Power-law Clauset test

POWER-LAW DISTRIBUTIONS IN EMPIRICAL DATA
https://arxiv.org/pdf/0706.1062.pdf

powerlaw: a Python package for analysis of heavy-tailed distributions
https://arxiv.org/pdf/1305.0215.pdf

Notes by: Mohammed Alasmar, University of Sussex, 2018

- This approach is about finding out if a tested data can be fitted using power-law distribution.

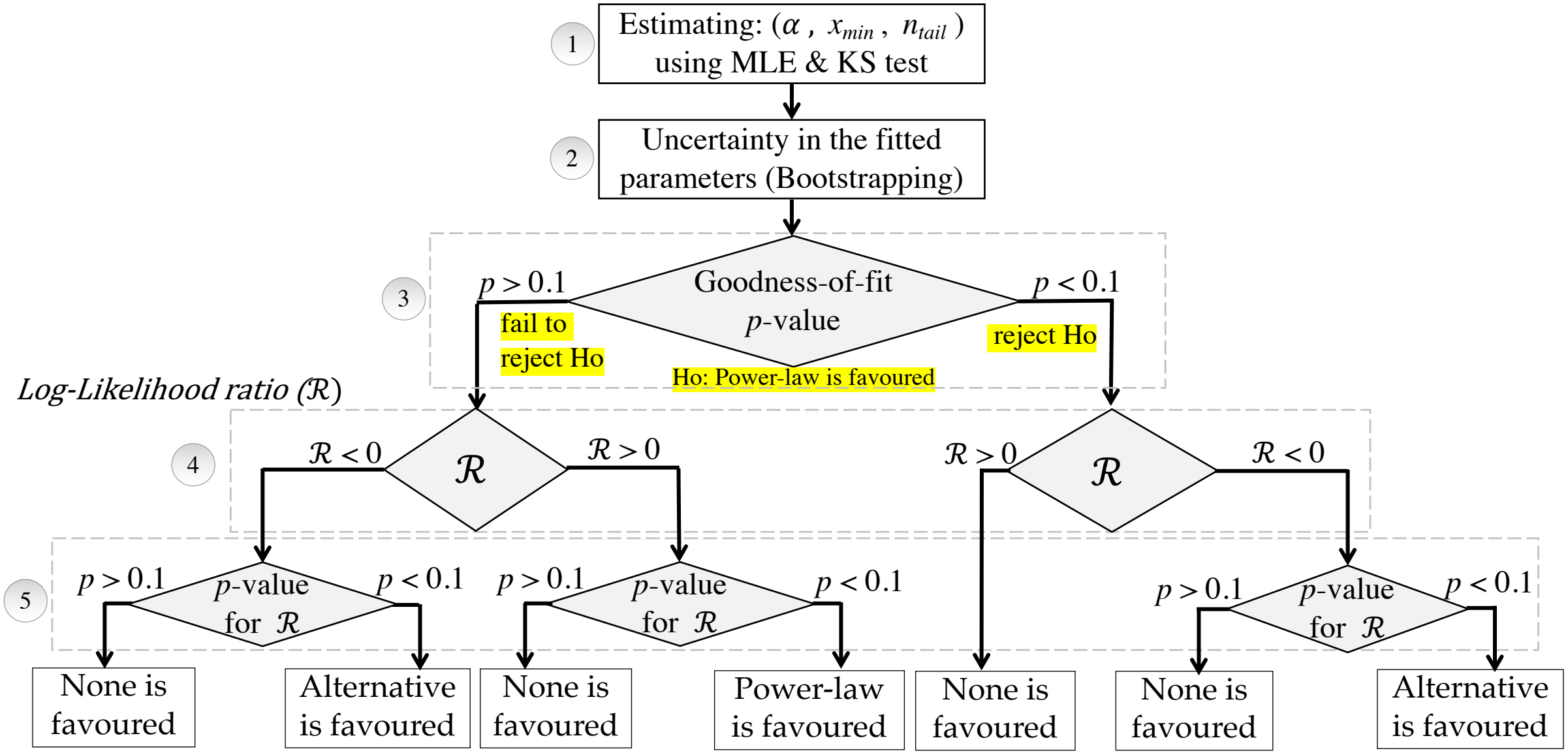
- Also, it gives a robust test to find an alternative distribution (e.g. lognormal) if the power-law is not a good fit.

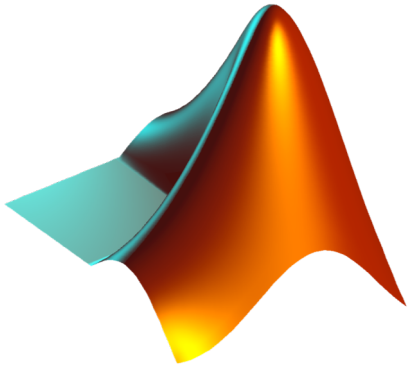
A. Clauset, C. R. Shalizi and M. E. J. Newman

	name	distribution $p(x) = Cf(x)$ $f(x)$	C
continuous	power law	$x^{-\alpha}$	$(\alpha-1)x_{\min}^{\alpha-1}$
	power law with cutoff	$x^{-\alpha}\mathrm{e}^{-\lambda x}$	$\frac{\lambda^{1-\alpha}}{\Gamma(1-\alpha,\lambda x_{\min})}$
	exponential	$\mathrm{e}^{-\lambda x}$	$\lambda \mathrm{e}^{\lambda x_{\min}}$
	stretched exponential	$x^{\beta-1}\mathrm{e}^{-\lambda x^{\beta}}$	$\beta\lambda \mathrm{e}^{\lambda x_{\min}^{\beta}}$
	log-normal	$\frac{1}{x}\exp\left[-\frac{(\ln x-\mu)^2}{2\sigma^2}\right]$	$\sqrt{\frac{2}{\pi\sigma^2}}\left[\mathrm{erfc}\left(\frac{\ln x_{\min}-\mu}{\sqrt{2}\sigma}\right)\right]^{-1}$

POWER-LAW DISTRIBUTIONS IN EMPIRICAL DATA

1. **MLE to estimate alpha**
2. **KS test to estimate xmin**
3. **Uncertainty in the fitted parameters**
4. **Goodness-of-fit test (p-value test)**
5. **Likelihood ratio test**


1
Estimating: (α , x_{min} , n_{tail}) using MLE & KS test
2
Uncertainty in the fitted parameters (Bootstrapping)
3
$p > 0.1$
fail to reject Ho
Goodness-of-fit p-value
$p < 0.1$
reject Ho
Ho: Power-law is favoured
Log-Likelihood ratio ($\mathcal{R}$)
4
$\mathcal{R} < 0$
$\mathcal{R}$
$\mathcal{R} > 0$
$\mathcal{R} > 0$
$\mathcal{R}$
$\mathcal{R} < 0$
5
$p > 0.1$
p-value for $\mathcal{R}$
$p < 0.1$
$p > 0.1$
p-value for $\mathcal{R}$
$p < 0.1$
$p > 0.1$
p-value for $\mathcal{R}$
$p < 0.1$
None is favoured
Alternative is favoured
None is favoured
Power-law is favoured
None is favoured
None is favoured
Alternative is favoured



See this file: test.m

Step 1

1

Estimating: $(\alpha\ ,\ x_{min}\ ,\ n_{tail}\ )$ using MLE & KS test

Use `plfit.m` Matlab function

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

Step 1

Estimating: (α , x_{min} , n_{tail})
using MLE & KS test

Estimating the scaling paramerter α

Maximum Likelihood Estimation: $\mathrm{L}(\alpha) = \prod_{i=1}^{n} p(x)$

Finding the value α that maximises the log-likelihood function: $\frac{d\,log(\mathrm{L}(\alpha))}{d(\alpha)} = 0$ ➔ solve for $\hat{\alpha}$

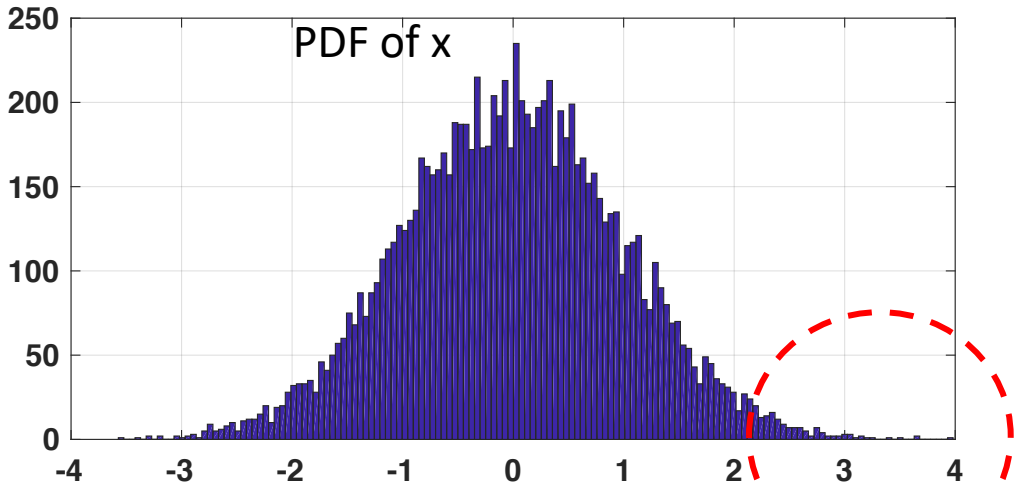
Estimating xmin using KS test

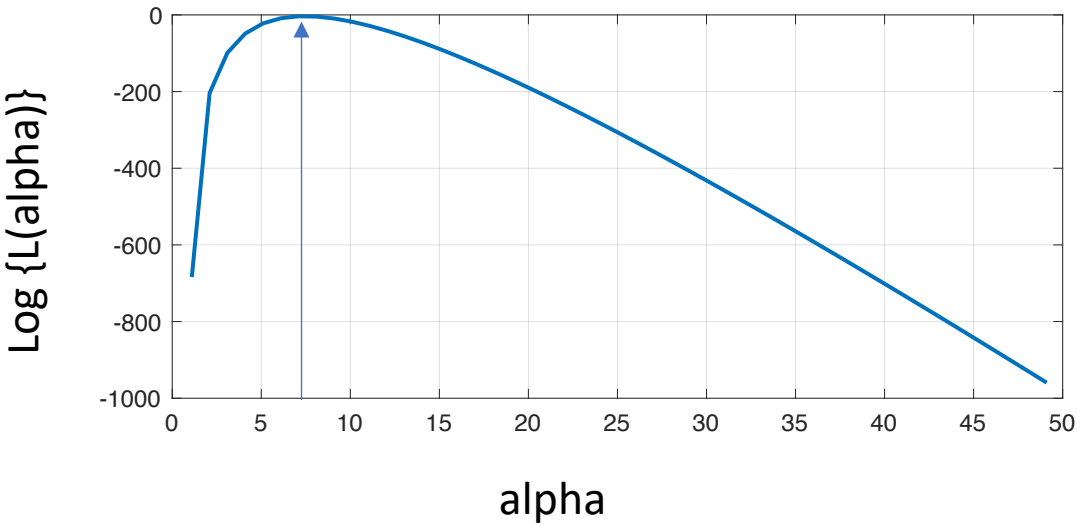
$D = \max_{x \geq x_{min}} |S(x) - P(x)|$, $S(x)$: input data CDF, $P(x)$:power-law CDF

$\hat{x}_{min}$: is the value that minimizes the distance D.

x_{min} is the minimum value for which the power law holds

Example:

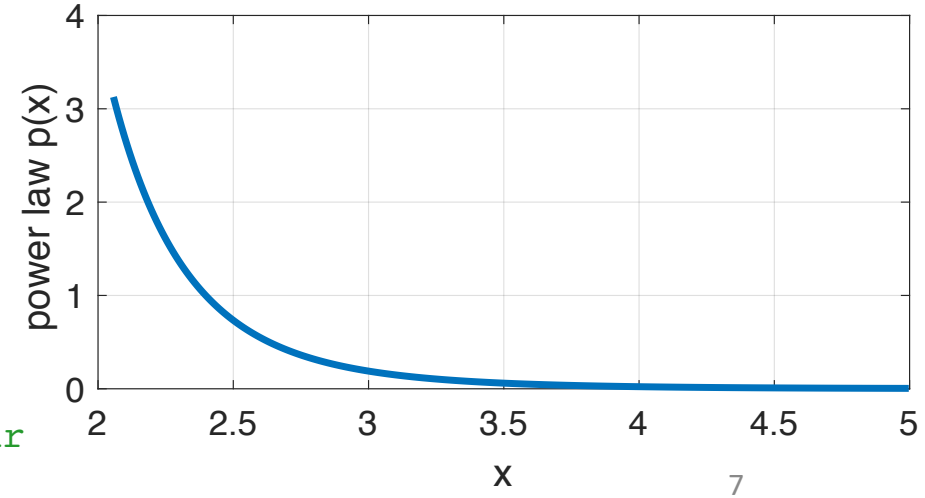




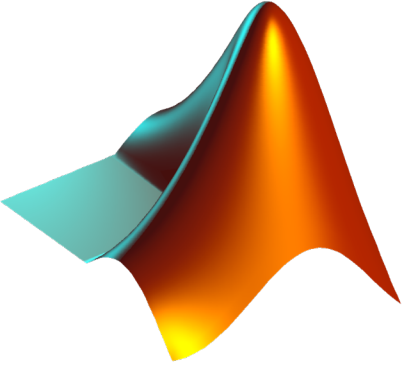

```
x= randn(1,10000);
histogram(x);
[alpha, xmin, ntail] = plfit(x)
alpha =    7.4311
xmin =     2.0570
ntail =    214 % out of 10000
```

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

```
x= xmin:0.01:5;
p= ((alpha-1)/xmin)* (x/xmin).^-alpha;
plot(x,p)
```



- 'alpha' is the maximum likelihood estimate of the scaling exponent,
- 'xmin' is the estimate of the lower bound of the power-law behaviour



Step 2

2

Uncertainty in the fitted parameters (Bootstrapping)

Use `plvar.m` Matlab function

Step 2

Uncertainty in the fitted parameters
Resampling (bootstrapping)

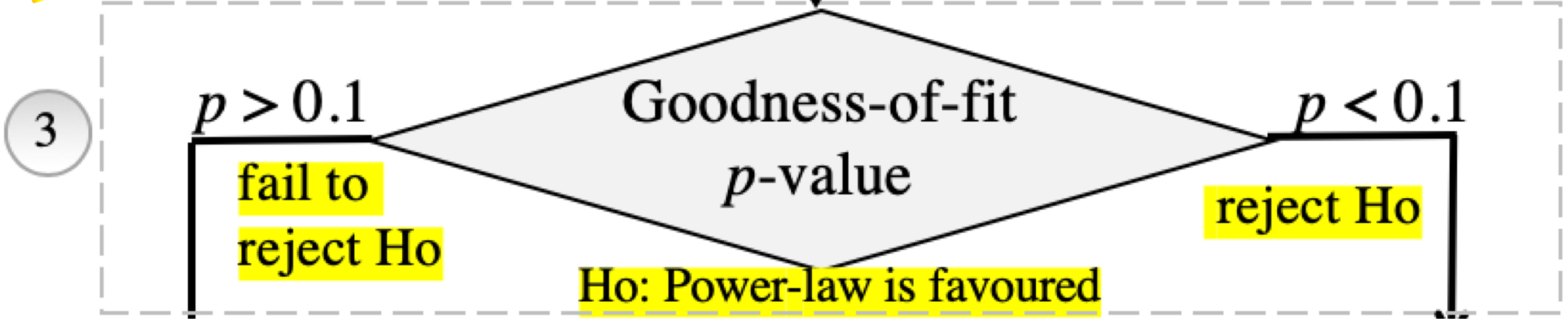
Bootstrapping: Resampling method in which observations are taken at random and with replacement from an existing sample.

```
[alpha, xmin, ntail] = plfit(x)
alpha =   7.4311
xmin =    2.0570
ntail =   214
```

```
plvar(x);
alpha_uncertainty =  1.1502
xmin_uncertainty =   0.2420
ntail_uncertainty =  203.2283
```

$$\hat{\alpha} = 7.4311 \pm 1.1502$$
$$\hat{x}_{min} = 2.0570 \pm 0.2420$$
$$n_{tail} = 214 \pm 203$$

Step 3



Use `plpva.m` Matlab function

Calculating p-value for fitted power-law model

```
p =plpva(x, xmin )
```

Run step 1,2 and 3 on a data from the paper:

Regenerating the results in Table 6.1

From Table 6.1

Run step 1,2 and 3 on a data from the paper (words count data):

quantity	n	$\langle x \rangle$	σ	$x_{\max}$	$\hat{x}_{\min}$	$\hat{\alpha}$	n_{tail}	p
count of word use	18 855	11.14	148.33	14 086	7 ± 2	1.95(2)	2958 ± 987	**0.49**

The frequency of occurrence of unique words in the novel …..

Matlab

1. **Fitting a power-law distribution**

```
[alpha, xmin] = plfit(x)
```

alpha = 1.9500 , xmin = 7 , n = 2958

2. **Estimating uncertainty in the fitted parameters**

```
[alphaun, xminun, nun]=plvar(x);
```

alphaun = 0.0254 , xminun = 1.8340 , nun = 987

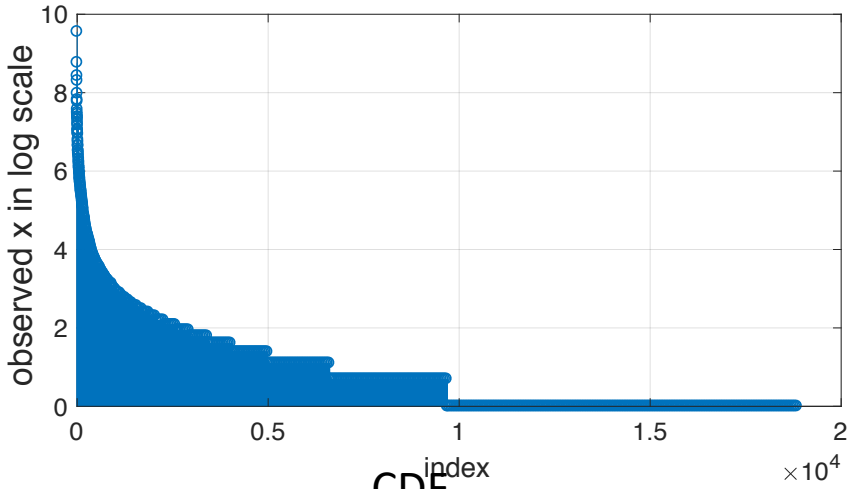
3. **Calculating p-value for fitted power-law model**

```
p =plpva(x, xmin )
```

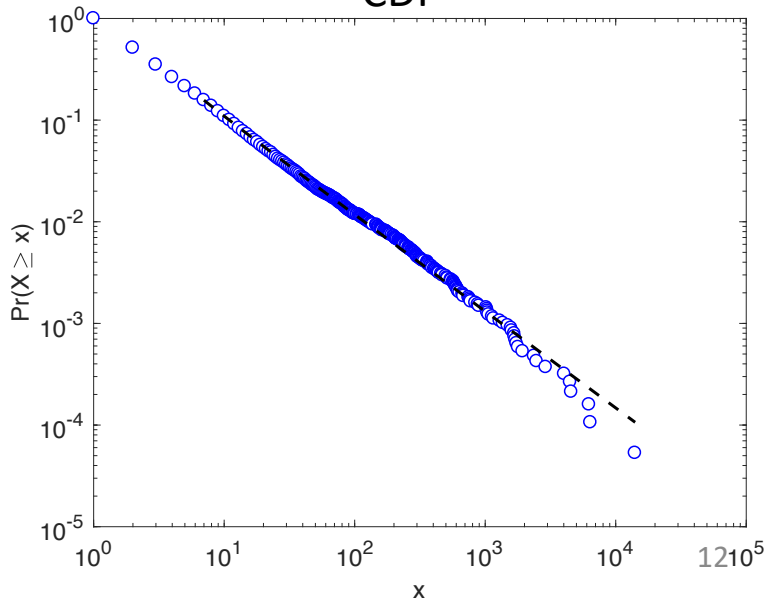
p= 0.49 ➔ p>0.1 fail to reject Ho

http://tuvalu.santafe.edu/~aaronc/powerlaws/data.htm
http://tuvalu.santafe.edu/~aaronc/powerlaws/

PDF



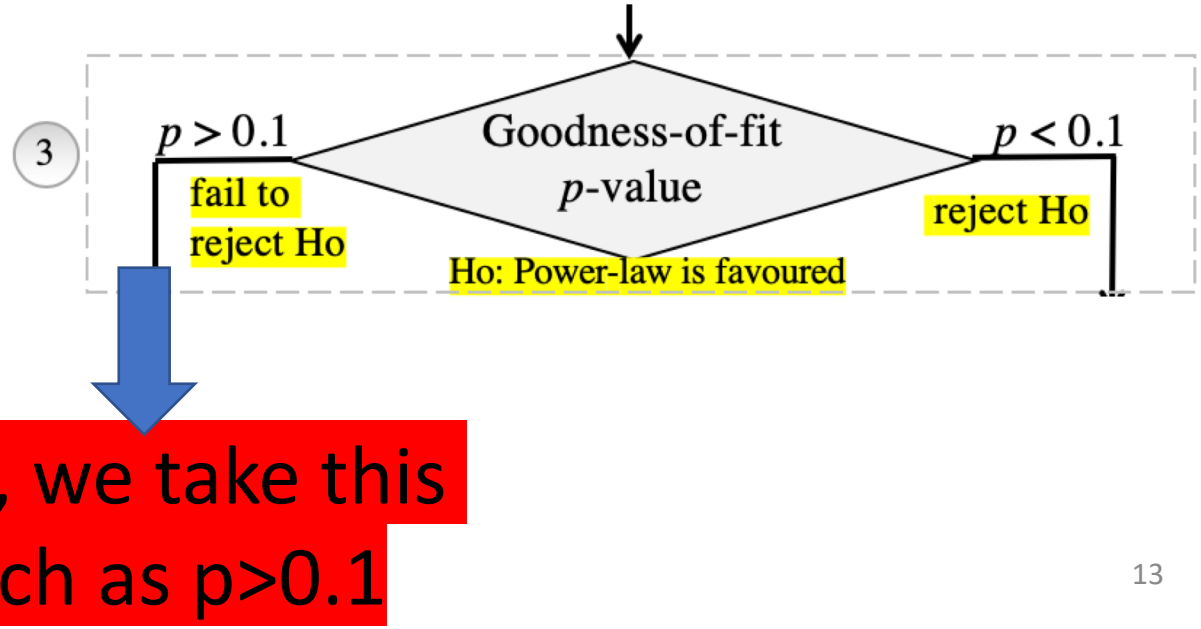
CDF



Ho: Power-law is favoured

p= 0.49 ➔ p>0.1 fail to reject Ho
(i.e., power-law is a good fit)

Now, is there any other good fit distribution?



Another example: blackouts dataset

Matlab

1. **Fitting a power-law distribution**

```
[alpha, xmin] = plfit(x)
```

2. **Estimating uncertainty in the fitted parameters**

```
[alphaun, xminun, nun]=plvar(x);
```

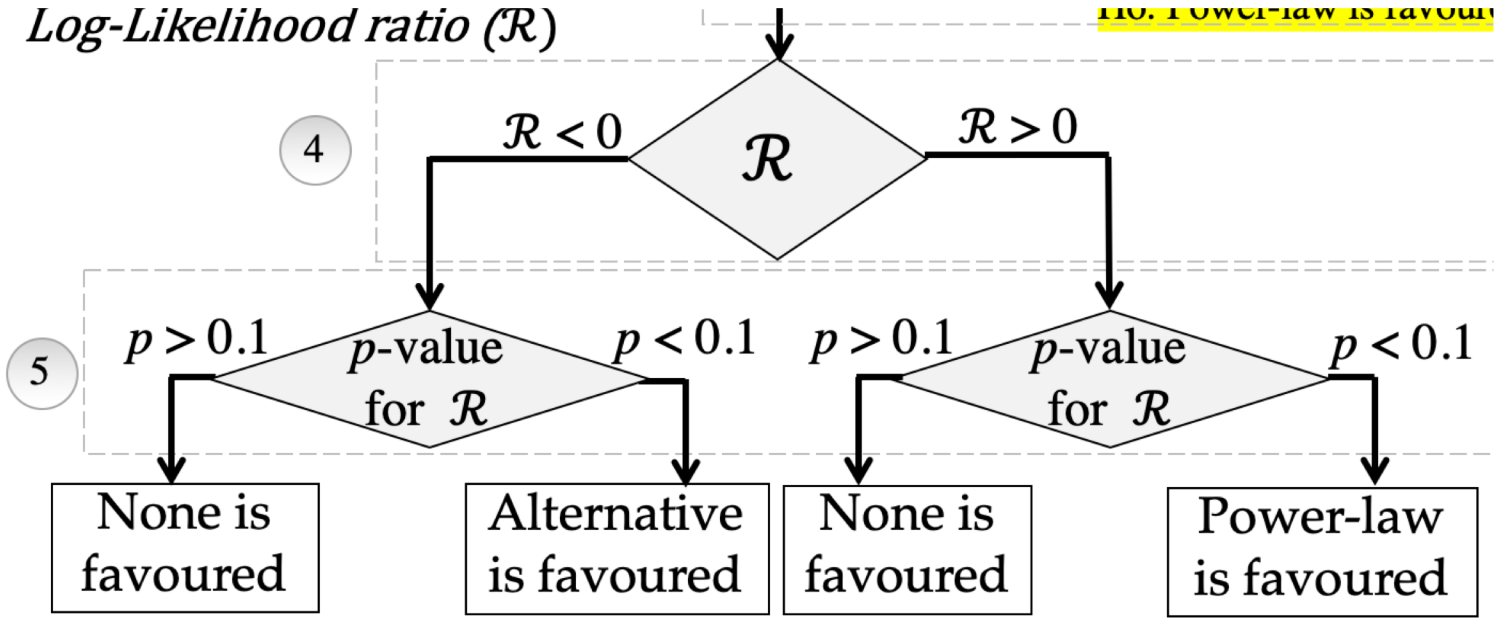
3. **Calculating p-value for fitted power-law model**

```
p =plpva(x, xmin )
```

quantity	n	$\langle x \rangle$	σ	$x_{\max}$	$\hat{x}_{\min}$	$\hat{\alpha}$	n_{tail}	p
blackouts ($\times 10^3$)	211	253.87	610.31	7500	230 ± 90	2.3(3)	59 ± 35	**0.62**

p = 0.62 ➔ p>0.1 fail to reject Ho

Step 4&5



`Use powerlaw.py function`

Likelihood ratio

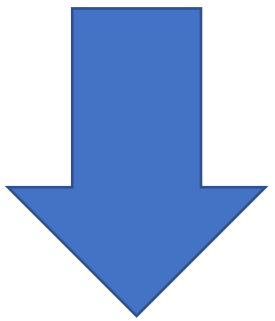
$$\Re, p = fit.\boldsymbol{distributionCompare}(powerlaw, alternative)$$

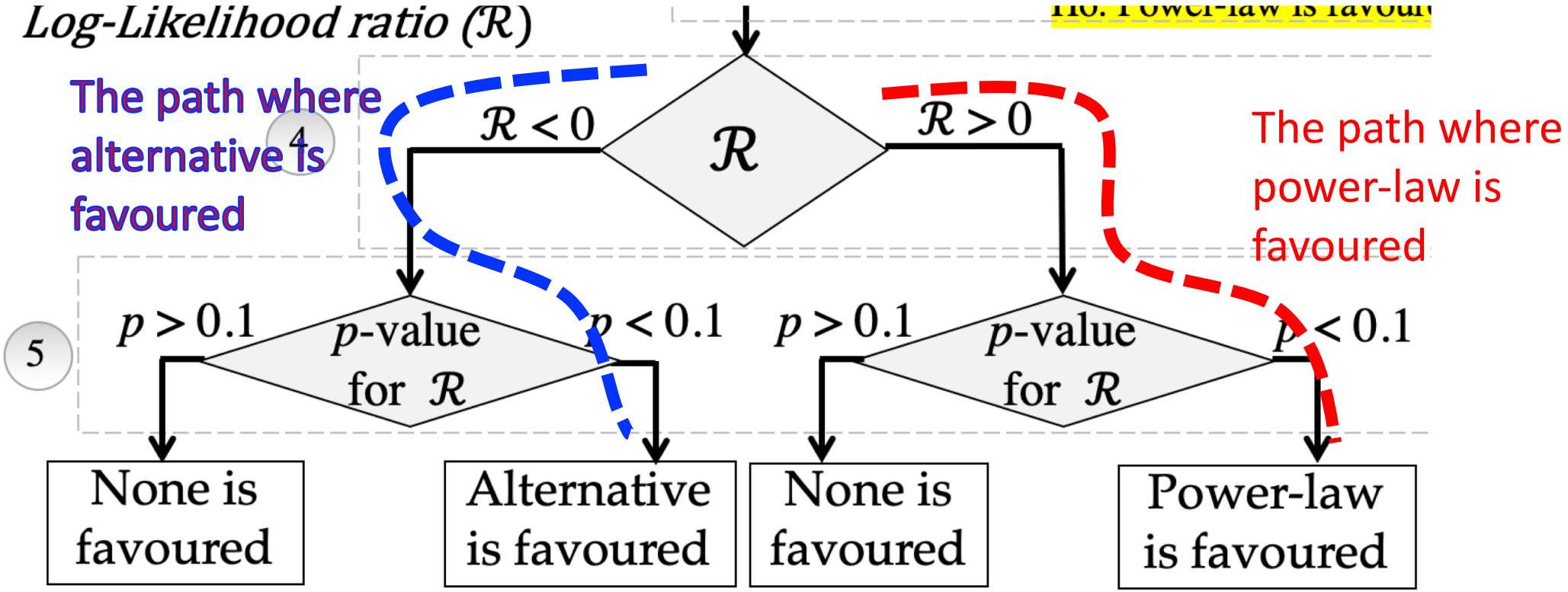
- **Weibull**
- **Lognormal**
- **Exponential**
- **..**

Likelihood ratio: $\Re = \frac{L_1}{L_2} = \frac{\prod_{i=1}^{n} p_1(x)}{\prod_{i=1}^{n} p_2(x)}$ → power-law likelihood function; → alternative likelihood function

- ***Log***−Likelihood ratio:
  - If $\Re > 0$, then the power-law is favoured.
  - If $\Re < 0$, then the alternative is favoured.
  - If $p < 0.1$, then the value of $\Re$ can be trusted.

i.e.





Example:

data set	p	Poisson LR	Poisson p	log-normal LR	log-normal p	exponential LR	exponential p	stretched exp. LR	stretched exp. p	power law + cut-off LR	power law + cut-off p	support for power law
words	0.49	4.43	0.00	0.395	0.69	9.09	0.00	4.13	0.00	−0.899	0.18	good
	PL	PL		none		PL		PL		none		

So, now we need to apply this test to get: $\mathfrak{R}$ and p values

$$\mathfrak{R}, p = fit.\boldsymbol{distributionCompare}(powerlaw, alternative)$$

Python

See this file: test.py

```python
# @Mohammed Alasmar, Univeristy of Sussex, 2018

import glob, os
import csv
import sys
import matplotlib.pyplot as plt
import powerlaw
import scipy.io
import numpy as np


os.chdir("/Users/mohammedalasmar/Desktop/power-law-codes/python")

print ("start ....")
mat = scipy.io.loadmat('blockouts.mat')
x=mat['x']
data= np.hstack(x)
#print (data)

fit = powerlaw.Fit(data)
print ("xmin = " ,fit.power_law.xmin)
print ("alpha = ", fit.power_law.alpha , "\n\n")


R_lognormal, p_lognormal = fit.distribution_compare('power_law', 'lognormal', normalized_ratio=True)
print ("lognormal: (LR , p) = (", R_lognormal,p_lognormal ,")\n\n")

R_exponential, p_exponential = fit.distribution_compare('power_law', 'exponential', normalized_ratio=True)
print ("exponential: (LR , p) = (", R_exponential,p_exponential ,")\n\n")

R_Weibull, p_Weibull = fit.distribution_compare('power_law', 'stretched_exponential',  normalized_ratio=True)
print ("Weibull: (LR , p) = (", R_Weibull,p_Weibull ,")\n\n")

R_cutoffPl, p_cutoffPl = fit.nested_distribution_compare('power_law', 'truncated_power_law', nested=True , nor
print ("truncated_power_law: (LR , p) = (", R_cutoffPl,p_cutoffPl ,")\n\n")
```

Results:

Console 1/A

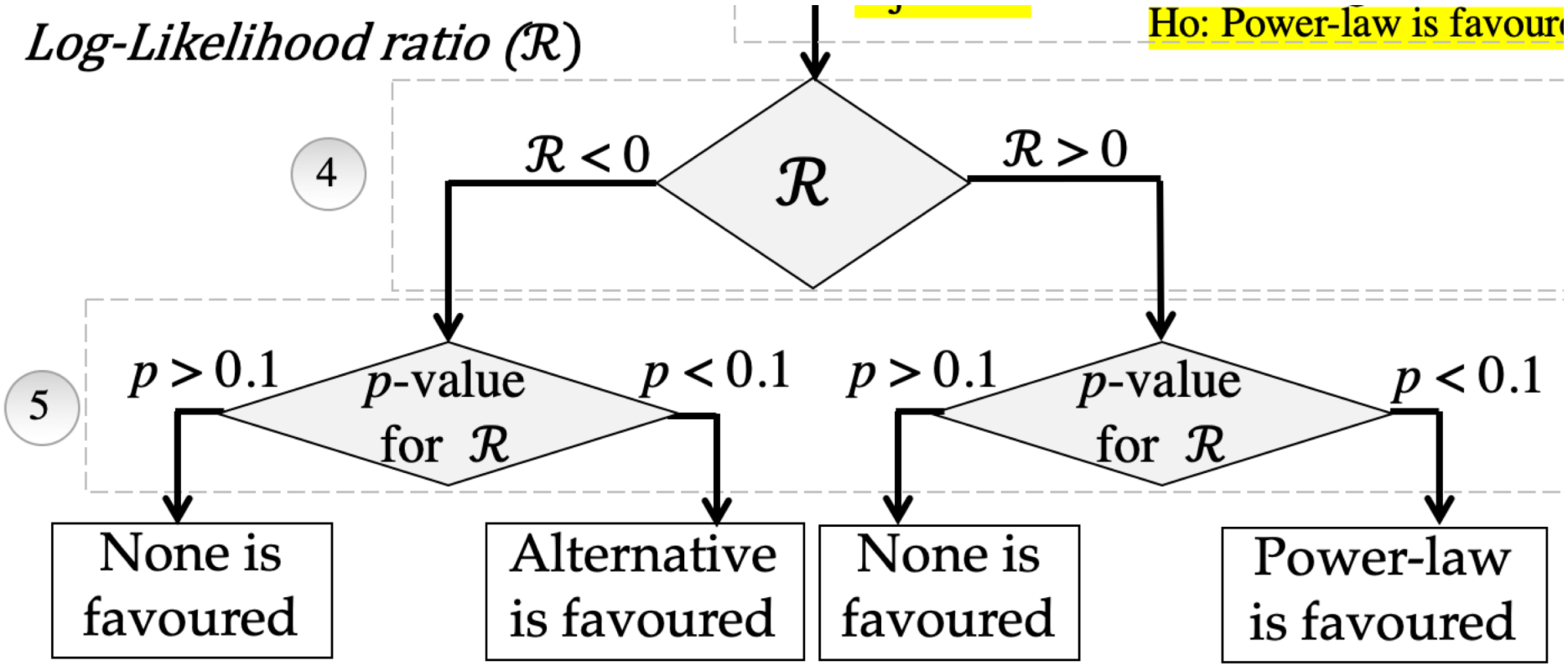
```
start ....
xmin =  230000.0
alpha =  2.272637219830288


lognormal: (LR , p) = ( -0.4157158274656643 0.677617958132759 )


exponential: (LR , p) = ( 1.4314804849576297 0.15229255604426487 )


Weibull: (LR , p) = ( -0.35821381794257084 0.720183307233291 )


truncated_power_law: (LR , p) = ( -0.6242890733211623 0.3821945857238682 )
```

See next slide

More examples

data set	power law p	log-normal LR	log-normal p	exponential LR	exponential p	stretched exp. LR	stretched exp. p	power law + cut-off LR	power law + cut-off p	support for power law
birds	**0.55** PL	-0.850 none	0.40	1.87 PL	**0.06**	-0.882 none	0.38	-1.24 none	0.12	moderate
blackouts	0.62 PL	-0.412 none	0.68	1.21 none	0.23	-0.417 none	0.68	-0.382 none	0.38	moderate
book sales	**0.66** PL	-0.267 none	0.79	2.70 PL	**0.01**	3.885 PL	**0.00**	-0.140 none	0.60	moderate
cities	**0.76** PL	-0.090 none	0.93	3.65 PL	**0.00**	0.204 ALT	0.84	-0.123 none	0.62	moderate
fires	0.05 ~~PL~~	-1.78 ALT	**0.08**	4.00 none	**0.00**	-1.82 ALT	**0.07**	-5.02 ALT	**0.00**	with cut-off
flares	**1.00** PL	-0.803 none	0.42	13.7 PL	**0.00**	-0.546 none	0.59	-4.52 ALT	**0.00**	with cut-off
HTTP	0.00 ~~PL~~	1.77 none	**0.08**	11.8 none	**0.00**	2.65 none	**0.01**	0.000 none	1.00	none
quakes	0.00 ~~PL~~	-7.14 ALT	**0.00**	11.6 none	**0.00**	-7.09 ALT	**0.00**	-24.4 ALT	**0.00**	with cut-off
religions	**0.42** PL	-0.073 none	0.94	1.59 none	0.11	1.75 PL	**0.08**	-0.167 none	0.56	moderate
surnames	**0.20** PL	-0.836 none	0.40	2.89 PL	**0.00**	-0.844 none	0.40	-1.36 ALT	**0.10**	with cut-off
wars	**0.20** PL	-0.737 none	0.46	3.68 PL	**0.00**	-0.767 none	0.44	-0.847 none	0.19	moderate
wealth	0.00 ~~PL~~	0.249 none	0.80	6.20 none	**0.00**	8.05 none	**0.00**	-0.142 none	0.59	none
web hits	0.00 ~~PL~~	-10.21 ALT	**0.00**	8.55 none	**0.00**	10.94 none	**0.00**	-74.66 ALT	**0.00**	with cut-off
web links	0.00 ~~PL~~	-2.24 ALT	**0.03**	25.3 none	**0.00**	-1.08 none	0.28	-21.2 ALT	**0.00**	with cut-off

~as in the previous slide



data set	p	Poisson LR	Poisson p	log-normal LR	log-normal p	exponential LR	exponential p	stretched exp. LR	stretched exp. p	power law + cut-off LR	power law + cut-off p	support for power law
words	0.49	4.43	0.00	0.395	0.69	9.09	0.00	4.13	0.00	−0.899	0.18	good
	PL	PL		none		PL		PL		none		

quantity	n	$\langle x \rangle$	σ	$x_{\max}$	$\hat{x}_{\min}$	$\hat{\alpha}$	n_{tail}	p
count of word use	18 855	11.14	148.33	14 086	7 ± 2	1.95(2)	2958 ± 987	**0.49**
protein interaction degree	1846	2.34	3.05	56	5 ± 2	3.1(3)	204 ± 263	**0.31**
metabolic degree	1641	5.68	17.81	468	4 ± 1	2.8(1)	748 ± 136	0.00
Internet degree	22 688	5.63	37.83	2583	21 ± 9	2.12(9)	770 ± 1124	**0.29**
telephone calls received	51 360 423	3.88	179.09	375 746	120 ± 49	2.09(1)	$102\,592 \pm 210\,147$	**0.63**
intensity of wars	115	15.70	49.97	382	2.1 ± 3.5	1.7(2)	70 ± 14	**0.20**
terrorist attack severity	9101	4.35	31.58	2749	12 ± 4	2.4(2)	547 ± 1663	**0.68**
HTTP size (kilobytes)	226 386	7.36	57.94	10 971	36.25 ± 22.74	2.48(5)	6794 ± 2232	0.00
species per genus	509	5.59	6.94	56	4 ± 2	2.4(2)	233 ± 138	**0.10**
bird species sightings	591	3384.36	10 952.34	138 705	6679 ± 2463	2.1(2)	66 ± 41	**0.55**
blackouts ($\times 10^3$)	211	253.87	610.31	7500	230 ± 90	2.3(3)	59 ± 35	**0.62**
sales of books ($\times 10^3$)	633	1986.67	1396.60	19 077	2400 ± 430	3.7(3)	139 ± 115	**0.66**
population of cities ($\times 10^3$)	19 447	9.00	77.83	8 009	52.46 ± 11.88	2.37(8)	580 ± 177	**0.76**
email address books size	4581	12.45	21.49	333	57 ± 21	3.5(6)	196 ± 449	**0.16**
forest fire size (acres)	203 785	0.90	20.99	4121	6324 ± 3487	2.2(3)	521 ± 6801	0.05
solar flare intensity	12 773	689.41	6520.59	231 300	323 ± 89	1.79(2)	1711 ± 384	**1.00**
quake intensity ($\times 10^3$)	19 302	24.54	563.83	63 096	0.794 ± 80.198	1.64(4)	$11\,697 \pm 2159$	0.00
religious followers ($\times 10^6$)	103	27.36	136.64	1050	3.85 ± 1.60	1.8(1)	39 ± 26	**0.42**
freq. of surnames ($\times 10^3$)	2753	50.59	113.99	2502	111.92 ± 40.67	2.5(2)	239 ± 215	**0.20**
net worth (mil. USD)	400	2388.69	4 167.35	46 000	900 ± 364	2.3(1)	302 ± 77	0.00
citations to papers	415 229	16.17	44.02	8904	160 ± 35	3.16(6)	3455 ± 1859	**0.20**
papers authored	401 445	7.21	16.52	1416	133 ± 13	4.3(1)	988 ± 377	**0.90**
hits to web sites	119 724	9.83	392.52	129 641	2 ± 13	1.81(8)	$50\,981 \pm 16\,898$	0.00
links to web sites	241 428 853	9.15	106 871.65	1 199 466	3684 ± 151	2.336(9)	$28\,986 \pm 1560$	0.00

TABLE 6.1

Basic parameters of the data sets described in this section, along with their power-law fits and the corresponding p-value (statistically significant values are denoted in ***bold****).*

Likelihood ratio test

	power law	log-normal		exponential		stretched exp.		power law + cut-off		support for
data set	p	LR	p	LR	p	LR	p	LR	p	power law
birds	**0.55**	-0.850	0.40	1.87	**0.06**	-0.882	0.38	-1.24	0.12	moderate
blackouts	**0.62**	-0.412	0.68	1.21	0.23	-0.417	0.68	-0.382	0.38	moderate
book sales	**0.66**	-0.267	0.79	2.70	**0.01**	3.885	**0.00**	-0.140	0.60	moderate
cities	**0.76**	-0.090	0.93	3.65	**0.00**	0.204	0.84	-0.123	0.62	moderate
fires	0.05	-1.78	**0.08**	4.00	**0.00**	-1.82	**0.07**	-5.02	**0.00**	with cut-off
flares	**1.00**	-0.803	0.42	13.7	**0.00**	-0.546	0.59	-4.52	**0.00**	with cut-off
HTTP	0.00	1.77	**0.08**	11.8	**0.00**	2.65	**0.01**	0.000	1.00	none
quakes	0.00	-7.14	**0.00**	11.6	**0.00**	-7.09	**0.00**	-24.4	**0.00**	with cut-off
religions	**0.42**	-0.073	0.94	1.59	0.11	1.75	**0.08**	-0.167	0.56	moderate
surnames	**0.20**	-0.836	0.40	2.89	**0.00**	-0.844	0.40	-1.36	**0.10**	with cut-off
wars	**0.20**	-0.737	0.46	3.68	**0.00**	-0.767	0.44	-0.847	0.19	moderate
wealth	0.00	0.249	0.80	6.20	**0.00**	8.05	**0.00**	-0.142	0.59	none
web hits	0.00	-10.21	**0.00**	8.55	**0.00**	10.94	**0.00**	-74.66	**0.00**	with cut-off
web links	0.00	-2.24	**0.03**	25.3	**0.00**	-1.08	0.28	-21.2	**0.00**	with cut-off

TABLE 6.2

data set	power law p	log-normal LR	log-normal p	exponential LR	exponential p	stretched exp. LR	stretched exp. p	power law + cut-off LR	power law + cut-off p	support for power law
birds	**0.55**	-0.850	0.40	1.87	**0.06**	-0.882	0.38	-1.24	0.12	moderate
blackouts	**0.62**	-0.412	0.68	1.21	0.23	-0.417	0.68	-0.382	0.38	moderate
book sales	**0.66**	-0.267	0.79	2.70	**0.01**	3.885	**0.00**	-0.140	0.60	moderate
cities	**0.76**	-0.090	0.93	3.65	**0.00**	0.204	0.84	-0.123	0.62	moderate
fires	0.05	-1.78	**0.08**	4.00	**0.00**	-1.82	**0.07**	-5.02	**0.00**	with cut-off
flares	**1.00**	-0.803	0.42	13.7	**0.00**	-0.546	0.59	-4.52	**0.00**	with cut-off
HTTP	0.00	1.77	**0.08**	11.8	**0.00**	2.65	**0.01**	0.000	1.00	none
quakes	0.00	-7.14	**0.00**	11.6	**0.00**	-7.09	**0.00**	-24.4	**0.00**	with cut-off
religions	**0.42**	-0.073	0.94	1.59	0.11	1.75	**0.08**	-0.167	0.56	moderate
surnames	**0.20**	-0.836	0.40	2.89	**0.00**	-0.844	0.40	-1.36	**0.10**	with cut-off
wars	**0.20**	-0.737	0.46	3.68	**0.00**	-0.767	0.44	-0.847	0.19	moderate
wealth	0.00	0.249	0.80	6.20	**0.00**	8.05	**0.00**	-0.142	0.59	none
web hits	0.00	-10.21	**0.00**	8.55	**0.00**	10.94	**0.00**	-74.66	**0.00**	with cut-off
web links	0.00	-2.24	**0.03**	25.3	**0.00**	-1.08	0.28	-21.2	**0.00**	with cut-off

data set	p	Poisson LR	Poisson p	log-normal LR	log-normal p	exponential LR	exponential p	stretched exp. LR	stretched exp. p	power law + cut-off LR	power law + cut-off p	support for power law
Internet	**0.29**	5.31	**0.00**	−0.807	0.42	6.49	**0.00**	0.493	0.62	−1.97	**0.05**	with cut-off
calls	**0.63**	17.9	**0.00**	−2.03	**0.04**	35.0	**0.00**	14.3	**0.00**	−30.2	**0.00**	with cut-off
citations	**0.20**	6.54	**0.00**	−0.141	0.89	5.91	**0.00**	1.72	**0.09**	−0.007	0.91	moderate
email	**0.16**	4.65	**0.00**	−1.10	0.27	0.639	0.52	−1.13	0.26	−1.89	**0.05**	with cut-off
metabolic	0.00	3.53	**0.00**	−1.05	0.29	5.59	**0.00**	3.66	**0.00**	0.000	1.00	none
papers	**0.90**	5.71	**0.00**	−0.091	0.93	3.08	**0.00**	0.709	0.48	−0.016	0.86	moderate
proteins	**0.31**	3.05	**0.00**	−0.456	0.65	2.21	**0.03**	0.055	0.96	−0.414	0.36	moderate
species	**0.10**	5.04	**0.00**	−1.63	0.10	2.39	**0.02**	−1.59	0.11	−3.80	**0.01**	with cut-off
terrorism	**0.68**	1.81	**0.07**	−0.278	0.78	2.457	**0.01**	0.772	0.44	−0.077	0.70	moderate
words	**0.49**	4.43	**0.00**	0.395	0.69	9.09	**0.00**	4.13	**0.00**	−0.899	0.18	good

Regenerating the results in Tables 6.2&6.3

Python

```
import powerlaw
import scipy.io
#mat = scipy.io.loadmat('blockouts.mat')
#mat = scipy.io.loadmat('cities.mat')
#mat = scipy.io.loadmat('words.mat')
mat = scipy.io.loadmat('loc3.mat')

x = mat['x']
data=x[0]
print (data)


fit = powerlaw.Fit(data)
print ("xmin" ,fit.power_law.xmin)
print ("alpha", fit.power_law.alpha)

#import plpva
#plpva.plpva(data,fit.power_law.xmin)

print (" ")

R, p = fit.distribution_compare('power_law', 'lognormal_positive', normalized_ratio=True)
print ("lognormal_positive", R,p)
print (" ")

R, p = fit.distribution_compare('power_law', 'lognormal', normalized_ratio=True)
print ("lognormal", R,p)
print (" ")

R, p = fit.distribution_compare('power_law', 'exponential', normalized_ratio=True)
print ("exponential", R,p)

print (" ")

R, p = fit.distribution_compare('power_law', 'stretched_exponential',  normalized_ratio=True)
print ("stretched_exponential", R,p)
print (" ")


R, p = fit.nested_distribution_compare('power_law', 'truncated_power_law',  normalized_ratio=False)
print ("truncated_power_law", R,p)
```