TP Introduction To Extreme Value Theory

BERGUIGA Oussama

March 22,2018

Introduction

The main purpose of that project is to use main results of the Extreme Value Theory on different data sets (portpirie, dow jones and glass)

Packages loading

```
setwd('C:\\Users\\oussa\\Downloads\\Data Science\\Master Statistique Big Data Dauphine\\Module 2\\Valeurs Extremes')
library(evir)
library(ismev)

## Warning: package 'ismev' was built under R version 3.3.3

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-15. For overview type 'help("mgcv-package")'.

library(fitdistrplus)

## Warning: package 'fitdistrplus' was built under R version 3.3.3

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 3.3.3

## Loading required package: survival

## Warning: package 'survival' was built under R version 3.3.3

library(fExtremes)

## Warning: package 'fExtremes' was built under R version 3.3.3

## Loading required package: timeDate

## Warning: package 'timeDate' was built under R version 3.3.3

## Loading required package: timeSeries

## Warning: package 'timeSeries' was built under R version 3.3.3

## Loading required package: fBasics
```

```
## Warning: package 'fBasics' was built under R version 3.3.3

##

## Rmetrics Package fBasics

## Analysing Markets and calculating Basic Statistics

## Copyright (C) 2005-2014 Rmetrics Association Zurich

## Educational Software for Financial Engineering and Computational Science

## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.

## https://www.rmetrics.org --- Mail to: info@rmetrics.org

## Loading required package: fGarch

## Warning: package 'fGarch' was built under R version 3.3.3

##
## Attaching package: 'fExtremes'

## The following objects are masked from 'package:evir':
##
##     dgev, dgpd, pgev, pgpd, qgev, qgpd, rgev, rgpd
```

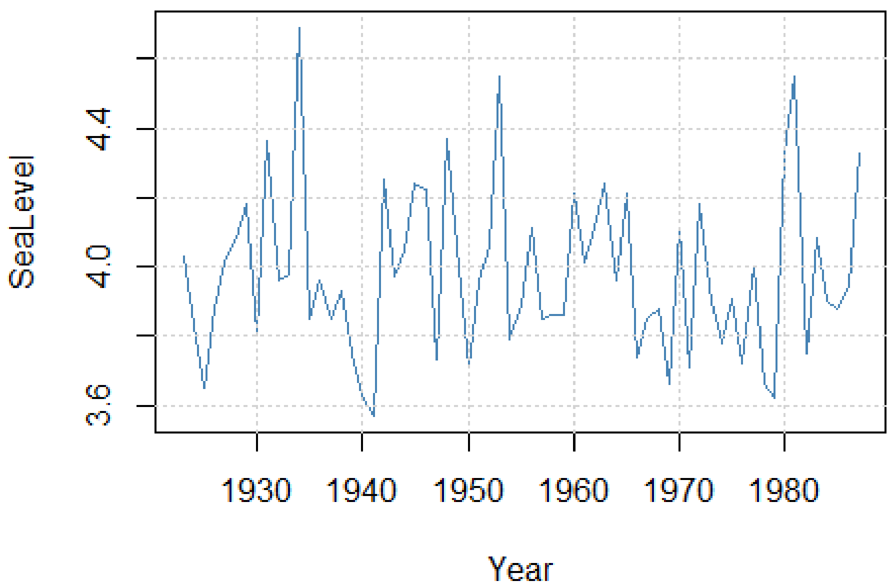
Study of PORTPIRIE Data Set

In that dataset we find annual maximum sea levels recorded at Port Pirie, in South Australia

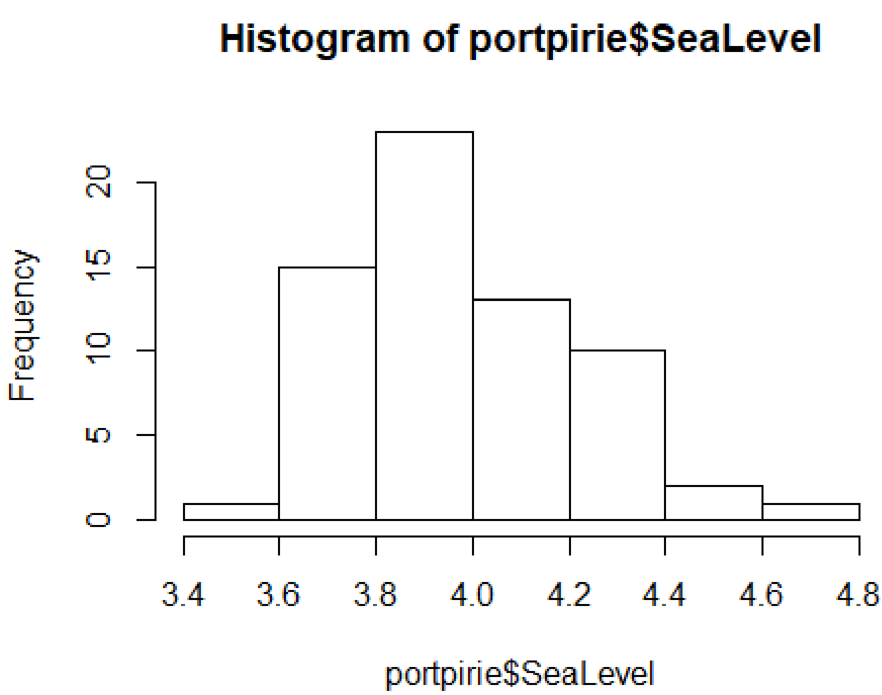
```
data(portpirie)
?portpirie

## starting httpd help server ... done
```

Descriptive Statistics

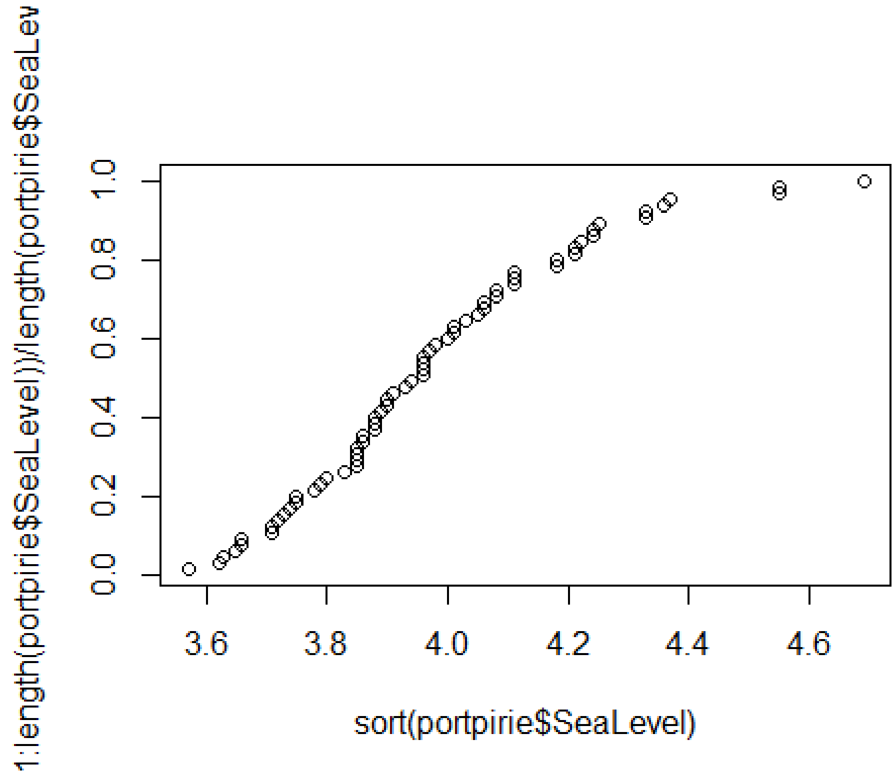
```
nrow(portpirie)

## [1] 65

summary(portpirie$SeaLevel)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.570   3.830   3.960   3.981   4.110   4.690

plot(portpirie,type="l",col="steelblue")

grid()
```




```
hist(portpirie$SeaLevel)
```




```
plot(sort(portpirie$SeaLevel),(1:length(portpirie$SeaLevel))
     /length(portpirie$SeaLevel))
```



Most of the sea levels maximum are around 4 meters, the maximum of the maximums is around 4.7 meters

Choice of modelisation

Let us try a GEV model for the portpirie case

Choice of the blocs

We already have the annual maximum of porpotries SeaLevel,here we have no choice, so it is annual blocks, which is a good pragmatic option.

Estimation of the parameters

First we try a gevfit

```
fitmle_gev_portpirie1=gevFit(portpirie$SeaLevel,type="mle",block=1)

fitmle_gev_portpirie1

##
## Title:
##  GEV Parameter Estimation
##
## Call:
##  gevFit(x = portpirie$SeaLevel, block = 1, type = "mle")
##
```

```
## Estimation Type:
##   gev mle
##
## Estimated Parameters:
##          xi          mu        beta
## -0.05013639  3.87475503  0.19804734
##
## Description
##   Thu Mar 22 22:50:10 2018
```

```
#gamma=-0.05 b=3.87 a = 0.2

#Weibull attraction domain

fitmle_gev_portpirie2=gev.fit(portpirie$SeaLevel)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] -4.339058
##
## $mle
## [1]  3.87474692  0.19804120 -0.05008773
##
## $se
## [1] 0.02793211 0.02024610 0.09825633
```

```
fitmle_gev_portpirie2
```

```
## $trans
## [1] FALSE
##
## $model
## $model[[1]]
## NULL
##
## $model[[2]]
## NULL
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
##
## $conv
## [1] 0
##
## $nllh
```

```
## [1] -4.339058
##
## $data
##  [1] 4.03 3.83 3.65 3.88 4.01 4.08 4.18 3.80 4.36 3.96 3.98 4.69 3.85 3.96
## [15] 3.85 3.93 3.75 3.63 3.57 4.25 3.97 4.05 4.24 4.22 3.73 4.37 4.06 3.71
## [29] 3.96 4.06 4.55 3.79 3.89 4.11 3.85 3.86 3.86 4.21 4.01 4.11 4.24 3.96
## [43] 4.21 3.74 3.85 3.88 3.66 4.11 3.71 4.18 3.90 3.78 3.91 3.72 4.00 3.66
## [57] 3.62 4.33 4.55 3.75 4.08 3.90 3.88 3.94 4.33
##
## $mle
## [1]  3.87474692  0.19804120 -0.05008773
##
## $cov
##               [,1]          [,2]         [,3]
## [1,]  0.0007802026  0.0001970387 -0.001074124
## [2,]  0.0001970387  0.0004099046 -0.000777249
## [3,] -0.0010741241 -0.0007772490  0.009654306
##
## $se
## [1] 0.02793211 0.02024610 0.09825633
##
## $vals
##           [,1]      [,2]        [,3]
##  [1,] 3.874747 0.1980412 -0.05008773
##  [2,] 3.874747 0.1980412 -0.05008773
##  [3,] 3.874747 0.1980412 -0.05008773
##  [4,] 3.874747 0.1980412 -0.05008773
##  [5,] 3.874747 0.1980412 -0.05008773
##  [6,] 3.874747 0.1980412 -0.05008773
##  [7,] 3.874747 0.1980412 -0.05008773
##  [8,] 3.874747 0.1980412 -0.05008773
##  [9,] 3.874747 0.1980412 -0.05008773
## [10,] 3.874747 0.1980412 -0.05008773
## [11,] 3.874747 0.1980412 -0.05008773
## [12,] 3.874747 0.1980412 -0.05008773
## [13,] 3.874747 0.1980412 -0.05008773
## [14,] 3.874747 0.1980412 -0.05008773
## [15,] 3.874747 0.1980412 -0.05008773
## [16,] 3.874747 0.1980412 -0.05008773
## [17,] 3.874747 0.1980412 -0.05008773
## [18,] 3.874747 0.1980412 -0.05008773
## [19,] 3.874747 0.1980412 -0.05008773
## [20,] 3.874747 0.1980412 -0.05008773
## [21,] 3.874747 0.1980412 -0.05008773
## [22,] 3.874747 0.1980412 -0.05008773
## [23,] 3.874747 0.1980412 -0.05008773
## [24,] 3.874747 0.1980412 -0.05008773
## [25,] 3.874747 0.1980412 -0.05008773
## [26,] 3.874747 0.1980412 -0.05008773
## [27,] 3.874747 0.1980412 -0.05008773
```

```
## [28,] 3.874747 0.1980412 -0.05008773
## [29,] 3.874747 0.1980412 -0.05008773
## [30,] 3.874747 0.1980412 -0.05008773
## [31,] 3.874747 0.1980412 -0.05008773
## [32,] 3.874747 0.1980412 -0.05008773
## [33,] 3.874747 0.1980412 -0.05008773
## [34,] 3.874747 0.1980412 -0.05008773
## [35,] 3.874747 0.1980412 -0.05008773
## [36,] 3.874747 0.1980412 -0.05008773
## [37,] 3.874747 0.1980412 -0.05008773
## [38,] 3.874747 0.1980412 -0.05008773
## [39,] 3.874747 0.1980412 -0.05008773
## [40,] 3.874747 0.1980412 -0.05008773
## [41,] 3.874747 0.1980412 -0.05008773
## [42,] 3.874747 0.1980412 -0.05008773
## [43,] 3.874747 0.1980412 -0.05008773
## [44,] 3.874747 0.1980412 -0.05008773
## [45,] 3.874747 0.1980412 -0.05008773
## [46,] 3.874747 0.1980412 -0.05008773
## [47,] 3.874747 0.1980412 -0.05008773
## [48,] 3.874747 0.1980412 -0.05008773
## [49,] 3.874747 0.1980412 -0.05008773
## [50,] 3.874747 0.1980412 -0.05008773
## [51,] 3.874747 0.1980412 -0.05008773
## [52,] 3.874747 0.1980412 -0.05008773
## [53,] 3.874747 0.1980412 -0.05008773
## [54,] 3.874747 0.1980412 -0.05008773
## [55,] 3.874747 0.1980412 -0.05008773
## [56,] 3.874747 0.1980412 -0.05008773
## [57,] 3.874747 0.1980412 -0.05008773
## [58,] 3.874747 0.1980412 -0.05008773
## [59,] 3.874747 0.1980412 -0.05008773
## [60,] 3.874747 0.1980412 -0.05008773
## [61,] 3.874747 0.1980412 -0.05008773
## [62,] 3.874747 0.1980412 -0.05008773
## [63,] 3.874747 0.1980412 -0.05008773
## [64,] 3.874747 0.1980412 -0.05008773
## [65,] 3.874747 0.1980412 -0.05008773
##
## attr(,"class")
## [1] "gev.fit"

# b = 3.87474692  a = 0.19804120 gamma = -0.05008773


#95% confidence intervalfor gamma:

IC=c(-0.05008773-0.09825633*2/sqrt(65),-0.05008773+0.09825633*2/sqrt(65))
IC
```

```
## [1] -0.07446213 -0.02571333
#0 does not seem to be part of the confidence interval
```

Let us try the probability weigthed moments method (better if we do not have enough data)

```
fitpwrgev_portprie = gevFit(portpirie$SeaLevel,type="pwm")
fitpwrgev_portprie

##
## Title:
##  GEV Parameter Estimation
##
## Call:
##  gevFit(x = portpirie$SeaLevel, type = "pwm")
##
## Estimation Type:
##   gev pwm
##
## Estimated Parameters:
##          xi          mu        beta
## -0.05119127  3.87314570  0.20321876
##
## Description
##   Thu Mar 22 22:50:10 2018

#results are clothes to those with Maximum Likelihood Estimator
```

Let us try a Gumbel modelisation

```
fitmle_gumbel_portpirie1=gumbelFit(portpirie$SeaLevel,type=c("mle"))

fitmle_gumbel_portpirie1

##
## Title:
##  Gumbel Parameter Estimation
##
## Call:
##  gumbelFit(x = portpirie$SeaLevel, type = c("mle"))
##
## Estimation Type:
##   gum mle
##
## Estimated Parameters:
##        mu      beta
## 3.8694924 0.1948879
##
## Description
##   Thu Mar 22 22:50:11 2018

fitmle_gumbel_portpirie2=gum.fit(portpirie$SeaLevel,type=c("pwm"))
```

```
## $conv
## [1] 0
##
## $nllh
## [1] -4.217682
##
## $mle
## [1] 3.8694426 0.1948867
##
## $se
## [1] 0.02549356 0.01885190
```

```
fitmle_gumbel_portpirie2
```

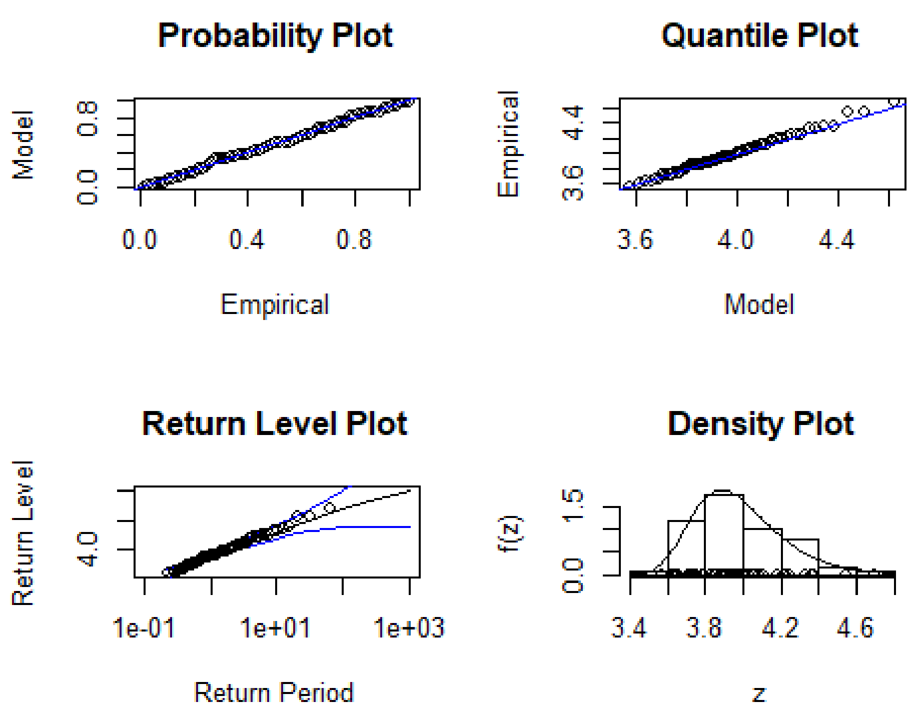
```
## $trans
## [1] FALSE
##
## $model
## $model[[1]]
## NULL
##
## $model[[2]]
## NULL
##
##
## $link
## [1] "identity" "identity"
##
## $conv
## [1] 0
##
## $nllh
## [1] -4.217682
##
## $data
##  [1] 4.03 3.83 3.65 3.88 4.01 4.08 4.18 3.80 4.36 3.96 3.98 4.69 3.85 3.96
## [15] 3.85 3.93 3.75 3.63 3.57 4.25 3.97 4.05 4.24 4.22 3.73 4.37 4.06 3.71
## [29] 3.96 4.06 4.55 3.79 3.89 4.11 3.85 3.86 3.86 4.21 4.01 4.11 4.24 3.96
## [43] 4.21 3.74 3.85 3.88 3.66 4.11 3.71 4.18 3.90 3.78 3.91 3.72 4.00 3.66
## [57] 3.62 4.33 4.55 3.75 4.08 3.90 3.88 3.94 4.33
##
## $mle
## [1] 3.8694426 0.1948867
##
## $cov
##              [,1]         [,2]
## [1,] 0.0006499218 0.0001526974
## [2,] 0.0001526974 0.0003553943
##
## $se
```

```
## [1] 0.02549356 0.01885190
##
## $vals
##           [,1]      [,2]
##  [1,] 3.869443 0.1948867
##  [2,] 3.869443 0.1948867
##  [3,] 3.869443 0.1948867
##  [4,] 3.869443 0.1948867
##  [5,] 3.869443 0.1948867
##  [6,] 3.869443 0.1948867
##  [7,] 3.869443 0.1948867
##  [8,] 3.869443 0.1948867
##  [9,] 3.869443 0.1948867
## [10,] 3.869443 0.1948867
## [11,] 3.869443 0.1948867
## [12,] 3.869443 0.1948867
## [13,] 3.869443 0.1948867
## [14,] 3.869443 0.1948867
## [15,] 3.869443 0.1948867
## [16,] 3.869443 0.1948867
## [17,] 3.869443 0.1948867
## [18,] 3.869443 0.1948867
## [19,] 3.869443 0.1948867
## [20,] 3.869443 0.1948867
## [21,] 3.869443 0.1948867
## [22,] 3.869443 0.1948867
## [23,] 3.869443 0.1948867
## [24,] 3.869443 0.1948867
## [25,] 3.869443 0.1948867
## [26,] 3.869443 0.1948867
## [27,] 3.869443 0.1948867
## [28,] 3.869443 0.1948867
## [29,] 3.869443 0.1948867
## [30,] 3.869443 0.1948867
## [31,] 3.869443 0.1948867
## [32,] 3.869443 0.1948867
## [33,] 3.869443 0.1948867
## [34,] 3.869443 0.1948867
## [35,] 3.869443 0.1948867
## [36,] 3.869443 0.1948867
## [37,] 3.869443 0.1948867
## [38,] 3.869443 0.1948867
## [39,] 3.869443 0.1948867
## [40,] 3.869443 0.1948867
## [41,] 3.869443 0.1948867
## [42,] 3.869443 0.1948867
## [43,] 3.869443 0.1948867
## [44,] 3.869443 0.1948867
## [45,] 3.869443 0.1948867
## [46,] 3.869443 0.1948867
```

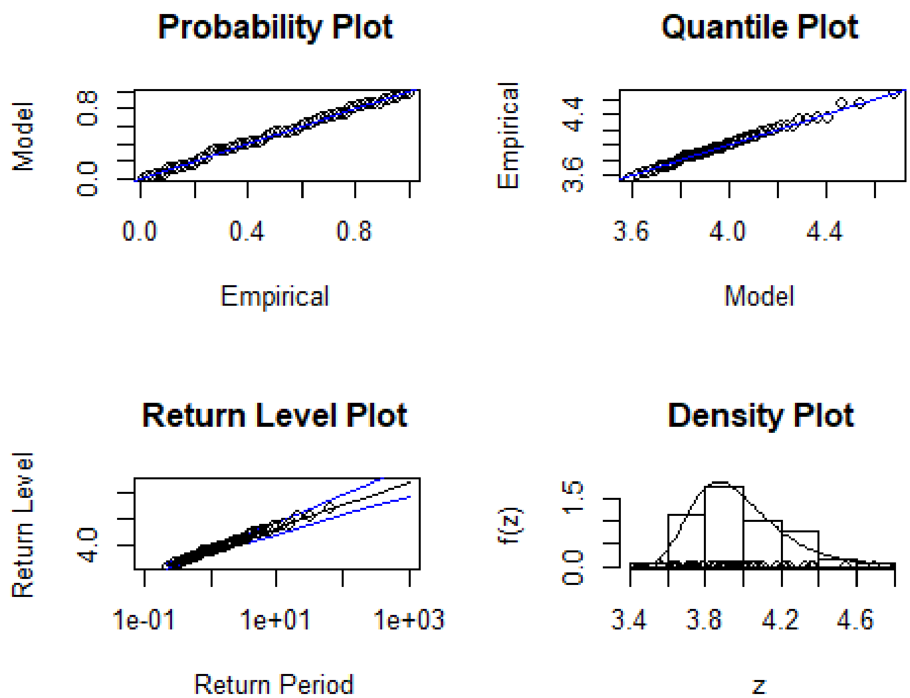
```
## [47,] 3.869443 0.1948867
## [48,] 3.869443 0.1948867
## [49,] 3.869443 0.1948867
## [50,] 3.869443 0.1948867
## [51,] 3.869443 0.1948867
## [52,] 3.869443 0.1948867
## [53,] 3.869443 0.1948867
## [54,] 3.869443 0.1948867
## [55,] 3.869443 0.1948867
## [56,] 3.869443 0.1948867
## [57,] 3.869443 0.1948867
## [58,] 3.869443 0.1948867
## [59,] 3.869443 0.1948867
## [60,] 3.869443 0.1948867
## [61,] 3.869443 0.1948867
## [62,] 3.869443 0.1948867
## [63,] 3.869443 0.1948867
## [64,] 3.869443 0.1948867
## [65,] 3.869443 0.1948867
##
## attr(,"class")
## [1] "gum.fit"

# b = 3.8694426  a = 0.1948867
```

Probability ,Quantile and return level plots interpretation

```
gev.diag(fitmle_gev_portpirie2)
```



```
gum.diag((fitmle_gumbel_portpirie2))
```



In view of the QQplots and the Probability plots, it seems that both Gumbel and Weibull modelisation are a good fit. But we can notice that the confidence interval for the return level plot is better for the Gumbel fit than for the Weibull model.Moreover with Gumbel we have less parameters, then with that parcimony principle, I prefer to choose Gumbel modelisation.

Moreover the Gumbel return Level plot is a linear function which is consistent with what we saw in the lesson.

Maximum domain of attraction

The maximum domain of attraction is Gumbel with gamma =0

Estimation of a return level corresponding to a return period 100 and 1000 or endpoint

```
xi=0
mu=3.8694924
beta= 0.1948879


#model Gumbel
#return level plot
```

```
fExtremes::qgev(1/100,xi=0,mu=mu,beta=beta,lower.tail=FALSE)

## [1] 4.766006
## attr(,"control")
##   xi       mu      beta lower.tail
##    0 3.869492 0.1948879      FALSE

#maximum sea level for the next 100 years : 4.77 meters

fExtremes::qgev(1/1000,xi=xi,mu=mu,beta=beta,lower.tail=FALSE)

## [1] 5.215633
## attr(,"control")
##   xi       mu      beta lower.tail
##    0 3.869492 0.1948879      FALSE

#maximum sea level for the next 1000 years : 5.22 meters
```

Small conclusion

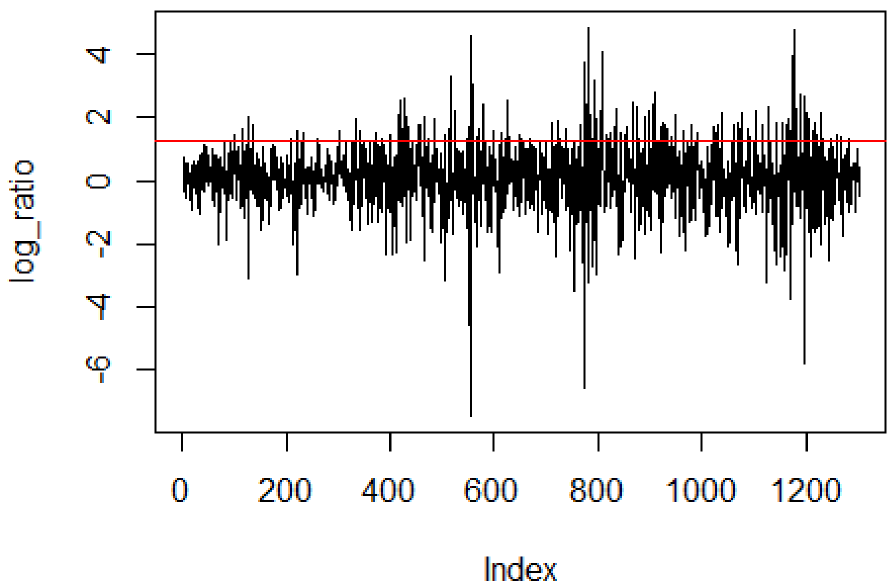
With a Gumbel model we could estimate that the maximum sea level of Port Pirie in a 100 and 1000 years are respectively around 4.77 and 5.22 meters,with a good confidence interval.It means that once every 100 years (respectively 1000 years) the PortPirie Sea Level annual maximum is above 4.77 meters (respectively 5.22 meters). Those results are consistent with the data since the maximum level in the past data is 4.69 meters.

Study of DowJones DataSet

Descriptive Statistics

In the DowJones dataset there are daily closing prices of the Dow Jones Index over the period 1996 to 2000

```
data(dowjones)
?dowjones


log_ratio=100*diff(log(dowjones$Index))[-1]

plot(log_ratio,type="l")
abline(h=1.25,col='red')#we will see later that the red line corresponds to a fixed optimal threshold of 1.25 "log-$"
```




```r
summary(log_ratio)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -7.45500 -0.47790  0.05287  0.06634  0.67700  4.86100
```

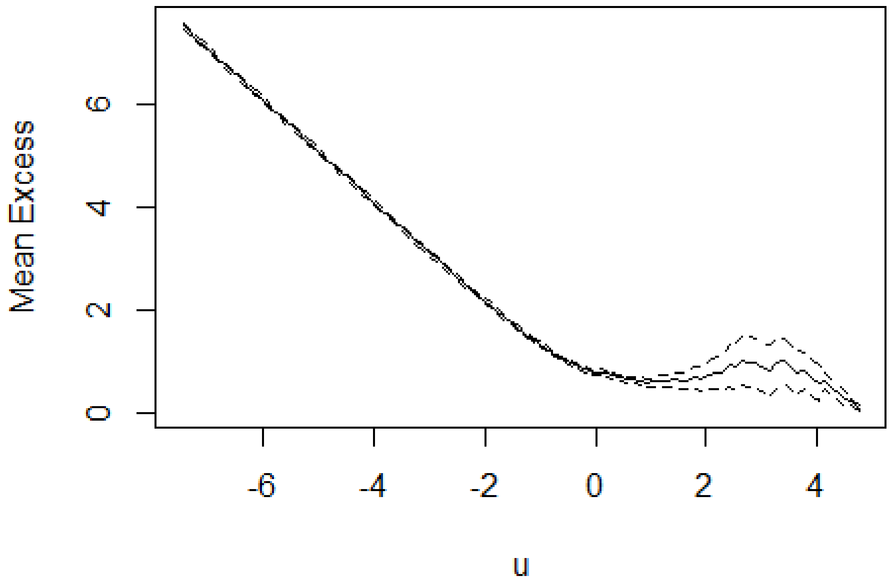
The log ratios are between -7.5 and 4.9 with a mean and a median around 0.06

Here we have daily data let us try a GPD model

Choice of the thresholds

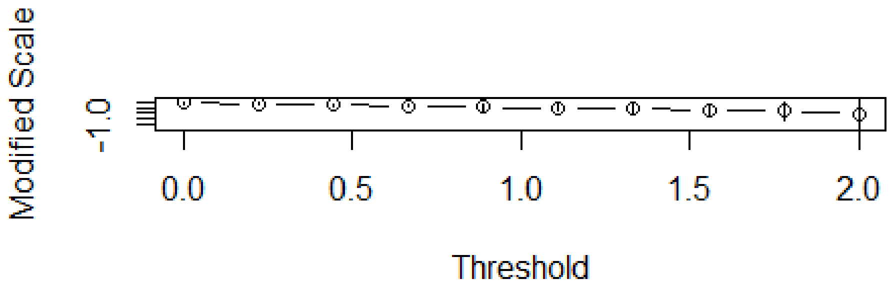
POT Method: We need the threshold to be close enough to the maximum (Pickand's theorem) but not too close (variance bias balance).

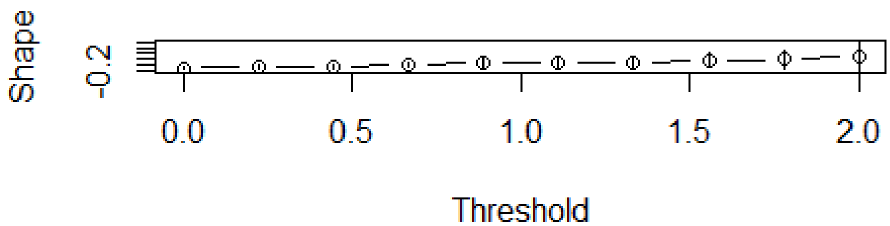
```r
findthresh(log_ratio,c(1,10,100))

## [1] 4.809520 2.709723 1.397623

mrl.plot(log_ratio)
```



The Mean Excess has to be linear with u at levels of u for which the Generalized Pareto model is appropriate. Based on the previous graphics, we can see that the linear tendancy ends for u between 0 and 2. Let us look for the confidence interval for σ (scale) and γ (shape) :

```
?gpd.fitrange
gpd.fitrange(log_ratio,0,2)
```





Thanks to that graphical method, it seems that 1.25 is a good threshold (we have a constant shape around 0 and the length of confidence intervals are low)

```
threshold=1.25
log_ratio_excess=log_ratio[log_ratio>threshold]
log_ratio_excess
```

```
##   [1] 1.426725 1.662865 2.000709 1.750577 1.323736 1.610578 1.509264
##   [8] 1.298631 1.574890 1.276691 1.979252 1.564724 1.263899 1.498175
##  [15] 1.349871 1.465522 2.074764 1.397623 2.569914 2.604867 1.348571
##  [22] 2.006131 1.704060 1.770474 1.774591 2.002369 1.280099 1.312468
##  [29] 1.940522 1.402783 1.459687 1.294050 3.320608 2.238415 1.739514
##  [36] 4.600841 3.073843 1.646851 1.310130 2.399422 1.462048 1.573205
##  [43] 1.530362 1.315576 1.276387 2.513889 1.397074 1.470134 1.343216
##  [50] 1.325365 1.282023 1.866790 1.876847 1.337908 1.628343 1.763002
##  [57] 1.617202 3.753540 4.860535 2.335575 1.904021 3.205062 1.973961
##  [64] 2.144608 1.282328 4.064700 1.404655 1.459207 1.318650 1.495400
##  [71] 2.317165 1.525078 1.457880 1.727170 1.372466 2.479741 2.379336
##  [78] 1.308964 2.008075 2.252084 2.043684 2.798425 1.266903 1.738730
##  [85] 1.861366 1.762358 1.615276 1.263642 1.368107 2.069903 1.610815
##  [92] 1.268681 1.777040 1.492308 1.427078 1.693370 1.778142 2.146270
##  [99] 1.789086 1.820096 1.661830 2.166288 1.628992 1.581920 1.392526
## [106] 2.213819 1.788511 2.364810 1.860344 1.866997 1.774151 1.970440
## [113] 1.552234 3.211285 4.809520 2.104159 2.302963 2.709723 2.649965
## [120] 1.732132 1.571667 1.985640 1.575805 1.704088 1.852773 2.188660
## [127] 1.329531 1.319780 1.463366 1.372262 1.336179
```

```r
length(log_ratio_excess)/length(log_ratio)

## [1] 0.1006144

#10 % of the data are above the excess
```

Estimation of the parameters

First we can use the probably weighted moment method

```r
gpdfit3=gpdFit(log_ratio,u=threshold,type="pwm")
gpdfit3

##
## Title:
##   GPD Parameter Estimation
##
## Call:
##  gpdFit(x = log_ratio, u = threshold, type = "pwm")
##
## Estimation Method:
##   gpd pwm
##
## Estimated Parameters:
##         xi       beta
## 0.07405389 0.57480697
##
## Description
##   Thu Mar 22 22:50:15 2018 by user: oussa

#gamma=0.07405389 sigma=0.57480697
```

But as we have a lot of data here we can prefere the MLE method

```r
gpdfit1=gpd.fit(log_ratio,threshold)

## $threshold
## [1] 1.25
##
## $nexc
## [1] 131
##
## $conv
## [1] 0
##
## $nllh
## [1] 68.02454
##
## $mle
## [1] 0.56689410 0.08680442
##
## $rate
```

```
## [1] 0.1006144
##
## $se
## [1] 0.07234195 0.09332016
```

```
gpdfit2=gpdFit(log_ratio,u=threshold,type="mle")
gpdfit2
```

```
##
## Title:
##   GPD Parameter Estimation
##
## Call:
##  gpdFit(x = log_ratio, u = threshold, type = "mle")
##
## Estimation Method:
##   gpd mle
##
## Estimated Parameters:
##          xi       beta
## 0.08680205 0.56688864
##
## Description
##   Thu Mar 22 22:50:16 2018 by user: oussa
```

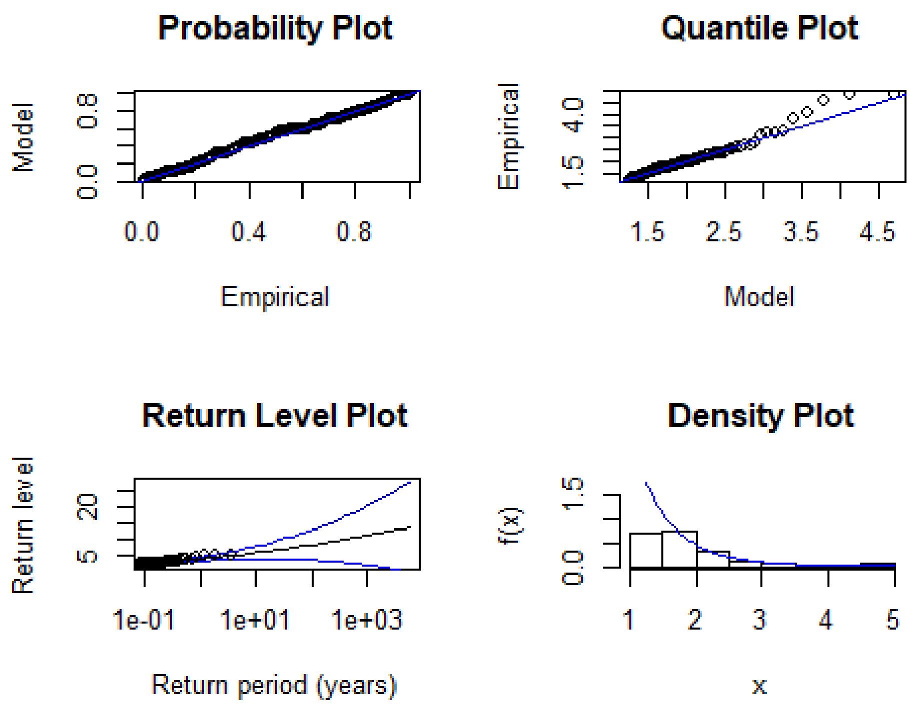
Maximum domain of attraction

Since gamma is approximatively equal to 0.087 > 0 this corresponds to a Frechet domain of attraction in the GEV theory. Then the distribution of excesses has no upper limit

```
gamma=0.08680205
sigma=0.56688864
```

Probability, Quantile and Return level plots and interpretation

```
gpd.diag(gpdfit1)
```



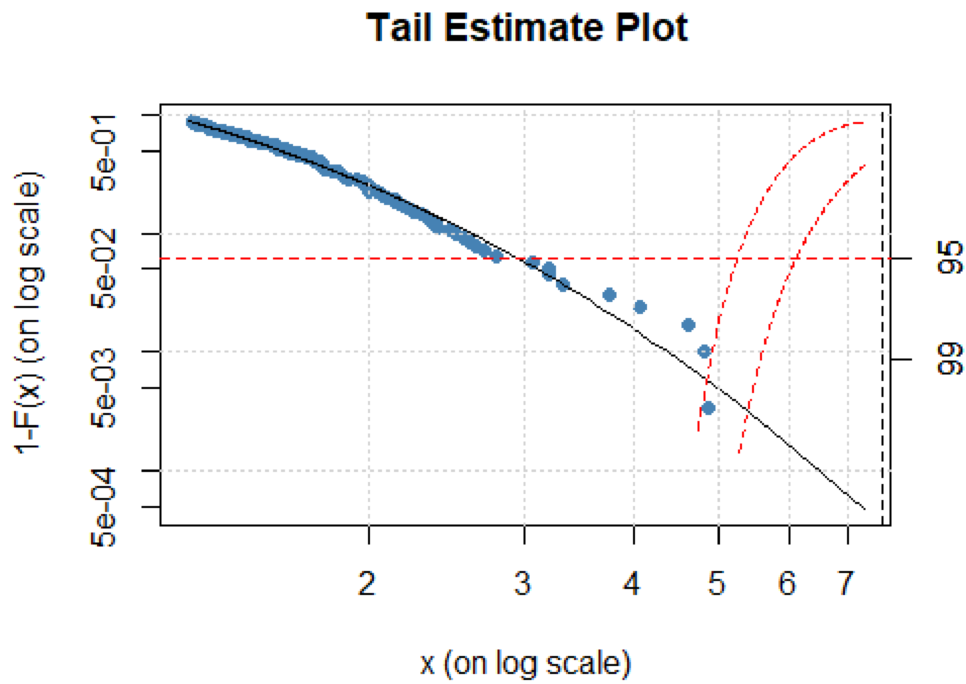
The probability plot is very good

The confidence interval in the return level plot seems to be good until T=1e1. Yet we have some few points out of the Quantile Plot (in the tail of the distribution) Moreover since gamma > 0 we can see convexity in the Return Level Plot

Estimation of a return level corresponding to a return period 100 and 1000 or endpoint

```
#1304 data from 1996 to 2000 means :
1304/5

## [1] 260.8

#261 datas a year

?gpdTailPlot
tail.mle=gpdTailPlot(gpdfit2)
gpdRiskMeasures(gpdfit2,p=1-1/(100*1304/5))# 7.65

##                 p quantile shortfall
## beta 0.9999617 7.653428  8.882864

gpdRiskMeasures(gpdfit2,p=1-1/(1000*1304/5))# 10.51

##                 p quantile shortfall
## beta 0.9999962 10.51504  12.01648
```

```
gpdQPlot(tail.mle,p=1-1/(100*1304/5),ci=0.95)

## Lower CI Estimate Upper CI
## 5.247828 7.653428 7.290802

gpdQPlot(tail.mle,p=1-1/(1000*1304/5),ci=0.95)
```




```
##   Lower CI  Estimate   Upper CI
##   6.114939 10.515038   7.290802
```

Small conclusion

The maximum Dow Jones log ratio excess in the data base was : 4.86-1.25=3.61 To conclude, thanks to the GPD theory, we could modelise with a fixed threshold of 1.25 “log-$” that the log ratio of dow jones , and that the return level for the excess of respectively 100 and 1000 days are 7.65 and 10.51.This corresponds to the quantile of a 0.9999617 probability (respectively 0.9999962) It means that once every 100 (respectively 1000) days, the log ratio excess above the threshold is higher than 7.65 (respectively 10.51). And when that excess is above those maximums, the mean of those excess is (shortfall) around 8.88 (respectively 12.01) But the confidence interval for that prediction (in view of the return level plot) is large..

Study of Glass DataSet

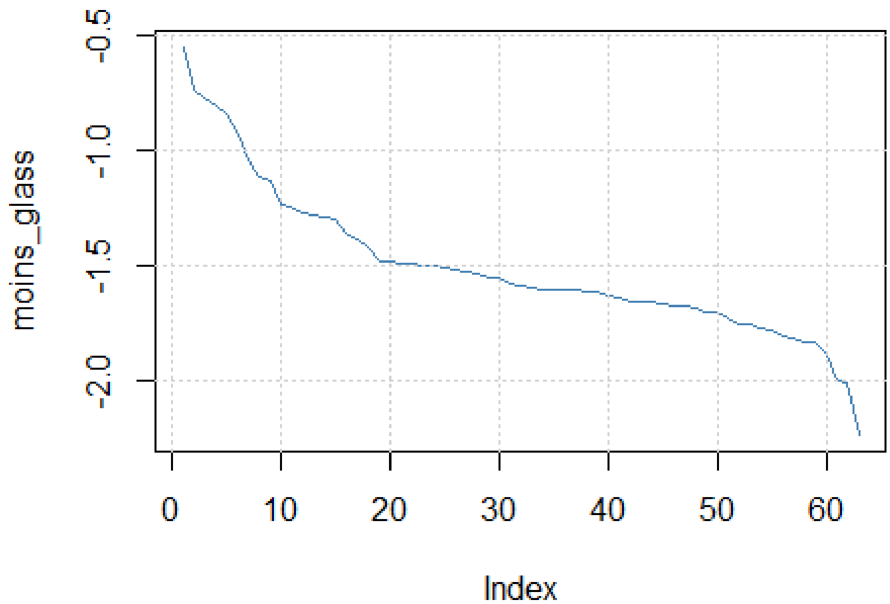
The glass dataset contains breaking strength of 63 glass fibres of length 1.5 centimeters

```
?glass
data(glass)
summary(glass)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.550   1.375   1.590   1.507   1.685   2.240
```

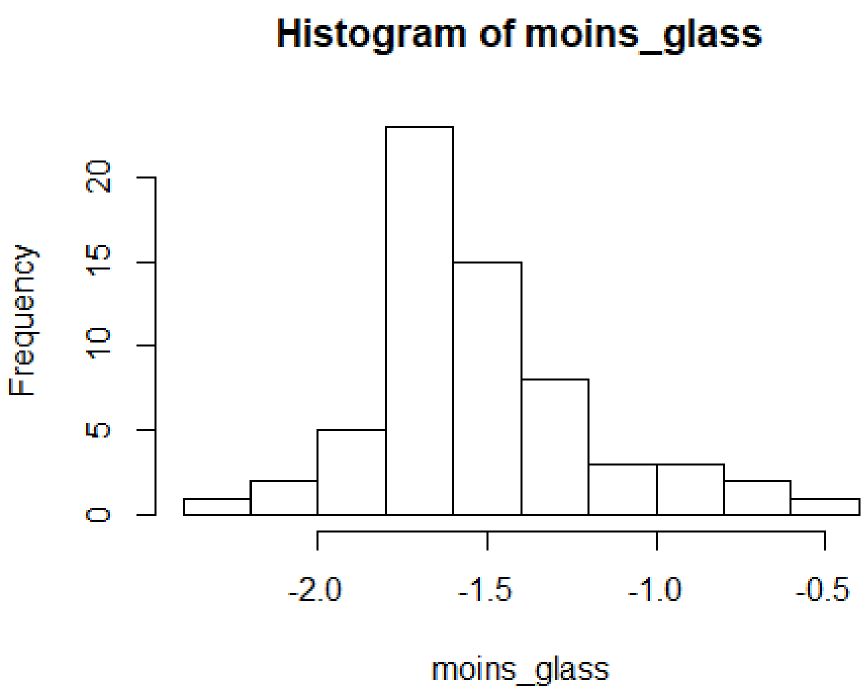
Here we want to study the minimum strength to break the glass.Let X1,X2,..Xn be the strengths, then the minimum of the serie is - (the maximum of (-X1,…,-Xn)

Descriptive Statistics

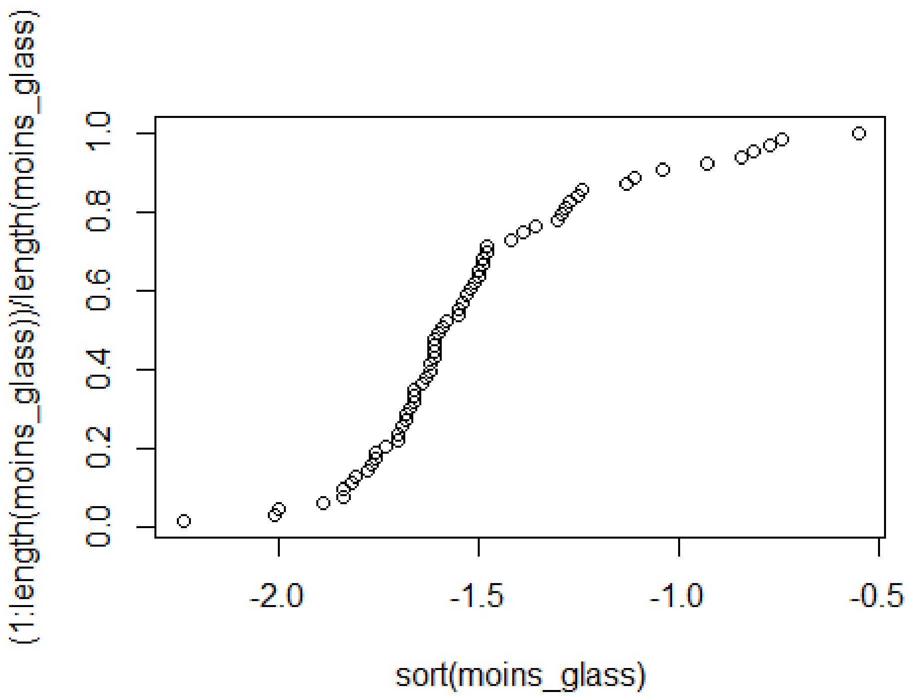
```
moins_glass=(-1)*(glass)





nrow(moins_glass)

## NULL

summary(moins_glass)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.240  -1.685  -1.590  -1.507  -1.375  -0.550

plot(moins_glass,type="l",col="steelblue")

grid()
```




```r
hist(moins_glass)
```




```r
plot(sort(moins_glass),(1:length(moins_glass))
     /length(moins_glass))
```




```
#most of the maximums are between -1.5 and -2
```

Choice of modelisation

Let us try a GEV model for the glass breaking strengths dataset

Choice of the blocs

We can consider that the data glass is the minimums of a lot of trials then we can consider the blocks = 1.

Estimation of the parameters

```
fitmle_gev_glass1=gevFit(moins_glass,type="mle",block=1)

fitmle_gev_glass1

##
## Title:
##  GEV Parameter Estimation
##
## Call:
##  gevFit(x = moins_glass, block = 1, type = "mle")
##
## Estimation Type:
##   gev mle
##
```

```
## Estimated Parameters:
##          xi          mu        beta
## -0.08435813 -1.64162098  0.27287140
##
## Description
##   Thu Mar 22 22:50:22 2018

#         gamma     b        a
#-0.08435813 -1.64162098  0.27287140

#Weibull attraction domain

fitmle_gev_glass2=gev.fit(moins_glass)

## $conv
## [1] 0
##
## $nllh
## [1] 14.28529
##
## $mle
## [1] -1.6416229  0.2728611 -0.0843667
##
## $se
## [1] 0.03752530 0.02552437 0.06994369

fitmle_gev_glass2

## $trans
## [1] FALSE
##
## $model
## $model[[1]]
## NULL
##
## $model[[2]]
## NULL
##
## $model[[3]]
## NULL
##
##
## $link
## [1] "c(identity, identity, identity)"
##
## $conv
## [1] 0
##
## $nllh
## [1] 14.28529
##
```

```
## $data
##  [1] -0.55 -0.74 -0.77 -0.81 -0.84 -0.93 -1.04 -1.11 -1.13 -1.24 -1.25
## [12] -1.27 -1.28 -1.29 -1.30 -1.36 -1.39 -1.42 -1.48 -1.48 -1.49 -1.49
## [23] -1.50 -1.50 -1.51 -1.52 -1.53 -1.54 -1.55 -1.55 -1.58 -1.59 -1.60
## [34] -1.61 -1.61 -1.61 -1.61 -1.62 -1.62 -1.63 -1.64 -1.66 -1.66 -1.66
## [45] -1.67 -1.68 -1.68 -1.69 -1.70 -1.70 -1.73 -1.76 -1.76 -1.77 -1.78
## [56] -1.81 -1.82 -1.84 -1.84 -1.89 -2.00 -2.01 -2.24
##
## $mle
## [1] -1.6416229  0.2728611 -0.0843667
##
## $cov
##               [,1]          [,2]          [,3]
## [1,]  0.0014081484  0.0002142133 -0.0007947444
## [2,]  0.0002142133  0.0006514936 -0.0004412636
## [3,] -0.0007947444 -0.0004412636  0.0048921195
##
## $se
## [1] 0.03752530 0.02552437 0.06994369
##
## $vals
##            [,1]      [,2]       [,3]
##  [1,] -1.641623 0.2728611 -0.0843667
##  [2,] -1.641623 0.2728611 -0.0843667
##  [3,] -1.641623 0.2728611 -0.0843667
##  [4,] -1.641623 0.2728611 -0.0843667
##  [5,] -1.641623 0.2728611 -0.0843667
##  [6,] -1.641623 0.2728611 -0.0843667
##  [7,] -1.641623 0.2728611 -0.0843667
##  [8,] -1.641623 0.2728611 -0.0843667
##  [9,] -1.641623 0.2728611 -0.0843667
## [10,] -1.641623 0.2728611 -0.0843667
## [11,] -1.641623 0.2728611 -0.0843667
## [12,] -1.641623 0.2728611 -0.0843667
## [13,] -1.641623 0.2728611 -0.0843667
## [14,] -1.641623 0.2728611 -0.0843667
## [15,] -1.641623 0.2728611 -0.0843667
## [16,] -1.641623 0.2728611 -0.0843667
## [17,] -1.641623 0.2728611 -0.0843667
## [18,] -1.641623 0.2728611 -0.0843667
## [19,] -1.641623 0.2728611 -0.0843667
## [20,] -1.641623 0.2728611 -0.0843667
## [21,] -1.641623 0.2728611 -0.0843667
## [22,] -1.641623 0.2728611 -0.0843667
## [23,] -1.641623 0.2728611 -0.0843667
## [24,] -1.641623 0.2728611 -0.0843667
## [25,] -1.641623 0.2728611 -0.0843667
## [26,] -1.641623 0.2728611 -0.0843667
## [27,] -1.641623 0.2728611 -0.0843667
## [28,] -1.641623 0.2728611 -0.0843667
```

```
## [29,] -1.641623 0.2728611 -0.0843667
## [30,] -1.641623 0.2728611 -0.0843667
## [31,] -1.641623 0.2728611 -0.0843667
## [32,] -1.641623 0.2728611 -0.0843667
## [33,] -1.641623 0.2728611 -0.0843667
## [34,] -1.641623 0.2728611 -0.0843667
## [35,] -1.641623 0.2728611 -0.0843667
## [36,] -1.641623 0.2728611 -0.0843667
## [37,] -1.641623 0.2728611 -0.0843667
## [38,] -1.641623 0.2728611 -0.0843667
## [39,] -1.641623 0.2728611 -0.0843667
## [40,] -1.641623 0.2728611 -0.0843667
## [41,] -1.641623 0.2728611 -0.0843667
## [42,] -1.641623 0.2728611 -0.0843667
## [43,] -1.641623 0.2728611 -0.0843667
## [44,] -1.641623 0.2728611 -0.0843667
## [45,] -1.641623 0.2728611 -0.0843667
## [46,] -1.641623 0.2728611 -0.0843667
## [47,] -1.641623 0.2728611 -0.0843667
## [48,] -1.641623 0.2728611 -0.0843667
## [49,] -1.641623 0.2728611 -0.0843667
## [50,] -1.641623 0.2728611 -0.0843667
## [51,] -1.641623 0.2728611 -0.0843667
## [52,] -1.641623 0.2728611 -0.0843667
## [53,] -1.641623 0.2728611 -0.0843667
## [54,] -1.641623 0.2728611 -0.0843667
## [55,] -1.641623 0.2728611 -0.0843667
## [56,] -1.641623 0.2728611 -0.0843667
## [57,] -1.641623 0.2728611 -0.0843667
## [58,] -1.641623 0.2728611 -0.0843667
## [59,] -1.641623 0.2728611 -0.0843667
## [60,] -1.641623 0.2728611 -0.0843667
## [61,] -1.641623 0.2728611 -0.0843667
## [62,] -1.641623 0.2728611 -0.0843667
## [63,] -1.641623 0.2728611 -0.0843667
##
## attr(,"class")
## [1] "gev.fit"

#b=-1.6416229  a=0.2728611 gamma=-0.0843667




#95% confidence intervalfor gamma:

IC=c(-0.0843667-0.06994369*2/sqrt(63),-0.0843667+0.06994369*2/sqrt(63))
IC

## [1] -0.10199085 -0.06674255
```

```
#0 does not seem to be part of the confidence interval
```

Let us try the probability weigthed moments method (better if we do not have enough data)

```
fitpwrgev_glass = gevFit(moins_glass,type="pwm")
fitpwrgev_glass

##
## Title:
##  GEV Parameter Estimation
##
## Call:
##  gevFit(x = moins_glass, type = "pwm")
##
## Estimation Type:
##   gev pwm
##
## Estimated Parameters:
##          xi          mu        beta
##  0.04624223 -1.65667089  0.23975505
##
## Description
##   Thu Mar 22 22:50:22 2018

#results are clothes to those with Maximum Likelihood Estimator
```

Note : X is positive (it is a force in Newton), -X<0

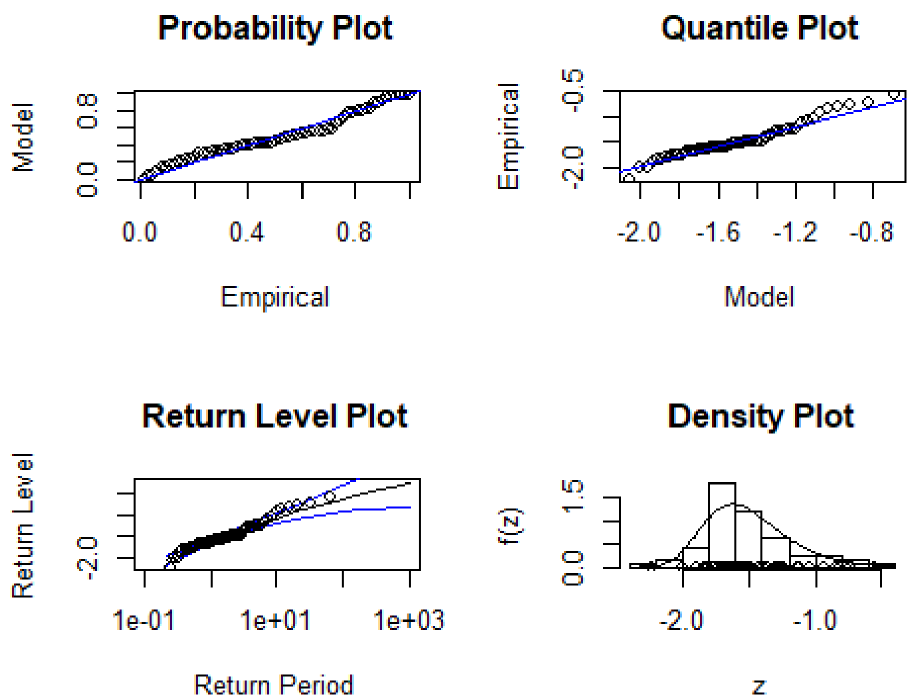
‘’’

```
#fitmle_gumbel_glass1=gumbelFit(moins_glass,type=c("mle"))

#fitmle_gumbel_glass1#this one does not work here, I stay with the Weibull model
```

‘’’

Probability, Quantile and Return level plots and interpretation

```
gev.diag(fitmle_gev_glass2)
```



The probability plot is good

The confidence interval in the return level plot seems to be good until T=1e2. Yet we have some few points out of the Quantile Plot (in the tail of the distribution) Moreover since gamma < 0 we can see concavity in the Return Level Plot

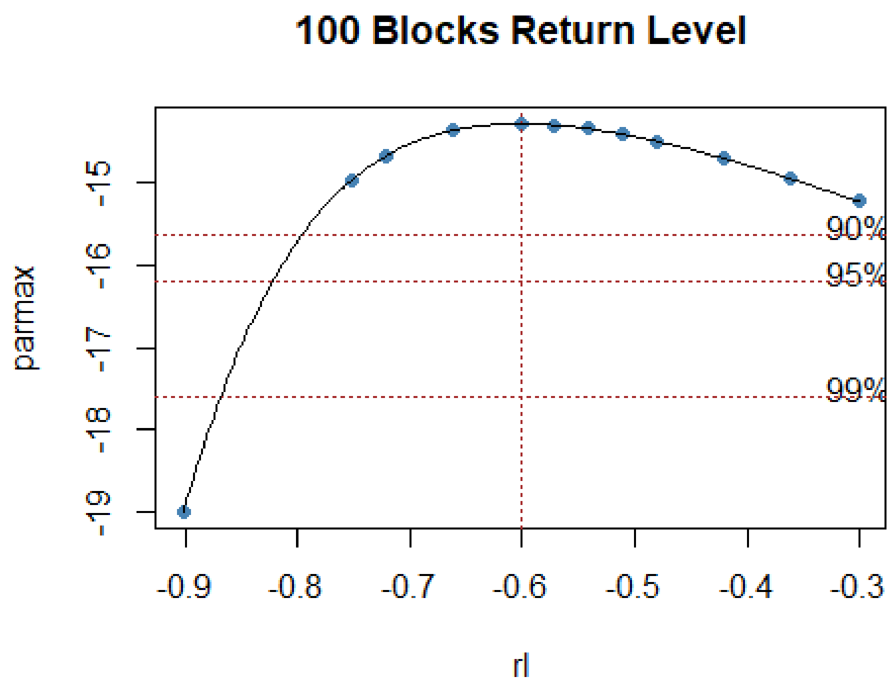
Maximum domain of attraction

The maximum domain of attraction Weibull with gamma<0

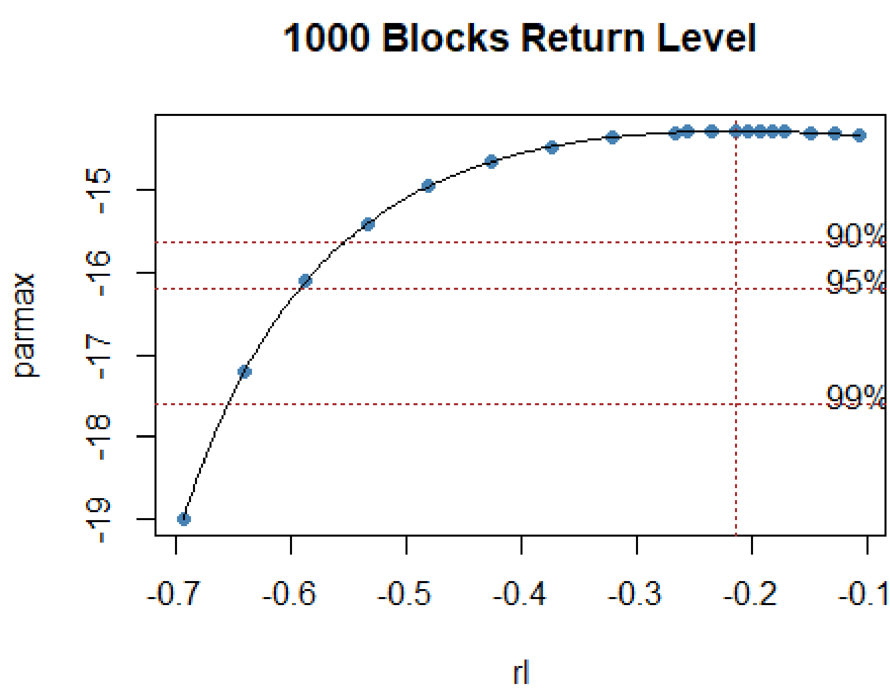
Estimation of a return level corresponding to a return period 100 and 1000 or endpoint

```
#model Weibull
#Estimated Parameters:

#          xi           mu         beta

#-0.08435813 -1.64162098  0.27287140

gamma=-0.0843667
b=-1.6416229
a=0.2728611
xf=b-a/gamma
xf#here xf is >0 so it does not make sense since X=-strenght is always
negative
```

```
## [1] 1.592605

xi=gamma
mu=b
beta=a




#Estimated Parameters:


#b-1.6416229  a0.2728611 gamma-0.0843667




#return level


fExtremes::qgev(1/100,xi=xi,mu=mu,beta=beta,lower.tail=FALSE)

## [1] -0.6013172
## attr(,"control")
##          xi        mu      beta lower.tail
##  -0.0843667 -1.641623 0.2728611      FALSE

#minimum strength to break the glass after 100 test  : 0.60 N

fExtremes::qgev(1/1000,xi=xi,mu=mu,beta=beta,lower.tail=FALSE)

## [1] -0.213275
## attr(,"control")
##          xi        mu      beta lower.tail
##  -0.0843667 -1.641623 0.2728611      FALSE

#minimum strength to break the glass after 1000 test  : 0.21 N

gevrlevelPlot(fitmle_gev_glass1,kBlocks = 100,  ci = c(0.90, 0.95, 0.99))
```




```
##                                min          v        max kBlocks
## GEV Return Level -0.7961365 -0.6012568 -0.3006284     100

gevrlevelPlot(fitmle_gev_glass1,kBlocks = 1000,  ci = c(0.90, 0.95, 0.99))
```



```
##                          min         v        max kBlocks
## GEV Return Level -0.5514317 -0.213181 -0.1065905    1000
```

Small conclusion

With a Weibull model we could estimate that the minimum strength once every 100 and 1000 breaking tests are respectively around 0.6 and 0.2 Newtons.Those results are consistent with the data since the minimum strength to break the glass was 0.55 Newtons.Yet the confidence interval is not good(for exemple for the 100 trials, the minimum strength confidence interval is [0,3;0,8]).Perhaps the problem here is that we do not have enough data (only 63 observations).