Handling data with R

fitting distributions, time-series analysis, and analysis of variance

Prof. Steve Uhlig

Professor of Networks

steve@eecs.qmul.ac.uk

R

Queen Mary University of London

What is R?

- Open-source statistical environment
- A framework for data analysis
- A programming language
- Free software

What can you do with R?

- Data manipulation:
  - Loading data
  - Selecting and modifying
  - Computing functions on data, e.g. statistics

Handling data with R

Queen Mary University of London

Loading data

- vector.small = read.table("ping-10")
- vector.medium = read.table("ping-100")
- vector.large = read.table("ping-1000")

Playing with data

- Vector Length: length(vector)
- Element n: vector[n]
- First k elements: vector[1:k]
- Last k elements: vector[(length(vector)-k+1), length(vector)]

Handling data with R

Queen Mary University of London

Playing with data

- Transpose: t(vector)
- All elements larger than x: vector(vector>x)
- Trimming data: mean(vector,trim=1/x)
- Function to trim data:

```
trim <- function(x,prop=.05) {
      trimx <- x[x < quantile(x,prob=(1-prop))]
      return(trimx)
}
```

Basic statistics

Summary statistics:

- Mean: mean(vector)
- Median: median(vector)
- Standard deviation: sd(vector)
- Summary: summary(vector)

Plotting densities:

- Basic plot: plot(vector)
- Histogram: hist(vector,x)
- CDF: plot(ecdf(vector))

Queen Mary
University of London

Fitting distributions

Fitting distributions

- Principle: finding a mathematical function that represents a statistical variable, e.g., your data
- How to proceed?
  1. Find the distribution from which the data might be drawn
  2. Estimate the parameters of that distribution
  3. Evaluate the quality of fit

Fitting distributions

Choosing a model:

- Graphically, e.g., qqplot
- Matching the empirical distribution
  - Mean: $\mu = \frac{\sum_{i=1}^{n} x_i}{n}$
  - Variability: $\text{var} = \sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}$
  - Skewness: $\gamma_1 = \frac{\sum_{i=1}^{n} (x_i - \mu)^3}{n\sigma^3}$
  - Kurtosis: $\gamma_2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^4}{n\sigma^4}$

Graphical method

From what distribution does your data come?

- Wrong approach:
  - mental model of the community
  - Uninformed assumptions
- Better:
  - Plotting sample quantiles against theoretical quantiles (qqplot)
  - If distributions are similar, quantiles should fall on the diagonal, e.g.,
    - x.norm1 = rnorm(1000)
    - x.norm2 = rnorm(1000)
    - qqplot(x.norm1,x.norm2)

Normal distribution

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

- Normal distribution = centered limit of most distributions
  - x.normal = rnorm(n=1000, m = mean(vector), sd = sd(vector))
  - hist(x.normal)
  - plot(ecdf(x.normal))
  - qqplot(vector,x.normal)

Poisson distribution

$$f(x,\lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$$

- Poisson distribution: arrivals of large numbers of independent sources
  - x.poisson = rpois(n=1000,lambda=mean(vector))
  - hist(x.poisson)
  - plot(ecdf(x.poisson))
  - qqplot(vector,x.poisson)

Beyond simple distributions

- Does it fit?
- Normal and Poisson have limited variations
- Rarely happens in real data, even after trimming
- Heavier distributions required
  - Gamma: sum of exponentials
  - Weibull: distribution of failures (extreme events)

Gamma distribution

$$f(x,\alpha,\beta) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^{\alpha}}$$

- Gamma distribution: sum of alpha exponential distributions
  - x.gamma = rgamma(n=1000,scale=0.83,shape=10.59)
  - hist(x.gamma)
  - qqplot(vector,x.gamma)

Weibull distribution

$$f(x,\lambda,k) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$$

- Weibull distribution: distribution of failures
  - x.weibull = rweibull(n=1000,scale=3.5,shape=14.1)
  - hist(x.weibull)
  - qqplot(vector,x.weibull)

Still no fit?

- Even extreme value distributions are not that heavy-tailed
- Real data often has very large variations, e.g., natural phenomena, large-scale systems (Internet)
- Strict heavy-tailed distribution = distribution whose tail is not exponentially bounded
- Examples: Pareto, Weibull with shape < 1, log-normal, Levy, alpha-stable

Pareto distribution

$$f(x,\alpha,x_m) = \alpha x_m^{\alpha} x^{-\alpha-1}$$

- Pareto distribution: very uneven distribution, e.g., income
- Generating Pareto distribution:
  - library(VGAM)
  - x.pareto1 <- rpareto(n=100000,location=10,shape=1)
  - x.pareto1.5 <- rpareto(n=100000,location=10,shape=1.5)
  - x.pareto2 <- rpareto(n=100000,location=10,shape=2)

Log-log plots

- Objective of log-log CCDF: see a heavy tail
- Plot CCDF on a log-log scale:

```
ccdf<-function(views,density=FALSE) {
    freq = table(views)
    X = rev(as.numeric(names(freq)))
    Y =cumsum(rev(as.list(freq)));
    data.frame(x=X,count=Y)
}
```

- Check various distributions
  - plot(ccdf(vector),log="xy")
  - plot(ccdf(x.pareto),log="xy")
  - plot(ccdf(x.normal),log="xy")
  - plot(ccdf(x.poisson),log="xy")
  - plot(ccdf(x.gamma),log="xy")
  - plot(ccdf(x.weibull),log="xy")

Moment estimation

Queen Mary University of London

- t^{th} moment of a distribution: $m_t = \sum_{i=1}^{n} x_i^t y_i$
- Example:
  - Estimating parameters of a gamma distribution using the first 2 moments:

$$\frac{\beta}{\alpha} = \bar{x} \qquad \frac{\beta}{\alpha^2} = s^2$$

  - Gives estimates for parameters:

$$\hat{\alpha} = \frac{\bar{x}}{s^2} \qquad \hat{\beta} = \frac{\bar{x}^2}{s^2}$$

Moment estimation: Gamma

$$f(x,\alpha,\beta) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^{\alpha}}$$

- Estimating parameters:
  - Alpha = mean(vector)/var(vector)
  - Beta = (mean(vector))**2/var(vector)
  - x.gamma = rgamma(n=1000,scale=alpha,shape=beta)
  - hist(x.gamma)
  - qqplot(vector,x.gamma)

Maximum Likelihood Estimation

•Given data x_i and a parameter theta to be estimated, what is the most likely value of ?

$$L(x_1, x_2, .., x_n, \theta) = \prod_{i=1}^{n} f(x_i, \theta)$$

•Estimating parameters of distributions with fitdistr():

- library(MASS)
- Normal: fitdistr(vector,‘‘normal‘‘)
- Gamma: fitdistr(vector,‘‘gamma‘‘)
- Weibull: fitdistr(vector,‘‘weibull‘‘)

Goodness of fit: theory

Queen Mary University of London

- Test: Is it reasonable to assume that the data comes from a specific distribution?
- 2 hypotheses:
  - H_0: Sample data comes from the stated distribution
  - H_A: Sample data does not come from the stated distribution
- Example: Kolmogorov-Smirnov test
  - Compares empirical distribution against theoretical one
  - Given n data points $x_1,\ldots x_n$, define $F_n(x_i) = N(i)/n$
  - Test statistic: $D_n = \sup_i |F(x_i) - F_n(x_i)|$
  - If D_n is too large for a given significance level, H_0 is rejected.

P-value

- The ***p*-value** is the probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis H_0 is true.
- One often "rejects the null hypothesis" when the *p*-value is less than the significance level α, which is often 0.05 or 0.01.
- Note: a statistical test NEVER accepts a hypothesis

Goodness of fit: practice

- Testing goodness of generated samples:
  - Normal: ks.text(vector, „rnorm“, mean=mean(vector), sd=sd(vector))
  - Gamma: ks.test(x.gamma, „pgamma“, scale=0.83,shape=10.59)
  - Weibull: ks.test(x.weibull,‘‘pweibull‘‘,scale=3.5,shape=14.14)
- Testing for normality:
  - Shapiro-Wilk test
  - Shapiro.test(x.normal)
- Now test the data against Normal, Gamma and Weibull
  - Normal?
  - Gamma?
  - Weibull?

Time series analysis

Time series

Queen Mary University of London

Definitions

- Time series: $X_1, X_2, \ldots, X_n$
- Increments: $(X_{i+1} - X_i)$, $i=1,\ldots,n-1$
- Aggregated process: $X^{(m)}$

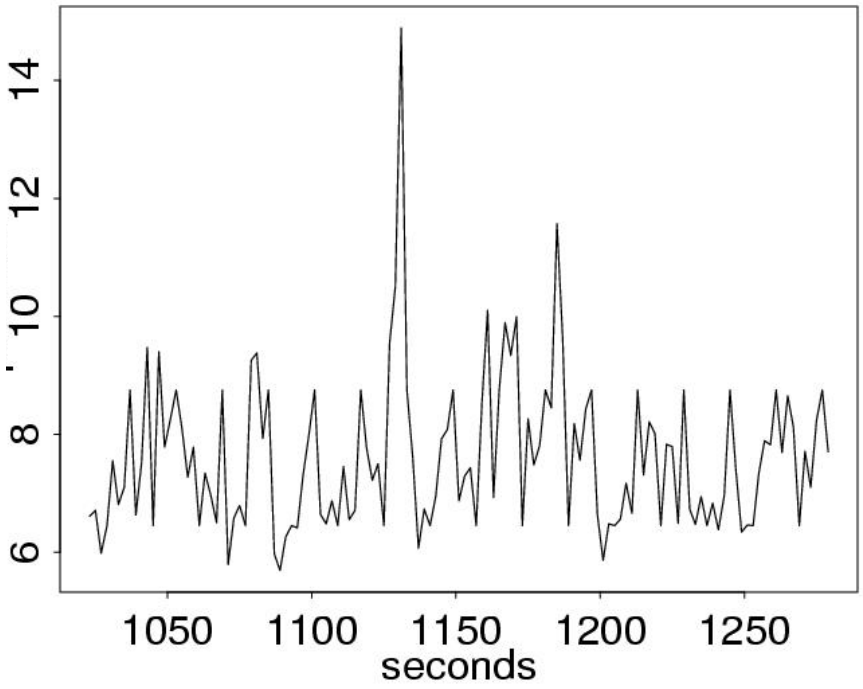
$$X^{(m)}(k) = \frac{1}{m}(X_{(k-1)m+1} + \ldots + X_{km}), k \geq 1$$

Time-series with R

- library(tseries)
- Computing m-aggregated time series:
  - vector.ts <- ts(vector,frequency=m)
  - vector.agg <- aggregate(vector.ts,FUN=mean)
  - decomposed.ts = stl(vector.ts,"periodic")
- Plot aggregated time series:
  - plot(vector.agg)
- Plot decomposed time series:
  - Plot(decomposed.ts)

Representation

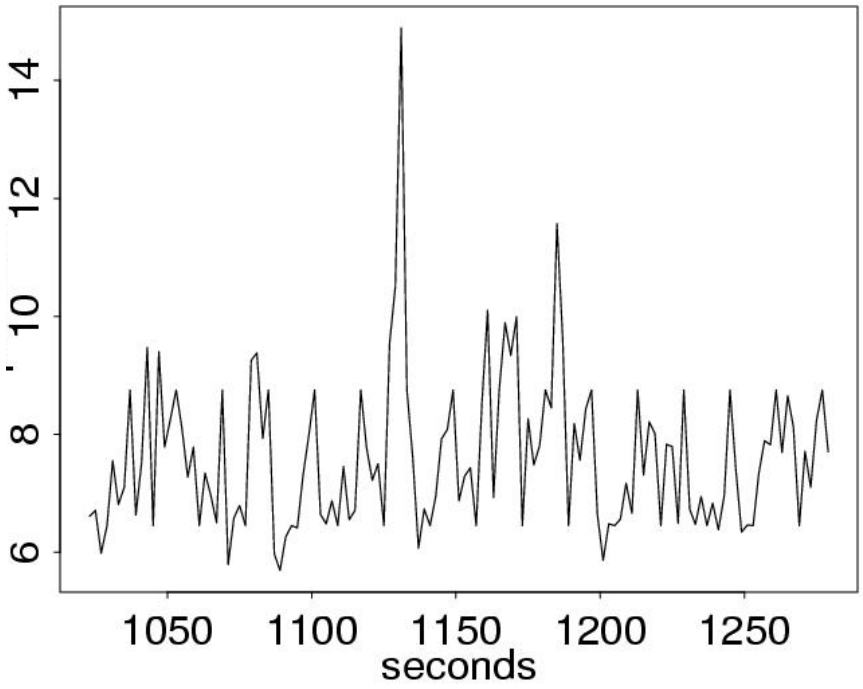
Timeseries



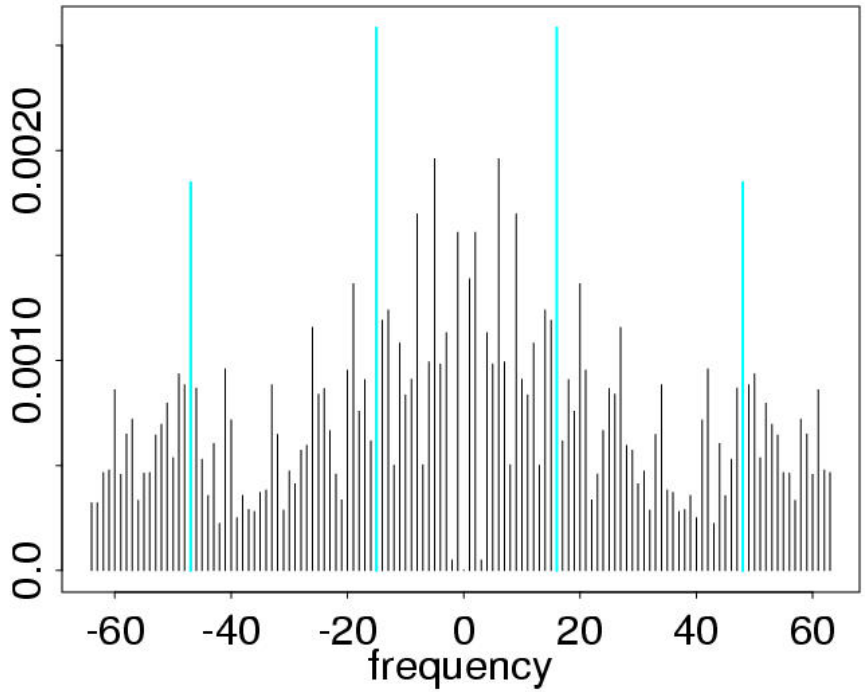
- Time series: information in time domain

Time or frequency

Timeseries



FFT



- Timeseries: information in time domain
- FFT: information in frequency (scale) domain

Time vs. frequency

- **Time**: assumptions about process increments, e.g., Gaussian, Pareto
- **Frequency**: assumes stationarity of frequency components of the process
- Check stationarity of low-order properties of process and its increments for subsets of the data:
  - mean(vector)
  - var(vector)
  - spectrum(vector)
  - diff(vector)

Auto-correlation

Auto-correlation function:

correlation between values of the process at different times s,t, as a function of the two times s,t or of the time difference t-s

$$R(s,t) = \frac{E[(X_t - \mu)(X_s - \mu)]}{\sigma^2}$$

Autocorrelation with R

- library(stats)
- plot(acf(vector))
- Compare Poisson, Pareto, and real data:
  - Poisson process is uncorrelated: 1 at lag 0, 0 otherwise
  - Pareto time-series is also uncorrelated: 1 at lag 0, 0 otherwise
  - Real data is correlated: acf(vector,log="x") shows slow decay

Root of correlation?

- Understanding the root of presence/absence of correlation is fundamental in time-series analysis:
  - Absence of correlation simplifies analysis/ modeling
  - Presence of correlation asks for detrending or more advanced modelling, e.g., heavy-tails
- Trick: differentiate the time series = detrending:
  - diff(vector, differences=x)
  - Until acf(diff(vector, differences=x)) shows signs of correlation, increase x.

Stationarity of increments

- Once the trends have been removed through differencing, increments should be checked for stationarity:
  - Library(fUnitRoots)

Normal:

- y = cumsum(c(0, xnorm1))
- adfTest(y)
- adfTest(xnorm1)

Real data:

- adfTest(vector.ts)

Short-Range Dependence

- A stationary process $X = (X_k : k \geq 1)$ with mean μ, variance σ^2 and autocorrelation function $r(k)$, $k \geq 1$, is said to exhibit short-range dependence (SRD) if there exists $0 < \rho < 1$ and $\tau > 0$ with

$$r(k)\frac{\tau}{\rho^k} \to 0 \quad as \quad k \to \infty$$

- Autocorrelations decay (at least) exponentially fast for large lags k

Short-Range Dependence

- The aggregated process $X^{(m)} = (X^{(m)}(k); k \geq 1)$ tends to second-order white noise, as $k \to \infty$

$$r^{(m)}(k) \to 0 \quad as \quad k \to \infty$$

for all $k \geq 1$, where $r^{(m)}$ denotes the autocorrelation function of $X^{(m)}$

- The variance-time function, i.e., the variance of the sample mean, as a function of m, satisfies:

$$\mathrm{var}(X^{(m)}) \sim \frac{c}{m} \quad as \quad m \to \infty$$

Advanced Time Series Analysis

Long-range dependence

- A stationary process $X = (X_k : k \geq 1)$ with mean μ, variance σ^2 and autocorrelation function $R(k)$, $k \geq 1$, is said to exhibit long-range dependence (LRD) if for some $1/2 < H < 1$ and

$$R(k) \sim ck^{2H-2} \quad as \quad k \to \infty$$

  where H is called the Hurst parameter

- Features of LRD
  - Infinite correlation length
  - Fluctuations over all time scales
  - No characteristic time scale

Long-range dependence

- The aggregated process $X^{(m)} = (X^{(m)}(k); k \geq 1)$ tends to non-degenerate limiting process, for m, k sufficiently large

$$r^{(m)}(k) \rightarrow r(k) \quad \text{as} \quad k \rightarrow \infty$$

- The variance-time function satisfies:

$$\text{var}(X^{(m)}) \sim cm^{2H-2} \quad as \quad m \rightarrow \infty$$

Asymptotic self-similarity

- Definition: A stationary process $X = (X_k : k \geq 1)$ is called asymptotically self-similar (with self-similarity parameter H, $0 < H < 1$), if for all *large enough m*,

$$X \approx m^{1-H} X^{(m)}$$

- Observations:
  - Asymptotic self-similarity is equivalent to long-range dependence of infinite correlation length
  - Asymptotic self-similarity does not specify the small-time scale behavior of a process

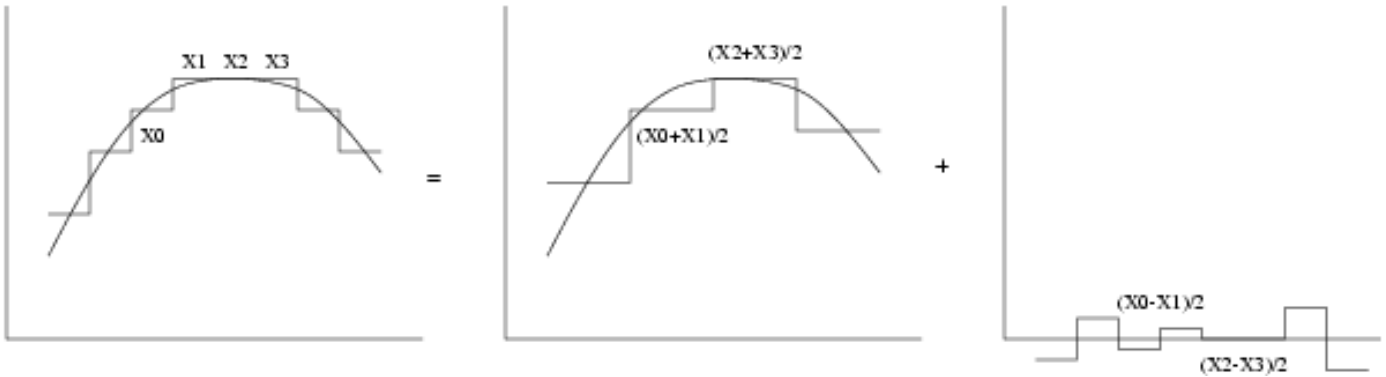
Wavelet analysis

- Intuition:
  - Finest scale:

    Compute averages of adjacent data points

    Compute differences between average and actual data
  - Next scale:

    Repeat based on averages from previous step



- Use wavelet coefficients to study scale or frequency dependent properties

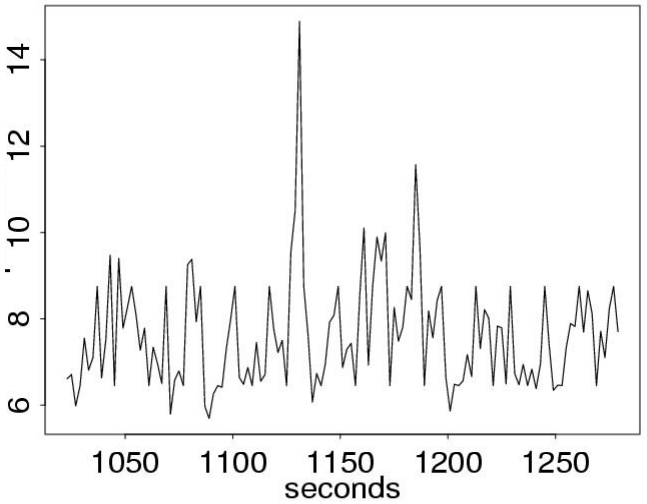
Wavelets

Time series: information in ***time*** domain

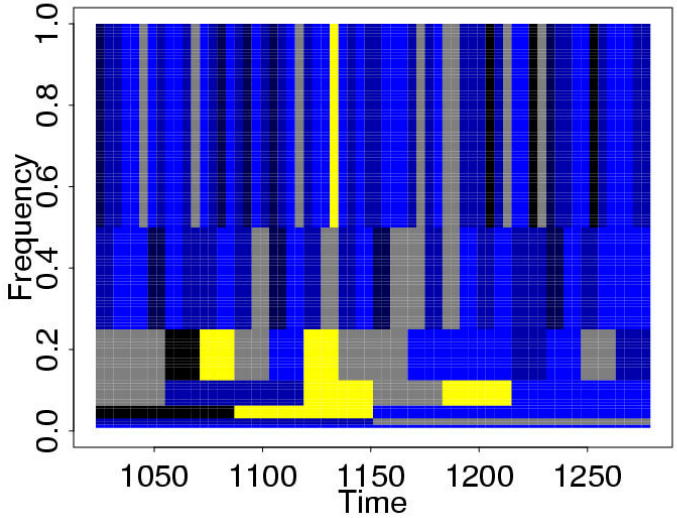
FFT: decomposition in ***frequency*** domain

Wavelets: localize a signal in both ***time*** and ***scale***

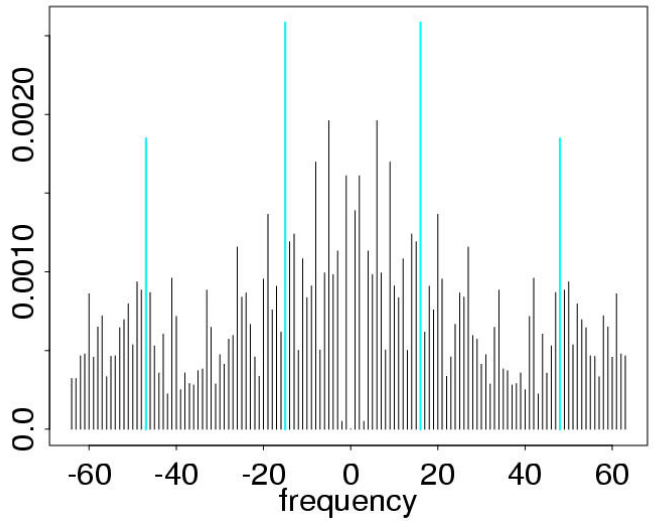
Timeseries



Wavelet transform



FFT



Discrete wavelet transform

- Coefficients at scale j and time $2^j k$:

$$d_{j,k} = \int X(s)\Psi_{j,k}(s)ds, \quad j \in Z, k \in Z$$

- Mother wavelet scaling: $\Psi_{j,k}(t) = 2^{-j/2}\Psi(2^{-j}t - k)$

- Wavelet decomposition:

$$X(t) = \sum_{j \in Z} \sum_{k \in Z} d_{j,k} \Psi_{j,k}(t)$$

Wavelets in R

- library(waveslim)
- Performing wavelet analysis:
  - vector.res <- dwt(vector[1:2**16],wf="d16",n.levels=10,boundary="periodic")
  - Length of analyzed time-series (L) must be a power of 2
  - Several wavelet functions available (wf)
  - Number of levels to be analyzed: $2^n <= L$

Wavelets

- Selecting wavelet coefficients at scale I
  - vector.res$di
- Computing log2 of their variance:
  - log2(var((vector.res$di)))
- Keeping energy of coefficients at each scale:
  - coefvar <- mat.or.vec(9,1)
  - coefvar[i] <- log2(var((vector.res$di)))
- Plot the log of the energy of the wavelet coefficients
  - Poisson process shows no scaling
  - Real traffic does

More Advanced Time Series Analysis

Scaling: intuition

~ absence of characteristic (time)scale

~ dependence among (time)scales

Many different types of scaling flavours :

- self-similarity
- long-range dependence
- multiscaling
- multifractality
- infinitely divisible cascades

Types of scaling

Let X(t) be a regularly sampled process :

- d(j,k) : process increment at timescale j and time k
- q : moment of the process
- j : timescale, j = 1,...,n.
- $E|X(t)|^q$: expectation of moment q of process X at time t (over all possible realizations of the process).

Assumption :

- d(j,k) is a homogeneous (stationary) zero-mean process.

Types of scaling

Queen Mary University of London

Self-similarity:

$$E|d(j,k)|^q \propto \exp(q\, H \ln(2^j))$$

Multiscaling:

$$E|d(j,k)|^q \propto \exp(H(q) \ln(2^j))$$

Infinitely divisible cascades:

$$E|d(j,k)|^q \propto \exp(H(q)\, n(2^j))$$

Mono-scaling

$$E|d(j,.)|^q = C_q(2^j)^{qH} \propto \exp(q\,H\,\ln(2^j))$$

- linear scaling among timescales and linear scaling among moments both driven by a single parameter H

$\Rightarrow$ called mono-scaling since H is constant

Multi-scaling

$$E|d(j,.)|^q = C_q(2^j)^{H(q)} \propto \exp(H(q)\ \ln(2^j))$$

- linear scaling among timescales j, but driven by a function H(.) of the moment q

⇒ mono-scaling becomes multi-scaling since scaling depends on the moment q

Cascades

$$E|d(j,.)|^q = C_q(2^j)^{H(q)} \propto \exp(H(q)\, n(2^j))$$

where

- H(q) represents 1 step of the cascade
- $n(2^j)$ represents cascade depth at timescale j

⇒ scaling does not depend linearly on the moment q nor the timescale j anymore

Separability

$$E|d(j,k)|^q \propto \exp(f(q)\, g(2^j))$$

- Separability of the moments (q) and the timescales (j)
- Higher-order moments q emphasize on larger irregularities
- Larger timescales j focus on smoothed versions of the process (zooming out)

⇒ framework to study irregularities and timescales independently

Link to probability theory

- Cascade : multiplicative process that breaks a process into smaller and smaller fragments according to some (deterministic or random) rule.
- Link with infinitely divisible distributions (Feller vol. 2) :

  F is infinitely divisible if for every n there exists a distribution F_n such that $F = F_n^{n*}$ where "*" denotes the convolution operator.

or equivalently

- F is infinitely divisible iff for each n it can be represented as the distribution of the sum $S_n = X_{1,n} + ... + X_{n,n}$ of n independent random variables with a common distribution F_n.

Analysis of Variance (ANOVA): Example

Example

- What aspects matter for BGP paths geographic length?
- Approach:
  - perform full-factorial simulations
  - Analyze the sensitivity of path choices to routing parameters

One-way analysis of variance

- R command:
  > aov.length = aov(depend~indep,data.geo)
- Parameters:
  - Dependent variable, e.g., path length
  - Independent variable(s), e.g., routing policies, AS size, iBGP mesh, IGP costs,…
  - Data structure
- Summary of the anova analysis:
  > summary(aov.length)
- Table of means:
  > print(model.tables(aov.length,"means"),digits=3)

One-way analysis of variance


```
> data.geo = read.table("anova.geo.best")
> aov.geo = aov(Y~POL,data.geo)
> summary(aov.geo)
             Df Sum Sq Mean Sq F value    Pr(>F)
POL           1 7.9350  7.9350  9235.2 < 2.2e-16 ***
Residuals  2398 2.0604  0.0009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Routing policies affect geographic path length

Multi-way analysis of variance

- In practice many parameters influence choice and length of BGP paths: policies, AS size, iBGP, IGP, peerings,…

> aov.geo = aov(Y~(POL*ASSIZE*HIER*IGP*CONN),data.geo)

> summary(aov.geo)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
POL	1	2.6279e+09	2627911820	78.8766	< 2.2e-16	***
ASSIZE	3	1.0815e+10	3604949017	108.2024	< 2.2e-16	***
HIER	1	2.7481e+09	2748059743	82.4829	< 2.2e-16	***
IGP	2	1.5599e+08	77994587	2.3410	0.096462	.
CONN	3	1.4506e+09	483542089	14.5135	2.299e-09	***
POL:ASSIZE	3	3.9356e+08	131186354	3.9376	0.008158	**
ASSIZE:CONN	9	8.2605e+08	91783497	2.7549	0.003324	**
Residuals	2280	7.5962e+10	33316733			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1