}{2\sigma^2}$$

On a log-log diagram a lognormal looks like a **parabola**!

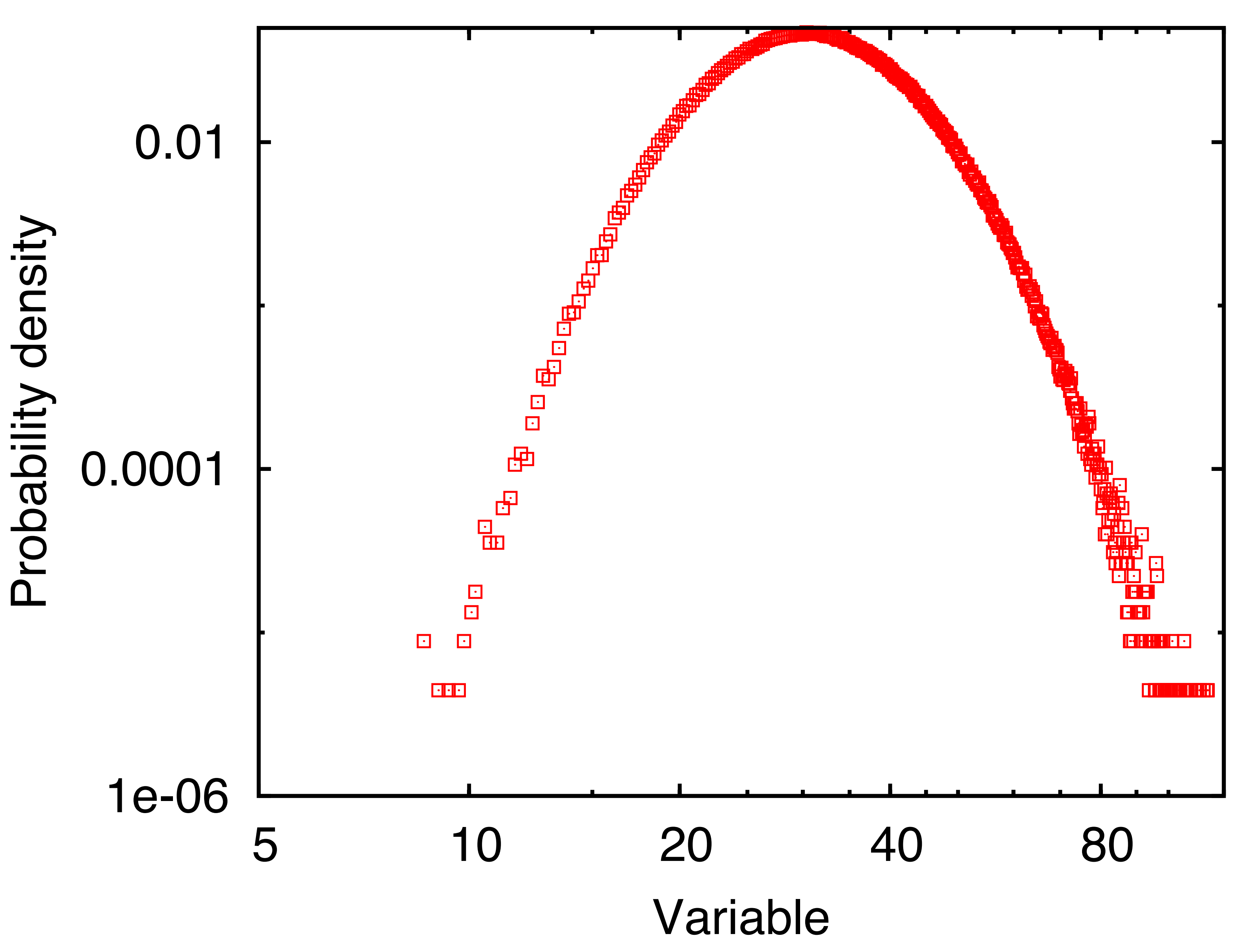
Using the log-scale

Lognormal distribution

Standard (lin-lin) scale



Log-log scale



The cumulative distribution

Distributions are generally noisy on the tail: events are rare!

$$P^{>}(x) = \int_{x}^{x_{max}} P(x')\, dx'$$

$$P^{<}(x) = \int_{x_{min}}^{x} P(x')\, dx' = 1 - P^{>}(x)$$

The integration averages fluctuations out, reducing the noise

The cumulative distribution

The cumulative of an exponential is still an exponential!

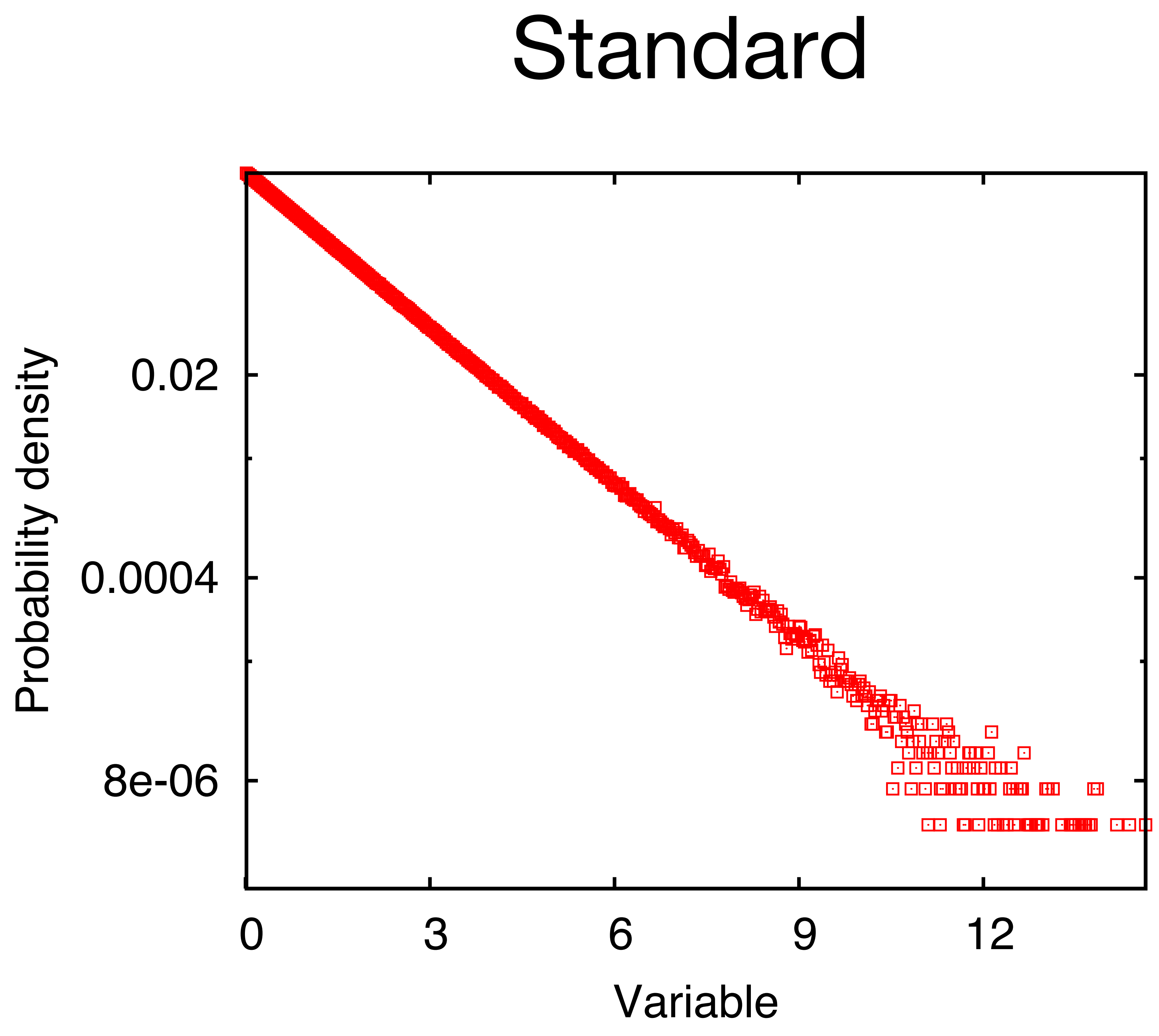
$$P^{>}_{exp}(x) = \int_{x}^{\infty} \lambda e^{-\lambda x'}\, dx' = \left[-e^{-\lambda x'}\right]_{x}^{\infty} = e^{-\lambda x}$$

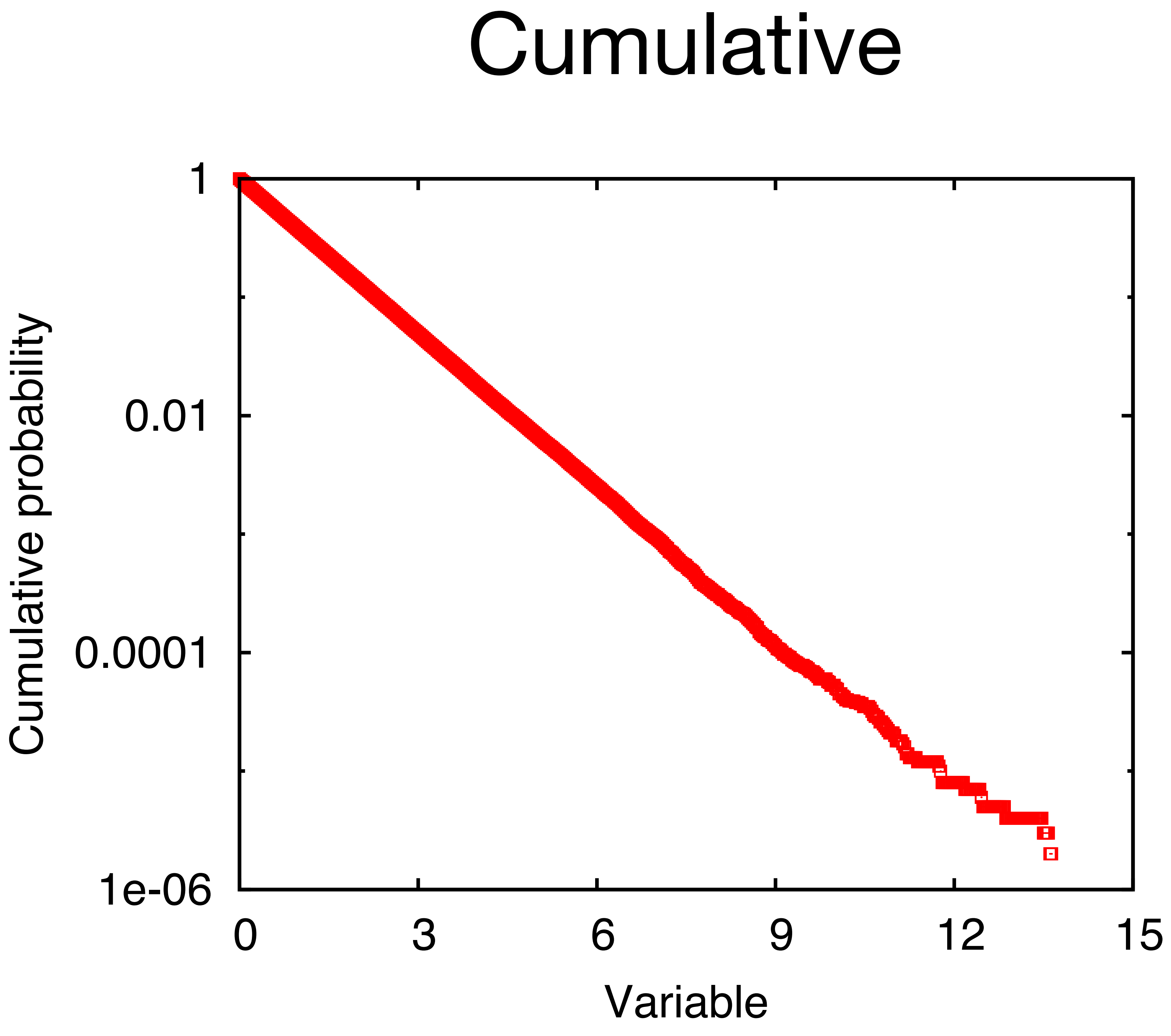
The cumulative of a power law is still a power law but with exponent $\alpha - 1$

$$P^{>}_{power}(x) = \int_{x}^{\infty} C{x'}^{-\alpha}\, dx' = \left[\frac{C{x'}^{1-\alpha}}{1-\alpha}\right]_{x}^{\infty} = \frac{Cx^{-(\alpha-1)}}{\alpha - 1}$$

The cumulative distribution

Exponential distribution

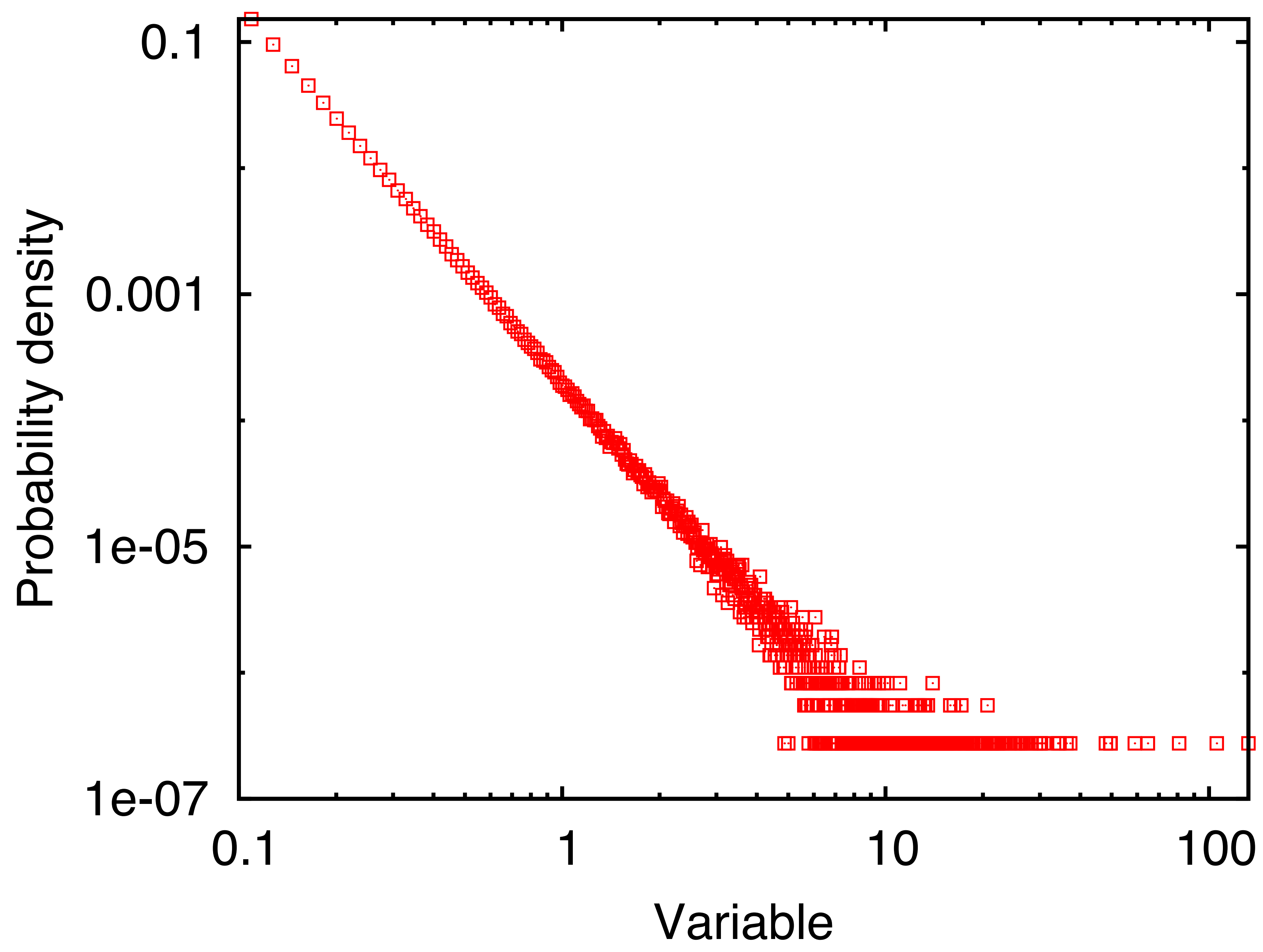




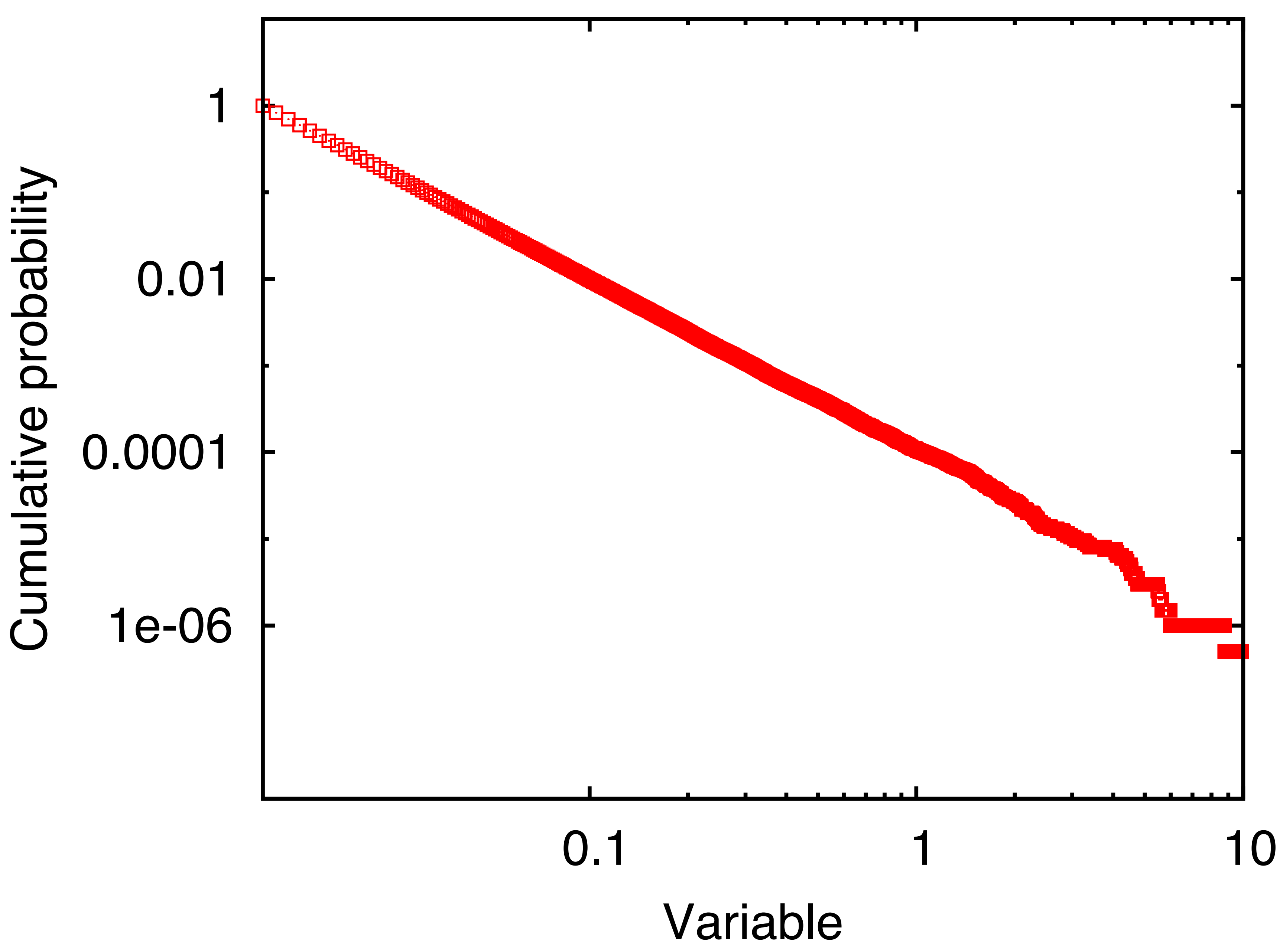
The cumulative distribution

Power law distribution

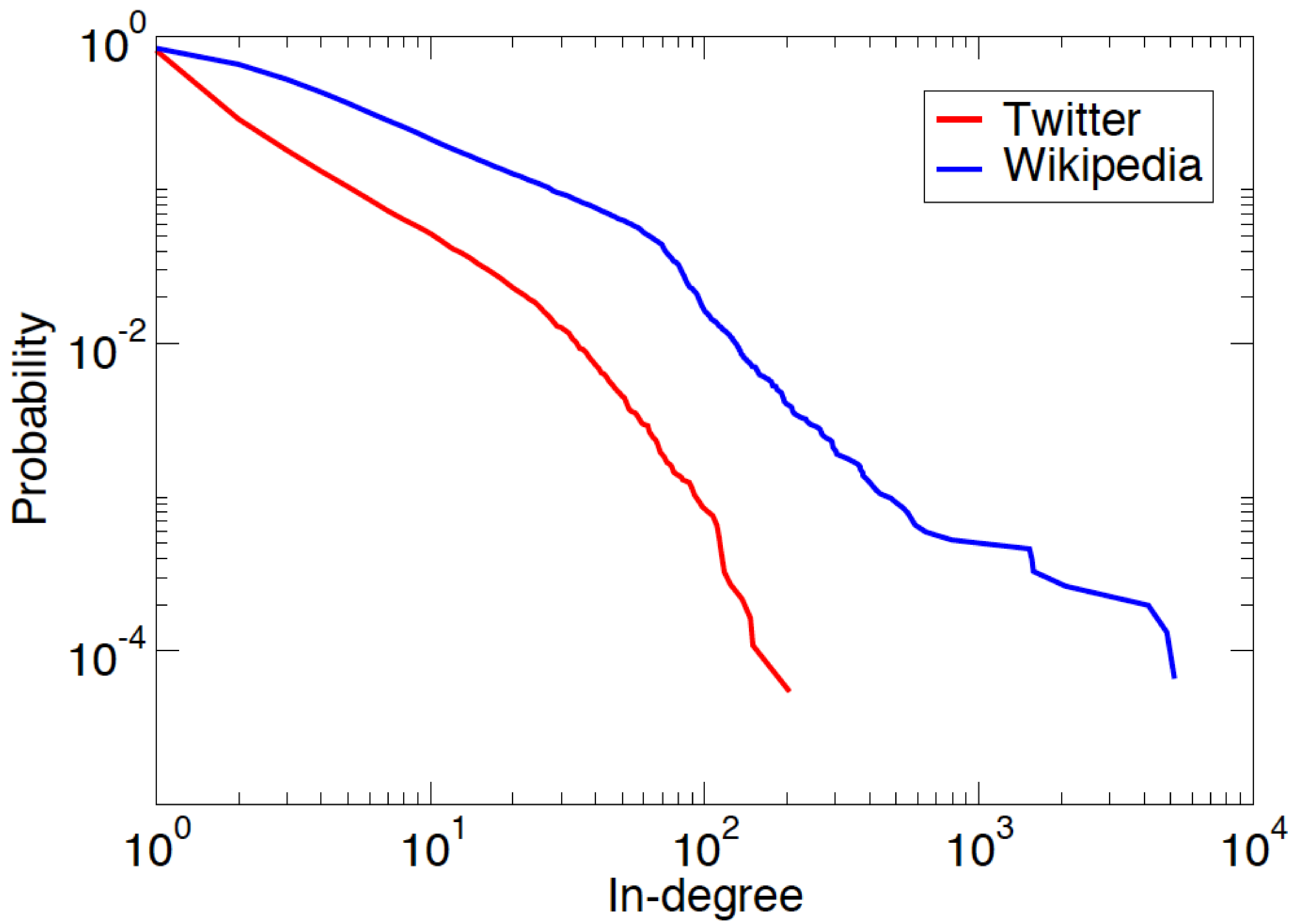
Standard



Cumulative



Degree distributions



Heavy-tail distributions: the variable goes from small to large values

Degree distributions

- The **heterogeneity parameter** $\boldsymbol{\kappa}$ says how broad the distribution is:

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle^2}$$

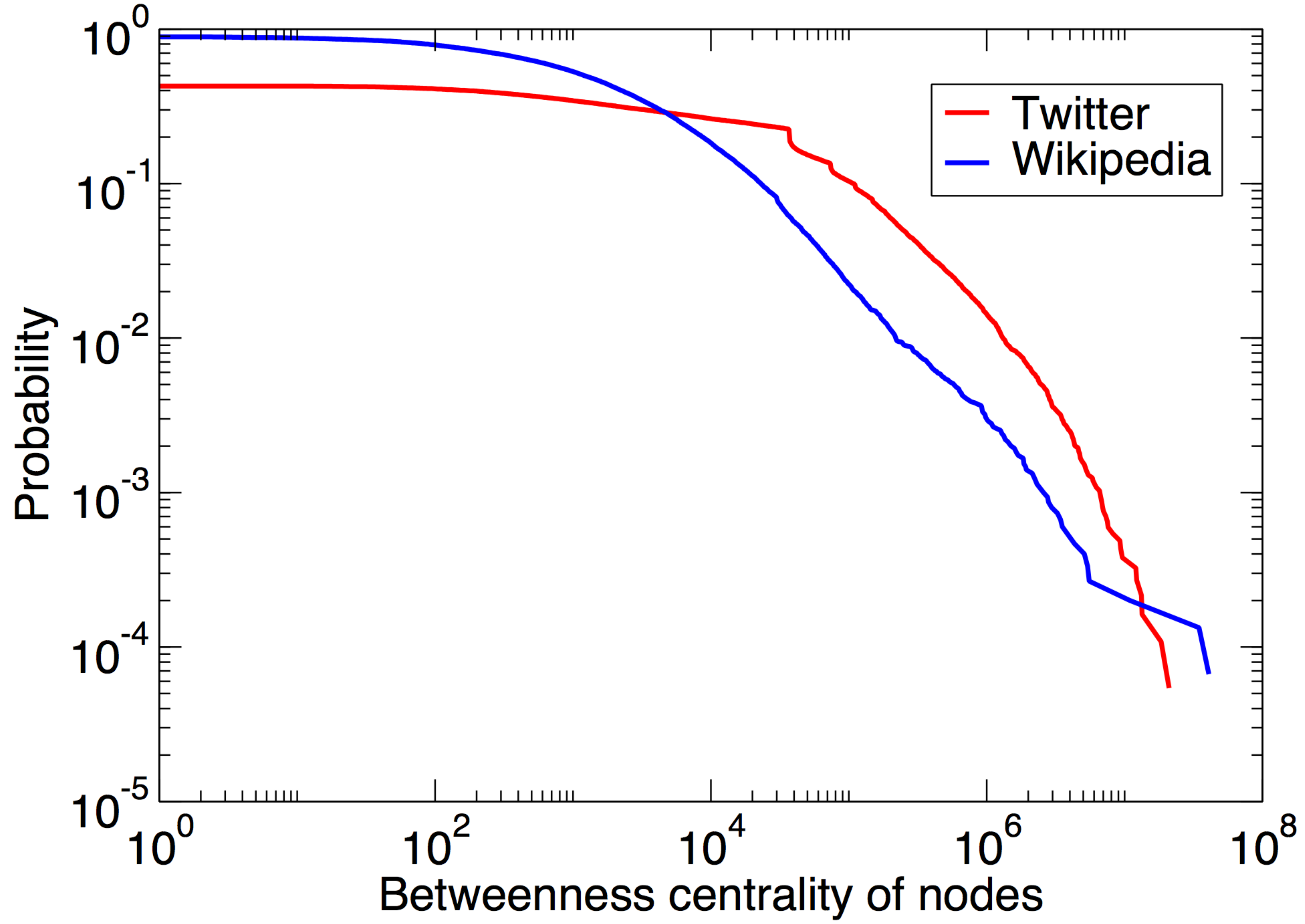
$$\langle k \rangle = \frac{\sum_i k_i}{N} = \frac{2L}{N}; \langle k^2 \rangle = \frac{\sum_i k_i^2}{N}$$

- If most degrees have the same value, say k_0 :

$$\langle k \rangle \approx k_0, \langle k^2 \rangle \approx k_0^2 \Longrightarrow \kappa \approx 1$$

- If the distribution is very heterogeneous: $\kappa \gg 1$

Betweenness distributions

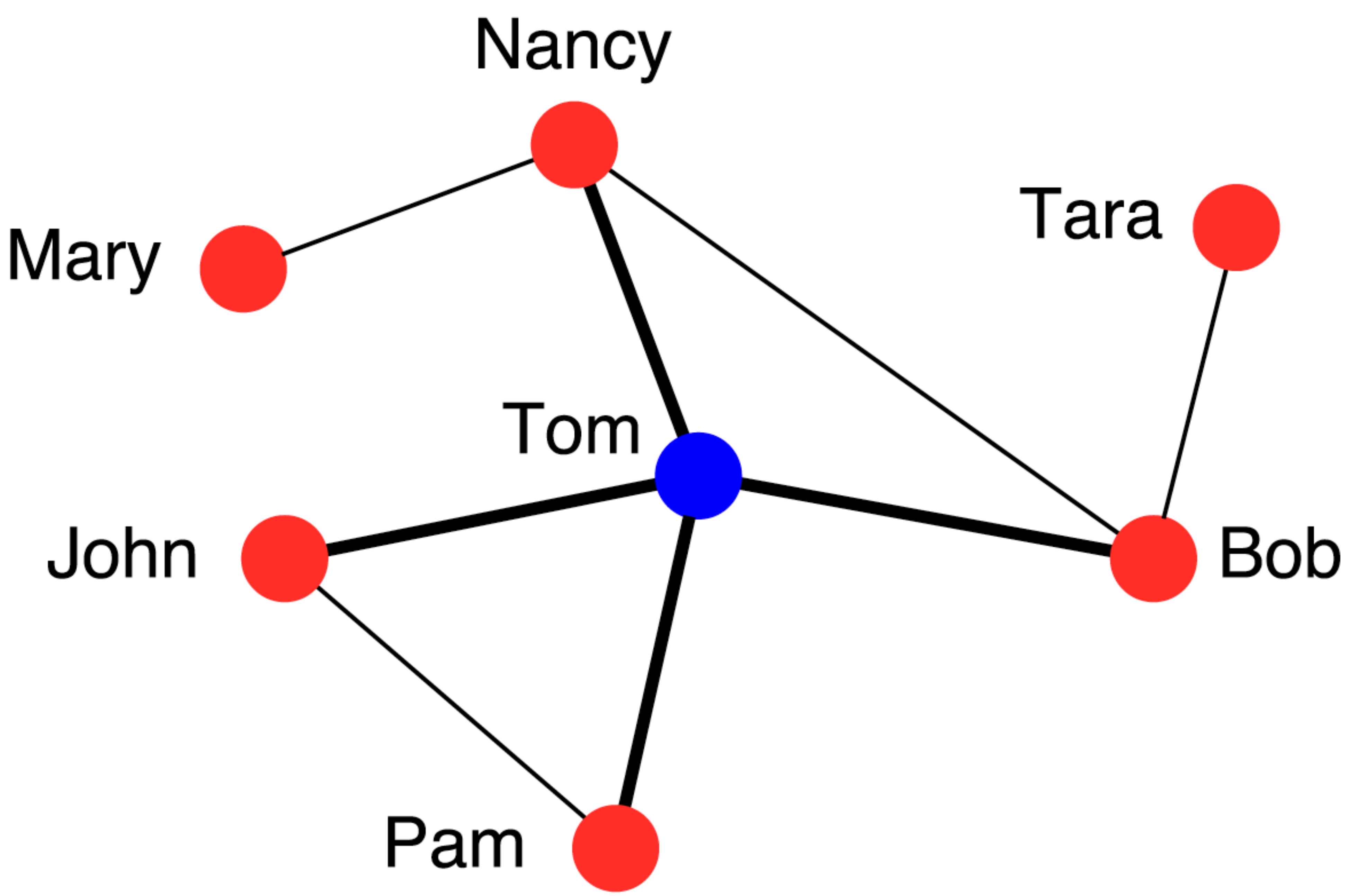


Heavy-tail distribution: the variable goes from small to large values

Degree centrality

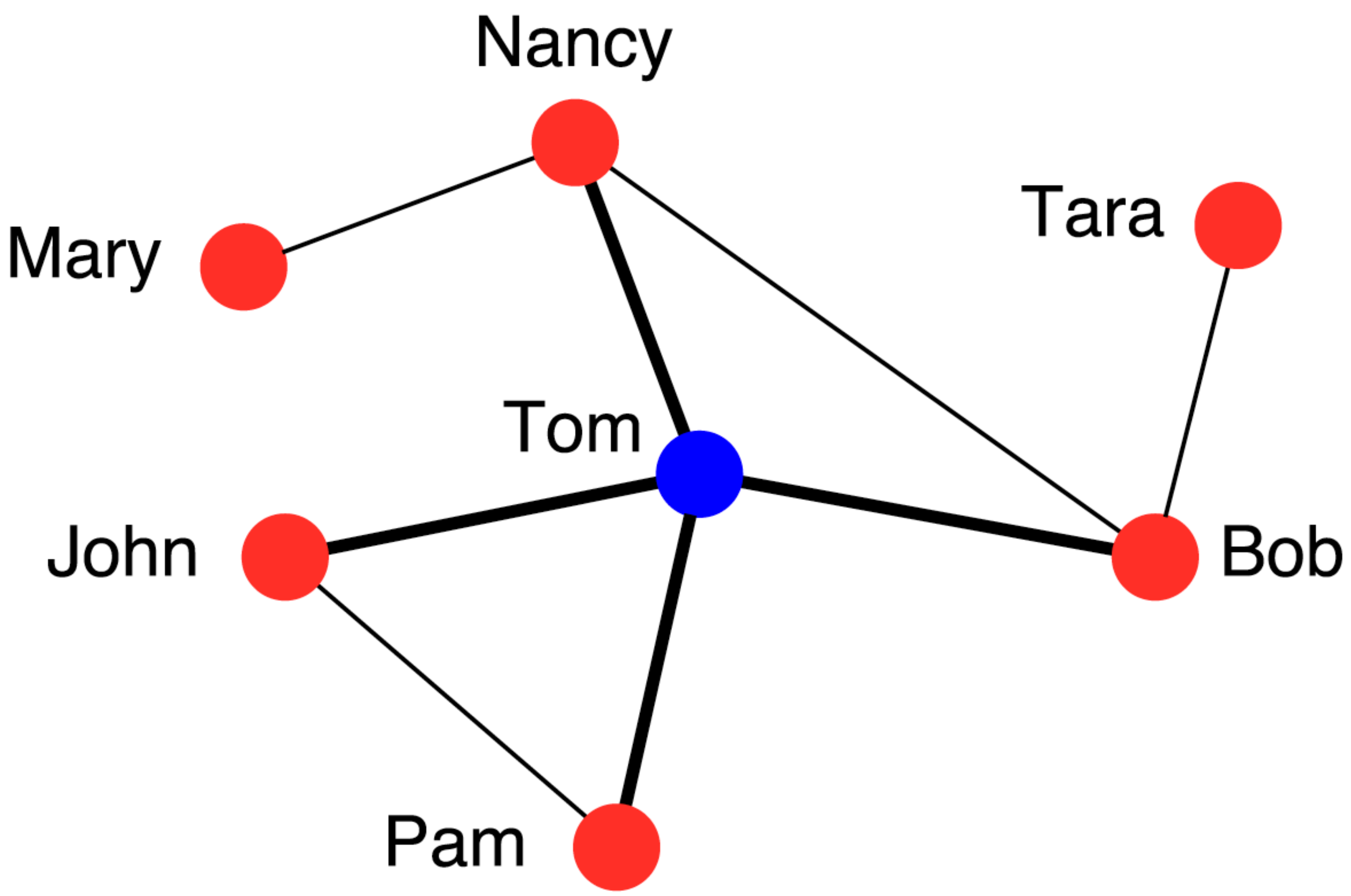
Network	Nodes (N)	Links (L)	Average degree ($\langle k \rangle$)	Maximum degree (k_{max})	Heterogeneity parameter (κ)
Facebook Northwestern Univ.	10,567	488,337	92.4	2,105	1.8
IMDB movies and stars	563,443	921,160	3.3	800	5.4
IMDB co-stars	252,999	1,015,187	8.0	456	4.6
Twitter US politics	18,470	48,365	2.6	204	8.3
Enron Email	36,692	367,662	10.0	1,383	14.0
Wikipedia math	15,220	194,103	12.8	5,171	38.2
Internet routers	190,914	607,610	6.4	1,071	6.0
US air transportation	546	2,781	10.2	153	5.3
World air transportation	3,179	18,617	11.7	246	5.5
Yeast protein interactions	1,870	2,277	2.4	56	2.7
C. elegans brain	297	2,345	7.9	134	2.7
Everglades ecological food web	69	916	13.3	63	2.2

Friendship paradox



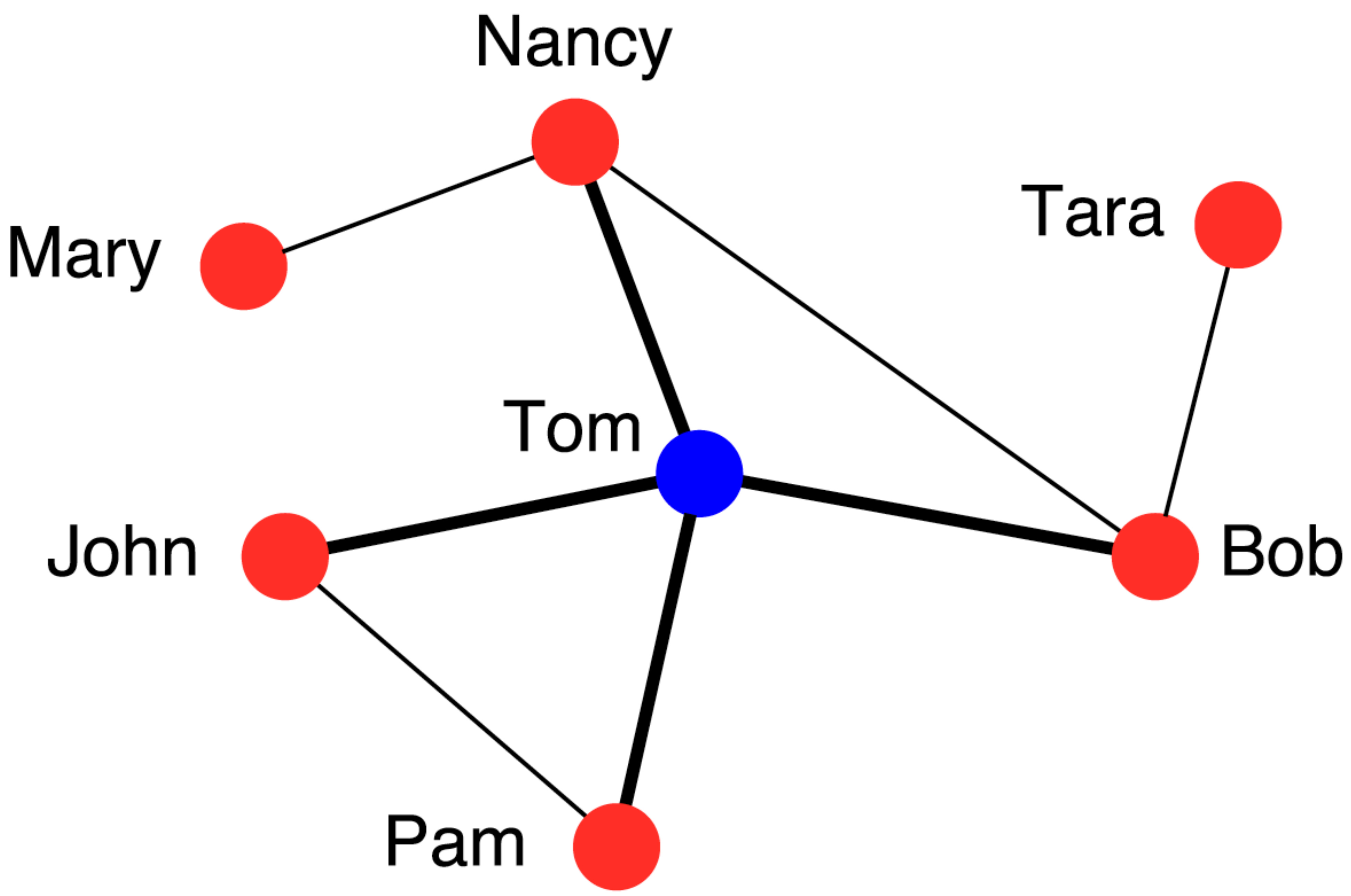
- By choosing *nodes at random*, Tom has **the same chance** to be picked as everybody else
- By choosing *links at random*, Tom has a **higher chance** to be picked than everybody else

Friendship paradox



By following links, **the chance to hit a hub increases**

Friendship paradox



- Average degree of a node = **2.29**
- Average degree of the neighbors of a node = **2.83 > 2.29**

- *Our friends have more friends than we do, on average* (**friendship paradox**)

Friendship paradox

- **Question:** Where does the friendship paradox come from?

- **Answer:**
  1. By averaging the degree of the nodes, we pick them at random
  2. By averaging the degree of the neighbors, we choose them by following links: nodes with degree k will be counted k times, which inflates the average

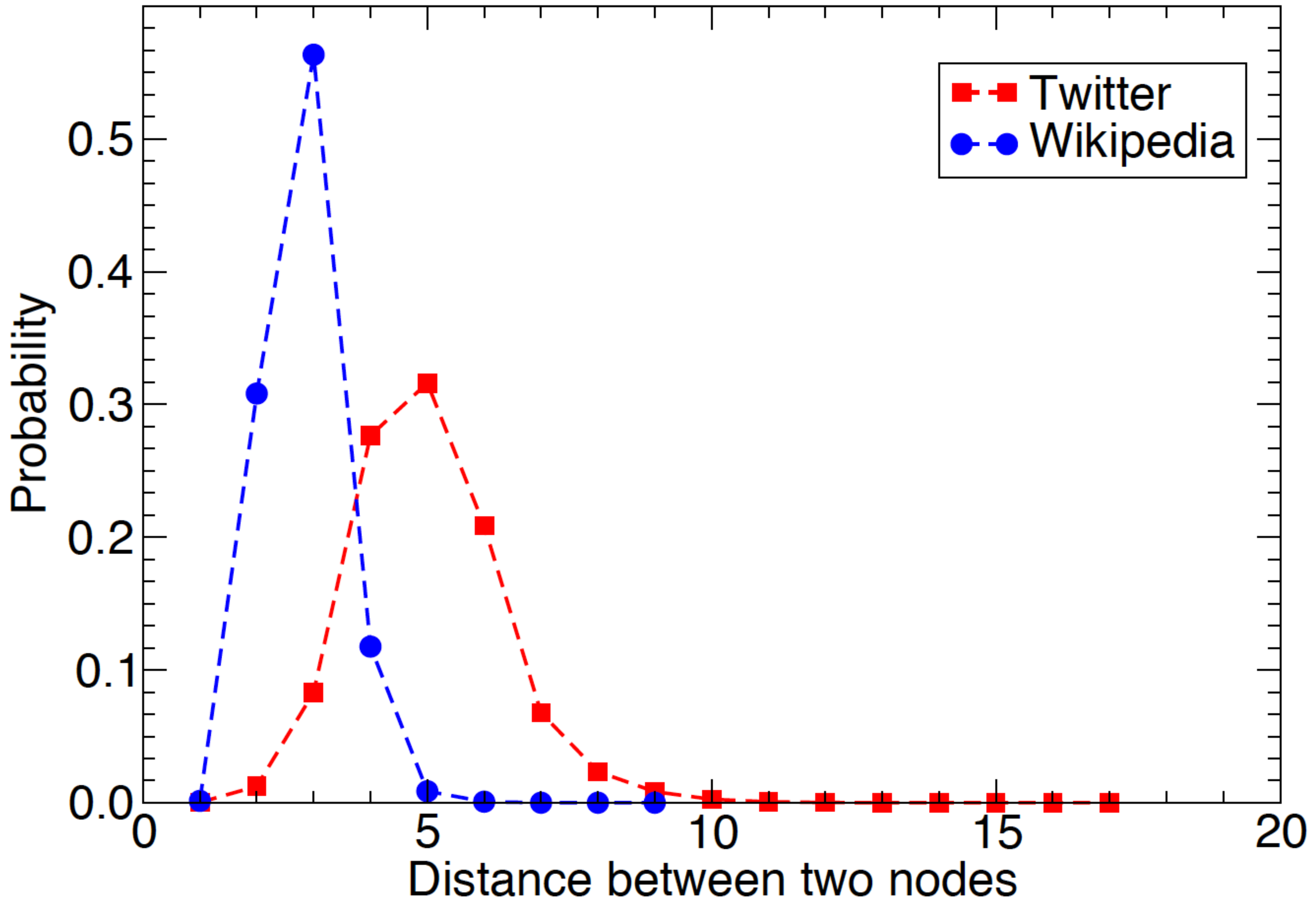
- The *more hubs, the stronger* the effect

Ultra-small worlds

- In real networks, many shortest paths go through hubs
- **Example:** air transportation
- There may be no routes between airport A and B (if they are small), but it may be possible to go from A to B via a hub airport C
- The small-world property is typical of most networks of interest: if the network has hubs, paths are ultra-short (**ultra-small world**)

Ultra-small worlds

Shortest-path length distribution



Robustness

- A system is **robust** if the failure of some of its components does not affect its function

- **Question:** how can we define the robustness of a network?

- **Answer:** we remove nodes and/or links and see what happens to its structure

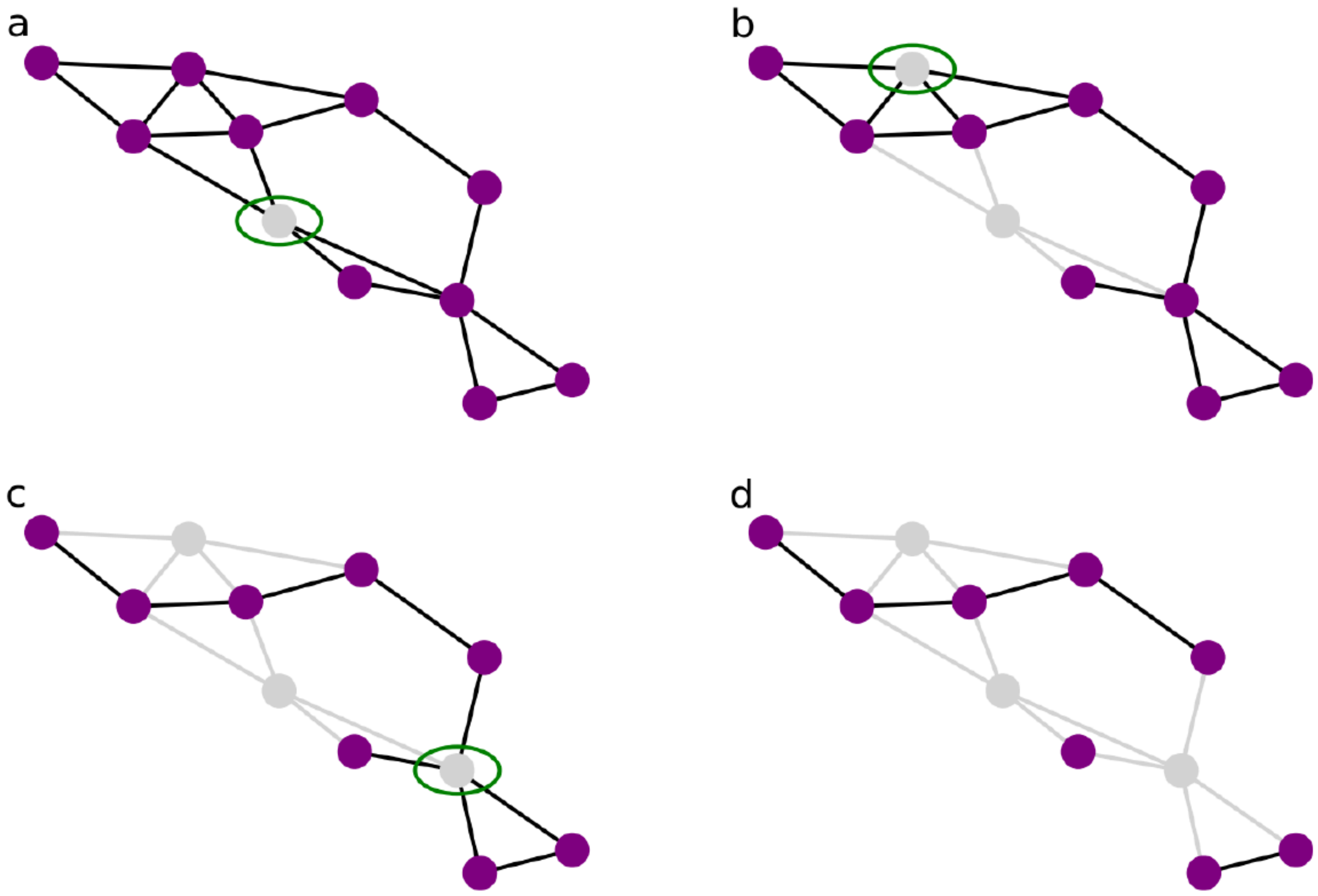
- **Key point:** *connectedness*

- If the Internet were not connected, it would be impossible to transmit signals (e.g., emails) between routers in different components

Robustness

- **Robustness test:** checking how the connectedness of the network is affected as more and more nodes are removed
- **How to do it:** plot the relative size *S* of the largest connected component as a function of the fraction of removed nodes
- We suppose that the network is initially connected: there is only one component and $S = 1$
- As more and more nodes (and their links) are removed, the network is progressively broken up into components and *S* goes down

Robustness



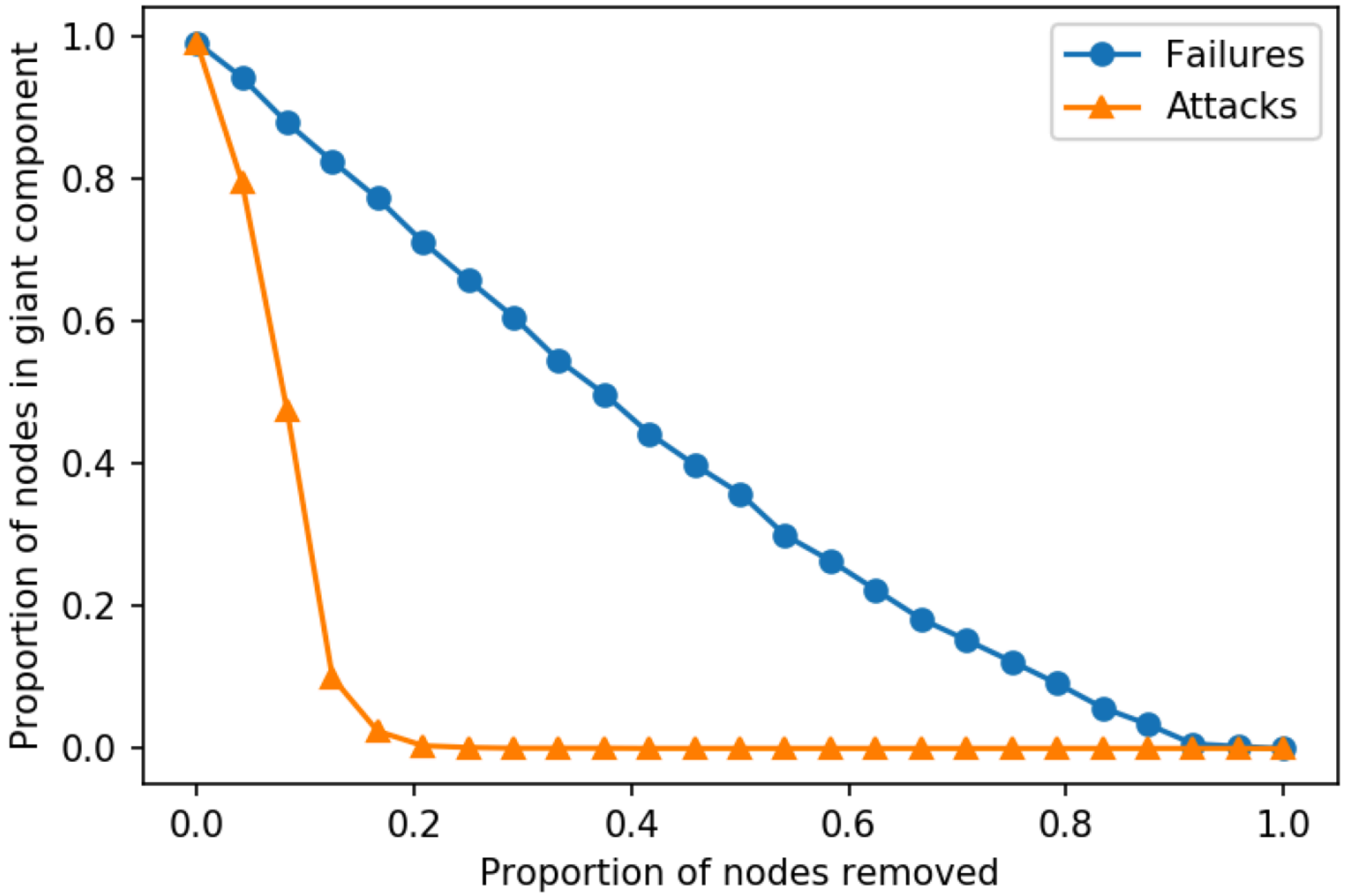
Robustness

- **Two strategies:**
  1. **Random failures:** nodes break down randomly, so they are all chosen with the **same probability**
  2. **Attacks:** hubs are deliberately targeted — the larger the **degree**, the higher the probability of removing the node

- In the first approach, we remove a fraction f of nodes, chosen at random

- In the second approach, we remove the fraction f of nodes with largest degree, from the one with largest degree downwards

Robustness



Conclusion: real networks are robust against random failures but fragile against targeted attacks!

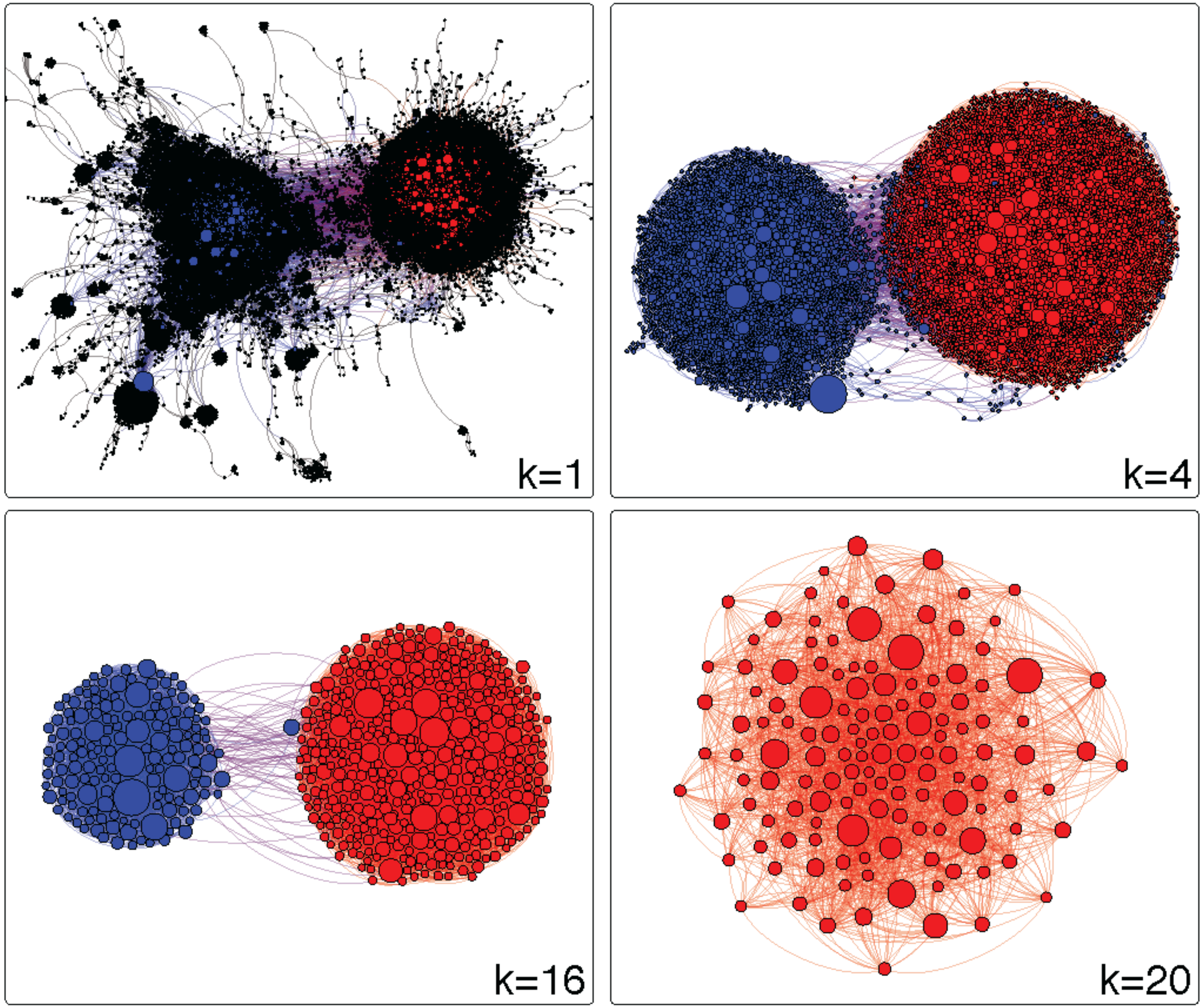
Core decomposition

- **Core:** dense part of the network, with high-degree nodes

- **Core decomposition:** procedure to identify denser and denser cores, by removing nodes of progressively higher degree. If we remove all nodes with degree $k - 1$ or lower, the remaining portion of the network is called $\boldsymbol{k}$**-core**

- $\boldsymbol{k}$**-core decomposition procedure:** start with k=0
  1. Recursively remove all nodes with degree k, until none are left
  2. The set of removed nodes is the $\boldsymbol{k}$**-th shell**, while the remaining ones form the $(\boldsymbol{k+1})$**-core**
  3. If there is no node left, terminate. Otherwise, increment k by one and repeat from step 1

Core decomposition



Core decomposition

Core decomposition helps to visualize large networks, by pruning low-degree nodes and showing only the densest parts

```
nx.core_number(G)   # return dict with core number of each node
nx.k_shell(G,k)     # subnetwork induced by nodes in k-shell
nx.k_core(G,k)      # subnetwork induced by nodes in k-core
nx.k_core(G)        # innermost (max-degree) core subnetwork
```