Principles of Database Systems

V. Megalooikonomou

Fractals and Databases

(based on notes by C. Faloutsos at CMU)

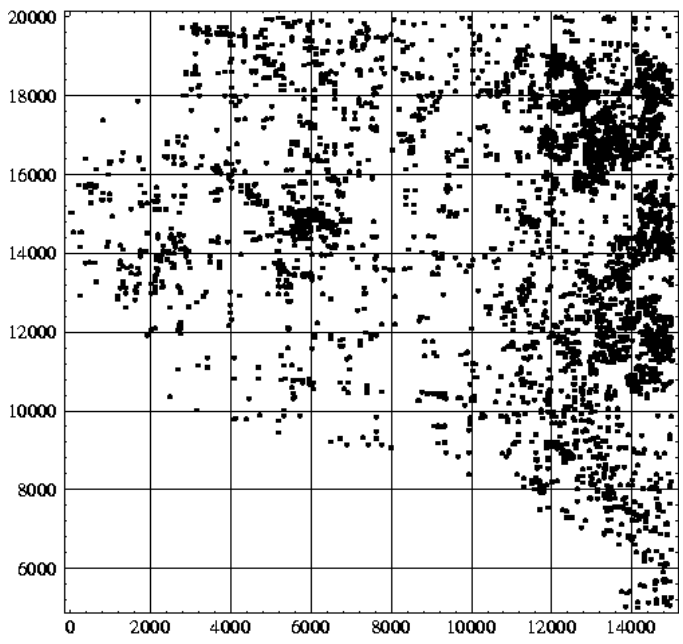
Indexing - Detailed outline

- fractals
  - intro
  - applications

Intro to fractals - outline

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner’s guide
- Appendix: gory details - boxcounting plots

Problem #1: GIS - points

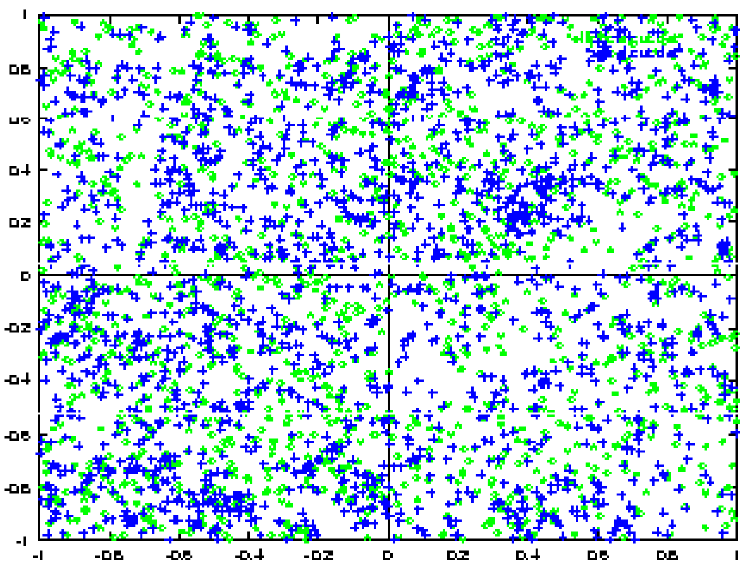


Road end-points of Montgomery county:

- Q1: how many d.a. for an R-tree?
- Q2 : distribution?
  - not uniform
  - not Gaussian
  - no rules??

Problem #2 - spatial d.m.

Galaxies (Sloan Digital Sky Survey -B. Nichol)



- ‘spiral’ and ‘elliptical’ galaxies

(stores and households ...)

- patterns?

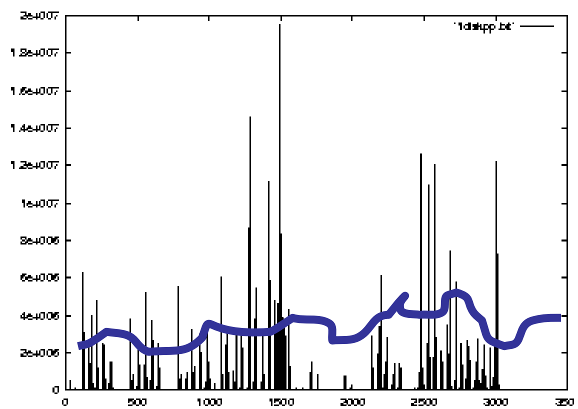
- attraction / repulsion?

- how many ‘spi’ within r from an ‘ell’?

Problem #3: traffic

- disk trace (from HP - J. Wilkes); Web traffic - fit a model

#bytes



time

Poisson

- how many explosions to expect?

- queue length distr.?

Common answer:

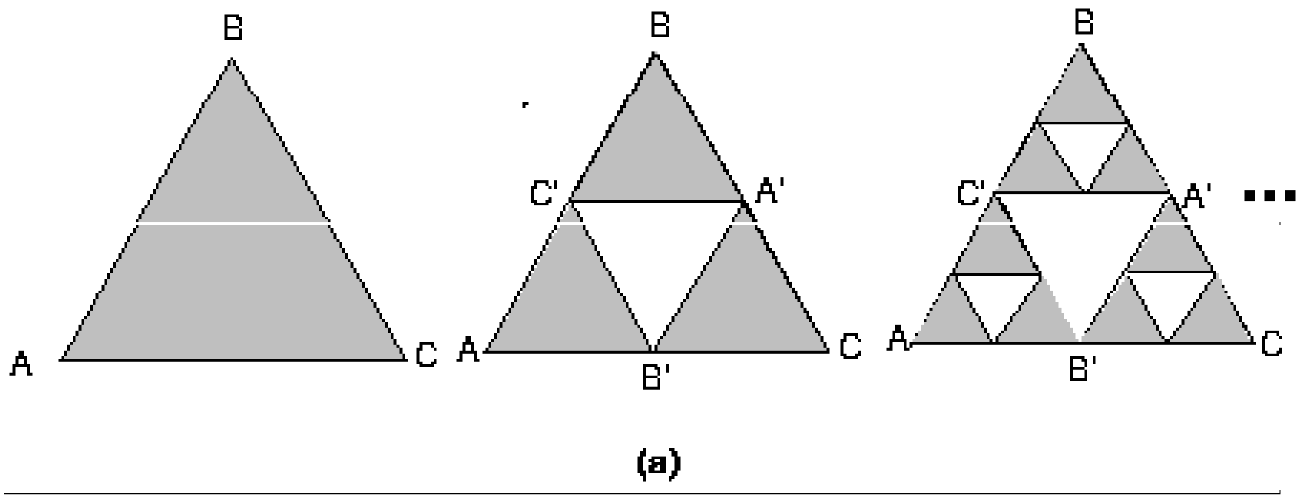
- Fractals / self-similarities / power laws
- Seminal works from Hilbert, Minkowski, Cantor, Mandelbrot, (Hausdorff, Lyapunov, Ken Wilson, ...)

Road map

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots

What is a fractal?

= self-similar point set, e.g., Sierpinski triangle:



zero area;

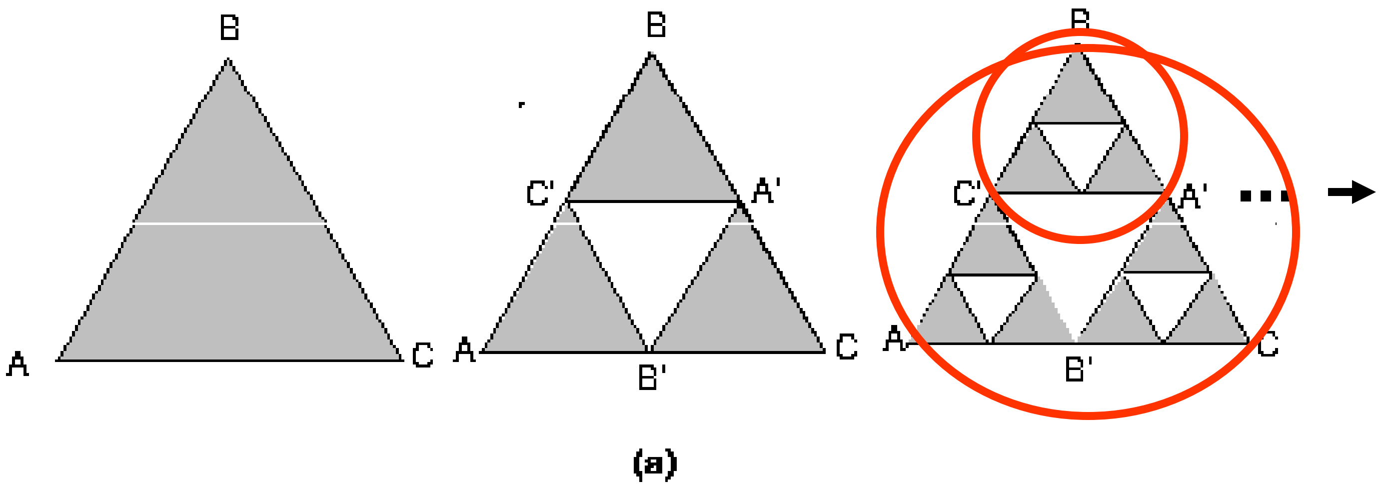
infinite perimeter!

Definitions (cont'd)

- Paradox: Infinite perimeter ; Zero area!
- 'dimensionality': between 1 and 2
- actually: Log(3)/Log(2) = 1.58...

Dfn of fd:

ONLY for a perfectly self-similar point set:



zero area;

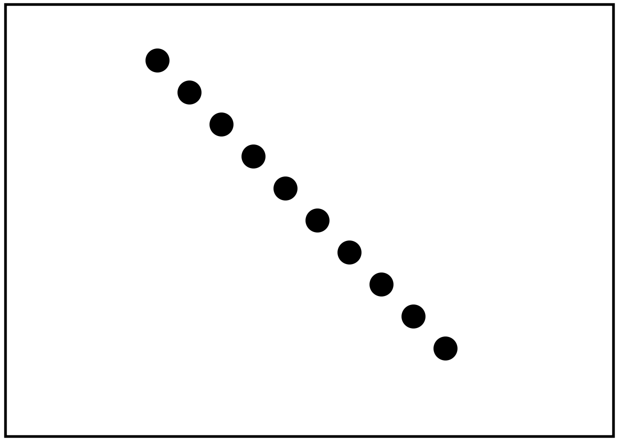
infinite length!

$$= \log(n) / \log(f) = \log(3) / \log(2) = \mathbf{1.58}$$

a perfectly self-similar object with n similar pieces each scaled down by a factor f

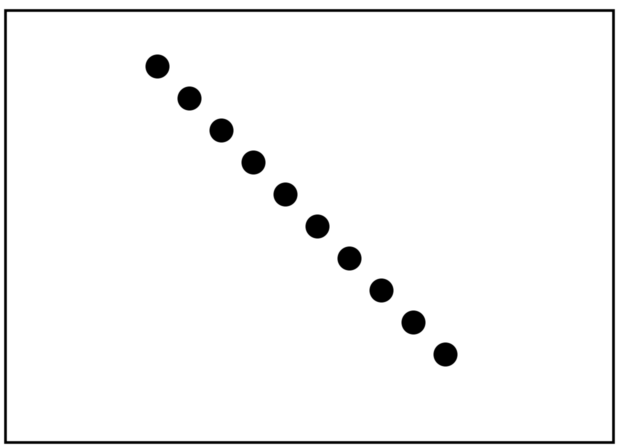
Intrinsic ('fractal') dimension

- Q: fractal dimension of a line?
- A: 1 (= log(2)/log(2)!)



Intrinsic ('fractal') dimension

- Q: dfn for a given set of points?



x	y
5	1
4	2
3	3
2	4

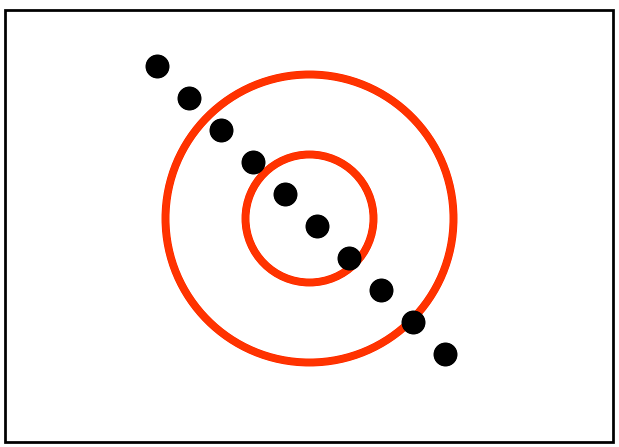
Intrinsic ('fractal') dimension

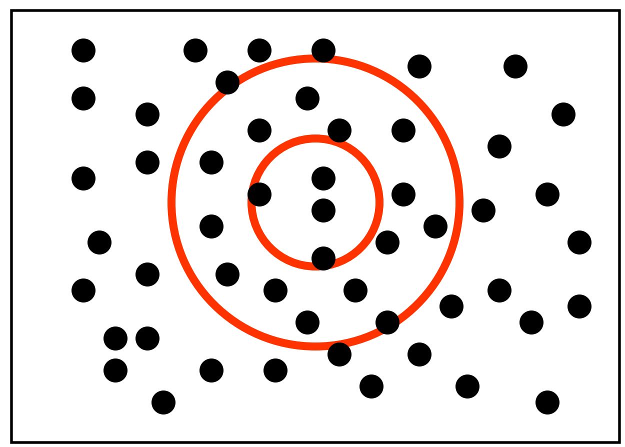
- Q: fractal dimension of a line?
- A: nn (<= r) ~ r^1

('power law': y=x^a)

- Q: fd of a plane?
- A: nn (<= r) ~ r^2

fd== slope of (log(nn) vs log(r))





Intrinsic ('fractal') dimension

- Algorithm, to estimate it?

Notice

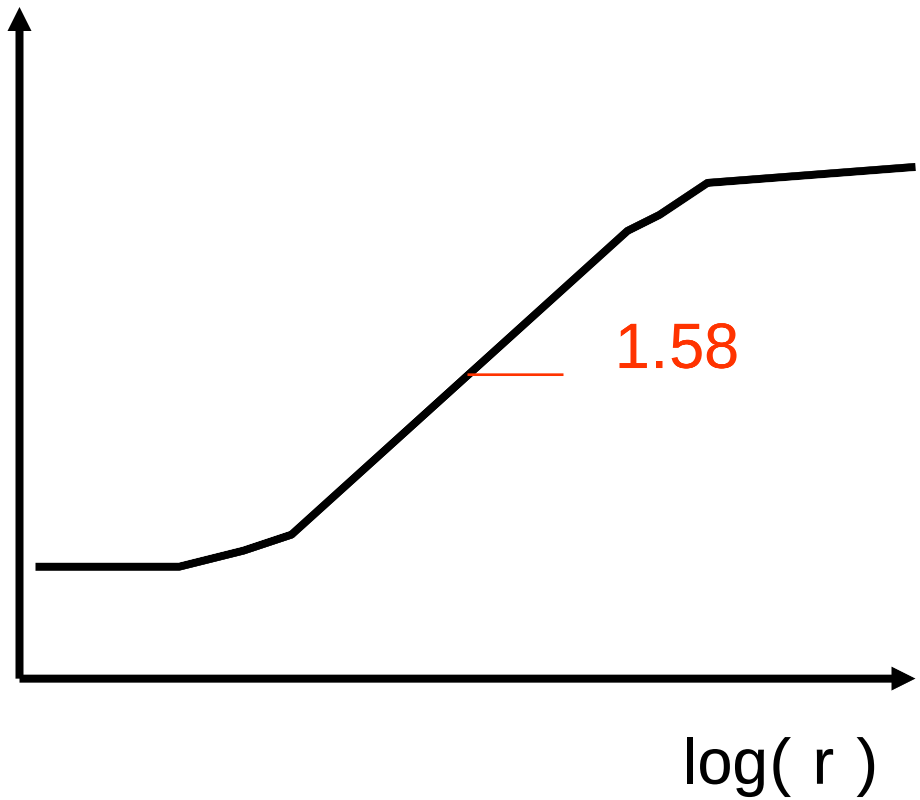
- *avg nn(<=r)* is exactly

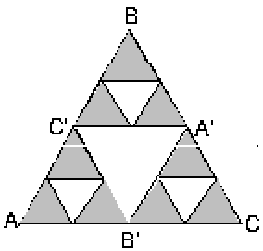
  *tot#pairs(<=r) / (2*N)*

including 'mirror' pairs

Sierpinsky triangle

== 'correlation integral'





Observations:

- Euclidean objects have **integer** fractal dimensions
  - point: 0
  - lines and smooth curves: 1
  - smooth surfaces: 2
- fractal dimension -> roughness of the periphery

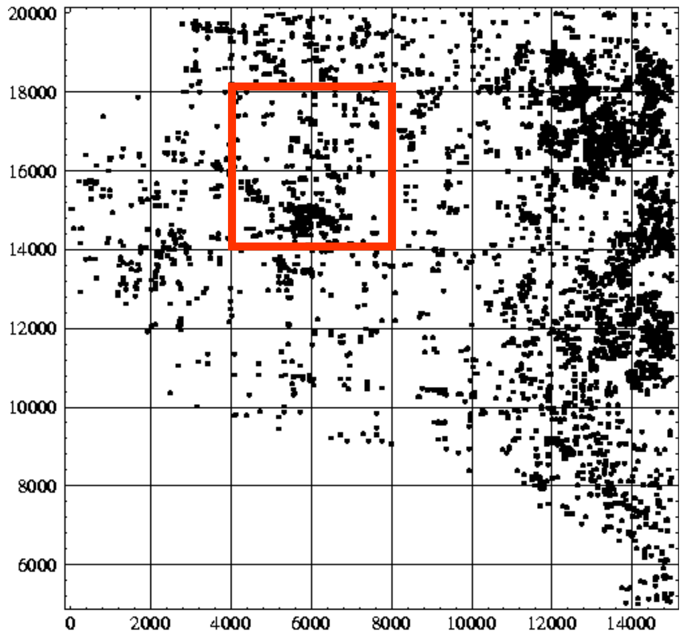
Important properties

- fd = embedding dimension -> uniform pointset
- a point set may have several fd, depending on scale

Road map

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots

Problem #1: GIS points

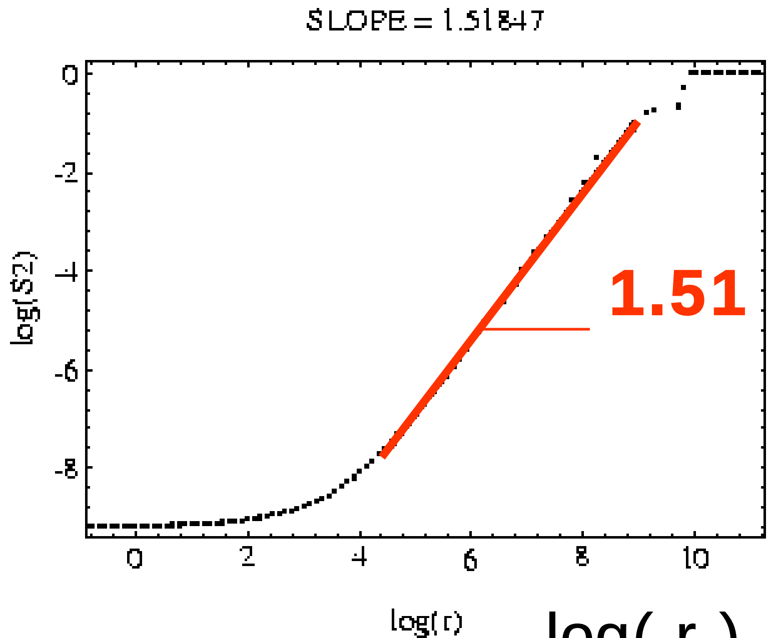


Cross-roads of Montgomery county:

- any rules?

Solution #1

log(#pairs(within <= r))



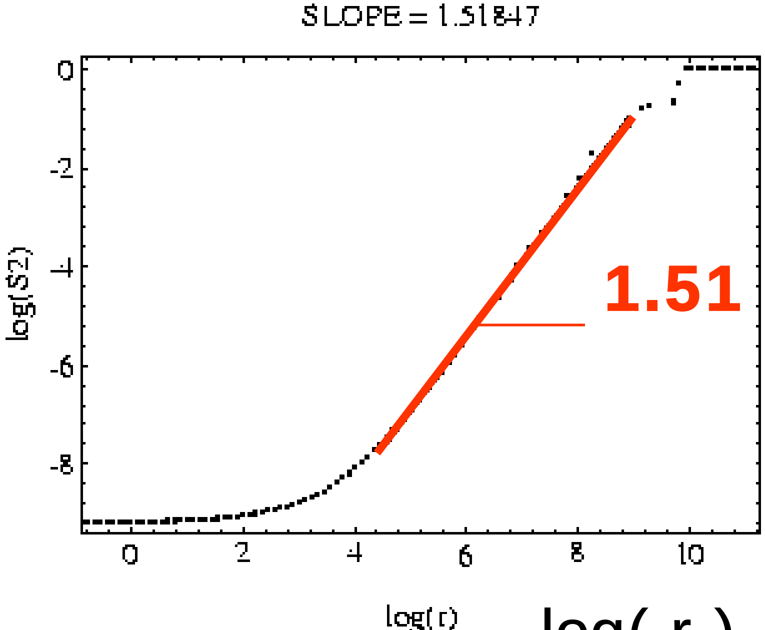
log(r)

A: self-similarity ->

- <=> fractals
- <=> scale-free
- <=> power-laws (y=x^a, F=C*r^(-2))
- avg#neighbors(<= r) = r^D

Solution #1

log(#pairs(within <= r))



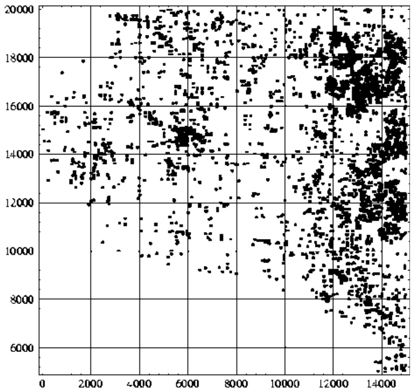
log(r)

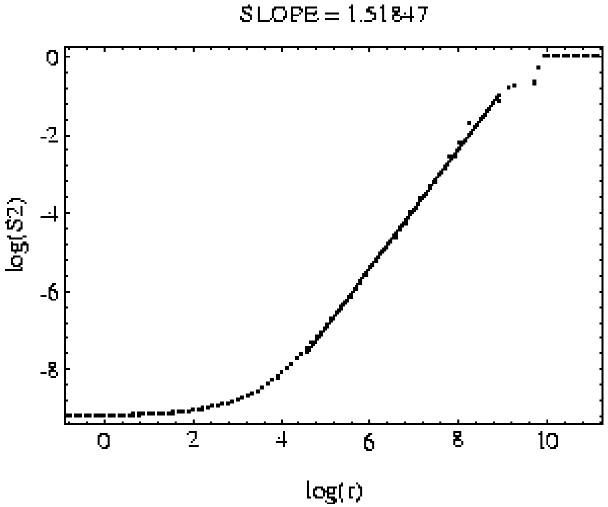
A: self-similarity

- avg#neighbors(<= r) ~ r^(1.51)

Examples:MG county

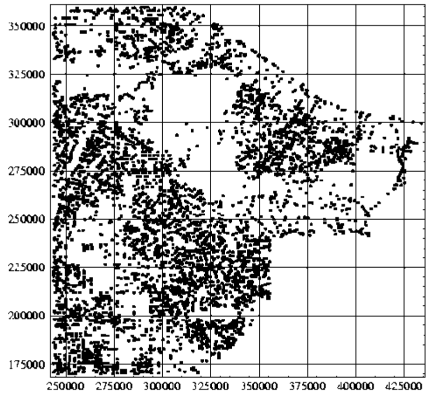
- Montgomery County of MD (road end-points)

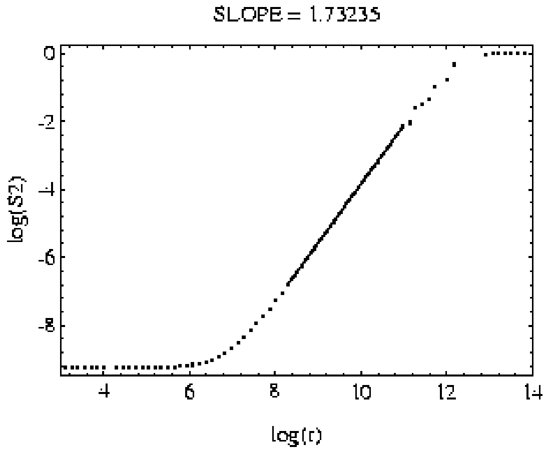




Examples:LB county

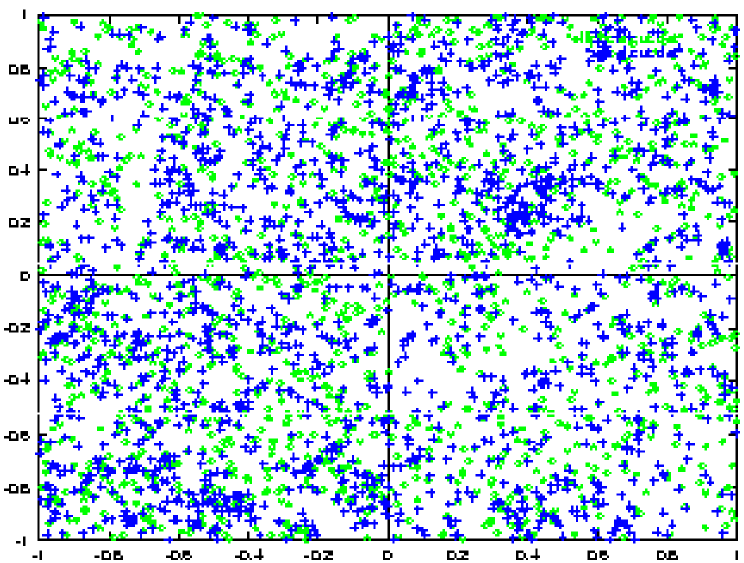
- Long Beach county of CA (road end-points)



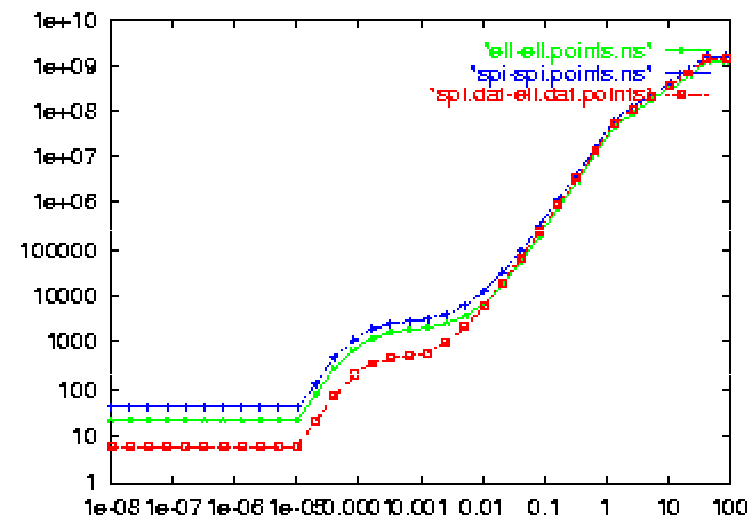


Solution#2: spatial d.m.

Galaxies (‘BOPS’ plot - [sigmod2000])



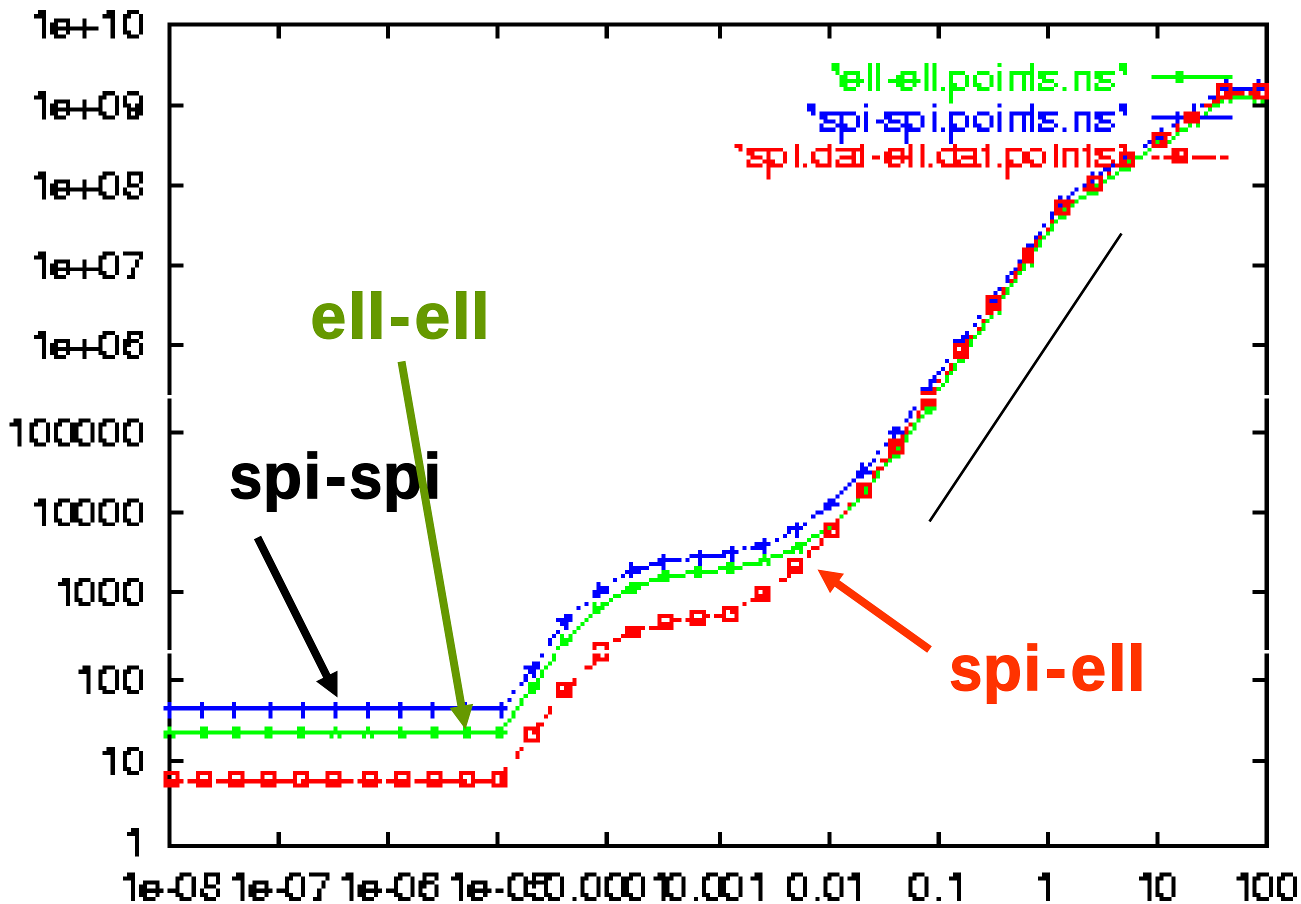
log(#pairs)



log(r)

Solution#2: spatial d.m.

log(#pairs within <=r)



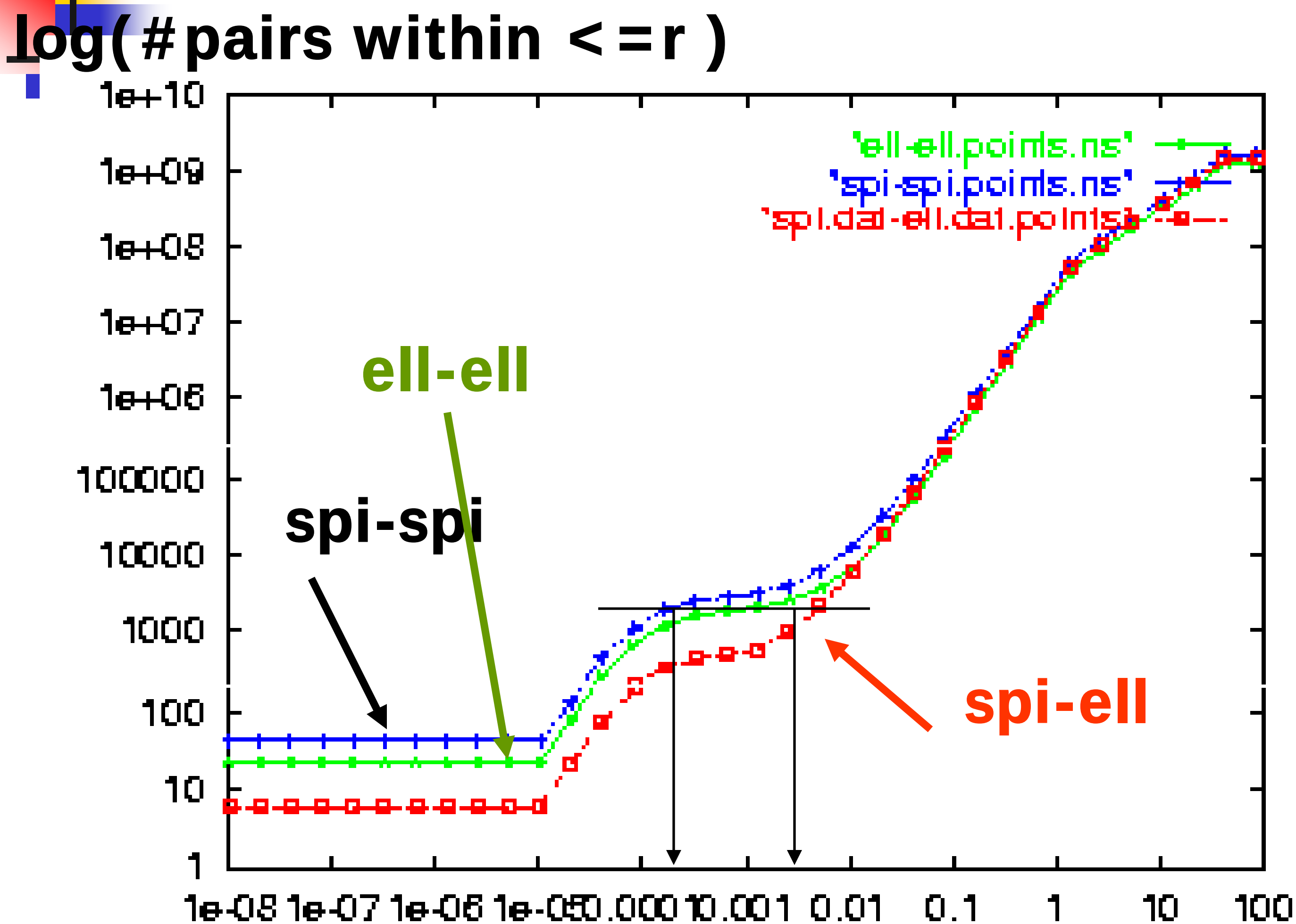
- 1.8 slope

- plateau!

- repulsion!

log(r)

spatial d.m.



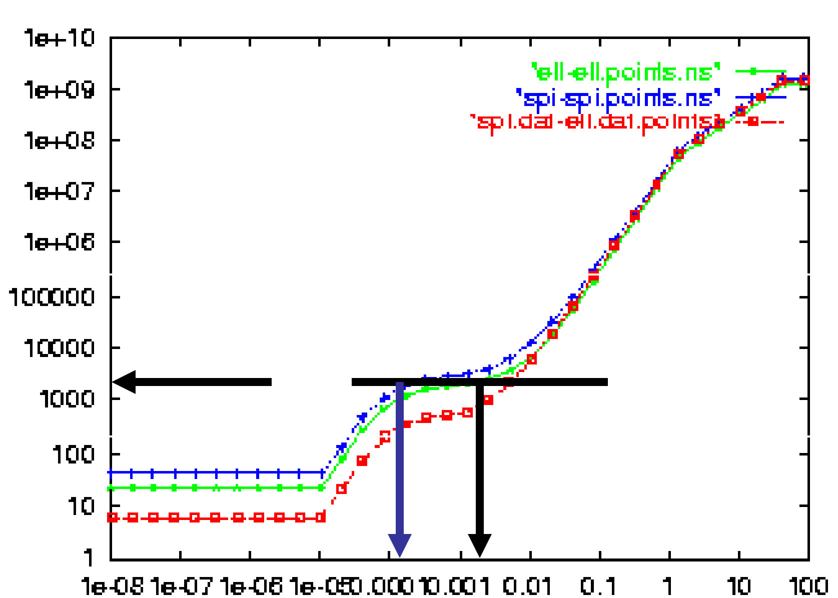
- 1.8 slope

- plateau!

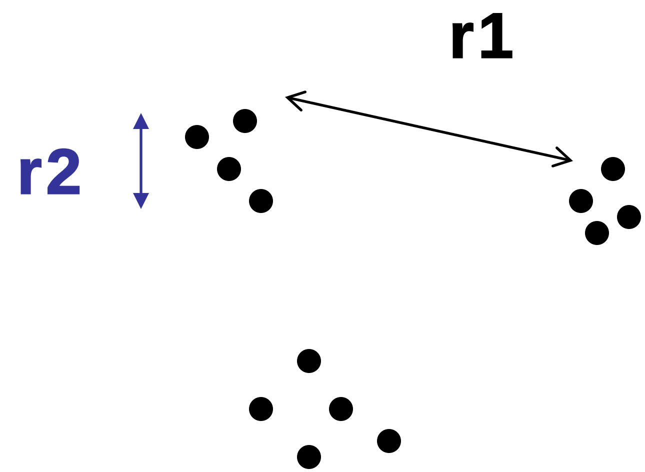
- repulsion!

log(r)

spatial d.m.

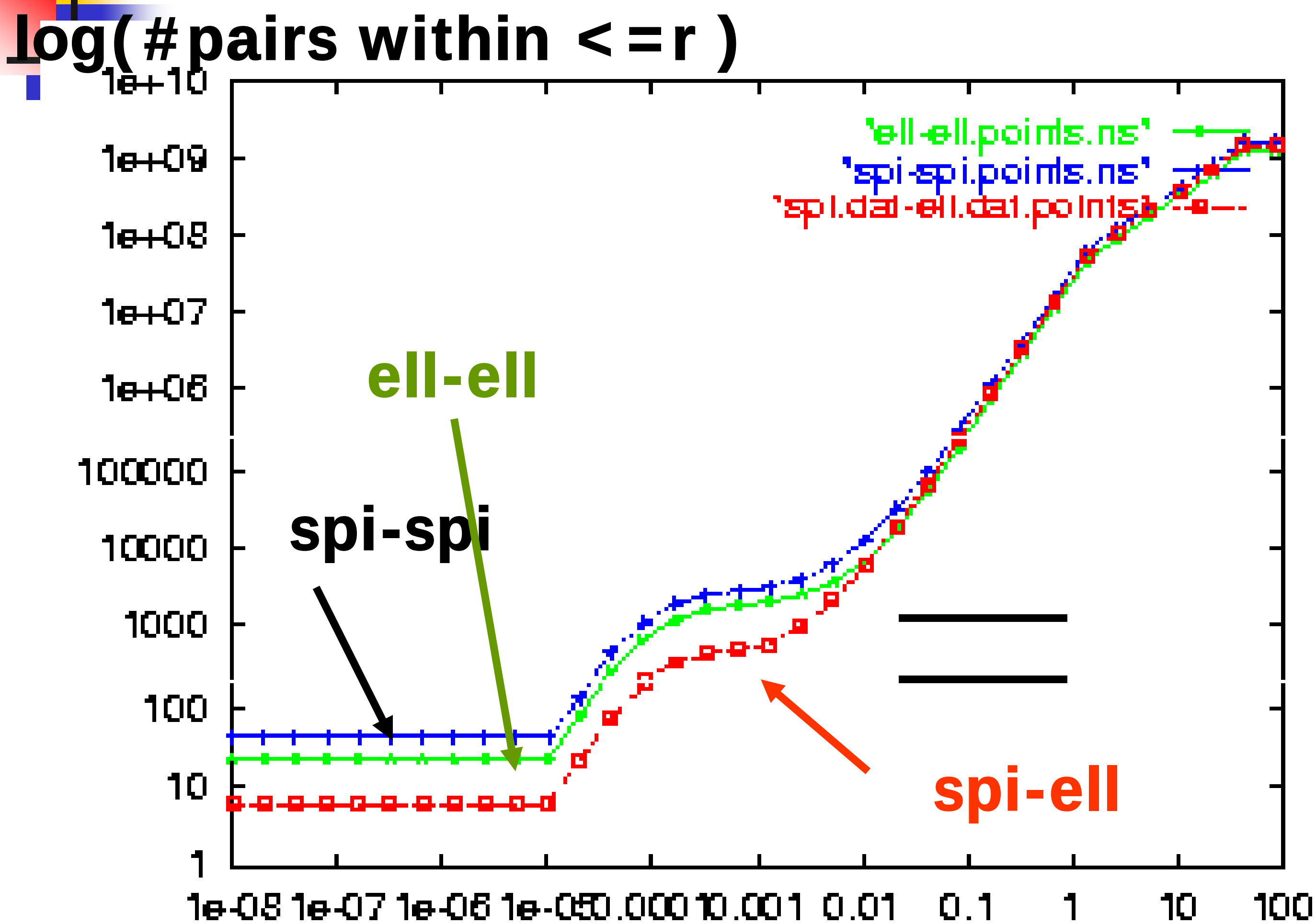


r2 **r1**



Heuristic on choosing # of clusters

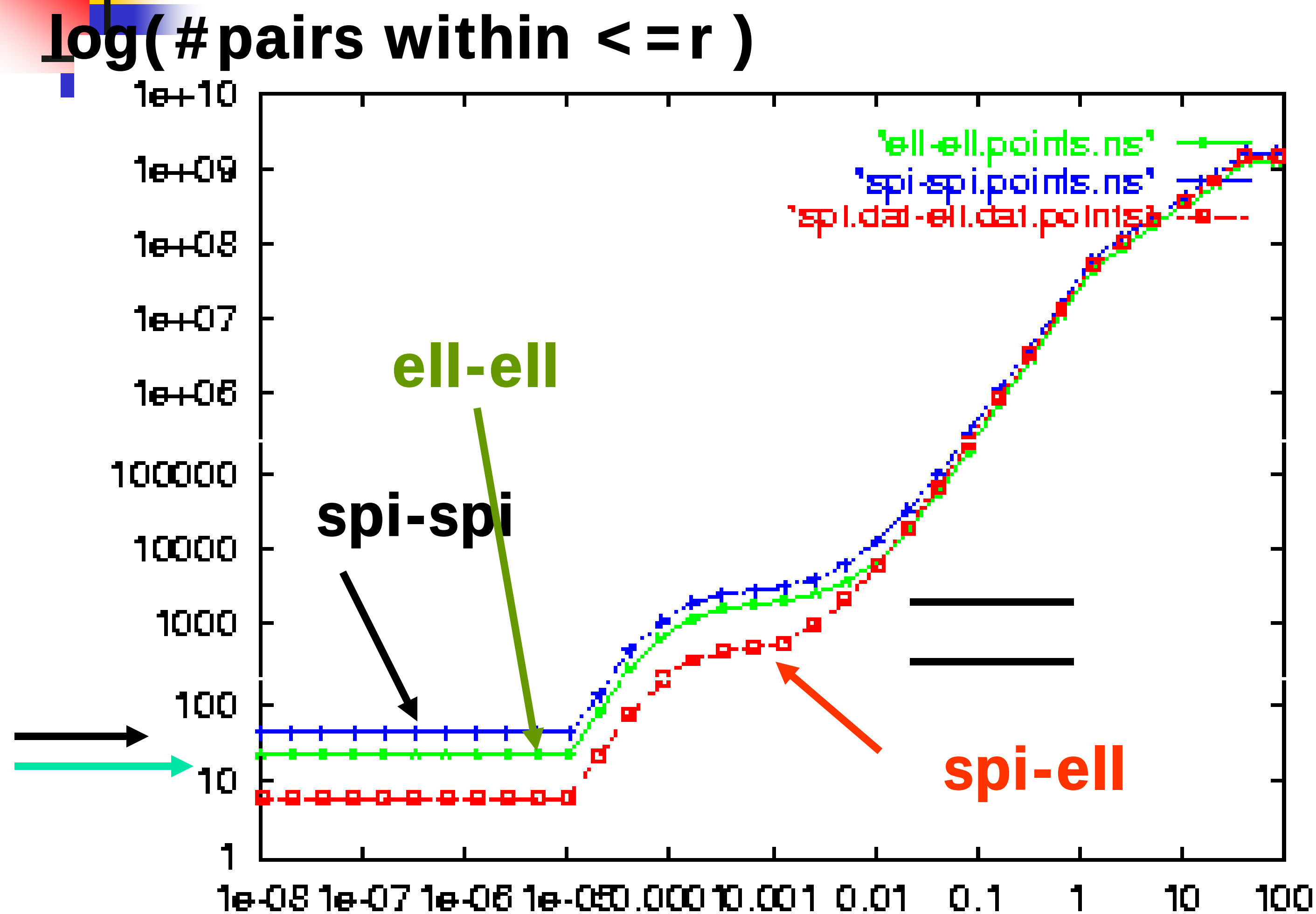
spatial d.m.



log(r)

- 1.8 slope
- plateau!
- repulsion!

spatial d.m.



- 1.8 slope

- plateau!

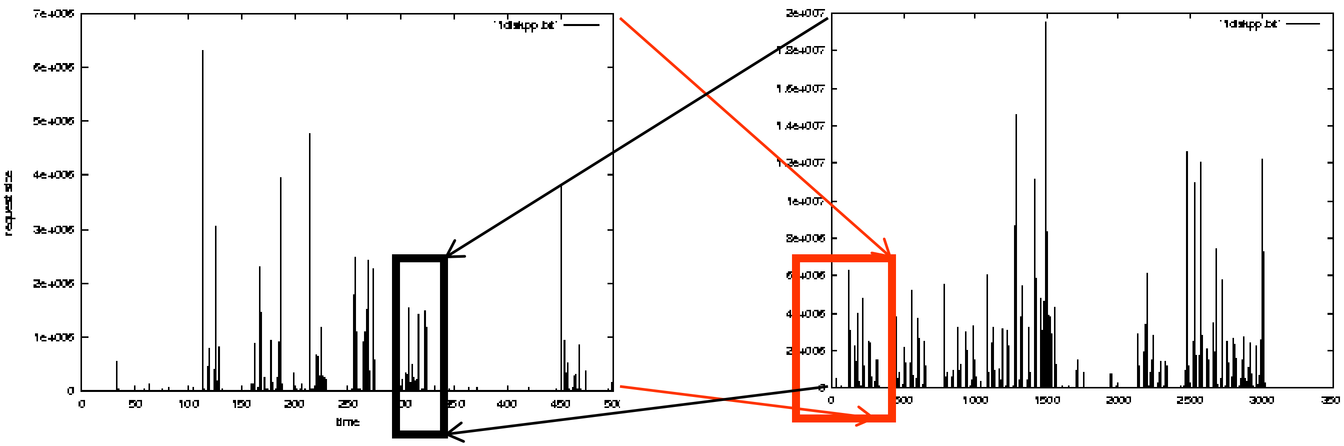
-repulsion

-duplicates !!

Solution #3: traffic

- disk traces: self-similar:

#bytes

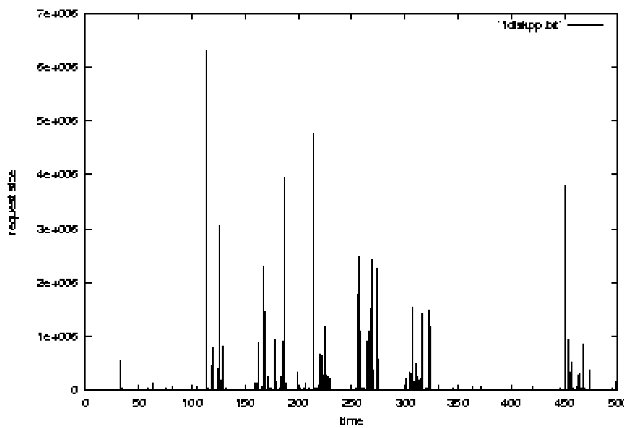


time

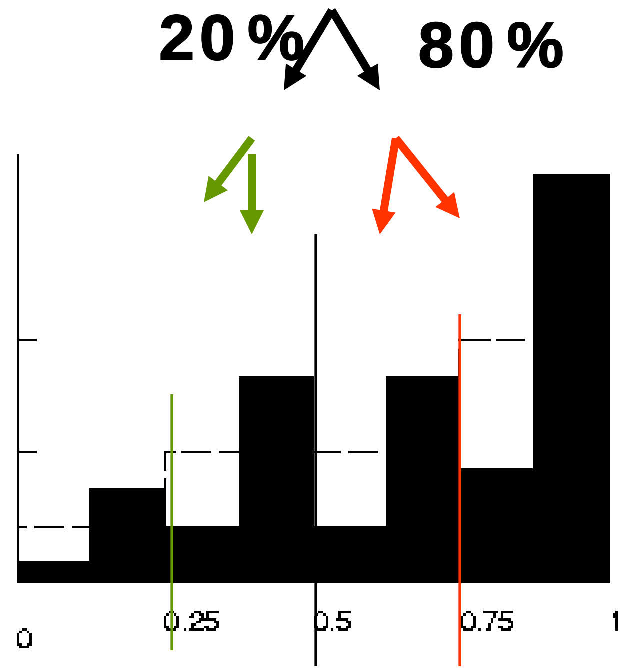
Solution #3: traffic

- disk traces (80-20 'law' = 'multifractal')

#bytes



time



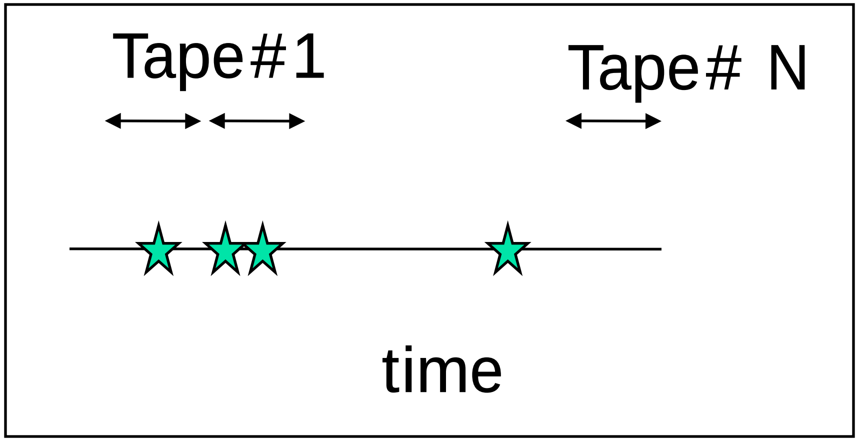
Solution#3: traffic

Clarification:

- fractal: a set of points that is self-similar
- multifractal: a probability density function that is self-similar

Many other time-sequences are bursty/clustered: (such as?)

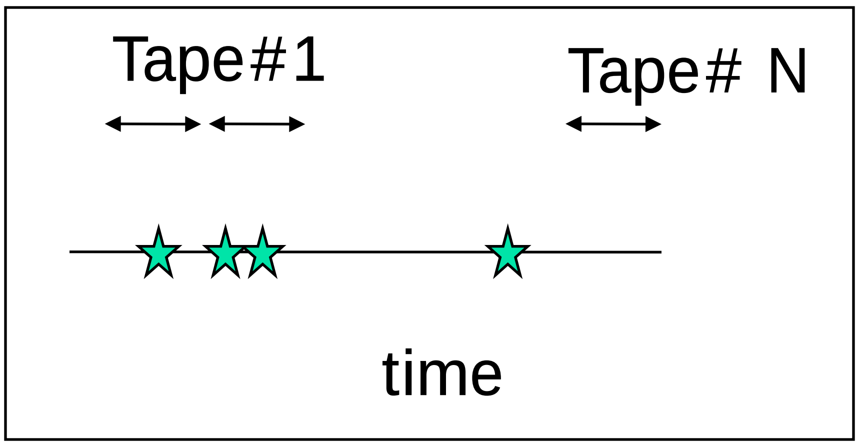
Tape accesses

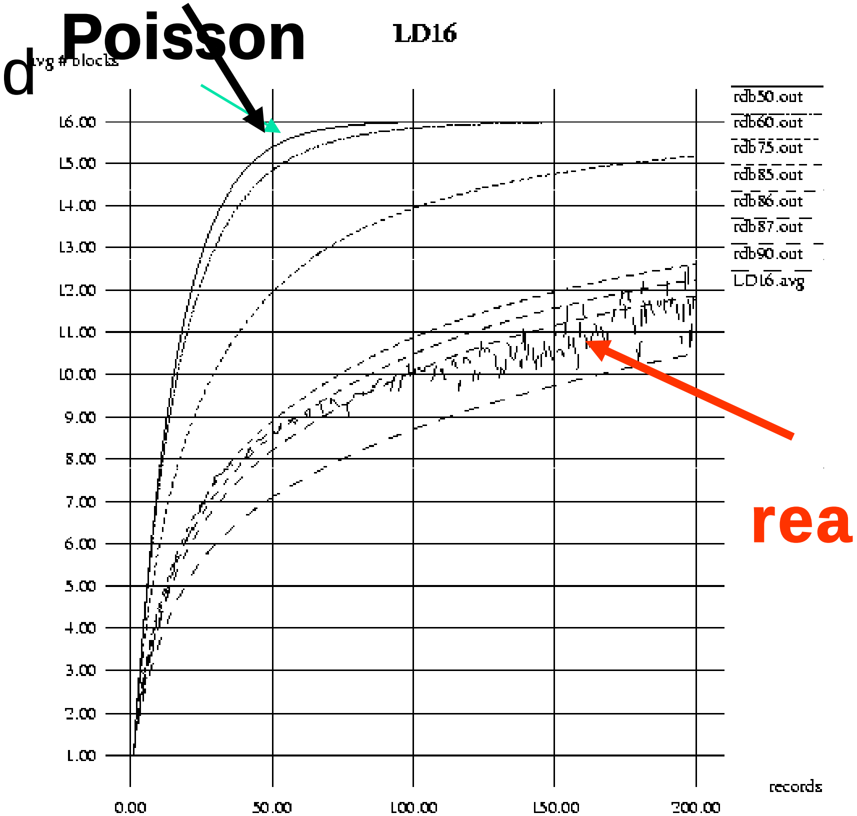


tapes needed, to retrieve *n* records?

(# days down, due to failures / hurricanes / communication noise…)

Tape accesses





Road map

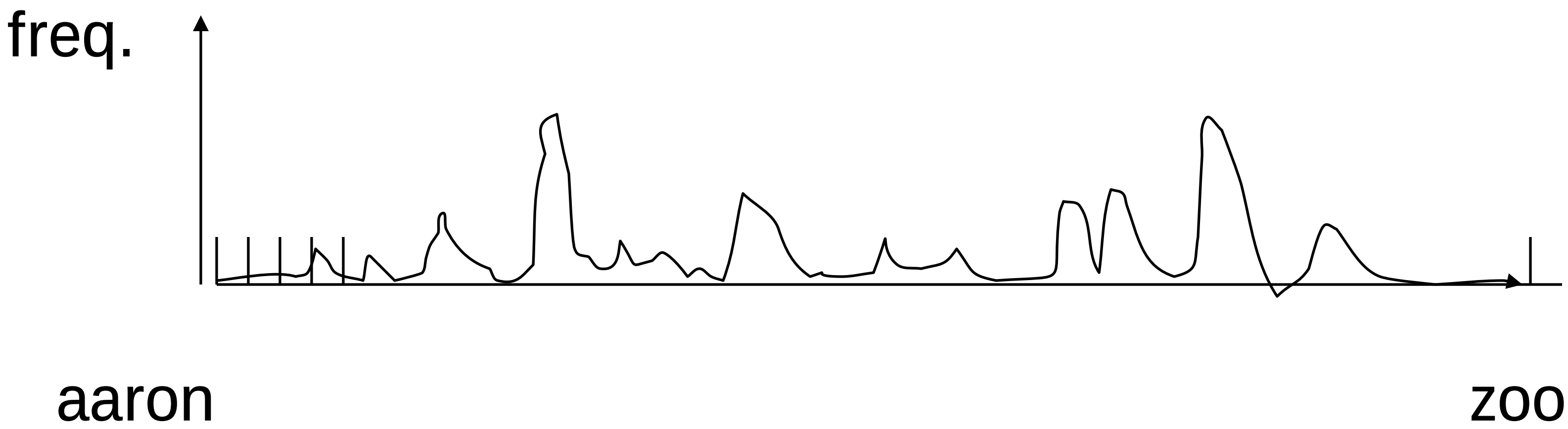
- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More **tools** and examples
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots

More tools

- Zipf’s law
- Korcak’s law / “fat fractals”

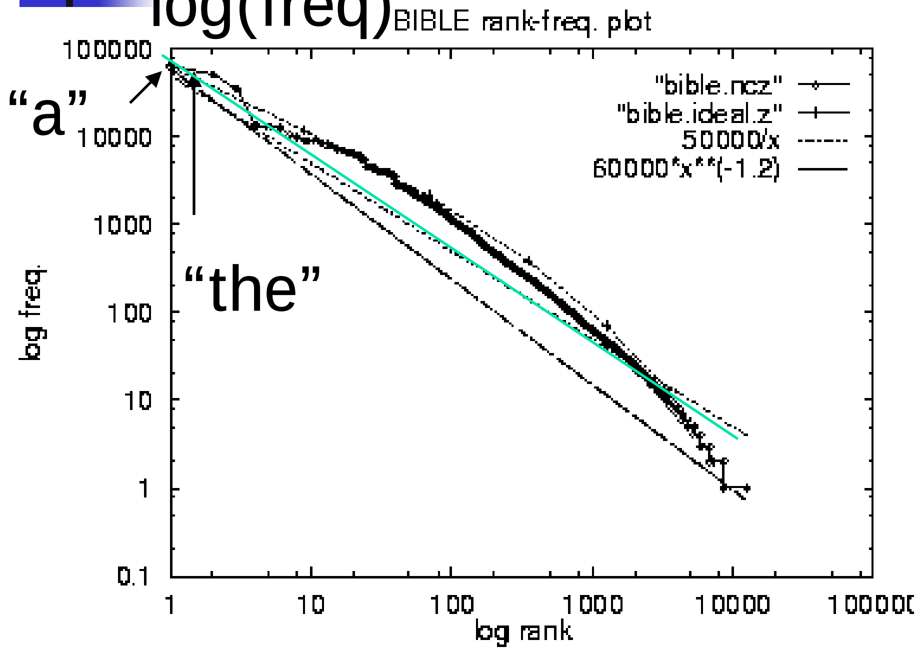
A famous power law: Zipf's law

- Q: vocabulary word frequency in a document - any pattern?



A famous power law: Zipf's law

log(freq)

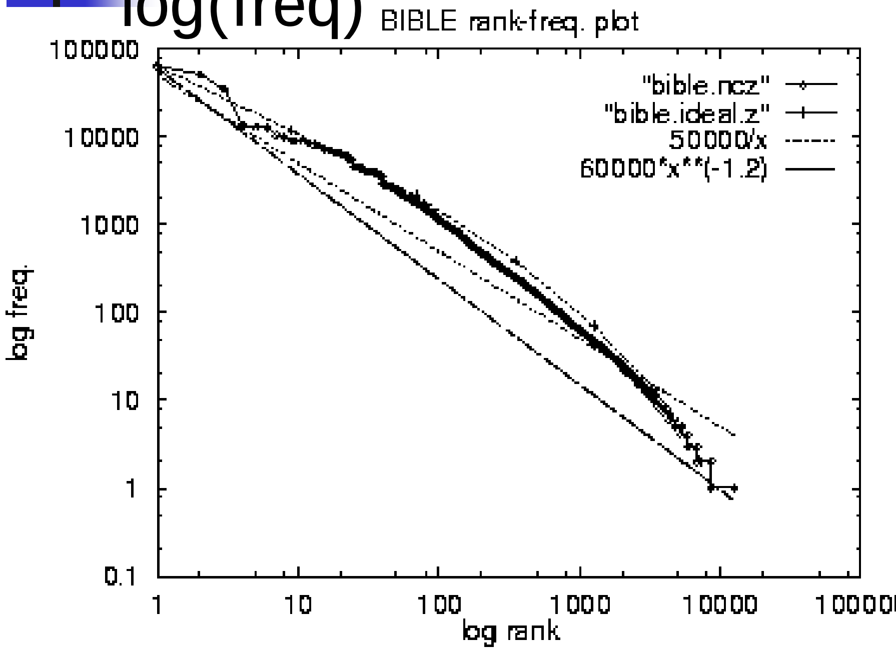


log(rank)

- Bible - **rank** vs **frequency** (log-log)

A famous power law: Zipf's law

log(freq)

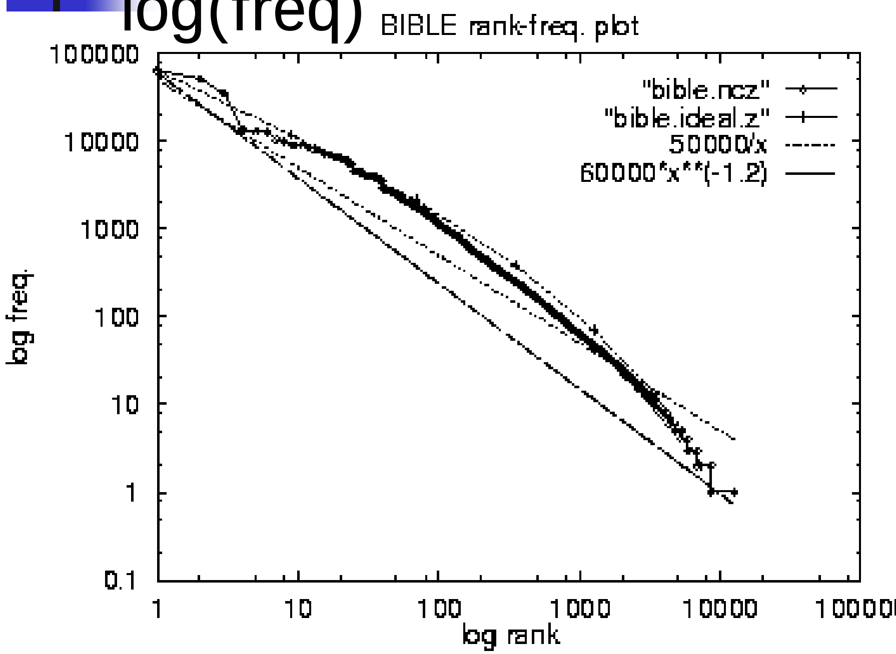


log(rank)

- Bible - rank vs frequency (log-log)
- similarly, in **many other** languages; for customers and sales volume; city populations etc etc

A famous power law: Zipf's law

log(freq)



log(rank)

- Zipf distr:

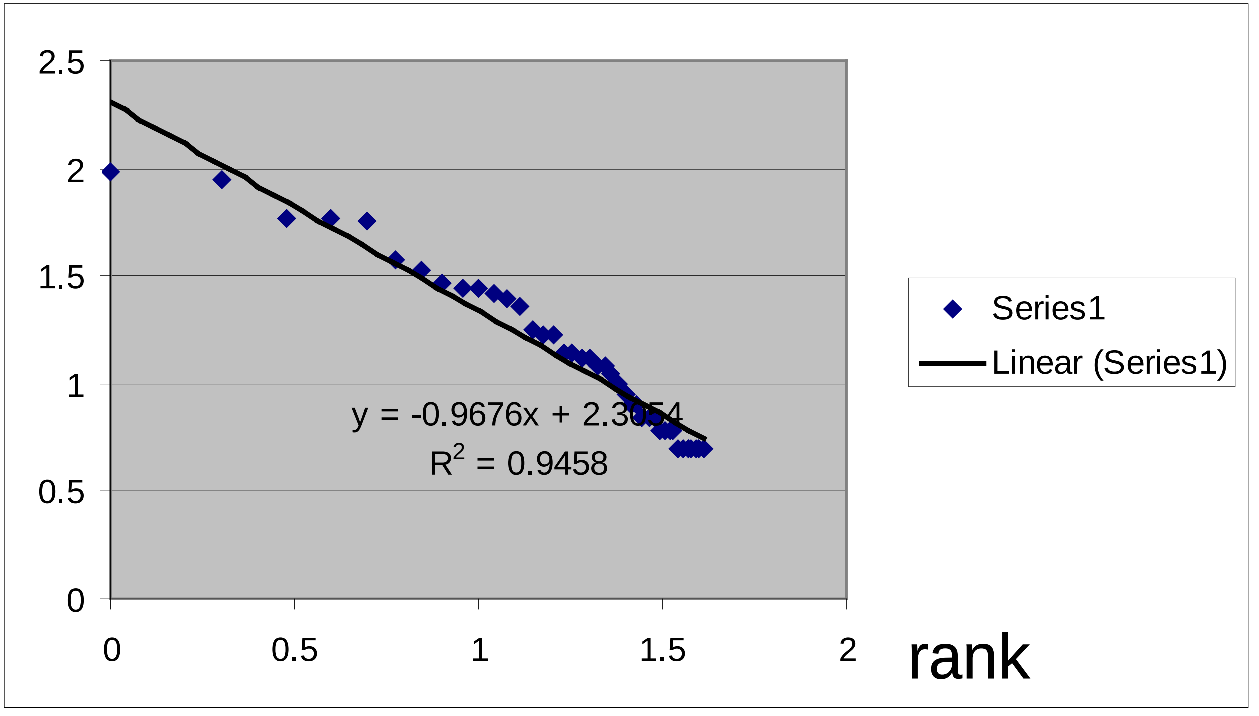
  freq = 1/ rank

- generalized Zipf:

  freq = 1 / (rank)^a

Olympic medals (Sidney):

log(#medals)



rank

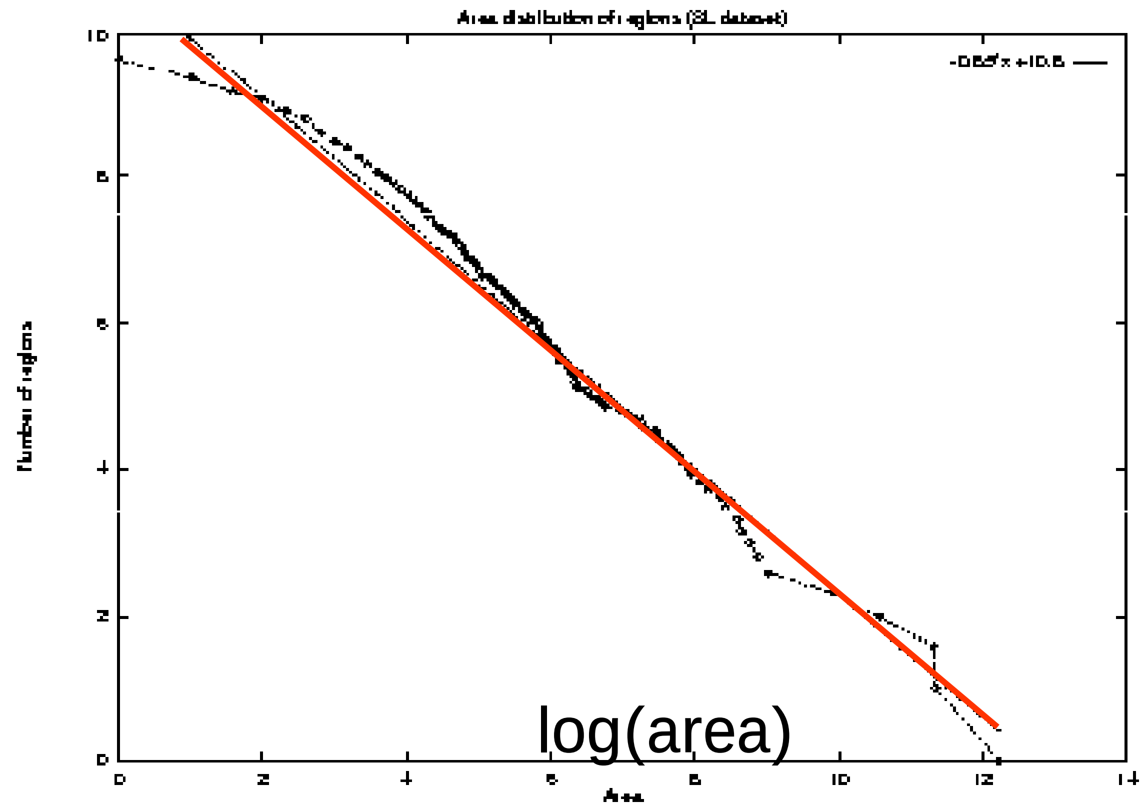
More power laws: areas – Korcak's law

Scandinavian lakes

Any pattern?

More power laws: areas – Korcak's law

log(count(>= area))



Scandinavian lakes area vs complementary cumulative count (log-log axes)

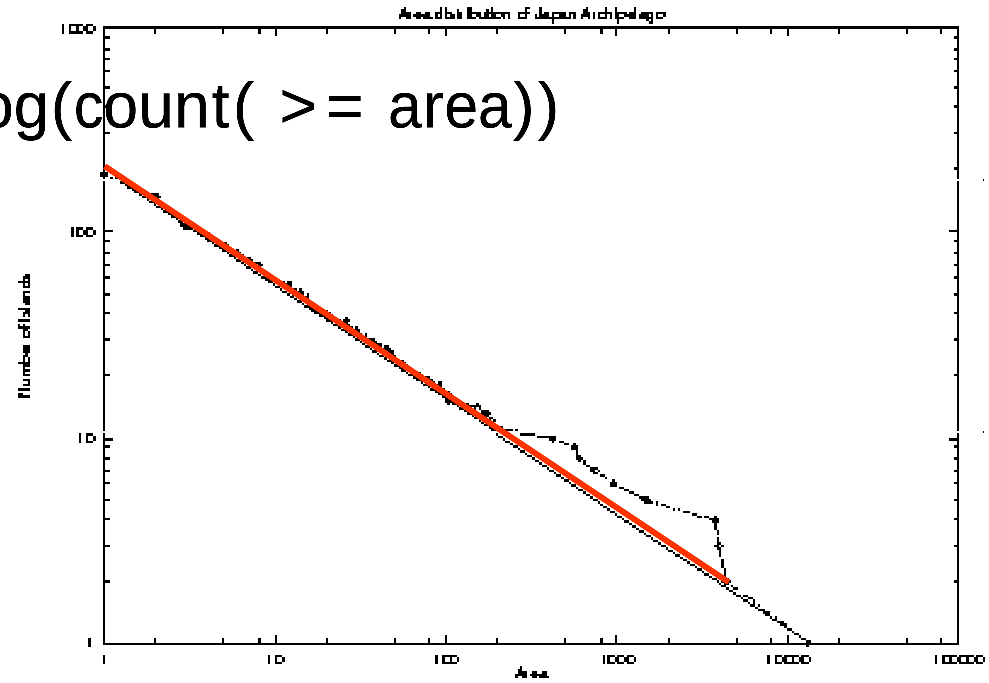
More power laws: Korcak

Japan islands

More power laws: Korcak

Japan islands;

area vs cumulative count (log-log axes)



log(count(>= area))

log(area)

(Korcak's law: Aegean islands)

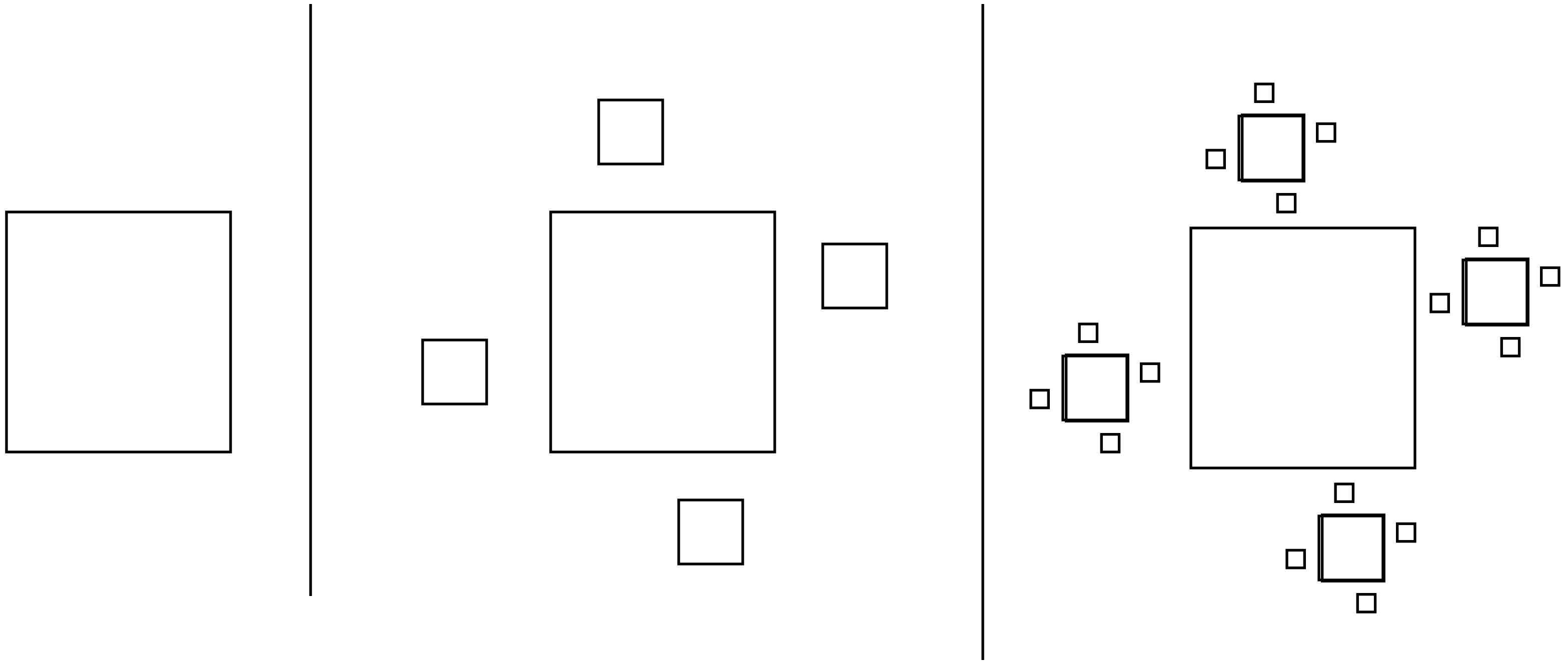
Korcak's law & "fat fractals"

How to generate such regions?

Korcak's law & "fat fractals"

Q: How to generate such regions?

A: recursively, from a single region



so far we've seen:

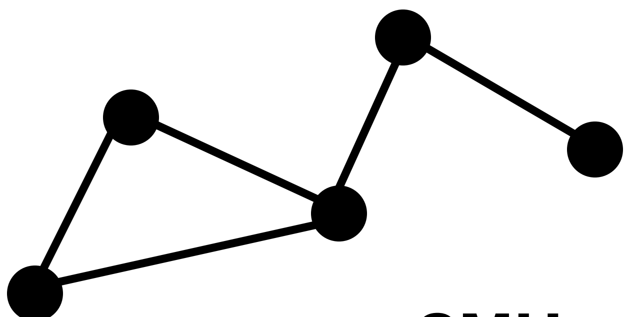
- concepts:
  - fractals, multifractals and fat fractals
- tools:
  - correlation integral (= pair-count plot)
  - rank/frequency plot (Zipf's law)
  - CCDF (Korcak's law)

Road map

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More tools and **examples**
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots

Other applications: Internet

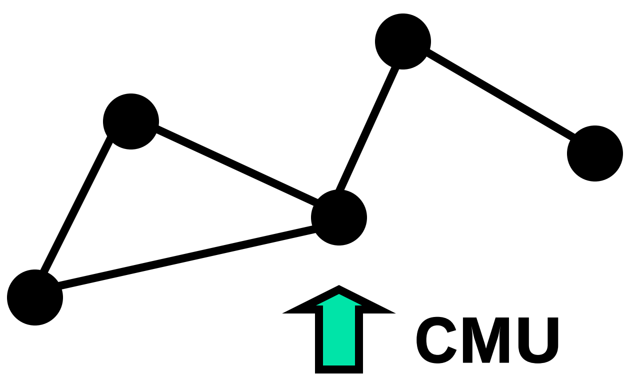
- How does the internet look like?



CMU

Other applications: Internet

- How does the internet look like?
- Internet routers: how many neighbors within *h* hops?

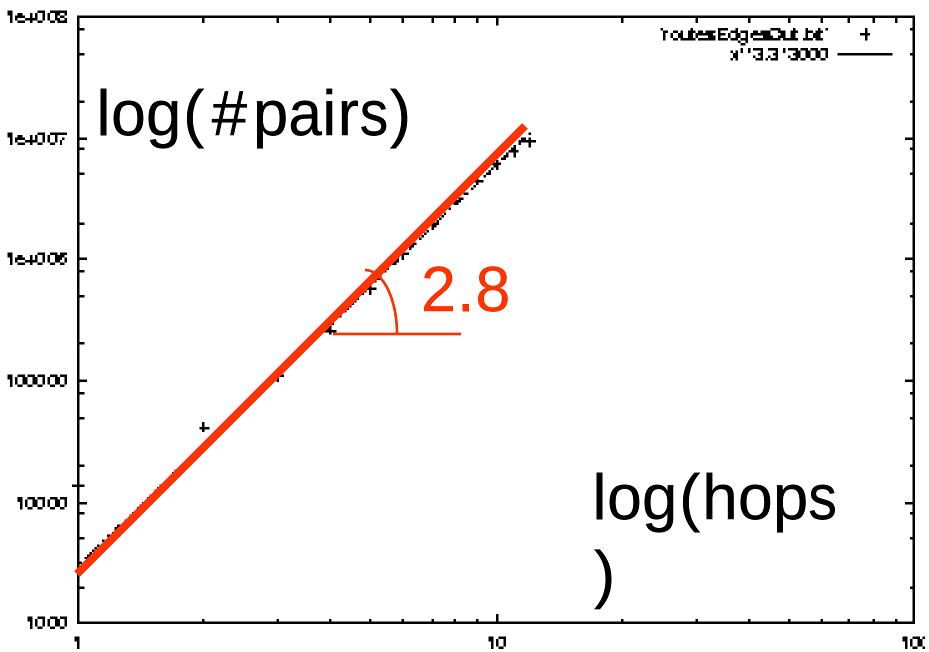


(reminder: our tool-box:)

- concepts:
  - fractals, multifractals and fat fractals
- tools:
  - correlation integral (= pair-count plot)
  - rank/frequency plot (Zipf's law)
  - CCDF (Korcak's law)

Internet topology

- Internet routers: how many neighbors within *h* hops?

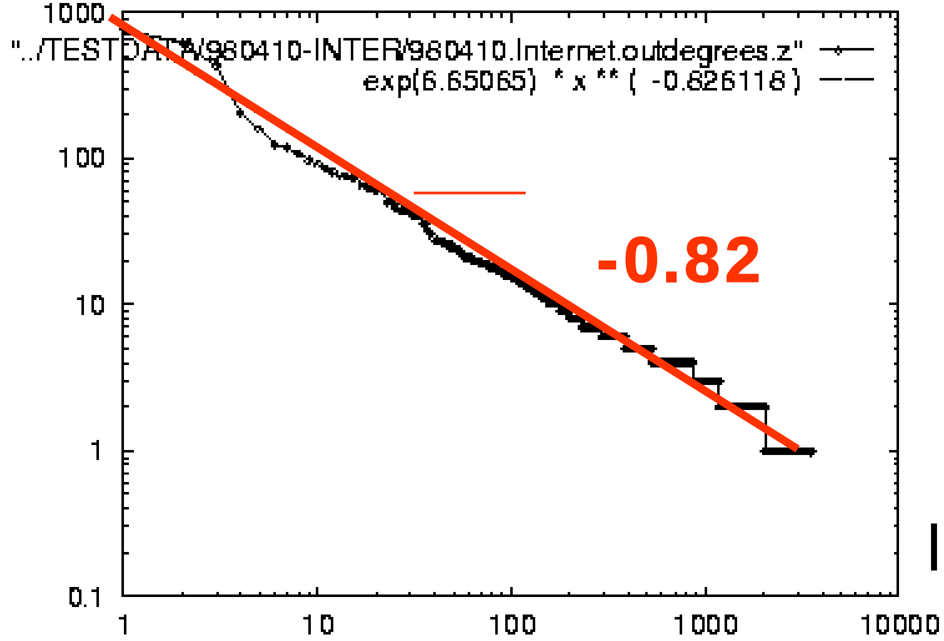


Reachability function: number of neighbors within r hops, vs r (log-log).

Mbone routers, 1995

More power laws on the Internet

log(degree)



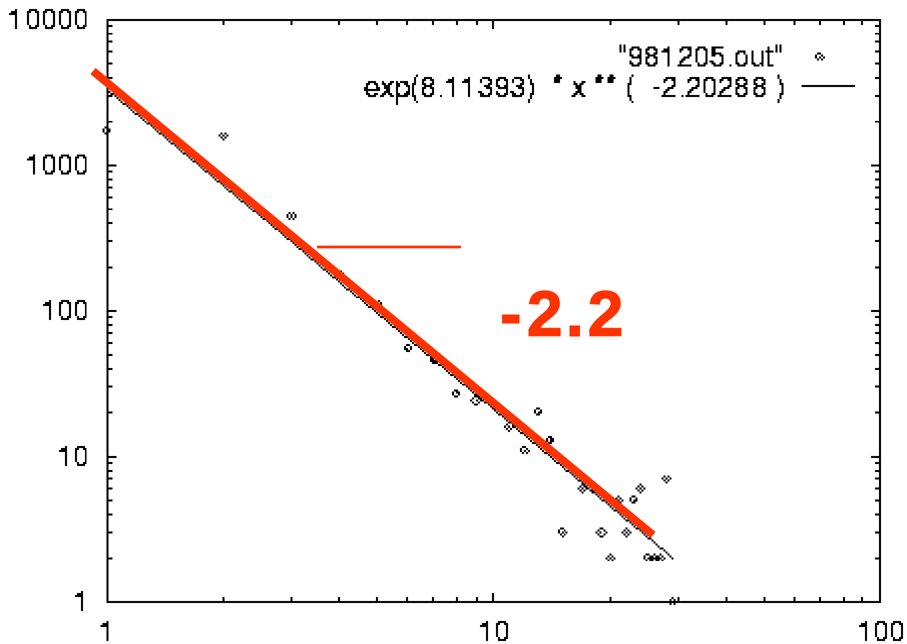
log(rank)

degree vs rank, for Internet domains (log-log) [sigcomm99]

More power laws - internet

- pdf of degrees: (slope: 2.2)

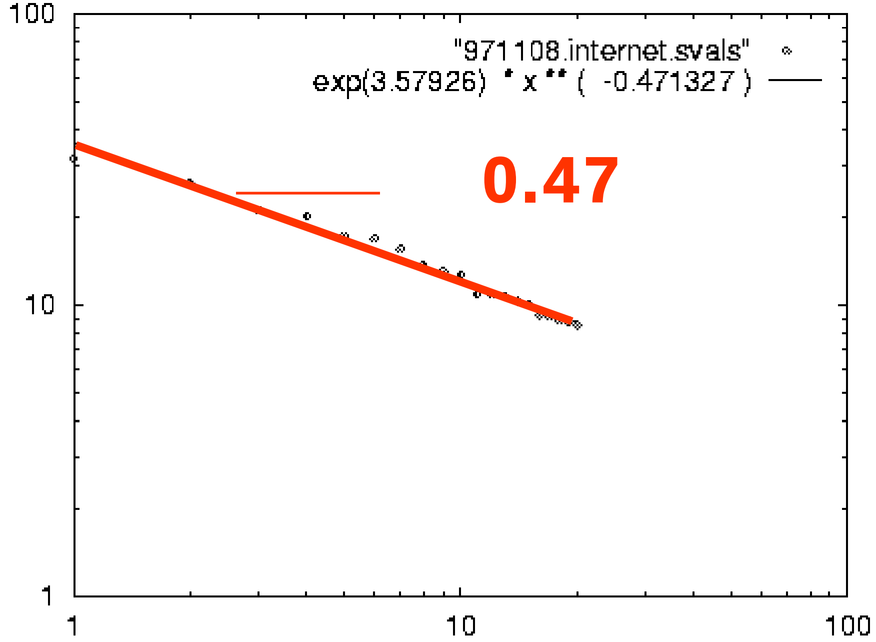
Log(count)



Log(degree)

Even more power laws on the Internet

log(i-th eigenvalue)

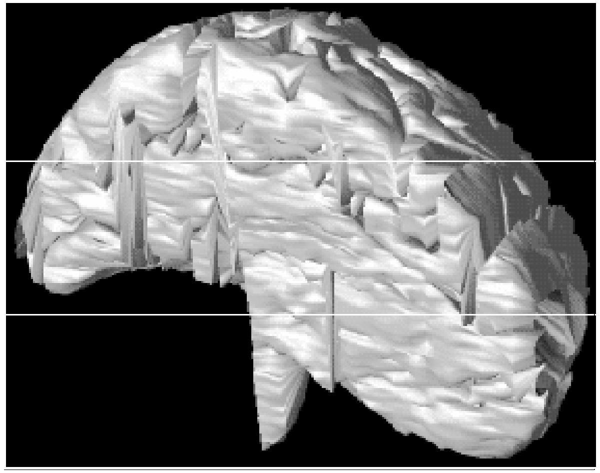


log(i)

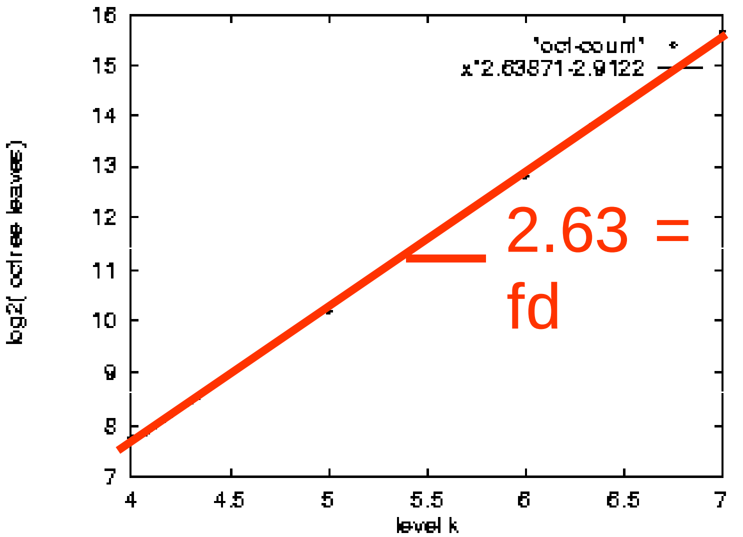
Scree plot for Internet domains (log-log) [sigcomm99]

More apps: Brain scans

- Oct-trees; brain-scans



Log(#octants)



octree levels

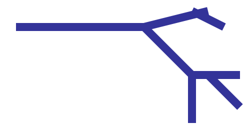
More apps: Medical images

[Burdett et al, SPIE ‘93]:

- benign tumors: fd ~ 2.37
- malignant: fd ~ 2.56

More fractals:

- cardiovascular system: 3 (!)



- stock prices (LYCOS) - random walks: 1.5

1 year

2 years

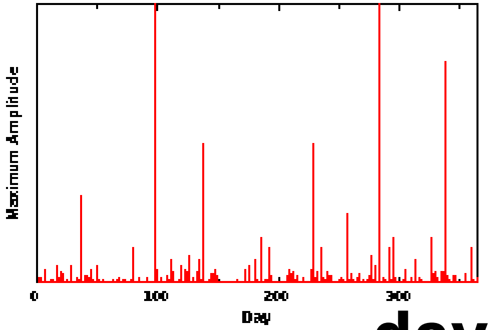
- Coastlines: 1.2-1.58 (Norway!)


Jokkmokk
Hornavan
Uddjaure
Storavan
Luleå
Mo i Rana
Sandnessjøen
Mosjøen
Storuman
Umeälven R.
Namsos
64°
Steinkjer
Umeå
Sweden
Trondheim
Molde
Östersund
Andalsnes
Ange
Härnösand
Glåma R.
Lågen R.
Ljusnan R.
Florø
Hudiksvall
Gulf of Bothnia
Gudvangen
Lillehammer
Siljan L.
Bergen
Falun
Gävle
Norway
Aland Islands
60°
Oslo
Uppsala
Karlstad
Mälaren L.
Stavanger
Stockholm
Otra R.
Örebro
Mariestad
Nyköping
Kristiansand
Vättern L.
Baltic
Vänersborg
North
Göteborg
Jönköping
Visby

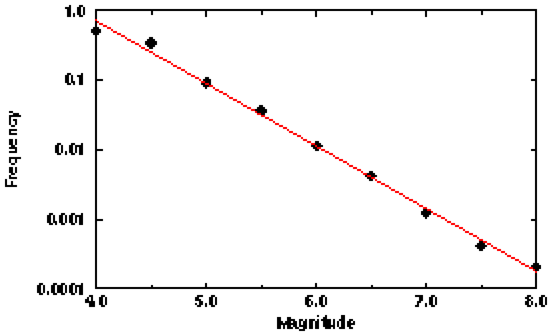
More power laws

- duration of UNIX jobs [Harchol-Balter]
- Energy of earthquakes (Gutenberg-Richter law) [simscience.org]

amplitude



day



log(freq)

magnitude

Even more power laws:

- publication counts (Lotka's law)
- Distribution of UNIX file sizes
- Income distribution (Pareto's law)
- web hit counts [Huberman]

Power laws, cont'ed

- In- and out-degree distribution of web sites [Barabasi], [IBM-CLEVER]
- length of file transfers [Bestavros+]
- Click-stream data (w/ A. Montgomery (CMU-GSIA) + MediaMetrix)

Road map

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner's guide
- Appendix: gory details - boxcounting plots

Settings for fractals:

Points; areas (-> fat fractals), eg:

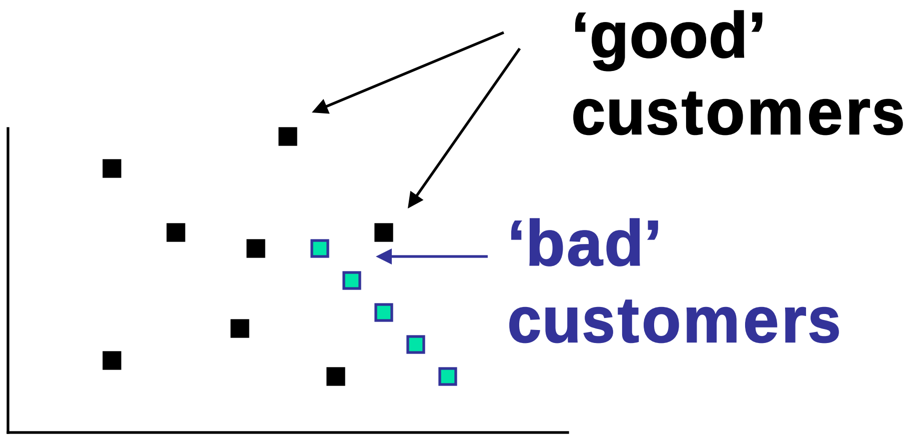
Settings for fractals:

Points; areas, eg:

- cities/stores/hospitals, over earth’s surface
- time-stamps of events (customer arrivals, packet losses, criminal actions) over time
- regions (sales areas, islands, patches of habitats) over space

Settings for fractals:

- customer feature vectors (age, income, frequency of visits, amount of sales per visit)



Some uses of fractals:

- Detect non-existence of rules (if points are uniform)
- Detect non-homogeneous regions (eg., legal login time-stamps may have different fd than intruders')
- Estimate number of neighbors / customers / competitors within a radius

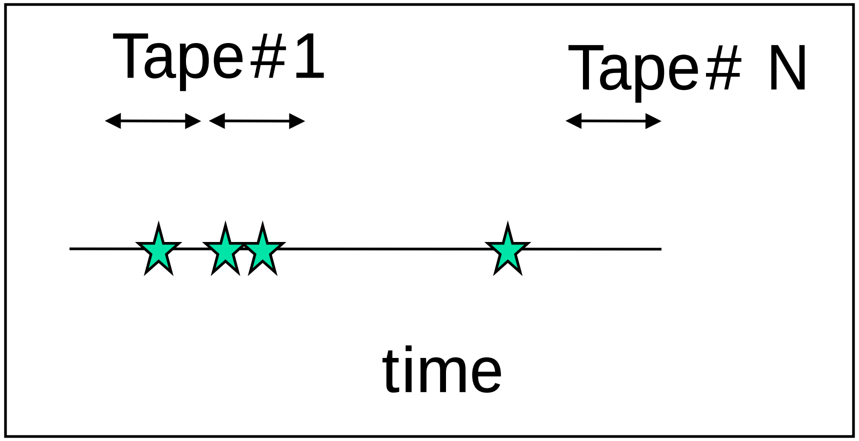
Multi-Fractals

Setting: points or objects, w/ some value, eg:

- cities w/ populations
- positions on earth and amount of gold/water/oil underneath
- product ids and sales per product
- people and their salaries
- months and count of accidents

Use of multifractals:

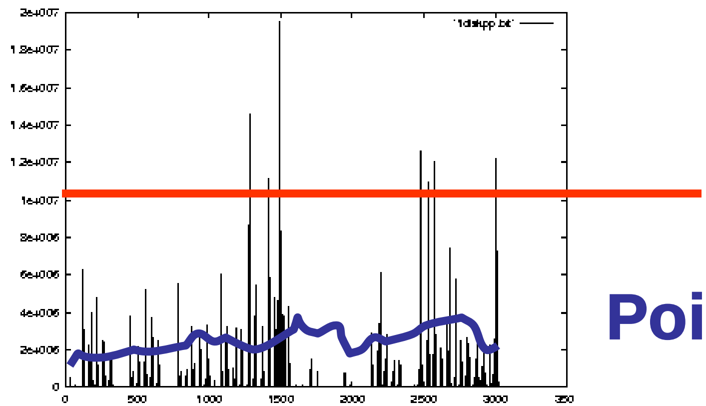
- Estimate tape/disk accesses
  - *how many of the 100 tapes contain my 50 phonecall records*?
  - *how many days without an accident?*



Use of multifractals

- how often do we exceed the threshold?

#bytes



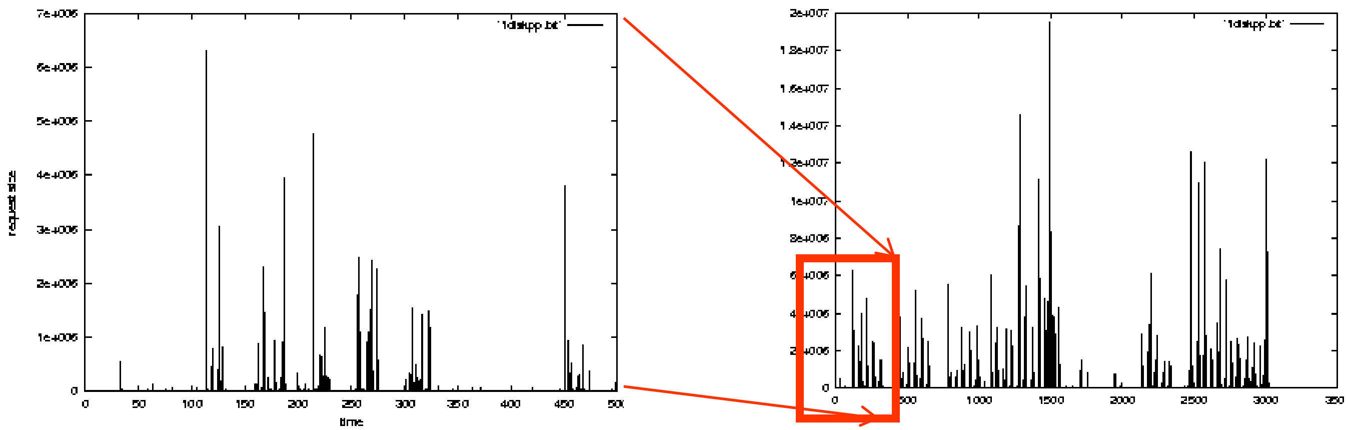
Poisson

time

Use of multifractals cont’d

- Extrapolations for/from samples

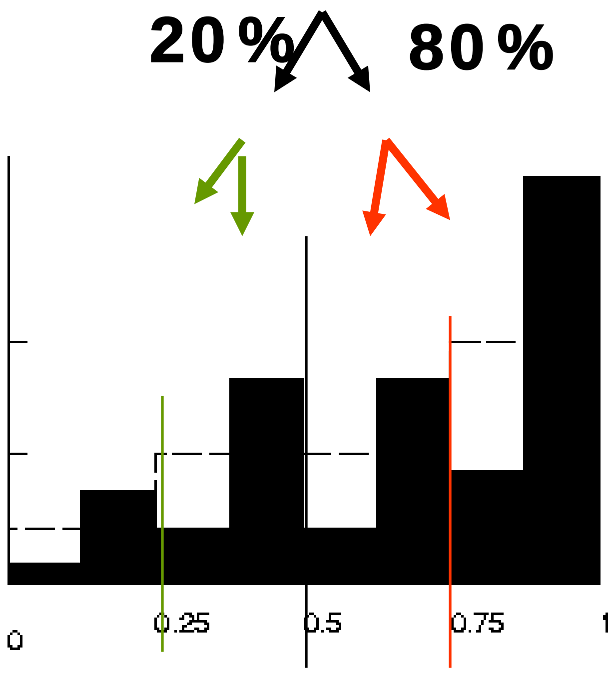
#bytes



time

Use of multifractals cont'd

- *How many distinct products account for 90% of the sales?*



Road map

- Motivation – 3 problems / case studies
- Definition of fractals and power laws
- Solutions to posed problems
- More examples and tools
- Discussion - putting fractals to work!
- Conclusions – practitioner’s guide
- Appendix: gory details - boxcounting plots

Conclusions

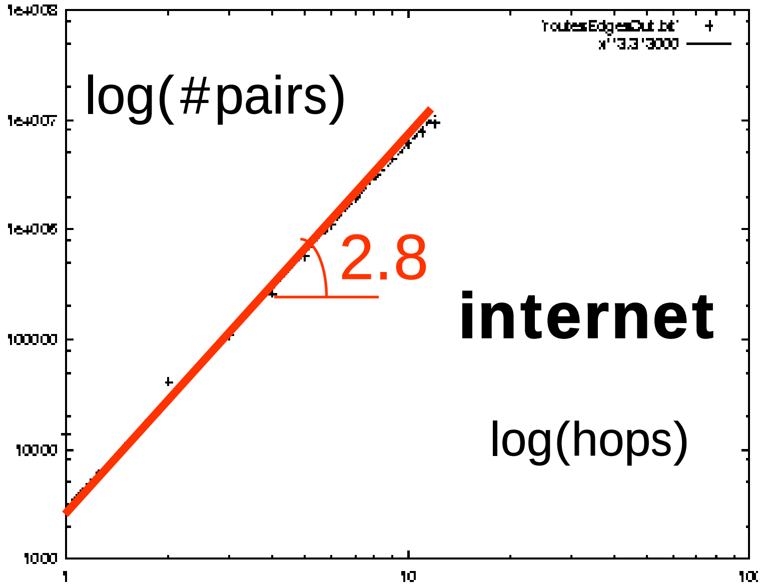
- **Real data often disobey textbook assumptions** (Gaussian, Poisson, uniformity, independence)
  - avoid ‘mean’ - use median, or even better, use:
- fractals, self-similarity, and power laws, to find patterns - specifically:

Conclusions

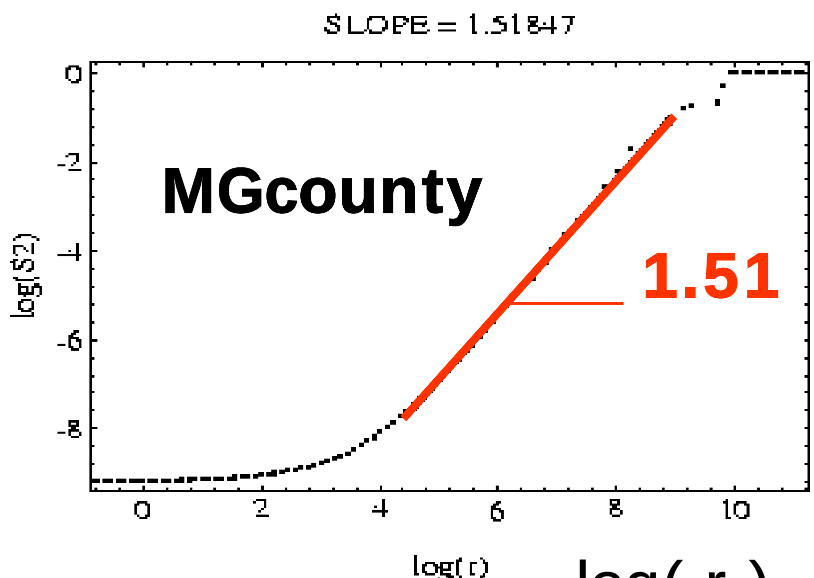
- **tool#1: (for points) 'correlation integral'**: (#pairs within <= *r*) vs (distance *r*)
- **tool#2: (for categorical values) rank-frequency** plot (a'la Zipf)
- **tool#3: (for numerical values) CCDF:** Complementary cumulative distr. function (#of elements with value >= *a*)

Practitioner's guide:

- **tool#1**: #pairs vs distance, for a **set of objects**, with a distance function (slope = intrinsic dimensionality)



log(#pairs(within <= r))



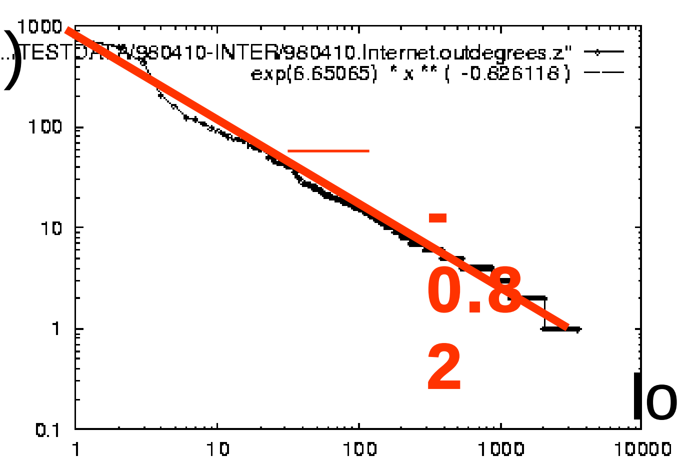
log(r)

Practitioner’s guide:

- **tool#2**: rank-frequency plot (for **categorical attributes**)

internet domains

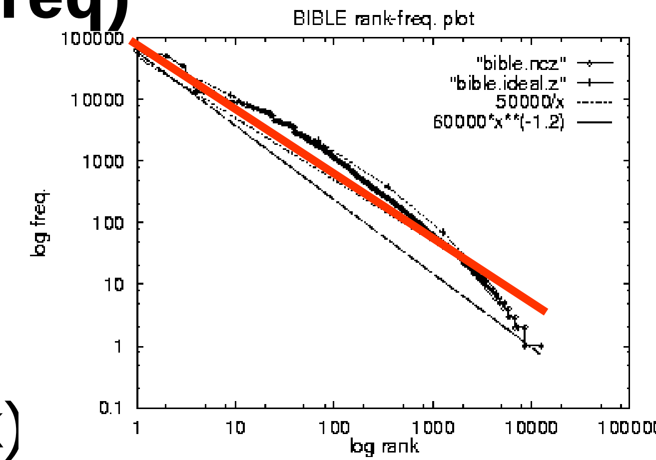
log(degree)



log(rank)

Bible

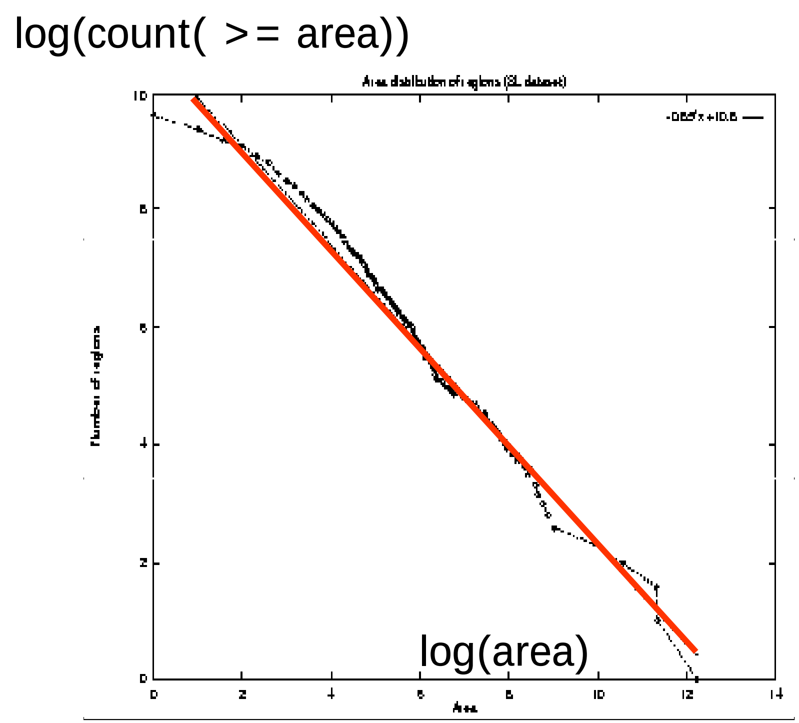
log(freq)



log(rank)

Practitioner's guide:

- **tool#3**: CCDF, for (skewed) **numerical attributes**, eg. areas of islands/lakes, UNIX jobs...)



scandinavian lakes

Books

- Strongly recommended intro book:
  - Manfred Schroeder *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise* W.H. Freeman and Company, 1991
- Classic book on fractals:
  - B. Mandelbrot *Fractal Geometry of Nature,* W.H. Freeman, 1977

References

- [ieeeTN94] W. E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, *On the Self-Similar Nature of Ethernet Traffic,* IEEE Transactions on Networking, 2, 1, pp 1-15, Feb. 1994.
- [pods94] Christos Faloutsos and Ibrahim Kamel, *Beyond Uniformity and Independence: Analysis of R-trees Using the Concept of Fractal Dimension,* PODS, Minneapolis, MN, May 24-26, 1994, pp. 4-13

References

- [vldb95] Alberto Belussi and Christos Faloutsos, *Estimating the Selectivity of Spatial Queries Using the `Correlation' Fractal Dimension* Proc. of VLDB, p. 299-310, 1995
- [vldb96] Christos Faloutsos, Yossi Matias and Avi Silberschatz, *Modeling Skewed Distributions Using Multifractals and the `80-20 Law’* Conf. on Very Large Data Bases (VLDB), Bombay, India, Sept. 1996.

References

- [vldb96] Christos Faloutsos and Volker Gaede *Analysis of the Z-Ordering Method Using the Hausdorff Fractal Dimension* VLD, Bombay, India, Sept. 1996
- [sigcomm99] Michalis Faloutsos, Petros Faloutsos and Christos Faloutsos, *What does the Internet look like? Empirical Laws of the Internet Topology,* SIGCOMM 1999

References

- [icde99] Guido Proietti and Christos Faloutsos, *I/O complexity for range queries on region data stored using an R-tree* International Conference on Data Engineering (ICDE), Sydney, Australia, March 23-26, 1999
- [sigmod2000] Christos Faloutsos, Bernhard Seeger, Agma J. M. Traina and Caetano Traina Jr., *Spatial Join Selectivity Using Power Laws*, SIGMOD 2000

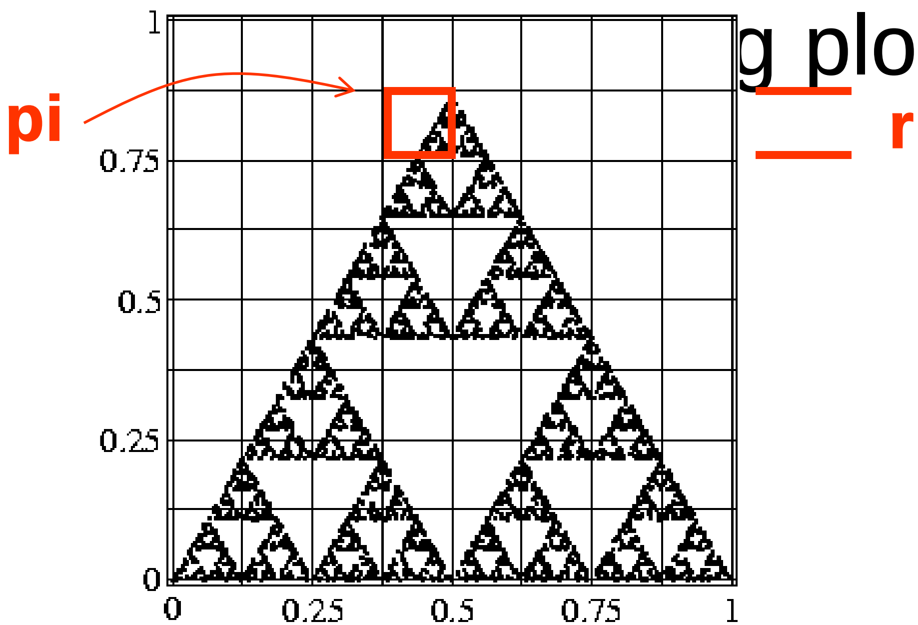
Appendix - Gory details

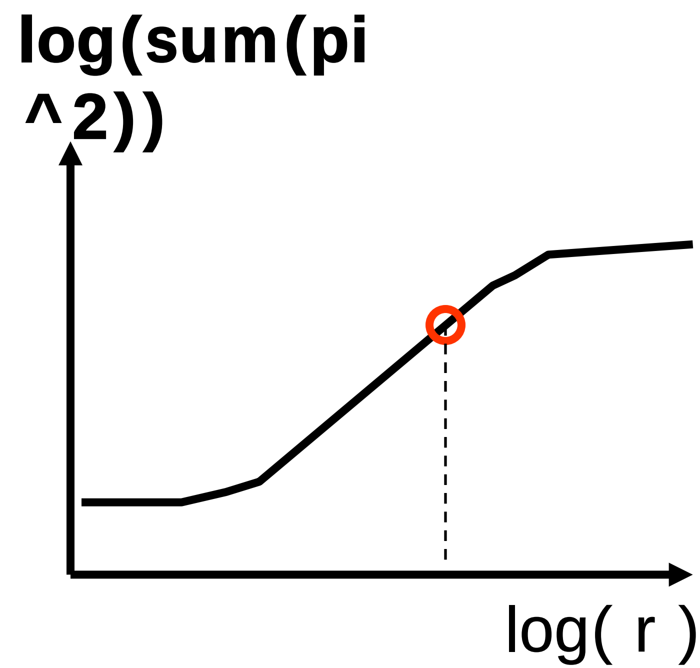
- Bad news: There are more than one fractal dimensions
  - Minkowski fd; Hausdorff fd; Correlation fd; Information fd
- Great news:
  - they can all be computed fast!
  - they usually have nearby values

Fast estimation of fd(s):

- How, for the (correlation) fractal dimension?

g plot:



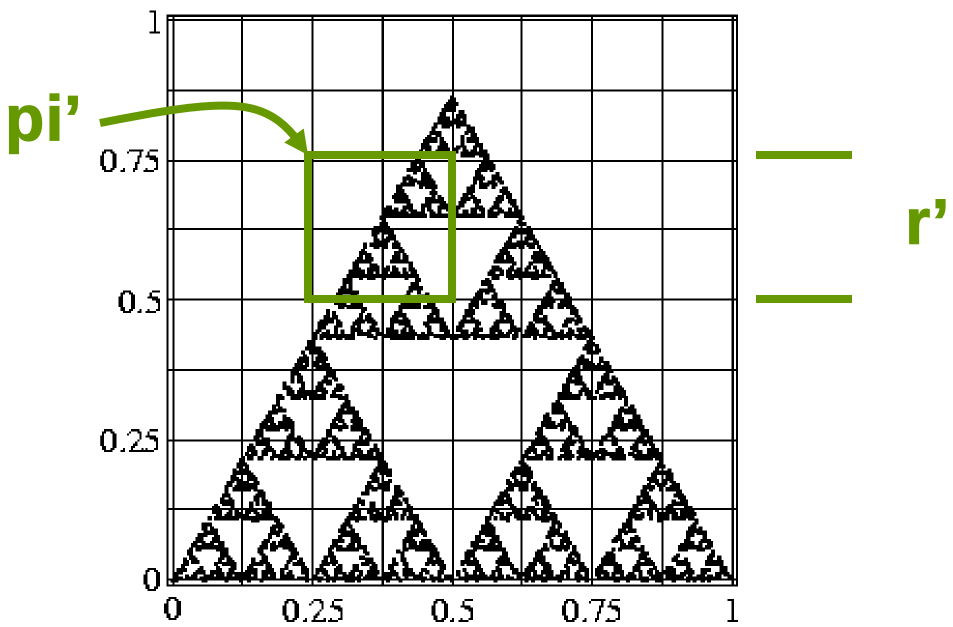


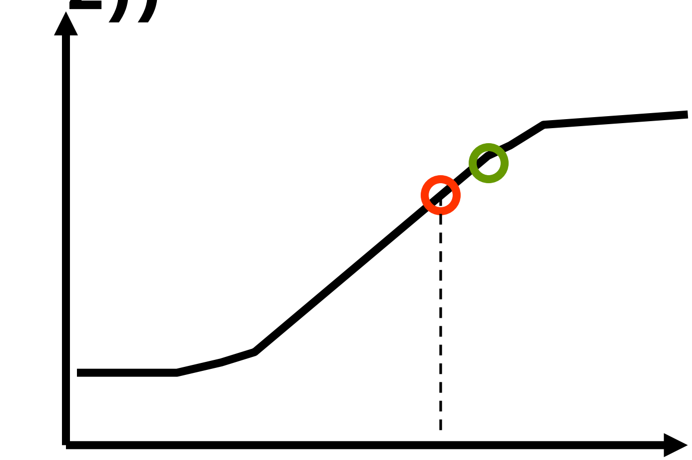
Definitions

- *pi* : the percentage (or count) of points in the *i*-th cell
- *r*: the side of the grid

Fast estimation of fd(s):

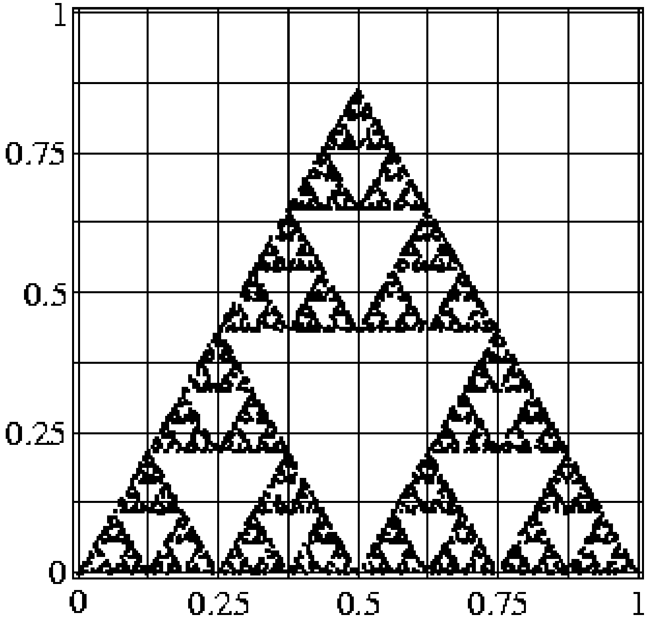
- compute sum(pi^2) for another grid side, r'

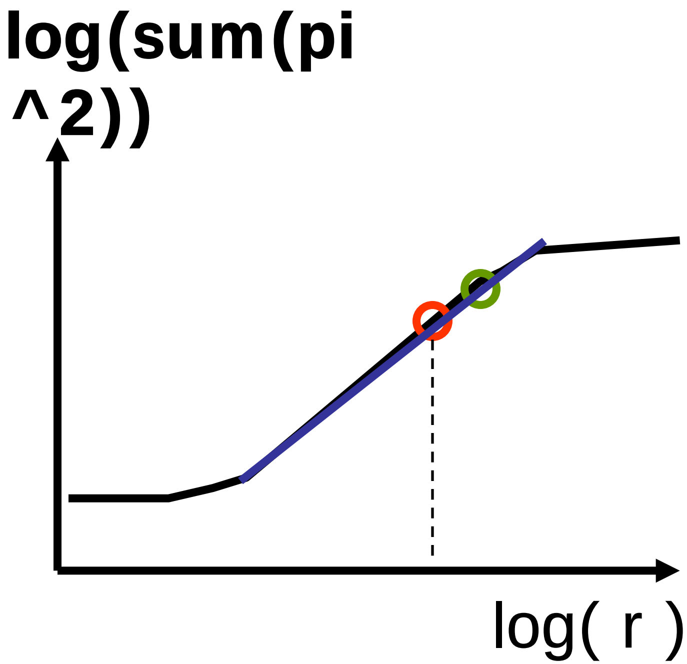




Fast estimation of fd(s):

etc; if the resulting plot has a linear part, its slope is the correlation fractal dimension D2





Definitions (cont'd)

- Many more fractal dimensions Dq (related to Renyi entropies):

$$D_q = \frac{1}{q-1} \frac{\partial \log(\sum p_i^q)}{\partial \log(r)} \qquad q \neq 1$$

$$D_1 = \frac{\partial \sum p_i \log(p_i)}{\partial \log(r)}$$

Hausdorff or box-counting fd:

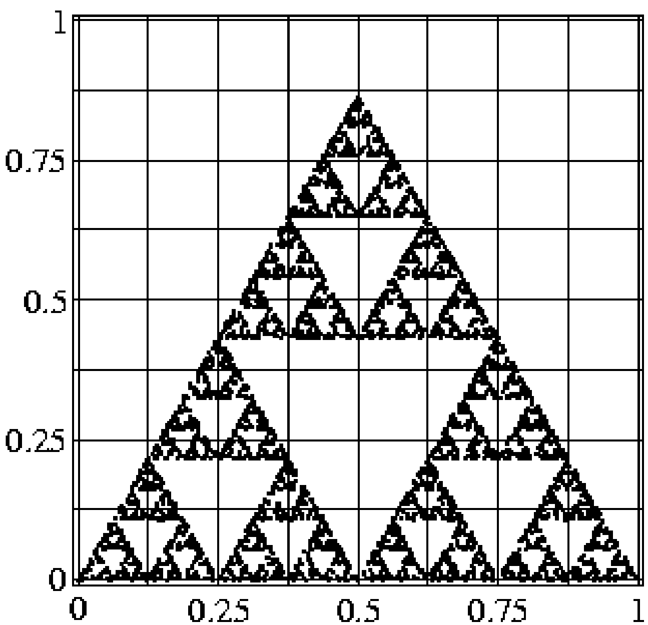
- Box counting plot: Log(N (r)) vs Log (r)
- r: grid side
- N (r): count of non-empty cells
- (Hausdorff) fractal dimension D0:

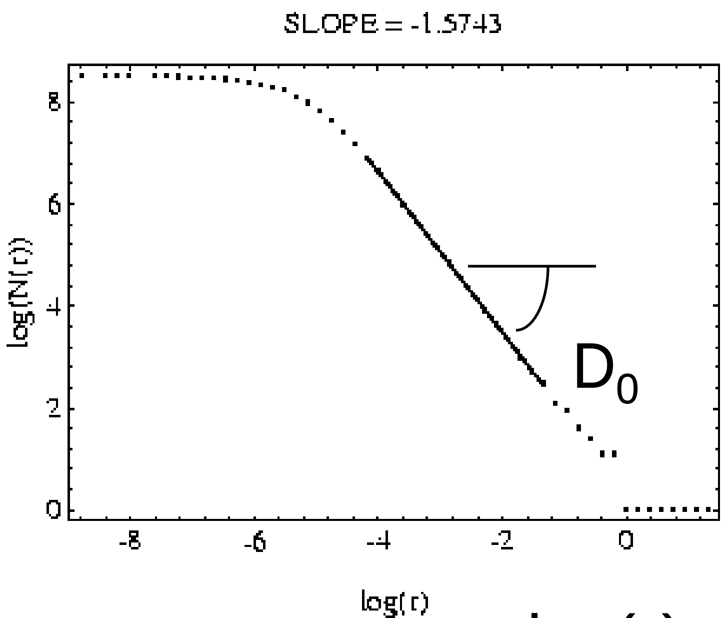
$$D_0 = -\frac{\partial \log(N(r))}{\partial \log(r)}$$

Definitions (cont'd)

- Hausdorff fd:

r log(#non-empty cells)





log(r)

Observations

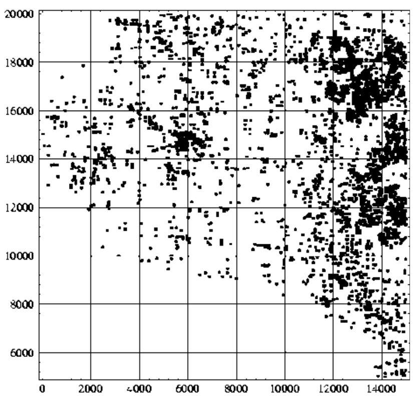
- q=0: Hausdorff fractal dimension
- q=2: Correlation fractal dimension (**identical** to the exponent of the number of neighbors vs radius)
- q=1: Information fractal dimension

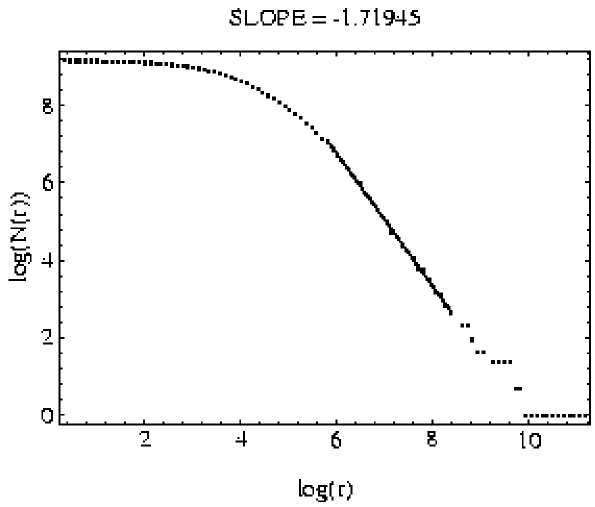
Observations, cont'd

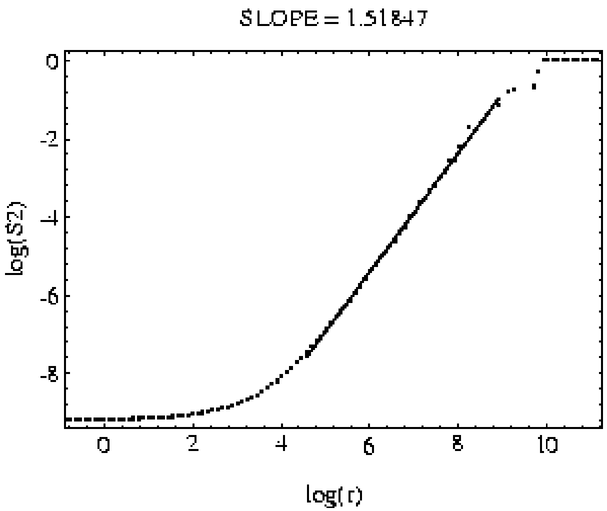
- in general, the Dq's take similar, but not identical, values.
- except for perfectly self-similar point-sets, where Dq=Dq' for any *q, q'*

Examples:MG county

- Montgomery County of MD (road end-points)

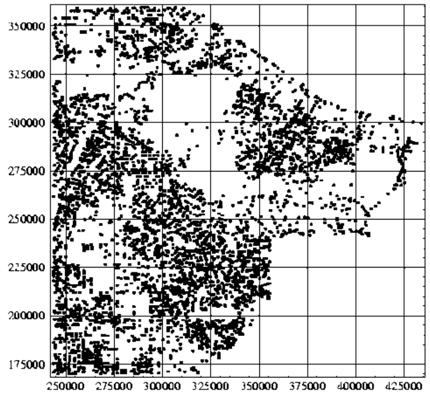


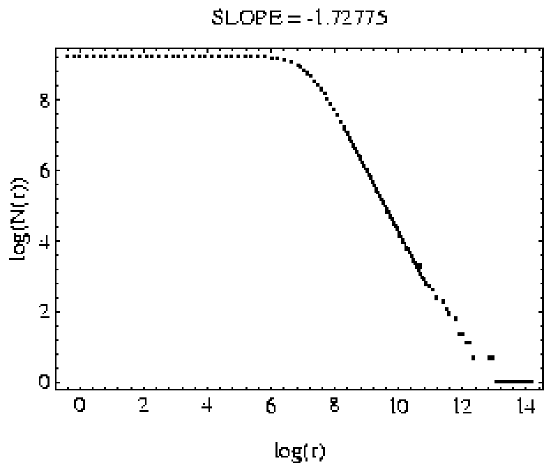


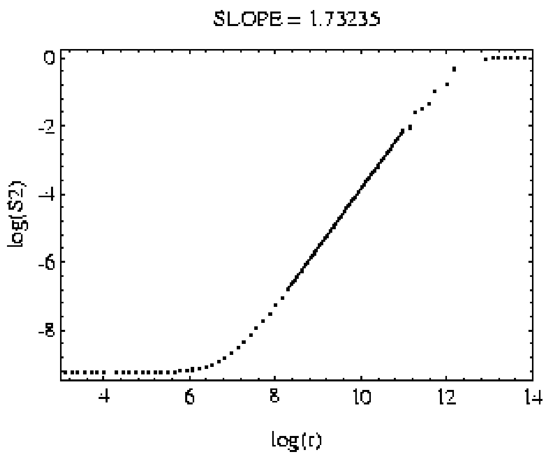


Examples:LB county

- Long Beach county of CA (road end-points)







Conclusions

- many fractal dimensions, with nearby values
- can be computed quickly

  (O(N) or O(N log(N))
- (code: on the web)