Nonstationary Extremes, II
Heteroscedastic Extremes in Practice

Manuele Leonelli

First LARS-IASC School on Computational Statistics and Data Science
Federal University of Bahia, Salvador, Brazil
November 15th, 2018

Plan for this session

1. Overview of available sofware
2. Implement univariate methods in R:
   (a) Block maxima approach
   (b) r-largest observations appraoch
   (c) Peaks over threshold
   (d) Heteroscedastic extensions
3. Discuss assumptions, diagnostics and model selection criteria

All code to implement the methods is available at

github.com/manueleleonelli.

Why R?

Three main reasons:

- R is free!
- R is open-source
- Largest set of functions for extreme value analysis (EVA)

Other sofware is available:

- In Matlab: `EVIM` and `WAFO`
- GUI software: `Xtremes` and `EXTREMES`
- Specialized software: `HYFRAN`, `GLSNet`

E. Gilleland, M. Ribatet, A.G. Stephenson (2013) A software review for extreme value analysis. *Extremes* 16:103-119.

R packages

Most functions for EVA are stored in R packages to perform a wide array of analyses:

- Univariate: `ismev`, `extRemes`, `evmix`, `POT` ...
- Bayesian: `evdbayes`, `revdbayes`, `MCMC4extremes` ...
- Multivariate: `evd`, `copula`, `extremis`, `texmex` ...
- Spatial: `spatialExtremes`, `spatialADAI` ...

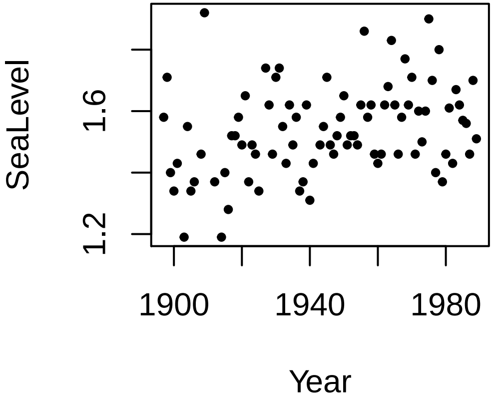
See this guide.

All these require at least some programming, as usual in R. One notable exception is in2extremes.

Let's get started!

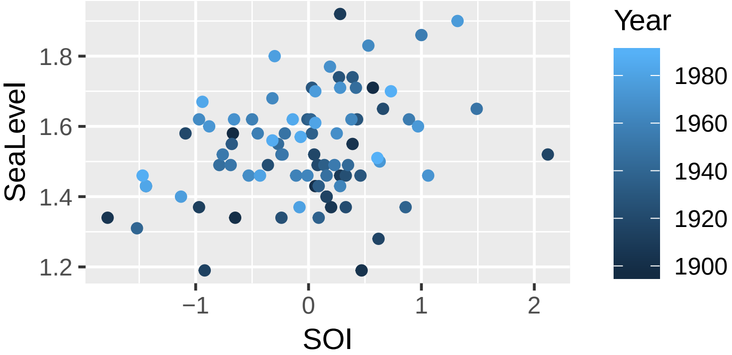
```
> library(ismev)
> data(fremantle)
> head(fremantle)
  Year SeaLevel   SOI
1 1897     1.58 -0.67
2 1898     1.71  0.57
3 1899     1.40  0.16
4 1900     1.34 -0.65
5 1901     1.43  0.06
7 1903     1.19  0.47
```

```
> plot(fremantle[,1:2], pch=19,cex=0.5)
```



Southern Oscillation Index (SOI)

```
> ggplot(fremantle,aes(x=SOI,y=SeaLevel,
+                       color=Year))+geom_point()
```



Fitting GEV distributions

```
> library(extRemes)
> ?fevd
```

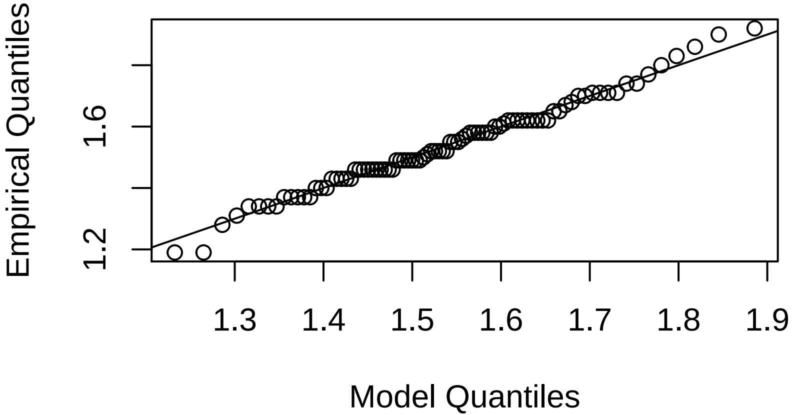
Let's look at the help

Another possibility is to use the `ismev` package.

E. Gilleland, R.W. Katz (2016) extRemes 2.0: An extreme value analysis package in R. *Journal of Statistical Software* 72:1-39.

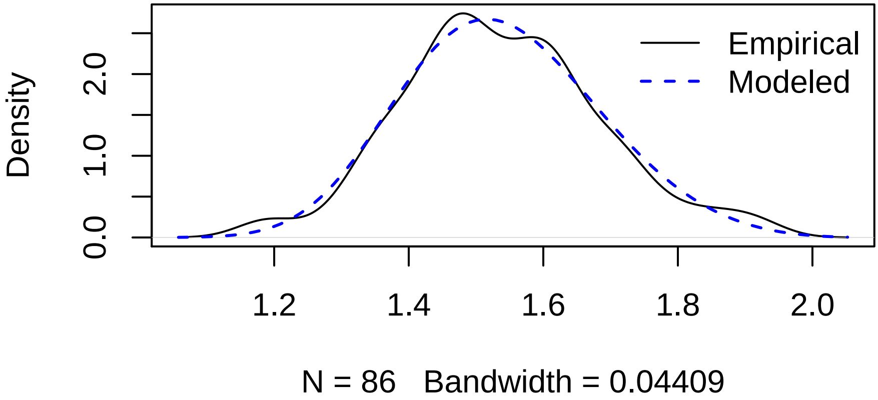
```
> fremantle_fit <- fevd(fremantle$SeaLevel,type="GEV")
> summary(fremantle_fit)
fevd(x = fremantle$SeaLevel, type = "GEV")

[1] "Estimation Method used: MLE"


 Negative Log-Likelihood Value:  -43.56663


 Estimated parameters:
  location      scale      shape
 1.4823417  0.1412723 -0.2174282

 Standard Error Estimates:
  location      scale      shape
0.01672527 0.01149706 0.06378114
```

```
> plot(fremantle_fit,type=c("qq"),main="")
```



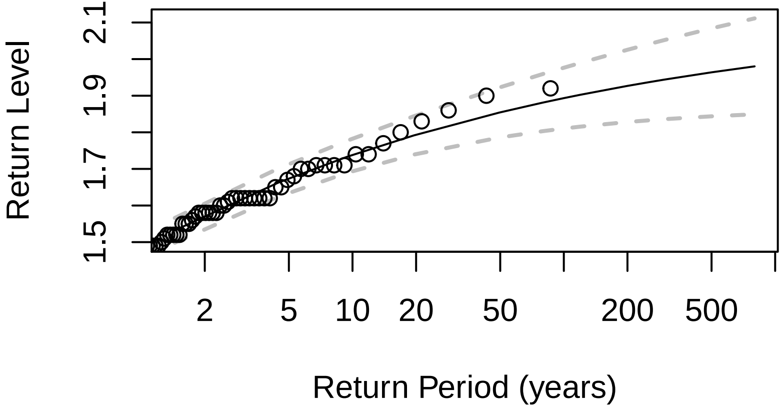
Diagnostics: densities

```
> plot(fremantle_fit,type=c("density"),main="")
```



Diagnostics: return levels

```
> plot(fremantle_fit,type=c("rl"),main="")
```



Inference: confidence intervals

```
> ci(fremantle_fit,type="parameter")
fevd(x = fremantle$SeaLevel, type = "GEV")

[1] "Normal Approx."

         95% lower CI   Estimate 95% upper CI
location    1.4495608  1.4823417   1.51512265
scale       0.1187385  0.1412723   0.16380615
shape      -0.3424370 -0.2174282  -0.09241948
```

Inference: return levels

```
> return.level(fremantle_fit,return.period=
+               c(2,10,20,50,100,500),do.ci=TRUE)
fevd(x = fremantle$SeaLevel, type = "GEV")

[1] "Normal Approx."

                     95% lower CI Estimate 95% upper CI
2-year return level      1.498259 1.532110     1.565962
10-year return level     1.689876 1.733753     1.777631
20-year return level     1.739108 1.791463     1.843817
50-year return level     1.785590 1.853927     1.922265
100-year return level    1.810194 1.893106     1.976017
500-year return level    1.843731 1.963815     2.083900
```

Bayesian GEV fitting

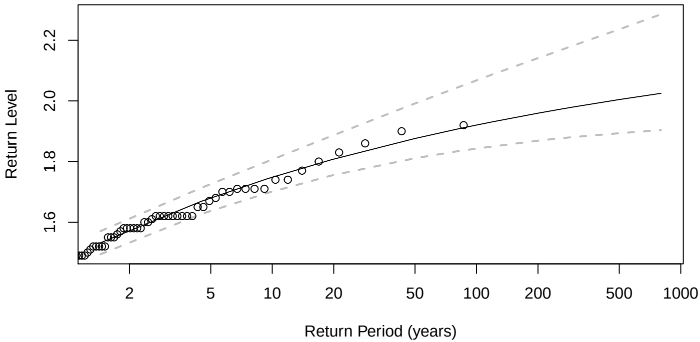
Let's look at the help again.

```
>  fremantle_fit_Bayes <- fevd(fremantle$SeaLevel,
+                                     type="GEV",method="Bayesian")
> summary(fremantle_fit_Bayes)
```

```
[1] "Quantiles of MCMC Sample from Posterior Distribution"

              2.5% Posterior Mean       97.5%
location  1.4432050      1.4806897  1.51878531
scale     0.1225513      0.1451410  0.17234688
shape    -0.3333519     -0.2014116 -0.05050336
```

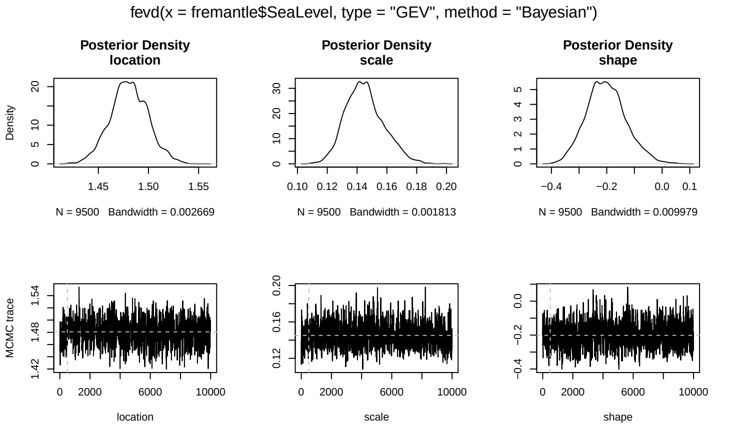
Bayesian return levels

```
> plot(fremantle_fit_Bayes,type="rl",main="")
```



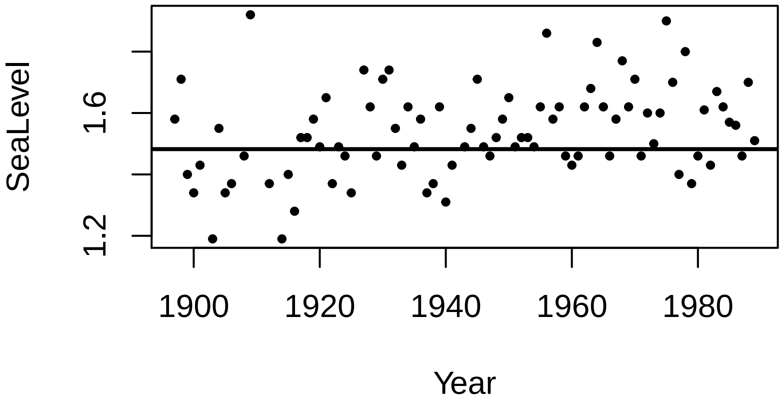
Convergence checking

```
> plot(fremantle_fit_Bayes,type="trace")
```



Estimated location of maximum Sea Level

```
> plot(fremantle[,1:2],pch=19,cex=0.5)
> abline(h=fremantle_fit$results$par[1],lwd=2)
```



Non-stationary fitting

```
> fremantle$Time <- fremantle$Year-min(fremantle$Year)
> fremantle_fit_time <- fevd(SeaLevel,data=fremantle,
+                              location.fun = ~Time)
> summary(fremantle_fit_time)
```

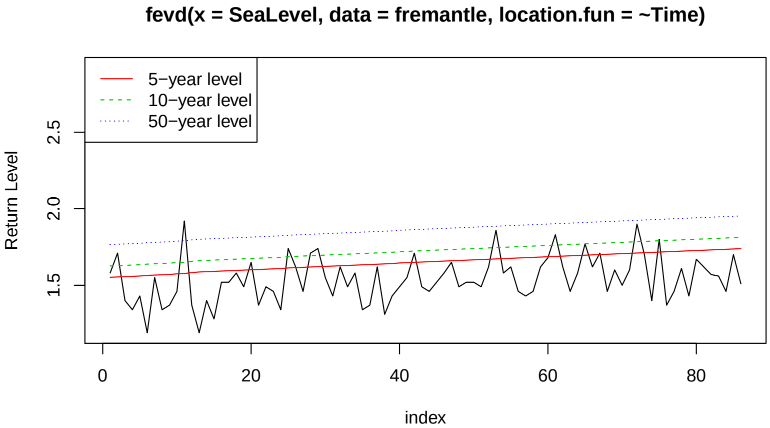
```
Estimated parameters:
        mu0          mu1        scale        shape
 1.382225781  0.002032151  0.124326256 -0.125309664

 Standard Error Estimates:
        mu0          mu1        scale        shape
0.0288458801 0.0004982185 0.0103749710 0.0682389684
```

Is there a time effect?

```
> ci(fremantle_fit_time,type="parameter")
fevd(x = SeaLevel, data = fremantle, location.fun = ~Time)

[1] "Normal Approx."

      95% lower CI     Estimate 95% upper CI
mu0     1.32568889  1.382225781  1.438762667
mu1     0.00105566  0.002032151  0.003008641
scale   0.10399169  0.124326256  0.144660825
shape  -0.25905558 -0.125309664  0.008436257
```

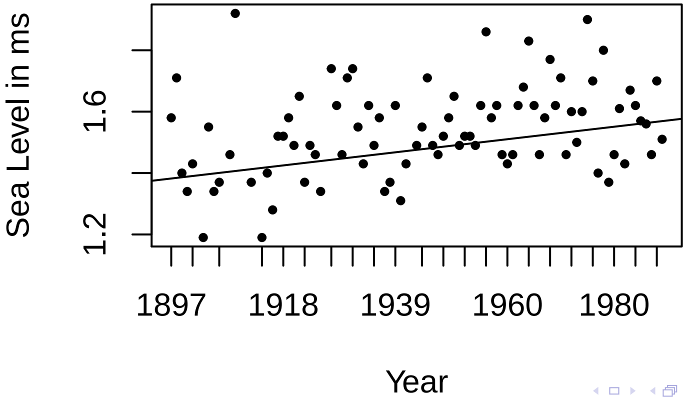
Time-varying return levels

```
> plot(fremantle_fit_time,"rl",rperiods=c(5,10,50))
```



Time-varying location of Sea Level

```
> plot(fremantle[,1:2],pch=19,cex=0.5)
> abline(a=fremantle_fit_time$results$par[1],
+                    b=fremantle_fit_time$results$par[2])
```

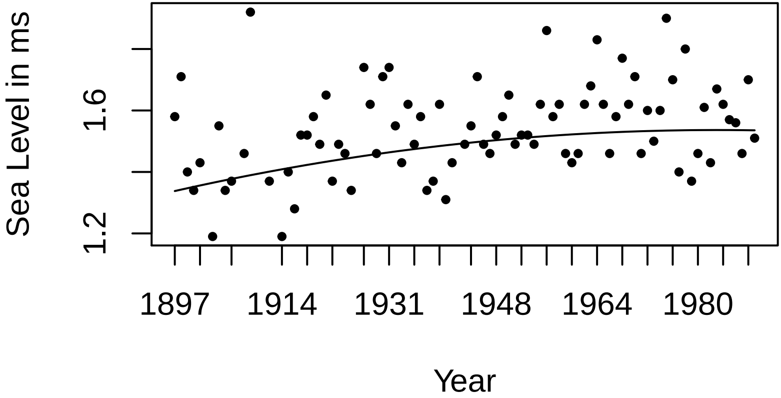


Model selection via hypothesis testing

```
> dev <- 2*(-fremantle_fit_time$results$value
+           +fremantle_fit$results$value)
> dev > qchisq(0.95,1)
[1] TRUE
> 1-pchisq(dev,1)
[1] 0.0003671508
```

Model	AIC	BIC
Stationary	-81.1	-73.8
Non-Stationary	-91.8	-82.0

A quadratic time effect?



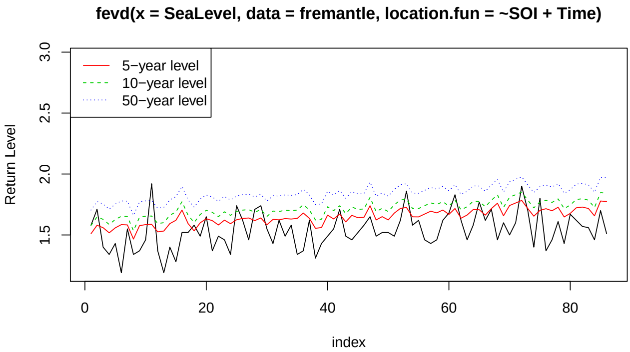
Deviance test has p-value 0.2231935 - *No significant quadratic effect*

Adding SOI to the model

```
> fremantle_fit_SOI <- fevd(SeaLevel,data=fremantle,
+                           location.fun = ~SOI+Time)
> ci(fremantle_fit_SOI,type="parameter")
fevd(x = SeaLevel, data = fremantle, location.fun = ~SOI + Tim

[1] "Normal Approx."

      95% lower CI     Estimate 95% upper CI
mu0    1.328079958  1.384328499   1.44057704
mu1    0.016019518  0.054519871   0.09302022
mu2    0.001141007  0.002114038   0.00308707
scale  0.101075964  0.120733366   0.14039077
shape -0.276802887 -0.149992773  -0.02318266
```

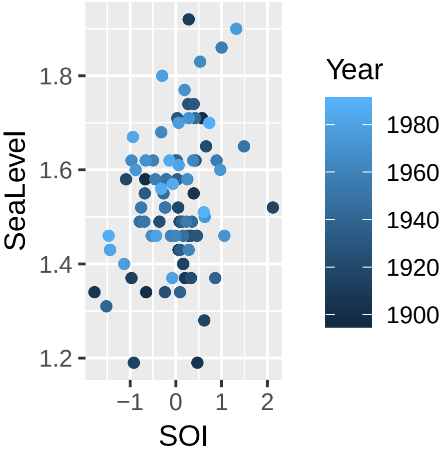
Time and SOI dependent returns

```
> plot(fremantle_fit_SOI,type="rl",rperiod=c(5,10,50))
```



Deviance test has p-value 5.47286e-06 - *Significant SOI effect*

What about an interaction?



We can test this easily using `location.fun = SOI+Time`.

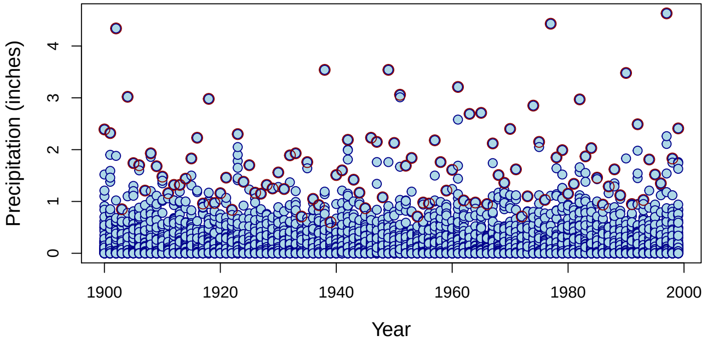
However the deviance test has p-value 0.6162236, thus no interaction effect.

Extracting maxima

```
> data(Fort)
> head(Fort)
  obs tobs month day year Prec
1   1    1     1   1 1900    0
2   2    2     1   2 1900    0
3   3    3     1   3 1900    0
4   4    4     1   4 1900    0
5   5    5     1   5 1900    0
6   6    6     1   6 1900    0
```

Extracting maxima

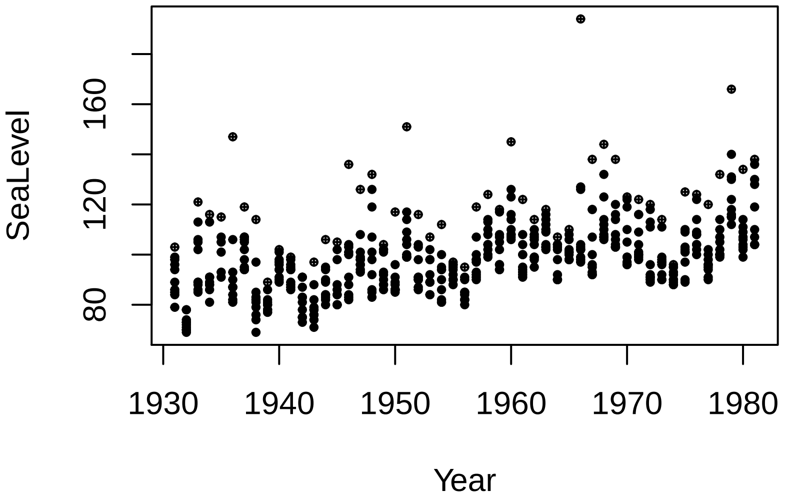
```
> bmFort <- blockmaxxer(Fort,
+                 blocks = Fort$year, which="Prec")
```



Venice sea levels

```
> data(venice)
> head(venice)
  Year  r1  r2  r3  r4  r5 r6 r7 r8 r9 r10
1 1931 103  99  98  96  94 89 86 85 84  79
2 1932  78  78  74  73  73 72 71 70 70  69
3 1933 121 113 106 105 102 89 89 88 86  85
4 1934 116 113  91  91  91 89 88 88 86  81
5 1935 115 107 105 101  93 91 NA NA NA  NA
6 1936 147 106  93  90  87 87 87 84 82  81
```

Venice sea levels



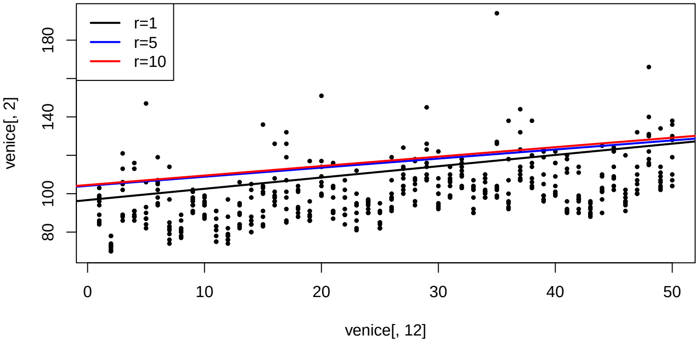
Fitting r-largest observations

```
> venice <- venice[-5,] #omit NA
> venice$time <- seq(1:nrow(venice)) #add time variable
> venice_fit_time <-fevd(r1,type="GEV",
+                         data = venice,location.fun = ~time)
> # Using the package ismev
> venice_fit_time_5 <- rlarg.fit(venice[,2:6],
+                                 ydat=venice,mul=12)
> venice_fit_time_10 <- rlarg.fit(venice[,2:11],
+                                 ydat=venice, mul=12)
```

r	β_0	β_1	σ	ξ
1	96.7 (4.3)	0.59 (0.14)	14.6 (1.6)	-0.022 (0.084)
5	104.1 (2.0)	0.47 (0.06)	12.3 (0.8)	-0.033 (0.043)
10	104.54 (1.7)	0.49 (0.04)	11.75 (0.7)	-0.06 (0.028)

Time-dependent location

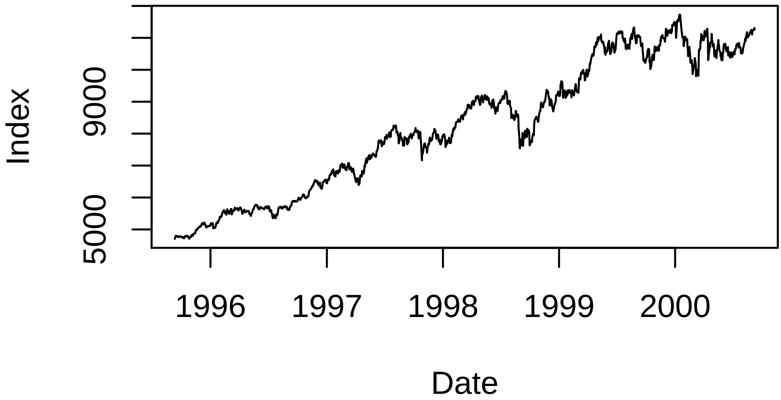
Various diagnostics can be plotted using `rlarg.diag`



Perhaps a “seasonal” effect?

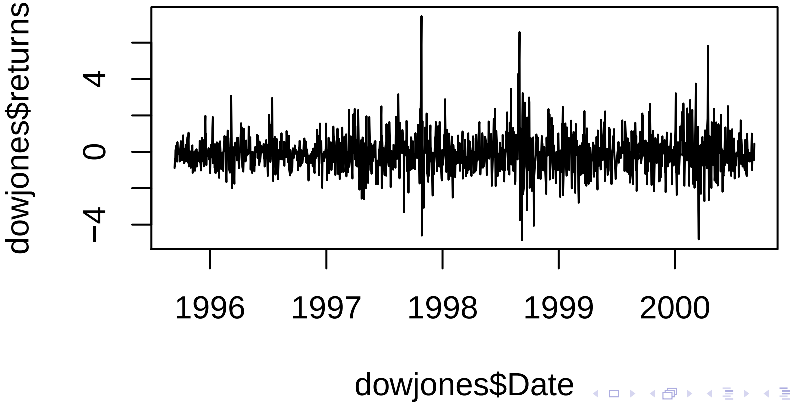
A financial application: the Dow Jones Index

```
> data(dowjones)
> plot(dowjones,type='l')
```



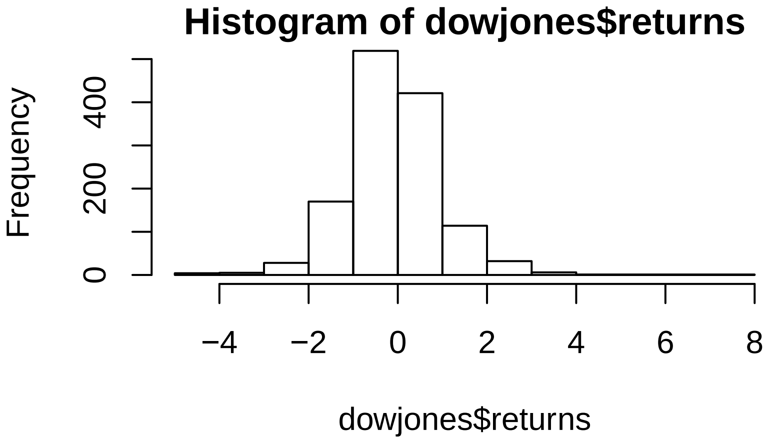
The negative log-returns

```
> dowjones <- data.frame(dowjones[-1,],
+                    returns=-100*diff(log(dowjones$Index)))
> plot(dowjones$Date,dowjones$returns,type="l")
```



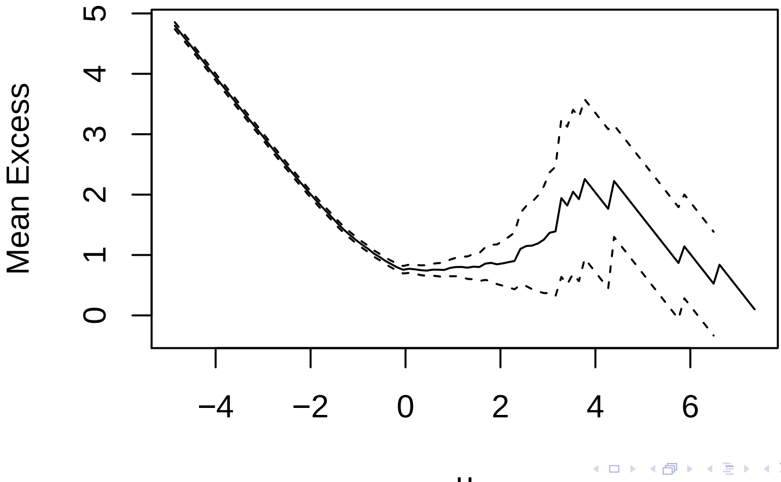
The returns' distribution

```
> hist(dowjones$returns)
```



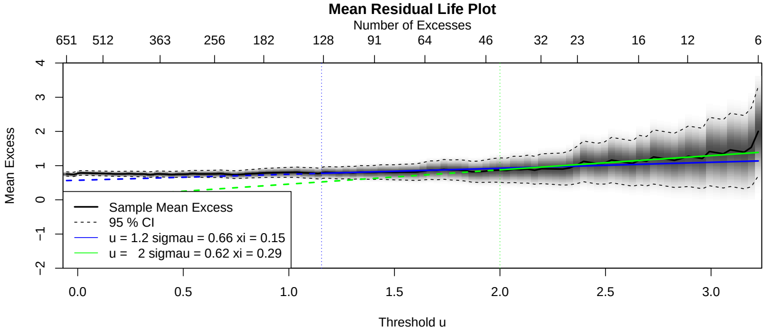
Choosing the threshold: MRL plot

```
> mrl.plot(dowjones$returns)
```



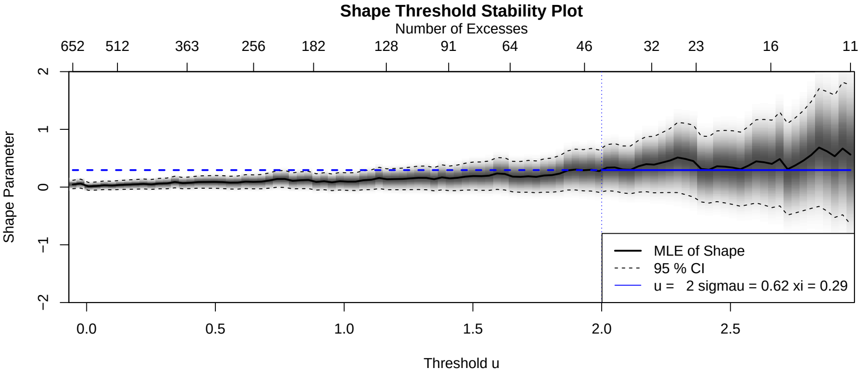
Choosing the threshold: MRL plot

```
> library(evmix)
> mrlplot(dowjones$returns,
+         try.thresh=c(quantile(dowjones$returns,0.9),2),
+         ylim = c(-2,4))
```



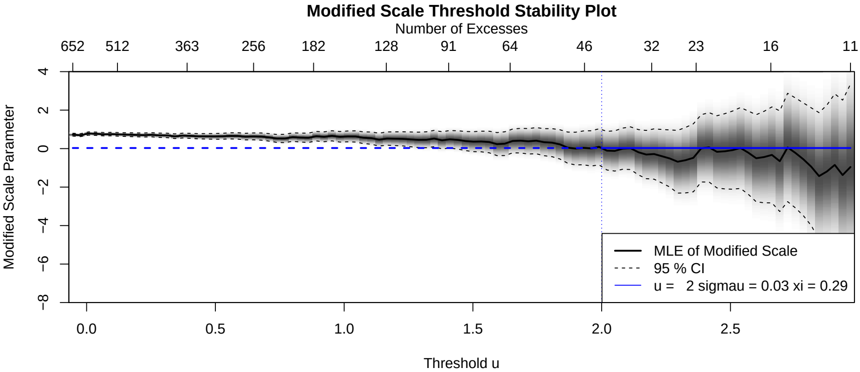
Choosing the threshold: stability plots

```
> tshapeplot(dowjones$returns,ylim=c(-2,2),try.thresh = 2)
```



Choosing the threshold: stability plots

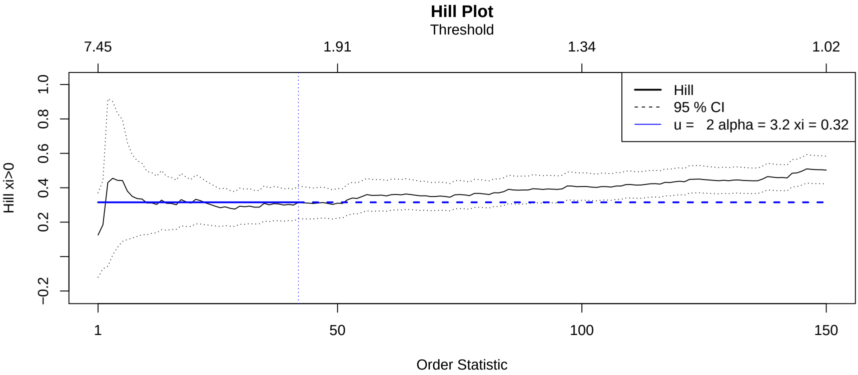
```
> tscaleplot(dowjones$returns,ylim=c(-8,4),try.thresh = 2)
```



Choosing the threshold: hill plot

This is only valid for heavy-tailed distributions $\xi > 0$

```
> hillplot(dowjones$returns,orderl=c(1,150),
+           try.thresh= 2,xlab="Order Statistic")
```



Fitting the GPD to exceedances

```
> dow_fit <- fevd(dowjones$returns,type="GP",threshold = 2)
> summary(dow_fit)
```

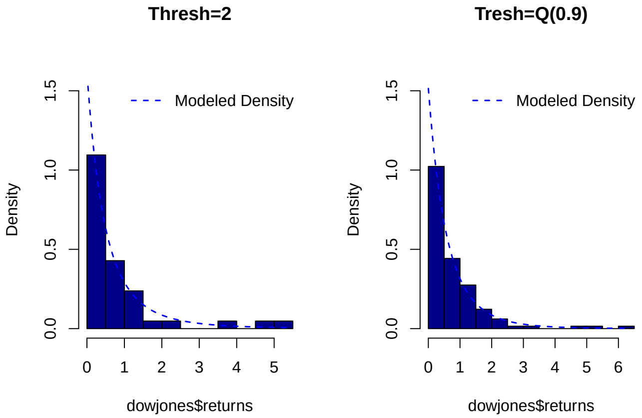
```
Estimated parameters:
    scale     shape
0.6183804 0.2941935

 Standard Error Estimates:
    scale     shape
0.1496571 0.1918915
```

Fitting the GPD to exceedances

```
> dow_fit2 <- fevd(dowjones$returns,type="GP",
+                   threshold=quantile(dowjones$returns,0.9))
> summary(dow_fit2)
```

```
Estimated parameters:
    scale     shape
0.6593380 0.1490148

 Standard Error Estimates:
     scale      shape
0.08408249 0.09380125
```

Estimated densities



Estimated parameters and return levels

Thresh = 2

	lower	mean	upper
scale	0.325	0.618	0.911
shape	-0.081	0.294	0.670

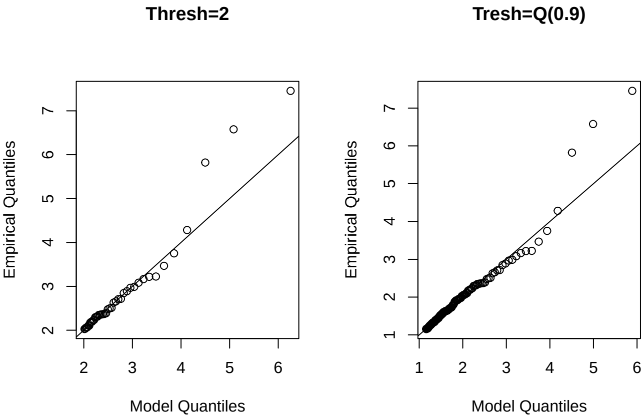
	lower	mean	upper
2-year	3.43	5.22	7.00
20-year	1.35	10.4	19.40
100-year	-5.74	16.7	39.20

Thresh = q(0.9)

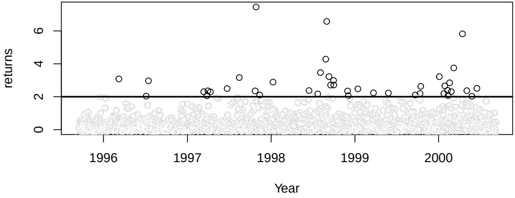
	lower	mean	upper
scale	0.494	0.659	0.824
shape	-0.034	0.194	0.333

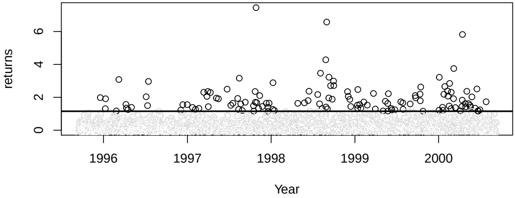
	lower	mean	upper
2-year	3.77	5.22	7.00
20-year	4.25	8.56	12.86
100-year	3.56	11.7	19.97

Quantile plots



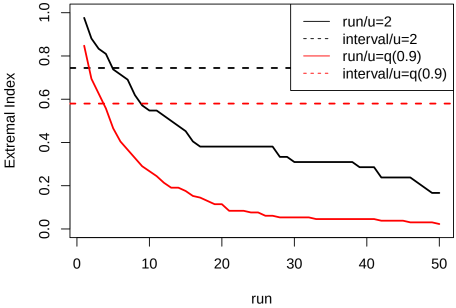
Are returns independent?





Extremal index

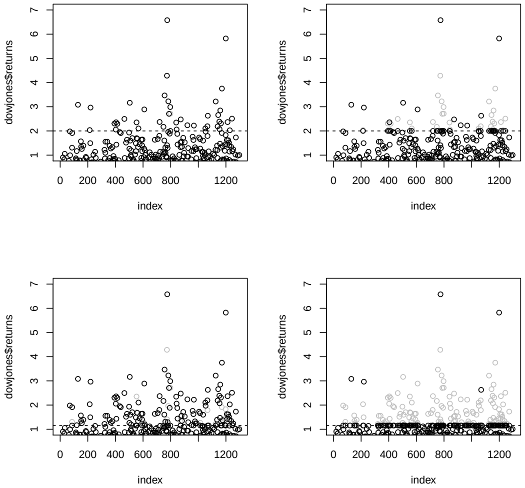
The extremal index θ measures the level of clustering of extreme events ($\theta = 1$ corresponding to independence)



C.A.T. Ferro,J. Segers (2003) Inference for clusters of extreme values . *Journal of the Royal Statistical Society Series B* 65:545-556.

Declustering

We can use the function `decluster` to extract the cluster maxima.



Results after declustering

Thresh.	r	σ	ξ	Ext.Index
2	0	0.62 (0.15)	0.29 (0.19)	0.74
2	5	0.61 (0.15)	0.31 (0.20)	0.76
2	7	0.79 (0.23)	0.24 (0.22)	1
q(0.9)	0	0.66 (0.08)	0.15 (0.09)	0.57
q(0.9)	4	0.72 (0.10)	0.12 (0.09)	0.70
q(0.9)	7	1.10 (0.22)	0.07 (0.14)	1

Heterоscedasticity

- Extremes may also not be identically distributed.
- The scedasis function gives a representation of tail risk.
- To compute this we need to restrict ourselves to the case $\xi > 0$ and constant.
- The `extremis` package gives an easy implementation for the scedasis.

J.H.J Einmahl, L. de Haan, C. Zhou (2016) Statistics of heteroscedastic extremes. *Journal of the Royal Statistical Society Series B* 78:31-51.

Testing for constant ξ

Here we implement the following test

1. Compute the Hill estimator $\hat{\xi}$ over the full time series using the k highest observations
2. Divide the time series into m blocks of equal length
3. Compute the Hill estimator $\hat{\xi}_l$ over each block using the $\lfloor k/m \rfloor + 1$ highest observations
4. Compute the test statistics

$$T = \frac{1}{m} \sum_{l=1}^{m} \left(\frac{\hat{\gamma}_l}{\hat{\gamma}} - 1 \right)^2$$

5. Compare kT with the quantile $\chi^2_{m-1}(1-\alpha)$ for a confidence level α

Results for the Dow Jones series

The values k were chosen to match the threshold used till now.

m	k	p-value
4	42	0.99
4	130	0.67
3	42	0.87
3	130	0.99

The assumption of constant ξ looks tenable.

Fitting the scedasis function

We can use the function `cdensity` to estimate the scedasis.

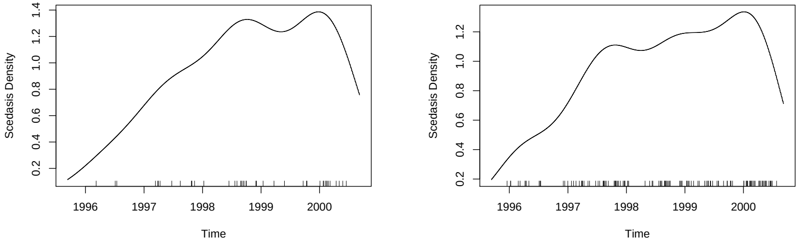


Figure: Scedasis function: threshold=2 (left), threshold=q(0.9) (right)

Fitting the scedasis cdf

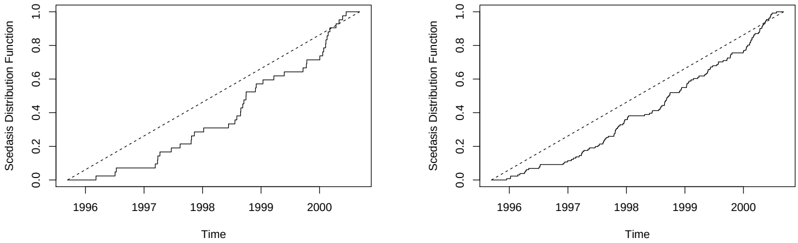


Figure: Scedasis cdf: threshold=2 (left), threshold=q(0.9) (right)

Conclusions

- R provide a set of packages to carry out a large number of inferential routines for extremes, including non-stationary, heteroscedastic extensions.
- Tomorrow we will look at the implementation of multivariate methods
- On Saturday you will actively fit extreme models to data.