VALEUR EXTREMES

SEFFANE Asmaa

2023-07-31

INTRODUCTION:

Extreme value theory is concerned with the statistical properties of the extreme events related to a random variable and the understanding and applications of their probability distributions.

In this project, I will study two dataset:

1. Fort Collins (Colorado) Daily precipitation amounts (inches) from a single rain gauge in Fort Collins, Colorado. The variable of interest is the daily precipitation amount (inches): Prec.

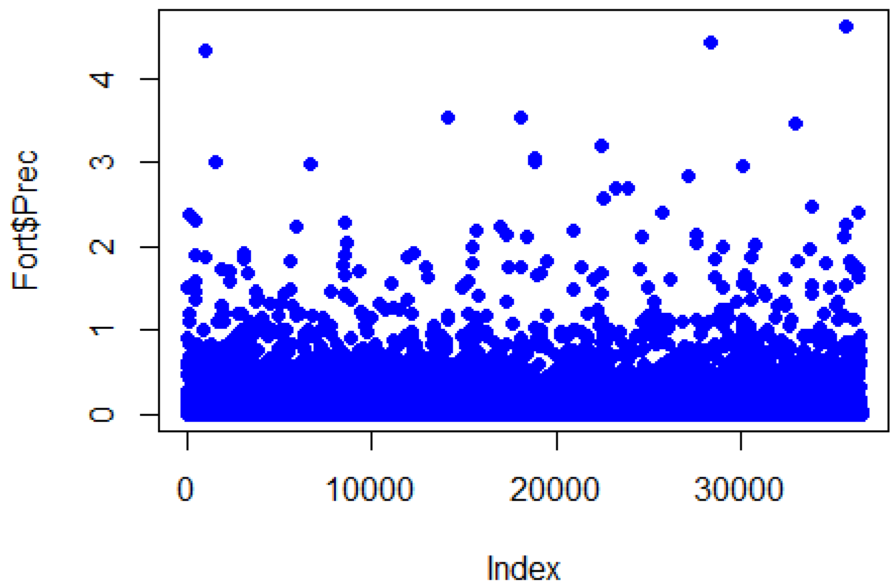
2. This dataset shows the annual maximum sea-levels recorded at Port Pirie, a location just north of Adelaide, South Australia, over the period 1923–1987. From such data it may be necessary to obtain an estimate of the maximum sea-level that is likely to occur in the region over the next 100 or 1000 years. It seems reasonable to assume that the pattern of variation has stayed constant over the observation period, so we model the data as independent observations from the GEV distribution.

Dataset Fort Collins Colorado:

Descriptives statistics:

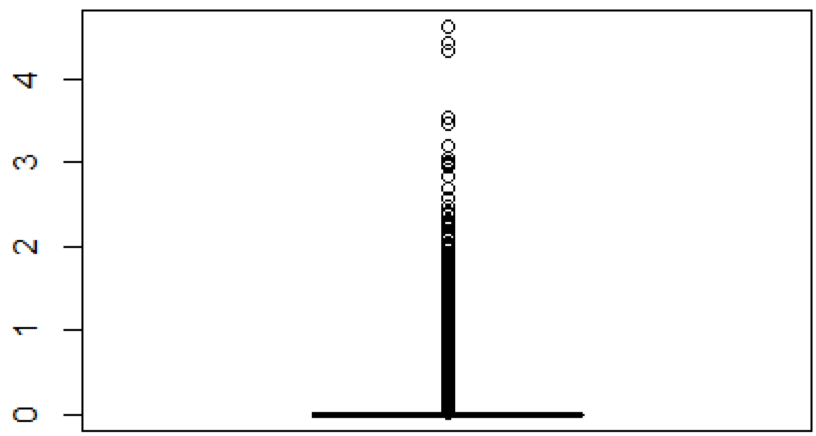
I start by requiring all the packages I will need and the dataset:

```
require(evd)

require(evir)

require(ismev)

require(fExtremes)

require(extRemes)



data(Fort)
str(Fort$Prec)

##  num [1:36524] 0 0 0 0 0 0 0 0 0 0 ...

length(Fort$Prec)

## [1] 36524

summary(Fort)

##       obs             tobs           month             day       
##  Min.   :    1   Min.   :  1.0   Min.   : 1.000   Min.   : 1.00  
##  1st Qu.: 9132   1st Qu.: 92.0   1st Qu.: 4.000   1st Qu.: 8.00  
##  Median :18263   Median :183.0   Median : 7.000   Median :16.00  
##  Mean   :18263   Mean   :183.1   Mean   : 6.523   Mean   :15.73  
##  3rd Qu.:27393   3rd Qu.:274.0   3rd Qu.:10.000   3rd Qu.:23.00  
##  Max.   :36524   Max.   :366.0   Max.   :12.000   Max.   :31.00  
##       year           Prec        
##  Min.   :1900   Min.   :0.00000  
##  1st Qu.:1925   1st Qu.:0.00000  
##  Median :1950   Median :0.00000  
##  Mean   :1950   Mean   :0.04181  
##  3rd Qu.:1974   3rd Qu.:0.00000  
##  Max.   :1999   Max.   :4.63000  

plot(Fort$Prec, col = "blue",pch=16)
```

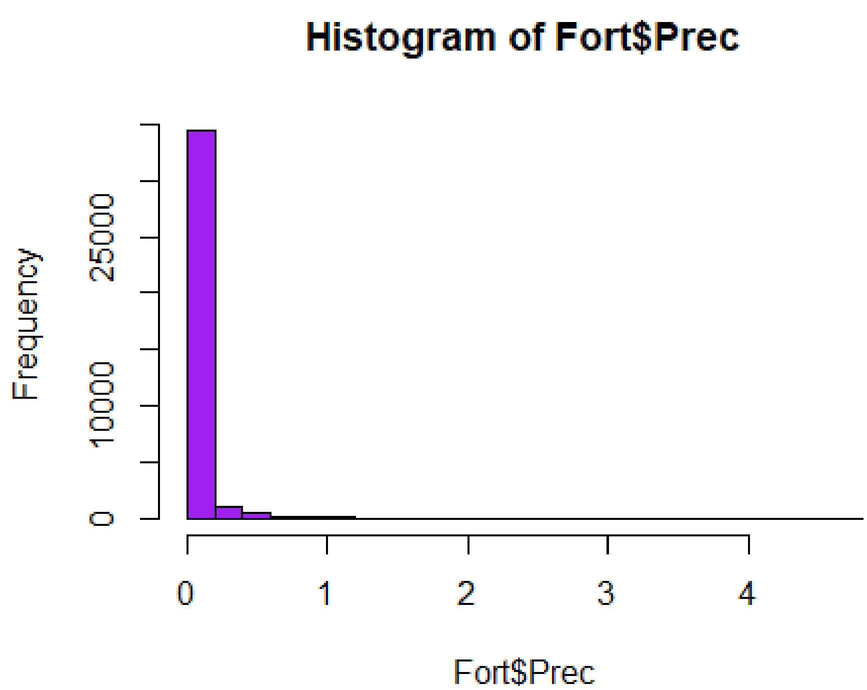


As we can see there is some values bigger than the mean 0.04181 with a big different, we can consider them as a extreme values.

```
boxplot(Fort$Prec)
```

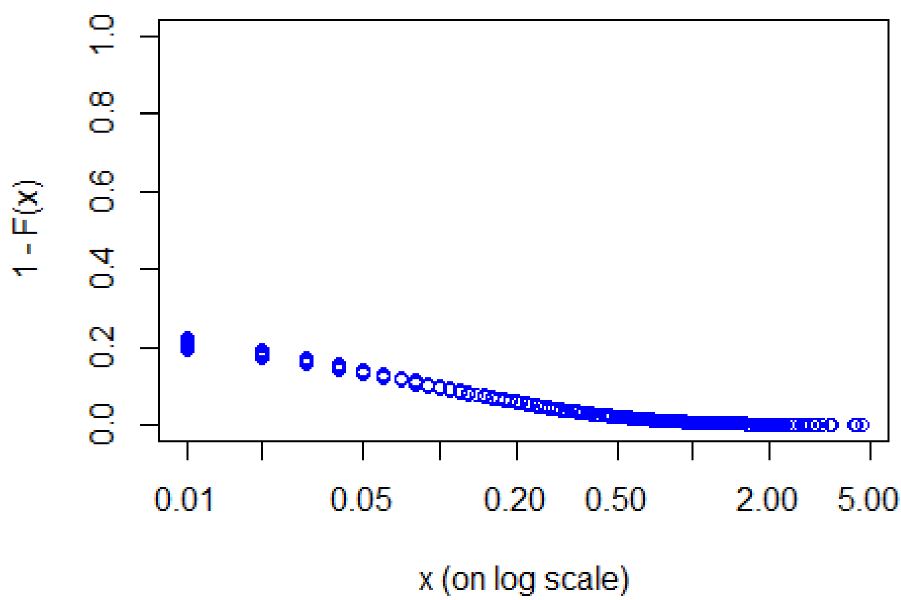



```r
hist(Fort$Prec, col = "purple")
```




```r
emplot(Fort$Prec, col = "blue")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 28366 x values <= 0
omitted
## from logarithmic plot
```

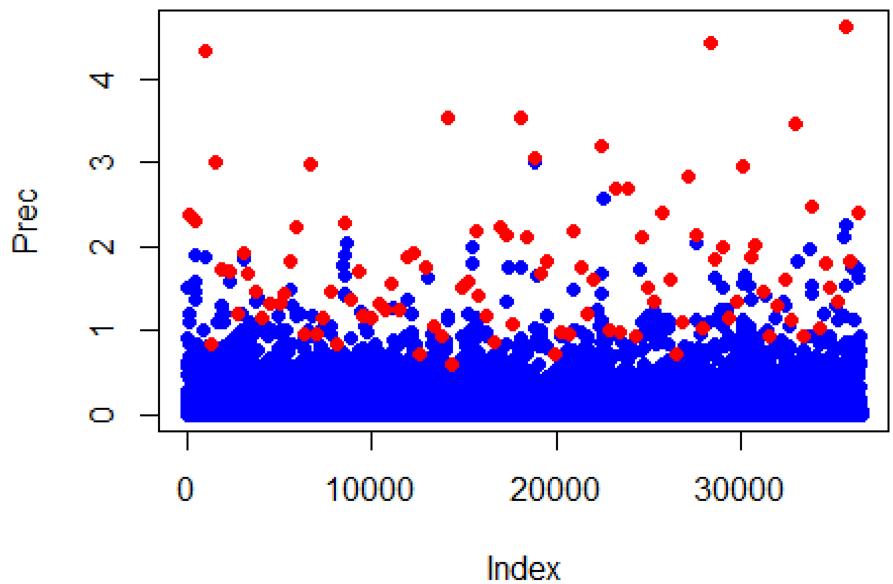


As it’s required, I must use both approaches (GEV and GPD) and compare them.

First approach: The Generalized Extreme Value distribution: GEV

The block maxima approach:

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

##       obs tobs month day year Prec
## 119   119  119     4  29 1900 2.39
## 507   507  142     5  22 1901 2.32
## 994   994  264     9  21 1902 4.34
## 1398 1398  303    10  30 1903 0.85
## 1583 1583  123     5   2 1904 3.02
## 1917 1917   91     4   1 1905 1.74

length(as.numeric(rownames(bmFort)))

## [1] 100

plot(Fort$Prec,ylab = "Prec",col = "blue",pch = 16)
points(as.numeric(rownames(bmFort)),bmFort$Prec, col="red", pch=16)
```



We have the maxima on red.

Maximum Likelihood Estimators:

```
FitFortMle = fevd(bmFort$Prec, type="GEV", method="MLE")
FitFortMle

##
## fevd(x = bmFort$Prec, type = "GEV", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  104.9645
##
##
##  Estimated parameters:
##  location     scale     shape
## 1.3466597 0.5328046 0.1736264
##
##  Standard Error Estimates:
##   location      scale      shape
## 0.06168793 0.04878843 0.09195458
##
##  Estimated parameter covariance matrix.
##                location          scale          shape
## location  0.003805401  0.0017067043 -0.0020838301
```

```
## scale     0.001706704  0.0023803113 -0.0008692638
## shape    -0.002083830 -0.0008692638  0.0084556445
##
##  AIC = 215.9291
##
##  BIC = 223.7446
```

```
ci(FitFortMle,type="parameter")
```

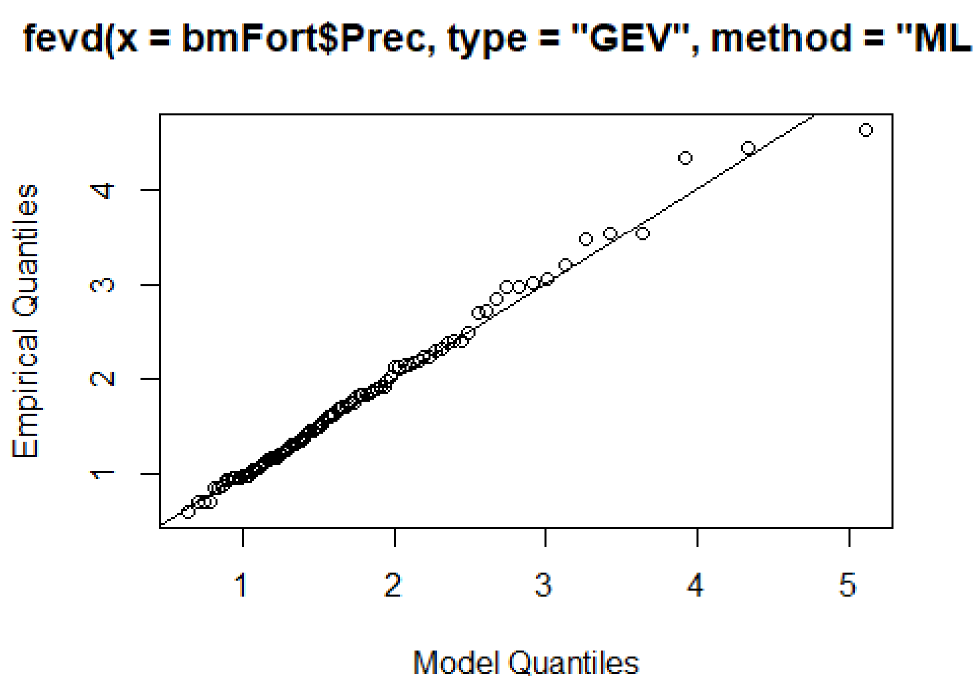
```
## fevd(x = bmFort$Prec, type = "GEV", method = "MLE")
##
## [1] "Normal Approx."
##
##          95% lower CI  Estimate 95% upper CI
## location  1.225753534 1.3466597    1.4675658
## scale     0.437181061 0.5328046    0.6284282
## shape    -0.006601292 0.1736264    0.3538540
```

In this approach, lambda equal to 0,17< ½ which mean maximum likelihood estimators are regular.

The interval of the lambda contains 0 what make us apply the Gamble approach.

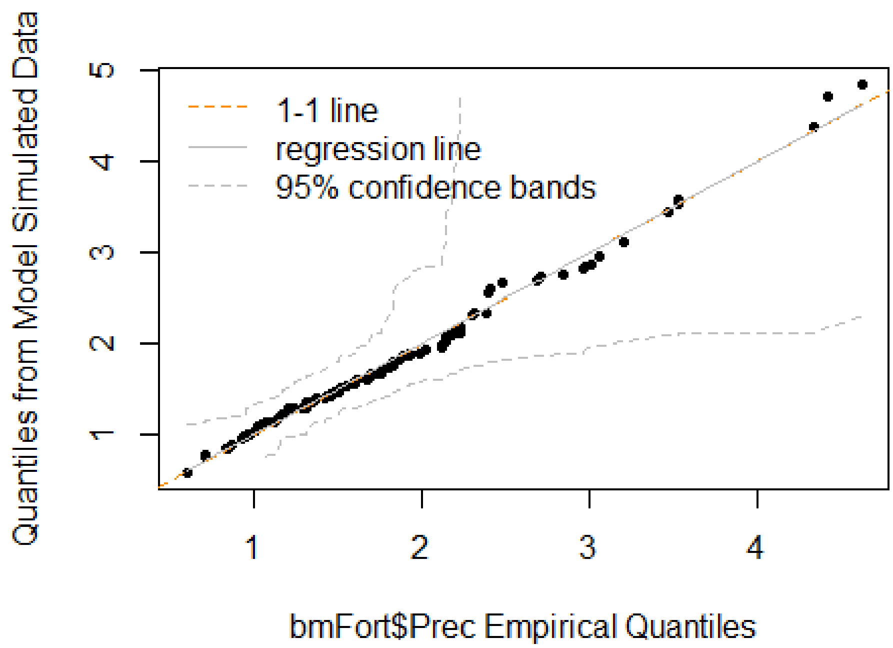
Let's take a look on the plots:

```
plot(FitFortMle,type="qq")
```



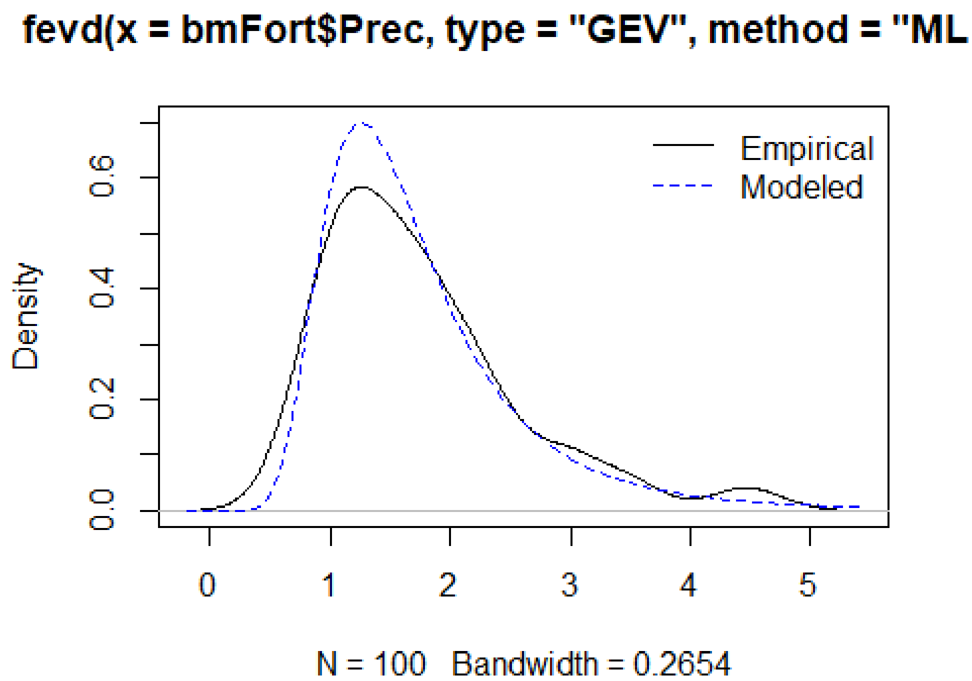
In this plot, we have just two values in the end of the plot who is not linear, but it’s not a big deal we can say that this plot is good.

```
plot(FitFortMle,type="qq2")
```



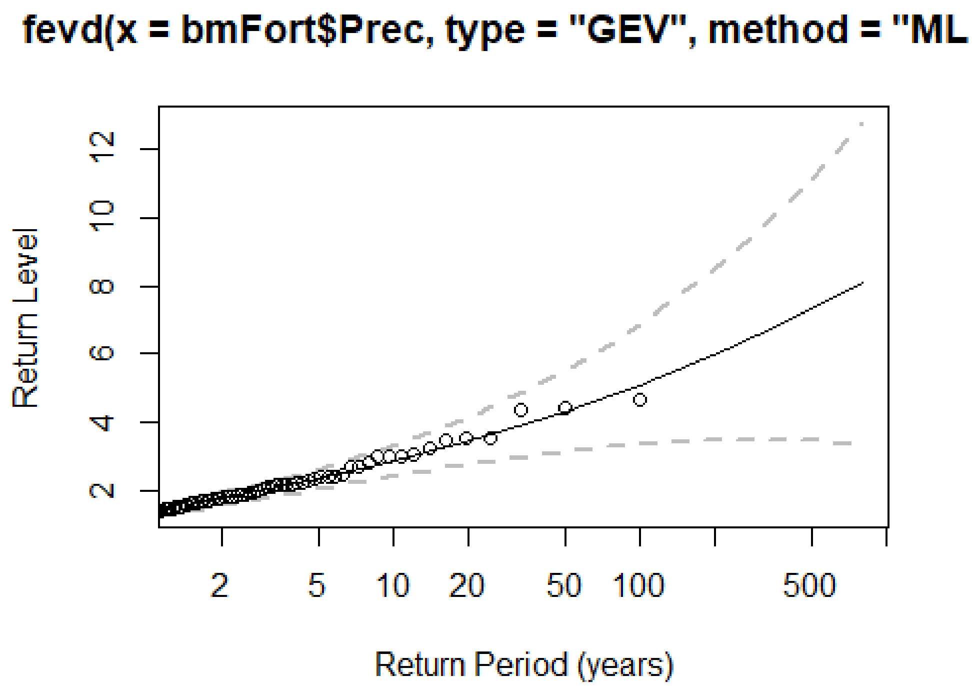
In this plot, all the values are in the confidence bands what make it perfect.

```
plot(FitFortMle,type="density")
```



Even if the modeled density over estimate the empirical density in some point and under estimate it in other point, there is not a lot of differences we can say that this plot is good

```
plot(FitFortMle,type="rl")
```



In this plot too, we have all the values in the bands what make it a perfect.

Let's see The Gumbel model:

```
FitFortGumbelMle=fevd(bmFort$Prec,type="Gumbel",method="MLE")
FitFortGumbelMle

##
## fevd(x = bmFort$Prec, type = "Gumbel", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  107.1278
##
##
##  Estimated parameters:
##  location     scale
## 1.3988265 0.5784564
##
##  Standard Error Estimates:
##   location      scale
## 0.06059952 0.04736171
##
##  Estimated parameter covariance matrix.
```

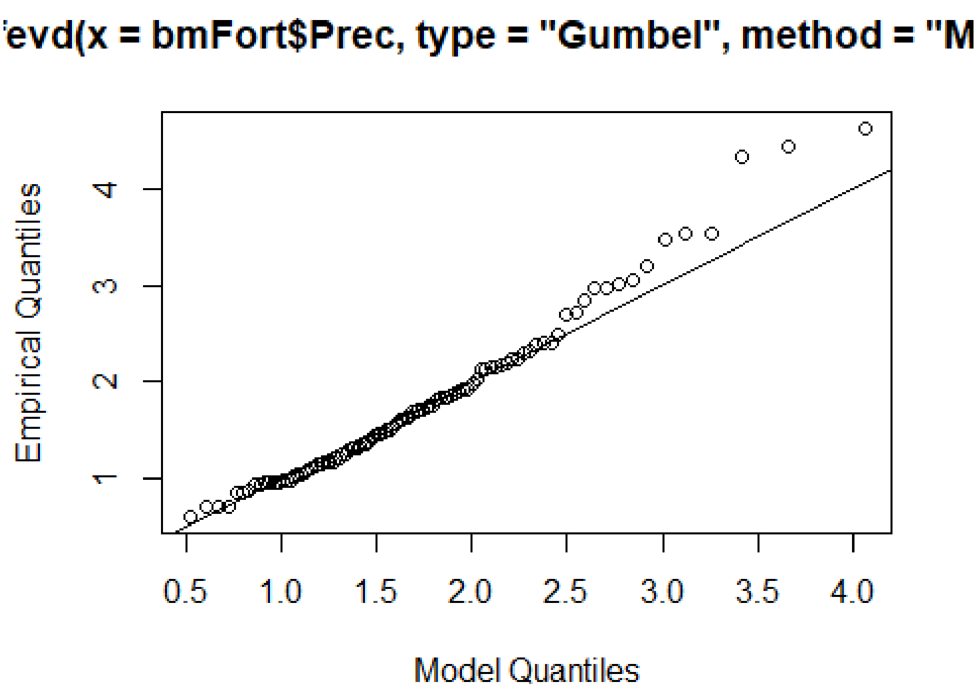
```
##                 location        scale
## location 0.0036723022 0.0008553822
## scale    0.0008553822 0.0022431316
##
##  AIC = 218.2555
##
##  BIC = 223.4659
```

```
ci(FitFortGumbelMle)
```

```
## fevd(x = bmFort$Prec, type = "Gumbel", method = "MLE")
##
## [1] "Normal Approx."
##
## [1] "100-year return level: 4.06"
##
## [1] "95% Confidence Interval: (3.5837, 4.5359)"
```

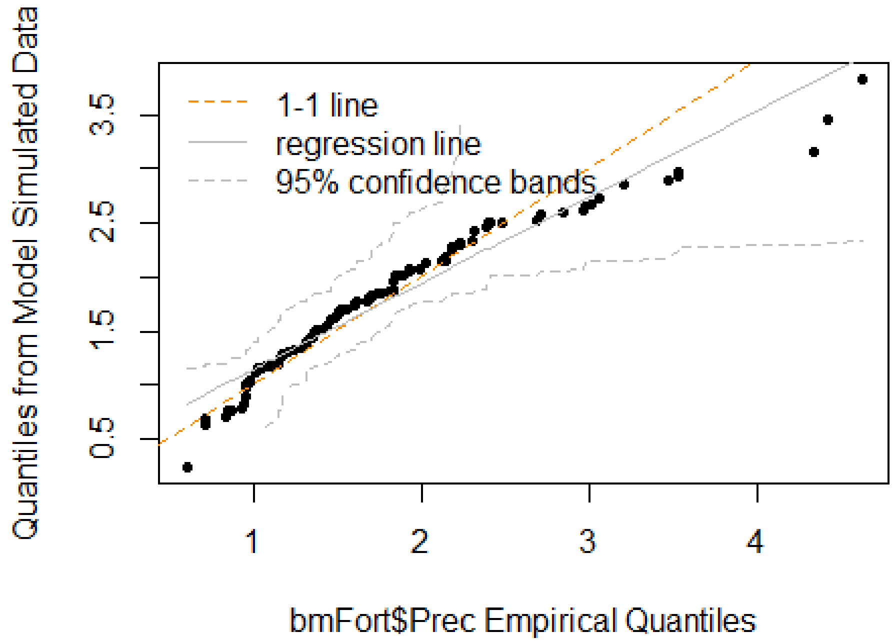
This approach has less parameters than the previous one, let's take a look on the plots:

```
plot(FitFortGumbelMle,type="qq2")
```



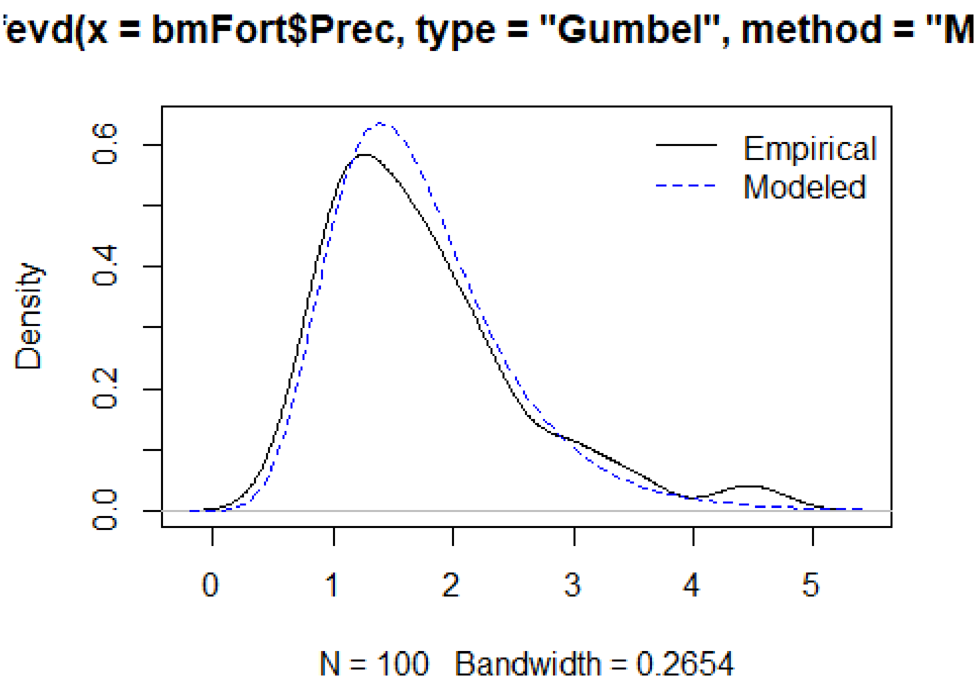
There is some values in the extreme which are not linear, 14 values.

```
plot(FitFortGumbelMle,type="qq2")
```



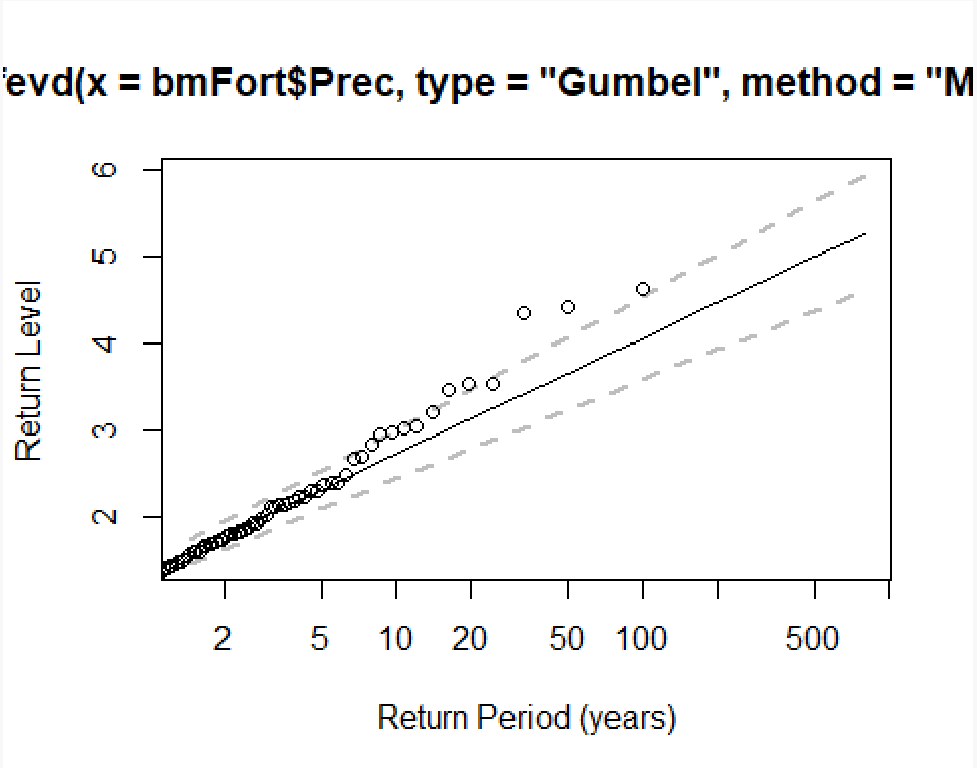
All the values are in the confidence bands what makes it perfect.

```
plot(FitFortGumbelMle,type="density")
```



In this Density plot, we can see that the modeled density overestimate the empirical density much more that the plot of the previous approach, what make this plot less good.

```r
plot(FitFortGumbelMle,type="rl")
```



Same notice here, there is some values that are out of the bands.

COMPARAISON:

```r
lr.test(FitFortMle,FitFortGumbelMle)

##
##  Likelihood-ratio Test
##
## data:  bmFort$PrecbmFort$Prec
## Likelihood-ratio = 4.3264, chi-square critical value = 3.8415, alpha =
## 0.0500, Degrees of Freedom = 1.0000, p-value = 0.03752
## alternative hypothesis: greater
```

The p-value of the test 0.03752 is less than 0.05, we can conclude that the full model provides a significantly better fit

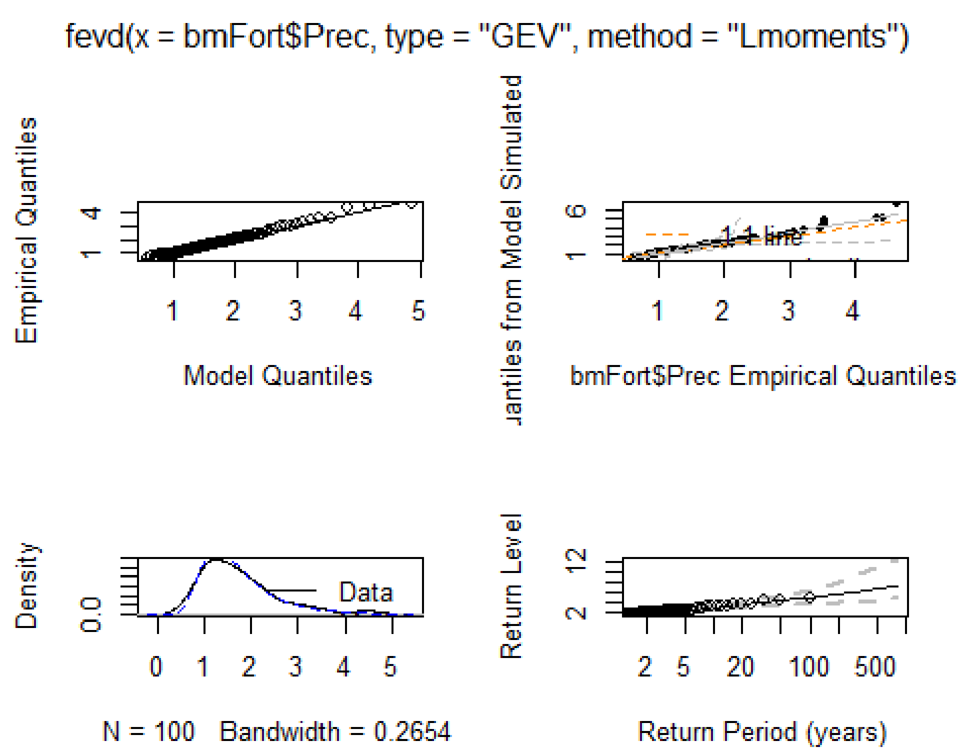
Probability Weighted Moment: PWM

```r
FitFortMoments=fevd(bmFort$Prec,type="GEV",method="Lmoments")
FitFortMoments

## fevd(x = bmFort$Prec, type = "GEV", method = "Lmoments")
## [1] "GEV  Fitted to  bmFort$Prec  using L-moments estimation."
##  location     scale     shape
## 1.3535282 0.5564335 0.1307426
```

```
ci(FitFortMoments,type="parameter")

## fevd(x = bmFort$Prec, type = "GEV", method = "Lmoments")
##
## [1] "Parametric Bootstrap"
## 502  iterations
##
##                    2.5%  Estimate     97.5%
## location  1.23404785 1.3535282 1.4972104
## scale     0.46453880 0.5564335 0.6560530
## shape    -0.03505252 0.1307426 0.2963131
```

Lambda equal to 0.1307, it's a normal value, and we can notice that the interval of lambda contains 0, what make us think of applying the Gumbel approach. Let's look on the plots:

```
plot(FitFortMoments)
```



All the plots look perfect, the values are in the confident shapes and the estimated density go well with the empirical density.

Return Level for Fort GEV MLE and PWM:

```
return.level(FitFortMle, return.period = c(100, 1000))

## fevd(x = bmFort$Prec, type = "GEV", method = "MLE")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
```

```
return.period)
##
##  GEV model fitted to  bmFort$Prec
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  100-year level 1000-year level
##        5.098635        8.459051
```

```
return.level(FitFortGumbelMle, return.period = c(100, 1000))
```

```
## fevd(x = bmFort$Prec, type = "Gumbel", method = "MLE")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  Gumbel model fitted to  bmFort$Prec
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  100-year level 1000-year level
##        4.059812        5.394372
```

```
return.level(FitFortMoments, return.period = c( 100, 1000))
```

```
## fevd(x = bmFort$Prec, type = "GEV", method = "Lmoments")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  GEV model fitted to  bmFort$Prec
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  100-year level 1000-year level
##        4.863562        7.597733
```

```
summary(Fort$Prec)
```

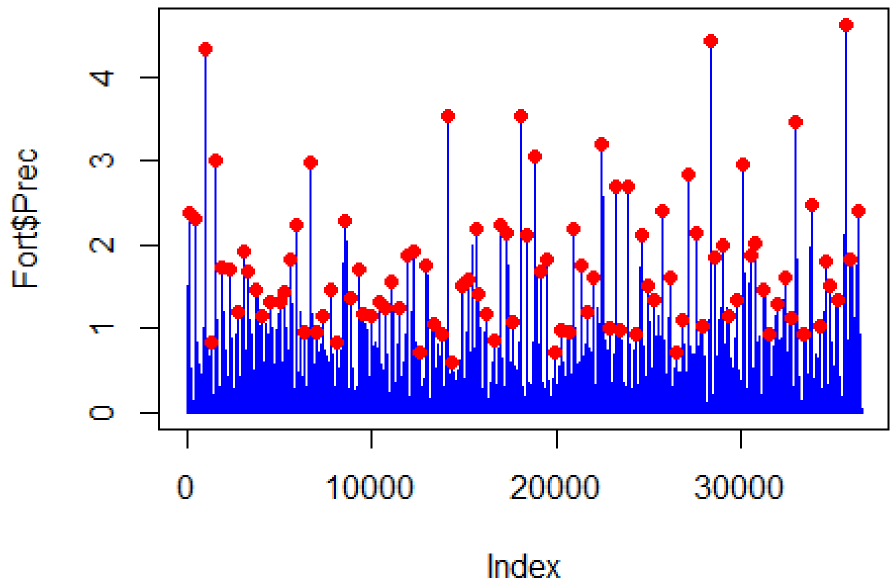
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.04181 0.00000 4.63000
```

Summary return level for a return period of 100 years:

Data set	Method of estimation	Fréchet $\gamma > 0$	Gumbel $\gamma = 0$	Weibull $\gamma < 0$ (Endpoint)
Fort colin colorado	MLE	5.093685	4.059812	X
	MOMENT	4.863562	X	X

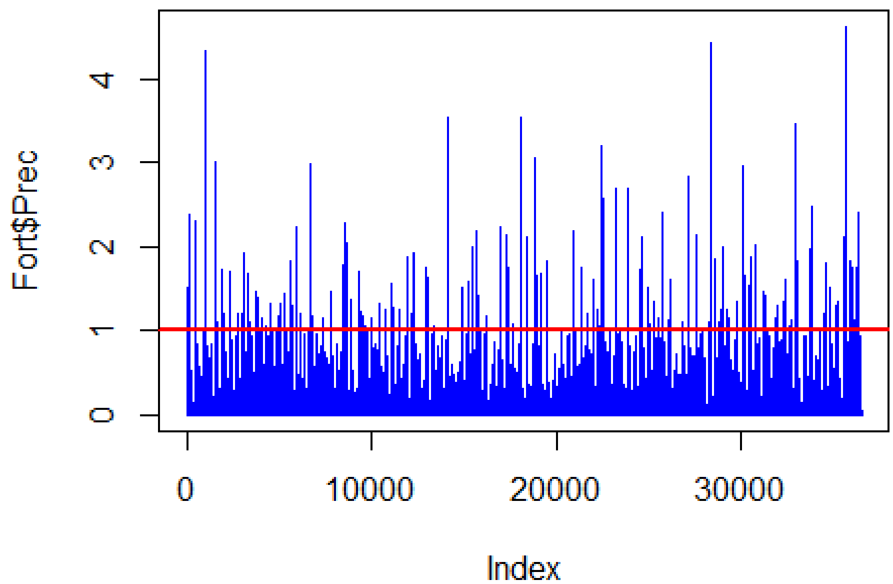
Second approach: The Generalized Pareto Distribution GPD:

```
plot(Fort$Prec,type="h",col="blue")
points(as.numeric(rownames(bmFort)),bmFort$Prec, col="red", pch=16)
```

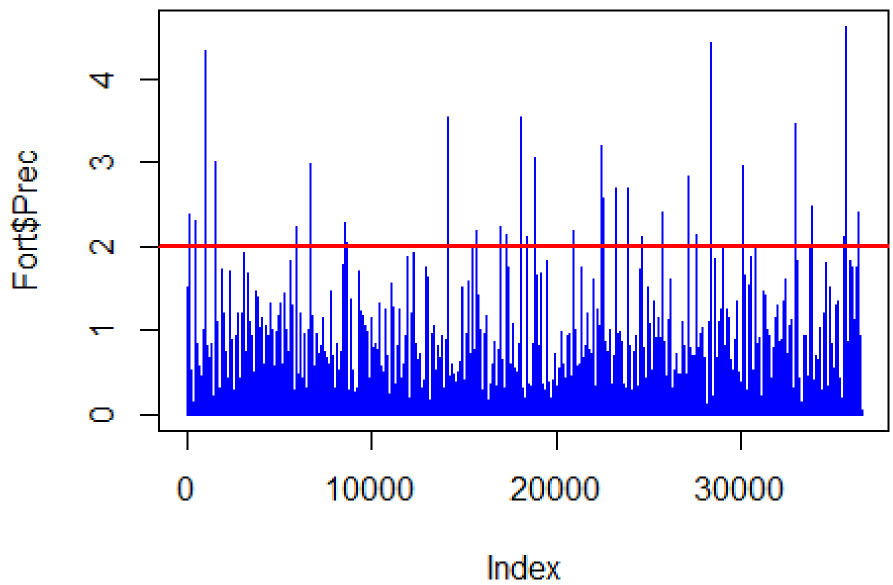


I will choose 1 then 2 as a threshold and see how much value I have:

```
plot(Fort$Prec,type="h",col="blue")
abline(h=1, col="red",lwd=2)
```

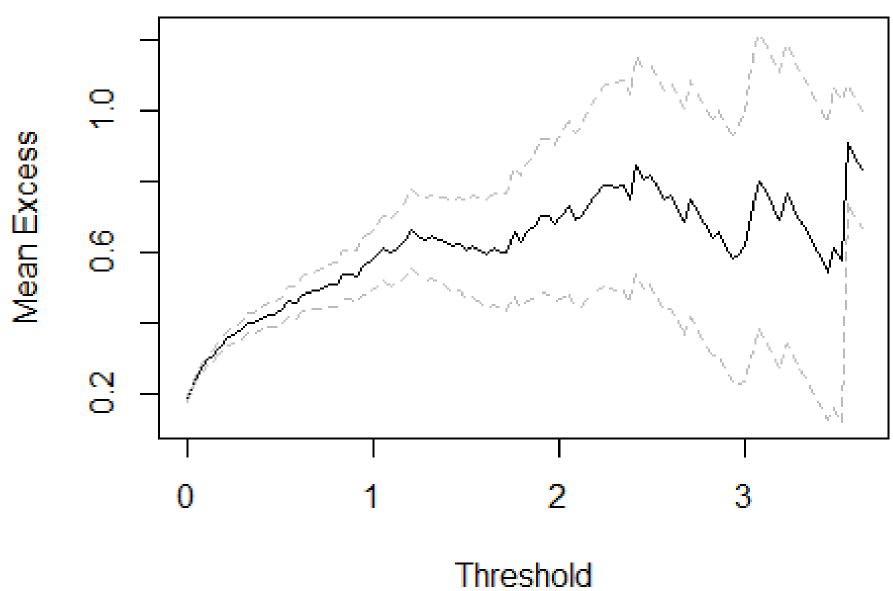



```
length(Fort$Prec > 2)

## [1] 36524

plot(Fort$Prec,type="h",col="blue")
abline(h=2, col="red",lwd=2)
```



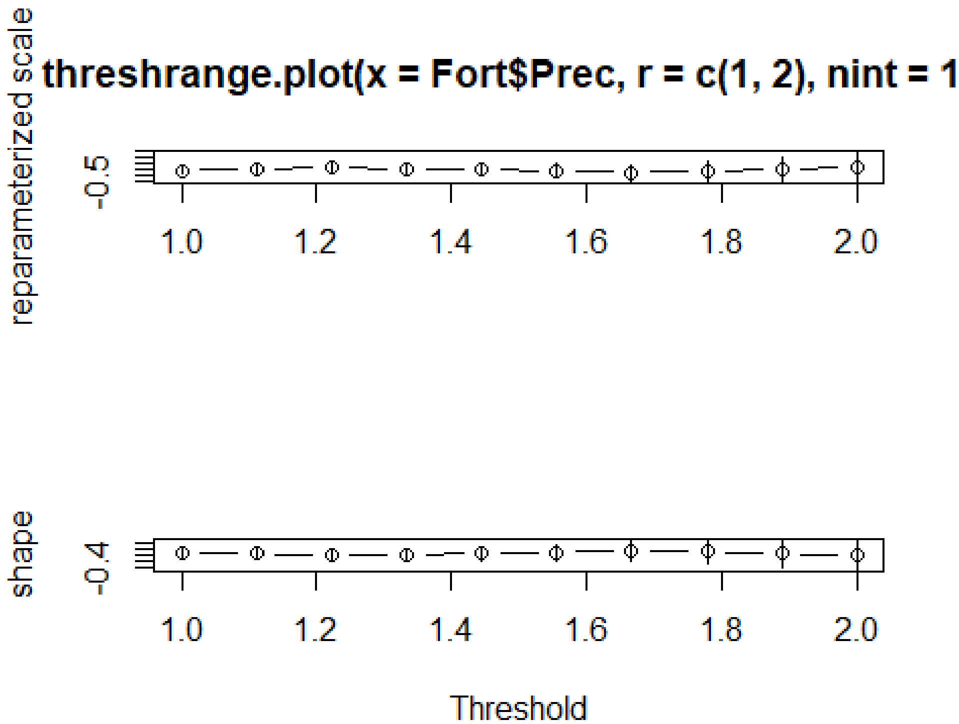
Mean Residual Life Plot:

```
mrlplot(Fort$Prec)
```



We can consider that befor1 we have a linear graph, but after 1, the graph is not linear anymore, so I will choose between 1 and 2:

```
threshrange.plot(Fort$Prec, r = c(1, 2), nint=10)
```



The variance gives us two value 1.2 or 1.5, I choose 1.2.

Maximum Likelihood Estimators: MLE

```
FitFortgpdMle
=fevd(Fort$Prec,threshold=1.2,type="GP",method="MLE",time.units="days")
FitFortgpdMle

##
## fevd(x = Fort$Prec, threshold = 1.2, type = "GP", method = "MLE",
##     time.units = "days")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  77.90375
##
##
##  Estimated parameters:
##       scale       shape
## 0.645148623 0.002790766
##
##  Standard Error Estimates:
```

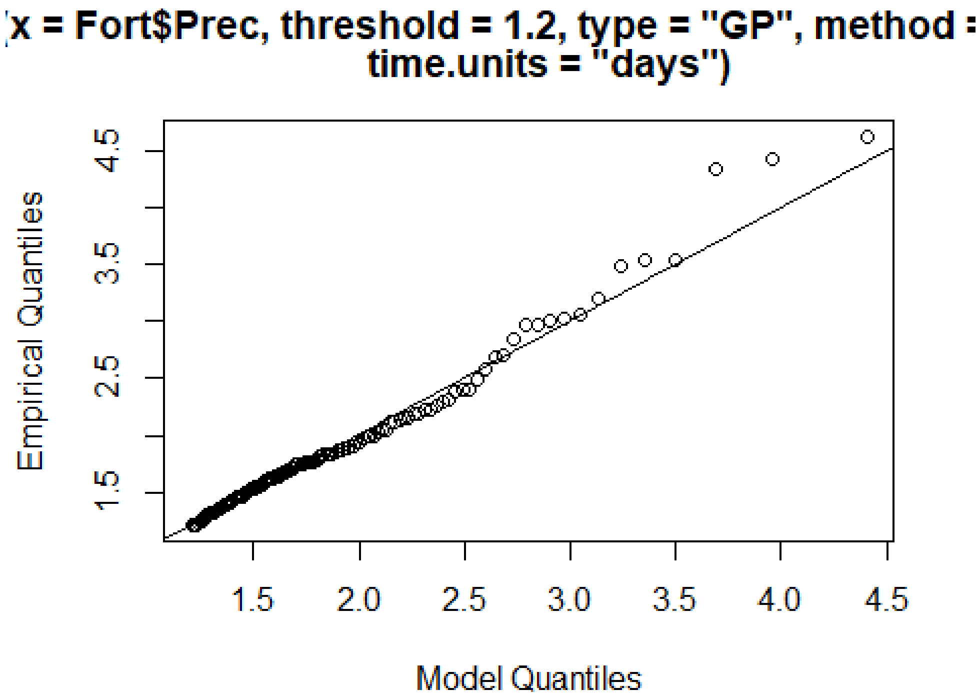
```
##      scale      shape
## 0.07783492 0.08549310
##
##  Estimated parameter covariance matrix.
##             scale        shape
## scale  0.006058275 -0.004702465
## shape -0.004702465  0.007309071
##
##  AIC = 159.8075
##
##  BIC = 165.662
```

```
ci(FitFortgpdMle,type="parameter")
```

```
## fevd(x = Fort$Prec, threshold = 1.2, type = "GP", method = "MLE",
##     time.units = "days")
##
## [1] "Normal Approx."
##
##       95% lower CI    Estimate 95% upper CI
## scale    0.4925950 0.645148623    0.7977023
## shape   -0.1647726 0.002790766    0.1703542
```

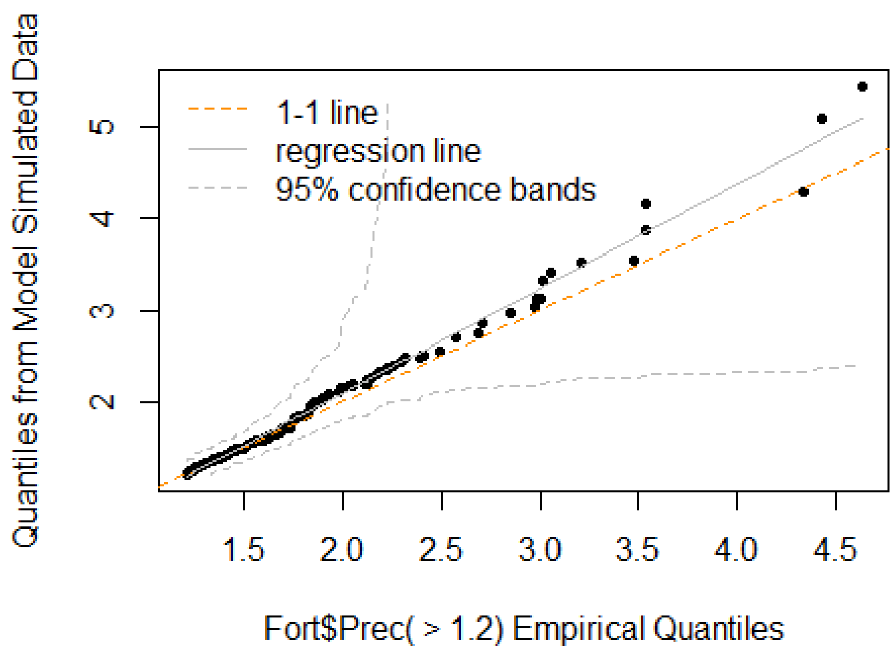
Lambda equal 0.0027 which is a small value and close to 0, we notice that the interval of lambda contains 0 what make us think about the exponential approach.

```
plot(FitFortgpdMle,type="qq")
```



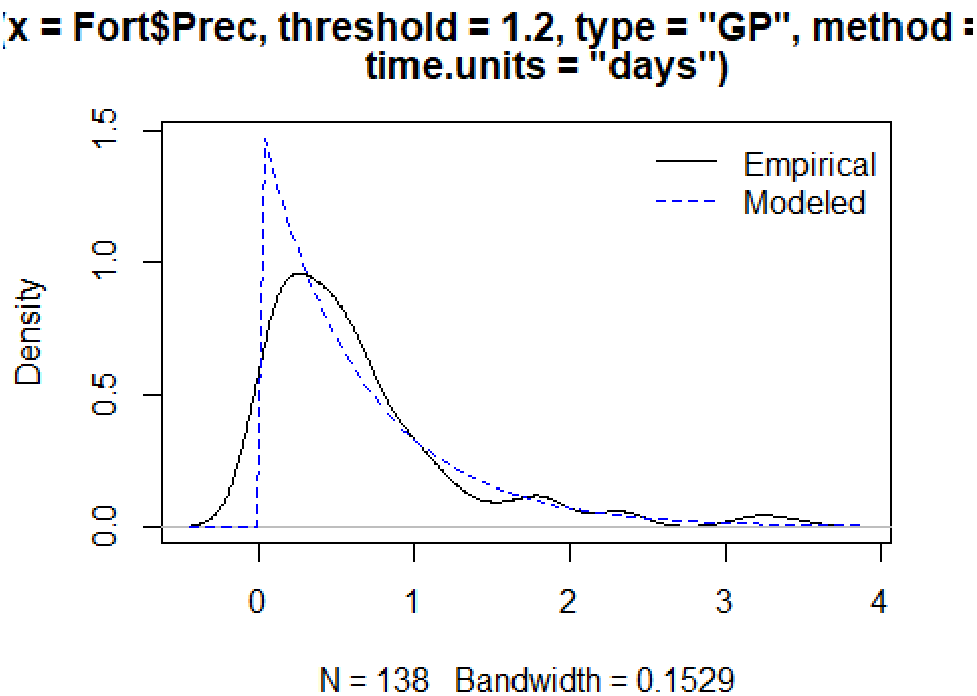
The plot is generally good except the 5 values that is not linear in the extreme.

```
plot(FitFortgpdMle,type="qq2")
```



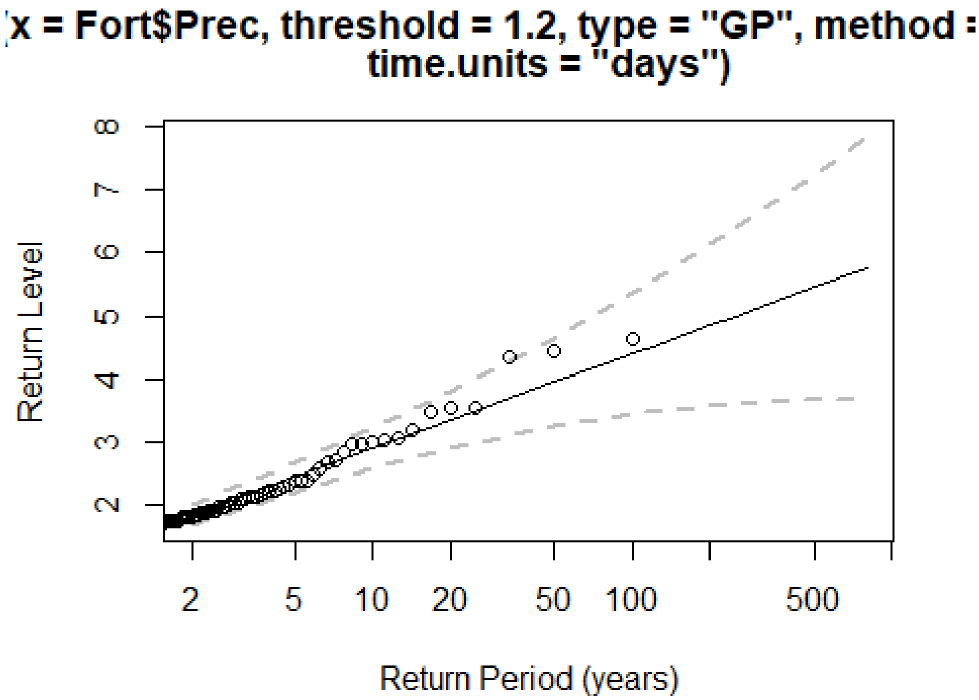
The plot is quite good, all the values are inside the confidence bands but, it's like the bands are going bigger.

```
plot(FitFortgpdMle,type="density")
```



The density in this approach is not good we have a lot of differences between the empirical density and the modeled one, some places underestimating other one over estimating.

```
plot(FitFortgpdMle,type="rl")
```



This plot is not perfect either, we have some value on the extremes of the bands, one is almost out of it.

```
FitFortgpdMleExp=fevd(Fort$Prec,threshold=1.2,type="Exponential",method="MLE"
,time.units="days")
FitFortgpdMleExp

##
## fevd(x = Fort$Prec, threshold = 1.2, type = "Exponential", method = "MLE",
##     time.units = "days")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  77.90429
##
##
##  Estimated parameters:
##     scale
## 0.6469565
##
##  Standard Error Estimates:
##     scale
## 0.0550722
##
##  Estimated parameter covariance matrix.
##               scale
```

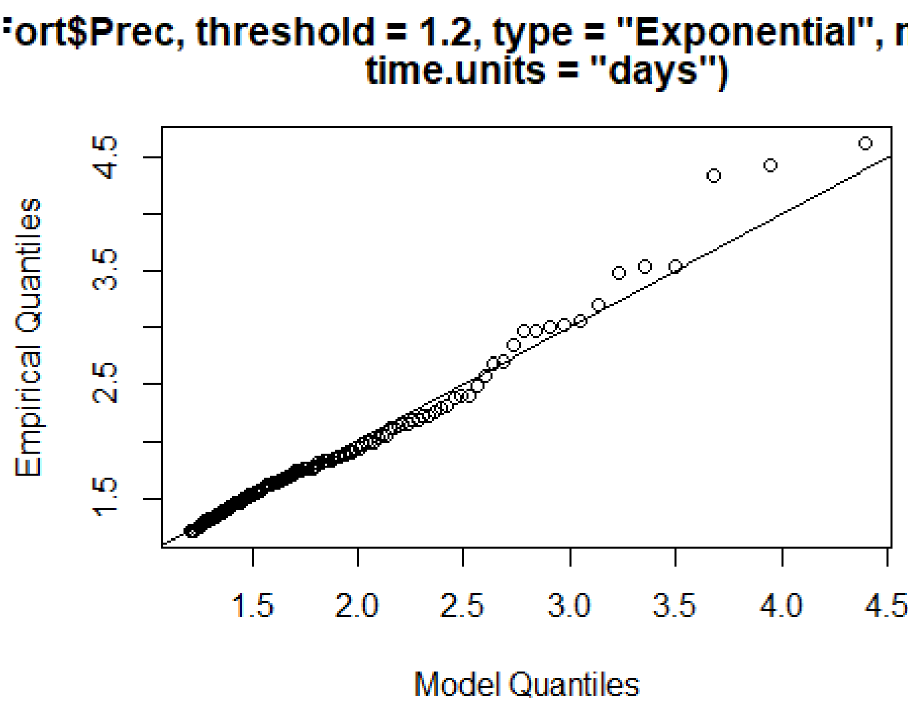
```
## scale 0.003032947
##
##  AIC = 157.8086
##
##  BIC = 160.7358
```

```
ci(FitFortgpdMleExp,type="parameter")
```

```
## fevd(x = Fort$Prec, threshold = 1.2, type = "Exponential", method = "MLE",
##     time.units = "days")
##
## [1] "Normal Approx."
##
## [1] "scale: 0.647"
##
## [1] "95% Confidence Interval: (0.539, 0.7549)"
```

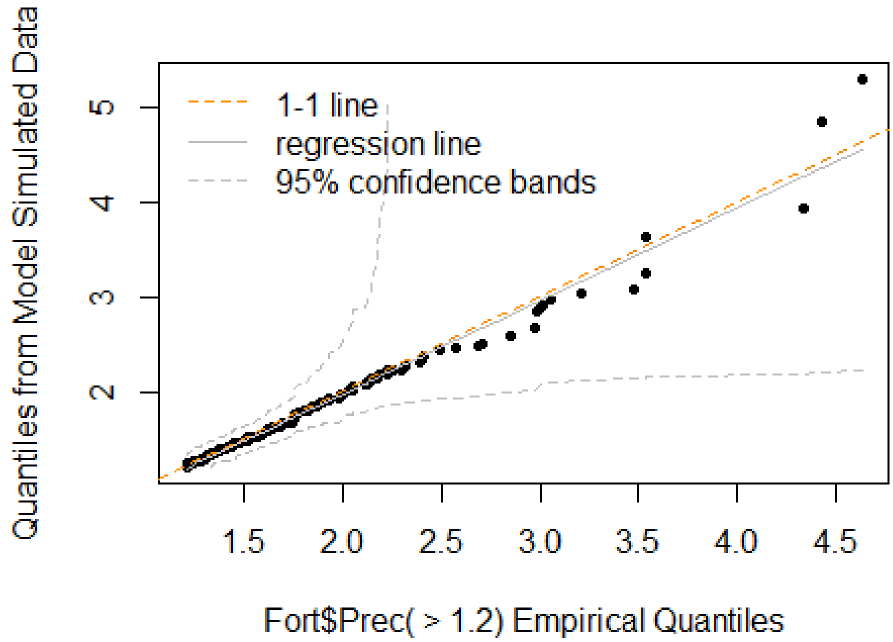
Let's take a look aver the plots:

```
plot(FitFortgpdMleExp,type="qq")
```



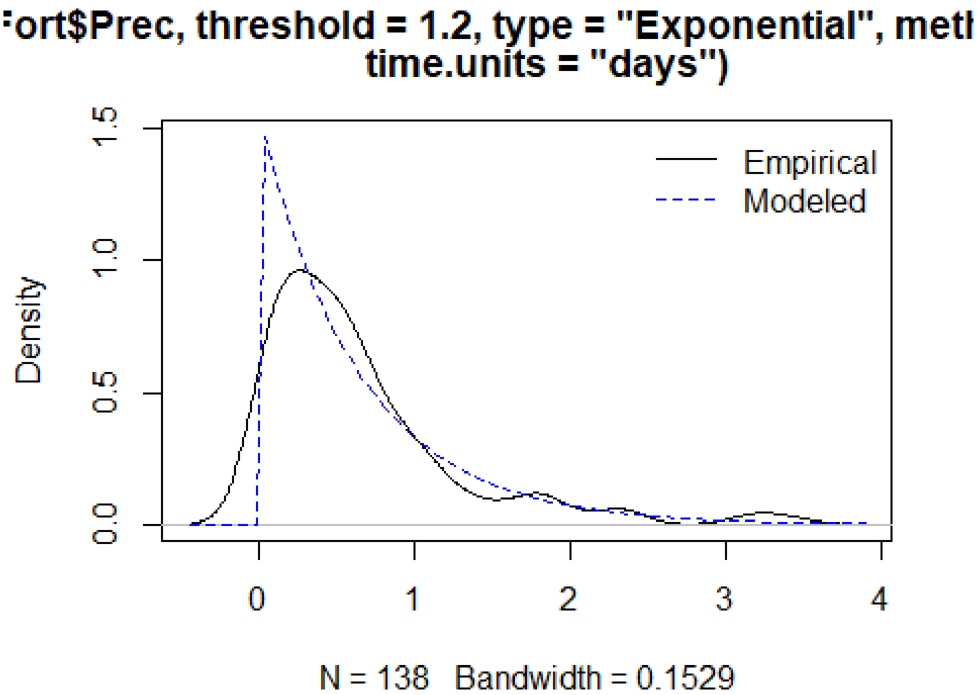
Same as the last approach, we have some values who are not linear in the extreme.

```
plot(FitFortgpdMleExp,type="qq2")
```



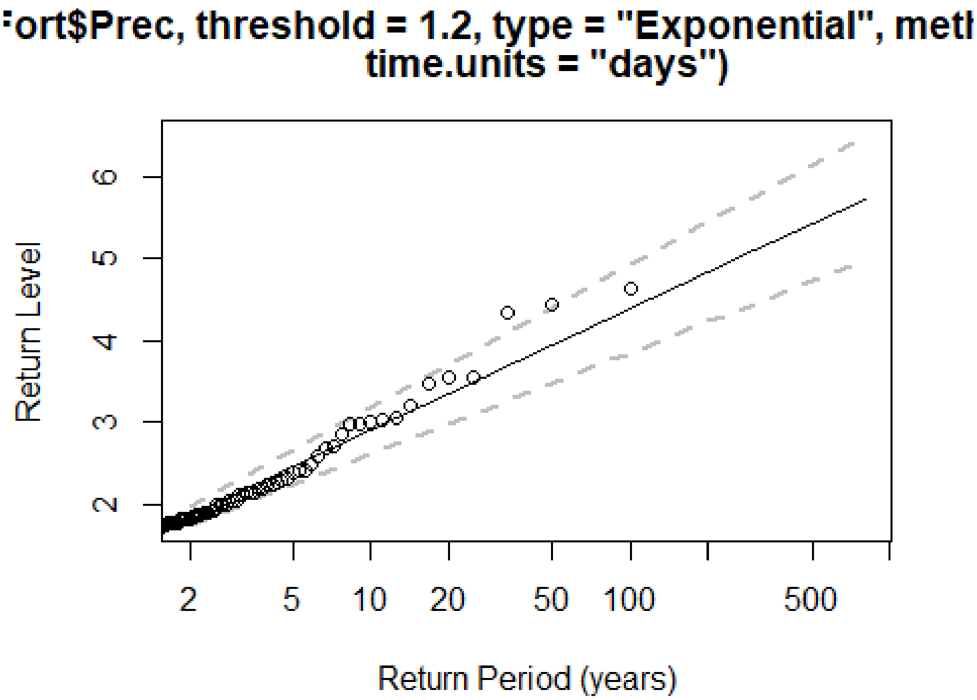
The plot is quite good, all the values are inside the confidence bands but, it’s like the bands are going bigger.

```
plot(FitFortgpdMleExp,type="density")
```



The density in this approach is not good we have a lot of differences between the empirical density and the modeled one, some places underestimating other one over estimating.

```
plot(FitFortgpdMleExp,type="rl")
```



This plot is not perfect either, we have some value on the extremes of the bands, one is out of it and the second is almost out.

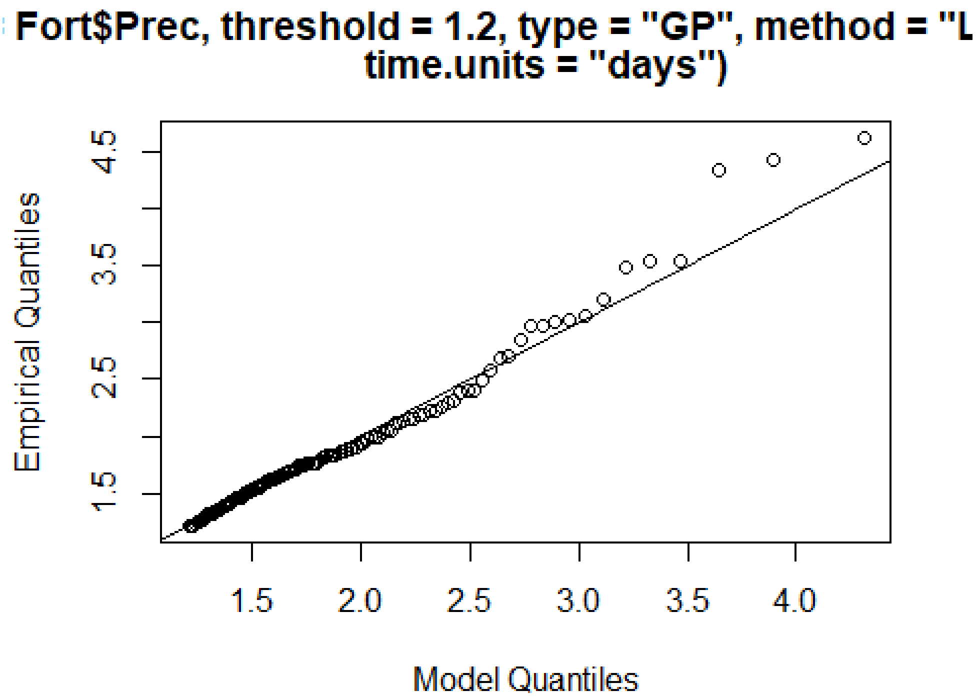
Probability Weighted Moment PWM:

```
FitFortgpdMoments=fevd(Fort$Prec,threshold=1.2,type="GP",method="Lmoments",ti
me.units="days")
FitFortgpdMoments

## fevd(x = Fort$Prec, threshold = 1.2, type = "GP", method = "Lmoments",
##     time.units = "days")
## [1] "GP  Fitted to  Fort$Prec  using L-moments estimation."
##       scale       shape
##  0.65861678 -0.01802325

ci(FitFortgpdMoments,type="parameter")

## fevd(x = Fort$Prec, threshold = 1.2, type = "GP", method = "Lmoments",
##     time.units = "days")
##
## [1] "Parametric Bootstrap"
## 502  iterations
##
##              2.5%    Estimate     97.5%
## scale  0.5024229  0.65861678 0.8258486
## shape -0.2030171 -0.01802325 0.1574323
```

We have lambda here is minus -0.01802325, and the interval contain 0 as we can notice.

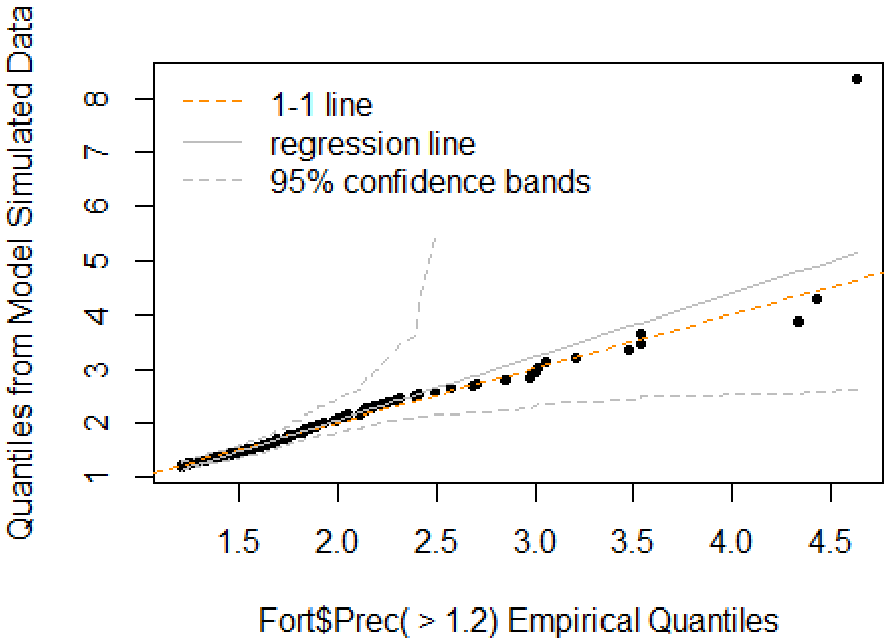
Let's take a look over the plots:

```
plot(FitFortgpdMoments,type="qq")
```



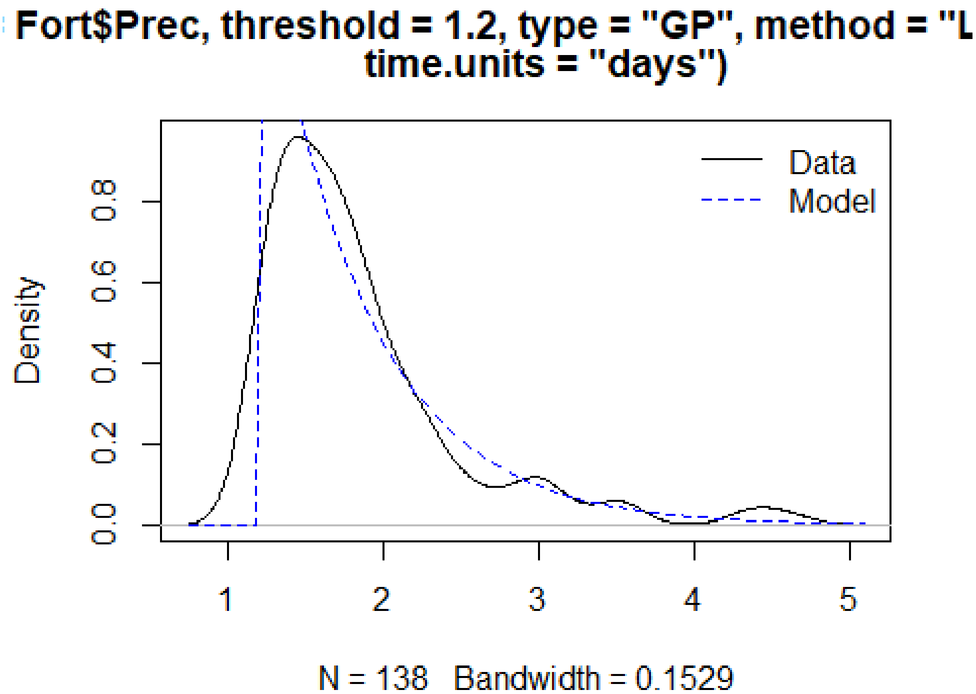
We have some values who are not linear in the extreme.

```
plot(FitFortgpdMoments,type="qq2")
```



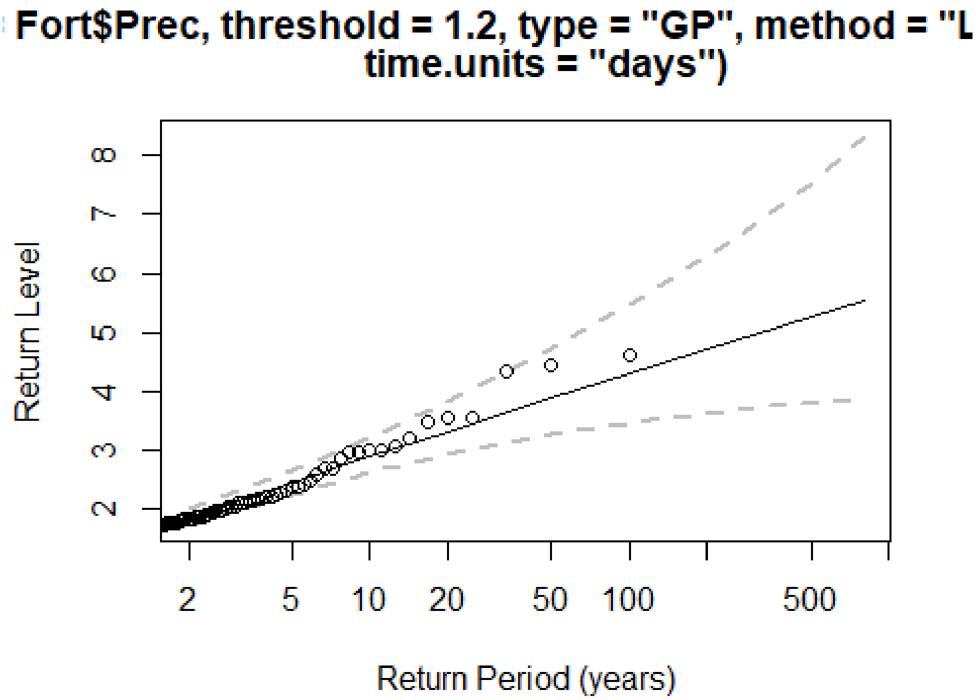
The plot is quite good except of the only value in the extreme, all the values are inside the confidence bands but, it's like the bands are going bigger.

```
plot(FitFortgpdMoments,type="density")
```



The density in this approach is not good we have a lot of differences between the empirical density and the modeled one, some places underestimating other one over estimating.

```
plot(FitFortgpdMoments,type="rl")
```



This plot is not perfect either, we have one value almost out of the bands.

Return Level:

```
return.level(FitFortgpdMle, return.period = c(20, 100, 1000))

## fevd(x = Fort$Prec, threshold = 1.2, type = "GP", method = "MLE",
##     time.units = "days")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  GP model fitted to  Fort$Prec
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##   20-year level  100-year level 1000-year level
##        3.350442        4.400785        5.911712

return.level(FitFortgpdMleExp, return.period = c(20, 100, 1000))

## fevd(x = Fort$Prec, threshold = 1.2, type = "Exponential", method = "MLE",
##     time.units = "days")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  Exponential model fitted to  Fort$Prec
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
```

```
##   20-year level  100-year level 1000-year level
##        3.346500        4.387737        5.877409
```

```
return.level(FitFortgpdMoments, return.period = c(20, 100, 1000))
```

```
## fevd(x = Fort$Prec, threshold = 1.2, type = "GP", method = "Lmoments",
##     time.units = "days")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  GP model fitted to  Fort$Prec
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##   20-year level  100-year level 1000-year level
##        3.321135        4.305267        5.664520
```

```
summary(Fort$Prec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.04181 0.00000 4.63000
```

Data set	Method of estimation	Fréchet γ > 0	Exp γ = 0	Weibull γ < 0 (Endpoint)
Fort colin colorado	MLE	4.400785	4.387737	X
	MOMENT	X	X	4.305267

COMPARAISON between GEV and GPD:

Data set/approach	Method of estimation	Fréchet γ > 0	γ = 0	Weibull γ < 0 (Endpoint)
Fort GP	MLE	5.093685	4.059812	X
Fort GP	MOMENT	4.863562	X	X
Fort GEV	MLE	4.400785	4.387737	X
Fort GEV	MOMENT	X	X	4.305267

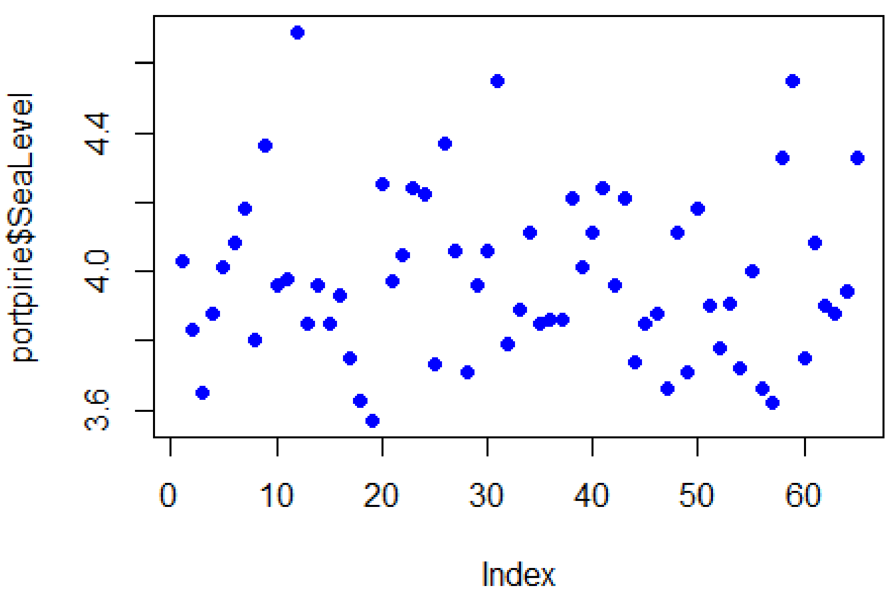
CONCLUSION:

Between the both approach, I choose the GEV, it is more consistent and return levels are not far from the max (4,63000).

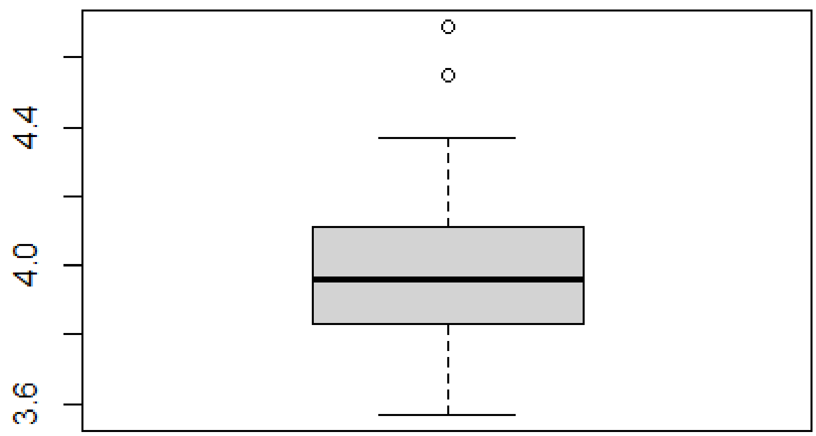
DATA OF port Pirie:

we must do the GEV approach.

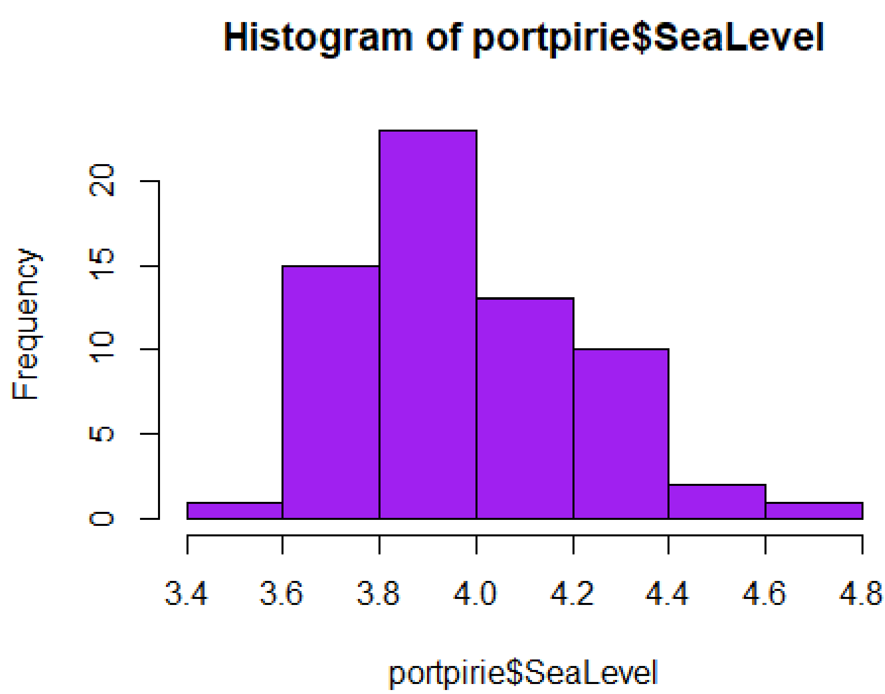
```
data(portpirie)
str(portpirie)

## 'data.frame':    65 obs. of  2 variables:
##  $ Year    : num  1923 1924 1925 1926 1927 ...
##  $ SeaLevel: num  4.03 3.83 3.65 3.88 4.01 4.08 4.18 3.8 4.36 3.96 ...

length(portpirie)

## [1] 2

summary(portpirie)

##       Year         SeaLevel
##  Min.   :1923   Min.   :3.570
##  1st Qu.:1939   1st Qu.:3.830
##  Median :1955   Median :3.960
##  Mean   :1955   Mean   :3.981
##  3rd Qu.:1971   3rd Qu.:4.110
##  Max.   :1987   Max.   :4.690

plot(portpirie$SeaLevel, col = "blue", pch = 16)
```



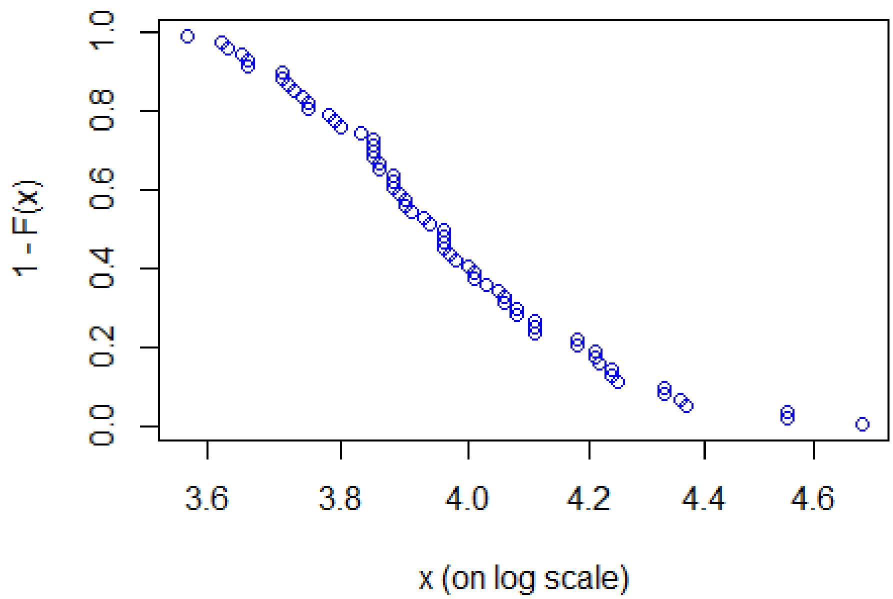

```
boxplot(portpirie$SeaLevel)
```




```
hist(portpirie$SeaLevel, col = "purple")
```




```
emplot(portpirie$SeaLevel, col = "blue")
```

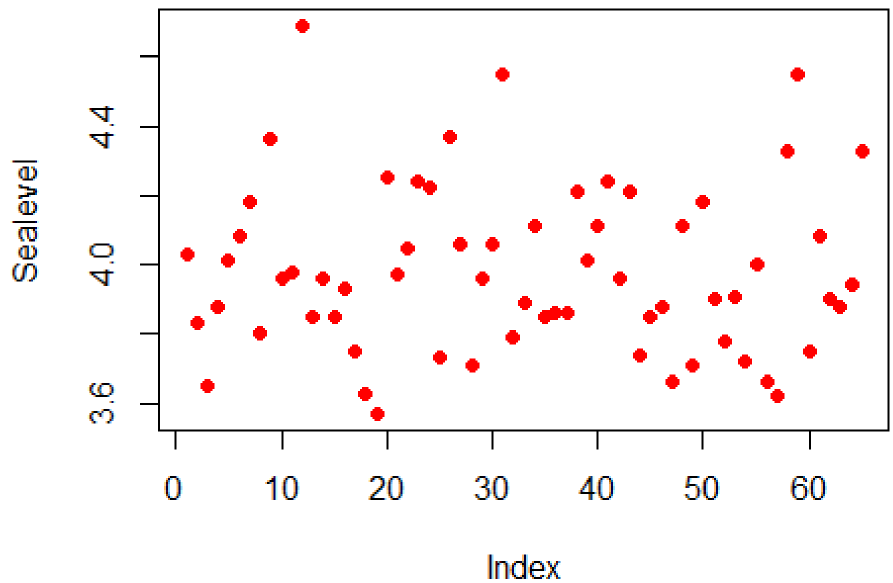


The Generalized Extreme Value distribution: GEV

The block maxima approach:

```
bmSea<- blockmaxxer(portpirie, blocks = portpirie$Year, which="SeaLevel")
head(bmSea)

##   Year SeaLevel
## 1 1923     4.03
## 2 1924     3.83
## 3 1925     3.65
## 4 1926     3.88
## 5 1927     4.01
## 6 1928     4.08

as.numeric(rownames(bmSea))

##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50
## [51] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

length(as.numeric(rownames(bmSea)))

## [1] 65
```

```
plot(portpirie$SeaLevel,ylab = "Sealevel",col = "blue",pch = 16)
points(as.numeric(rownames(bmSea)),portpirie$SeaLevel, col="red", pch=16)
```



Maximum Likelihood Estimators:

```
FitSeaMle = fevd(portpirie$SeaLevel, type="GEV", method="MLE")
FitSeaMle

##
## fevd(x = portpirie$SeaLevel, type = "GEV", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  -4.339058
##
##
##  Estimated parameters:
##   location      scale      shape
##  3.8747499  0.1980440 -0.0501095
##
##  Standard Error Estimates:
##   location      scale      shape
## 0.02793224 0.02024798 0.09825416
##
##  Estimated parameter covariance matrix.
##                 location         scale         shape
```

```
## location  0.0007802099  0.0001970436 -0.0010740750
## scale     0.0001970436  0.0004099808 -0.0007774225
## shape    -0.0010740750 -0.0007774225  0.0096538796
##
##  AIC = -2.678117
##
##  BIC = 3.845045
```

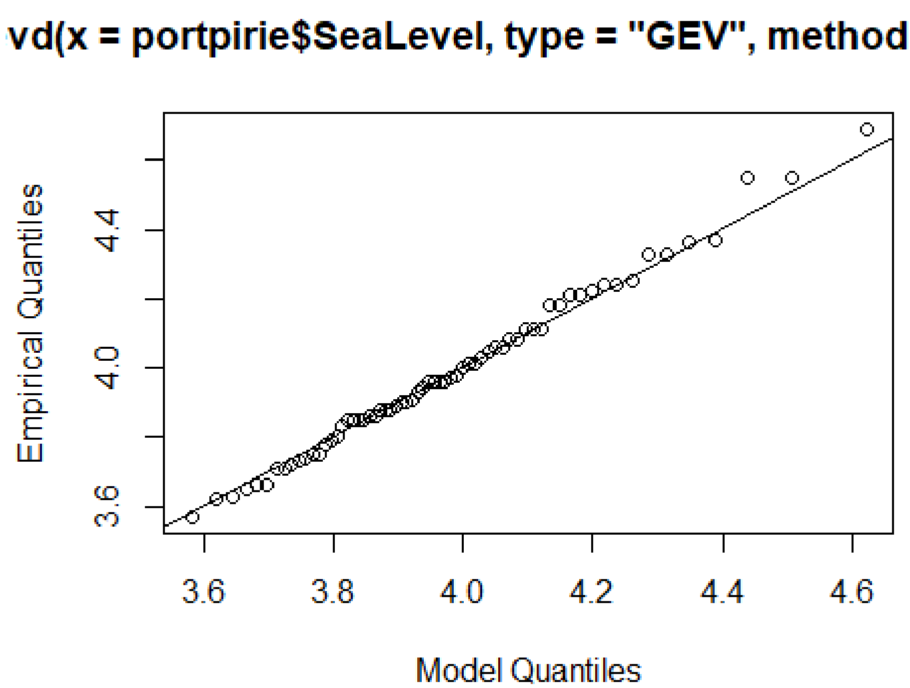
```
ci(FitSeaMle,type="parameter")
```

```
## fevd(x = portpirie$SeaLevel, type = "GEV", method = "MLE")
##
## [1] "Normal Approx."
##
##          95% lower CI   Estimate 95% upper CI
## location    3.8200037  3.8747499    3.9294960
## scale       0.1583586  0.1980440    0.2377293
## shape      -0.2426841 -0.0501095    0.1424651
```

Lambda equal to -0.0501095 <0, and the interval contains 0, it gives us the idea of Gumble approach.

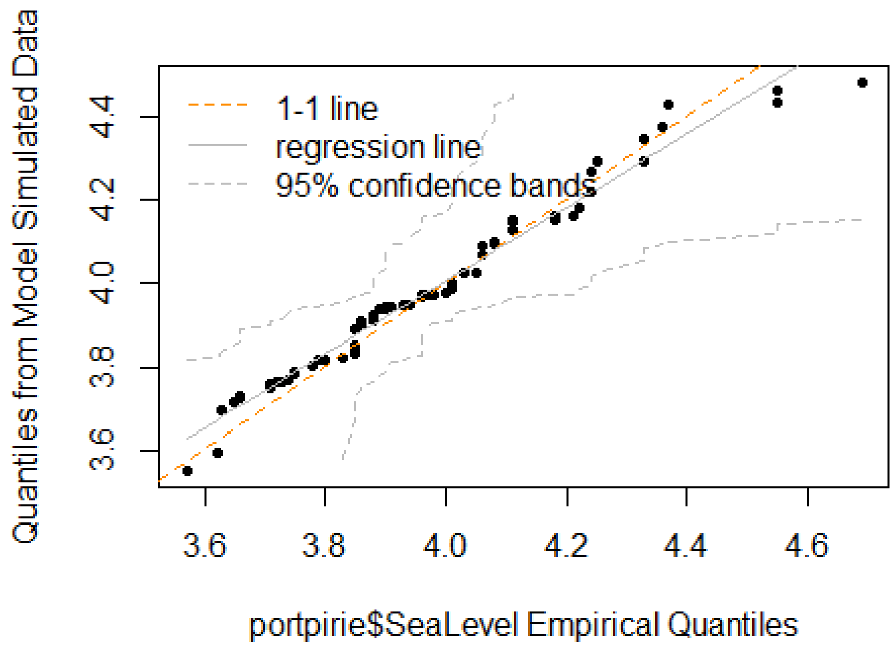
Let's take a look of the plots:

```
plot(FitSeaMle,type="qq")
```



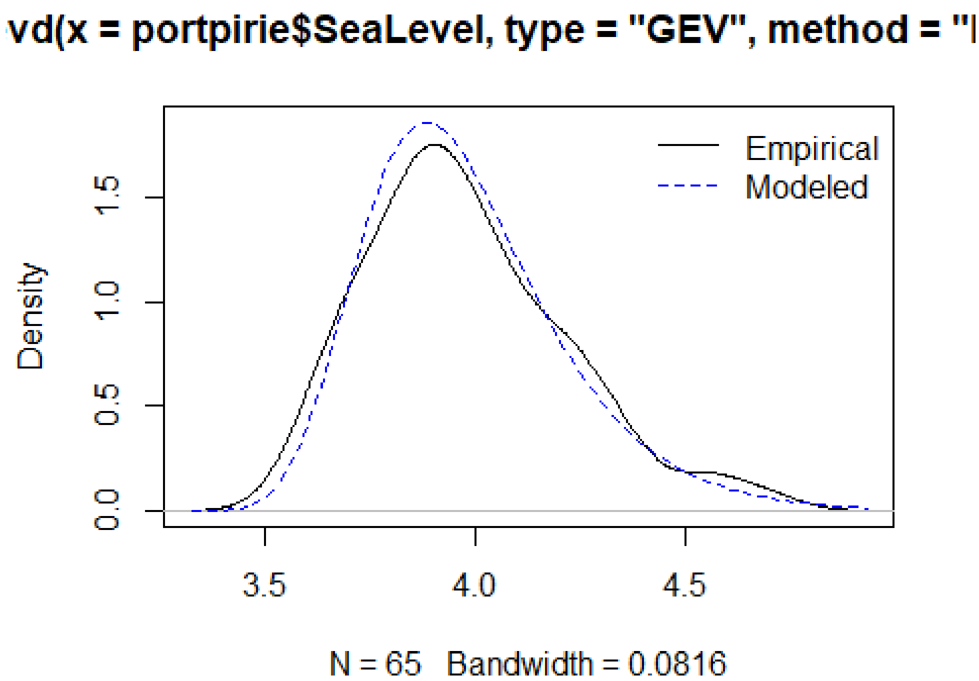
We have some values who are not linear in the extreme.

```
plot(FitSeaMle,type="qq2")
```



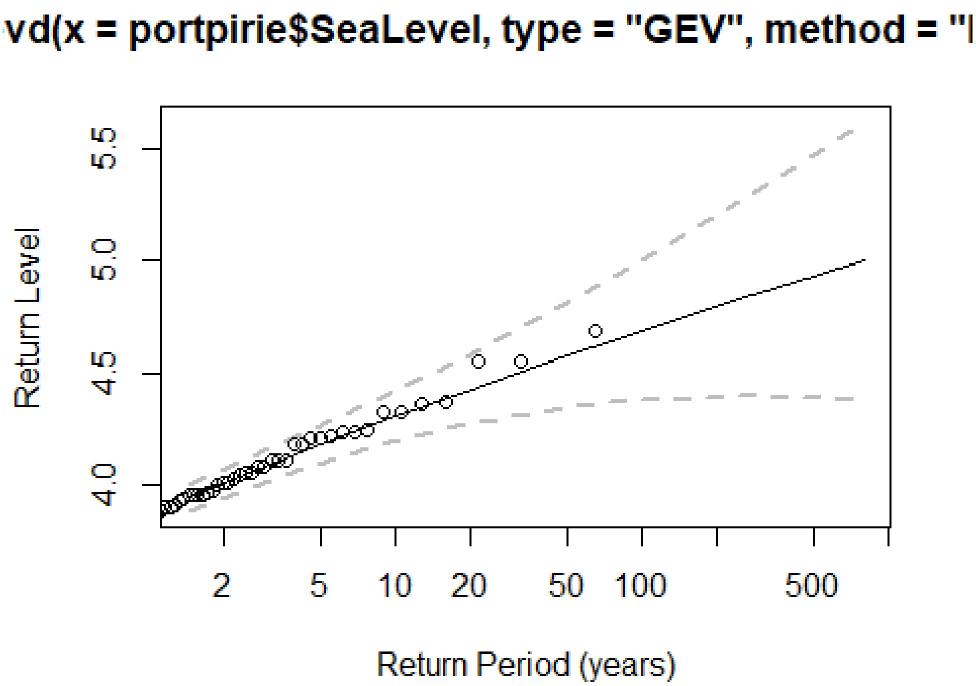
The plot is quite good except of the only value in the extreme, all the values are inside the confidence bands but, it's like the bands are going bigger.

```
plot(FitSeaMle,type="density")
```



The density empirical is almost same as the modeled one, we can say that it is good.

```
plot(FitSeaMle,type="rl")
```



This plot is perfect, all the values are in the bands.

The Gumbel model:

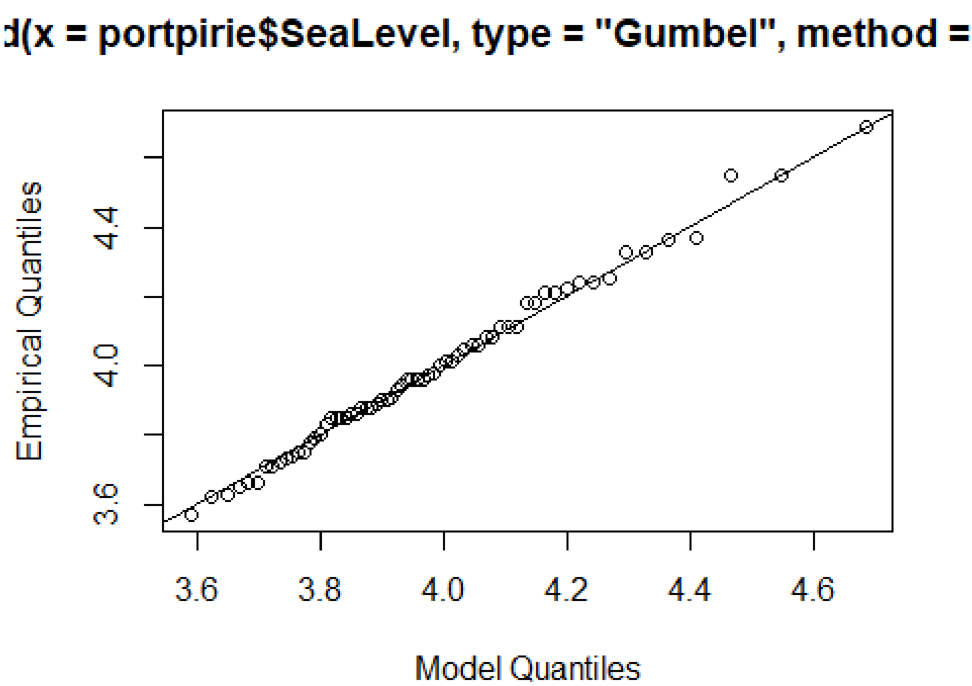
```
FitSeaGumbelMle=fevd(portpirie$SeaLevel,type="Gumbel",method="MLE")
FitSeaGumbelMle

##
## fevd(x = portpirie$SeaLevel, type = "Gumbel", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  -4.217682
##
##
##  Estimated parameters:
##  location     scale
## 3.8694436 0.1948895
##
##  Standard Error Estimates:
##   location      scale
## 0.02549389 0.01885368
##
##  Estimated parameter covariance matrix.
```

```
##                location        scale
## location 0.0006499383 0.0001527080
## scale    0.0001527080 0.0003554611
##
##  AIC = -4.435364
##
##  BIC = -0.08658925
```

```
ci(FitSeaGumbelMle)
```

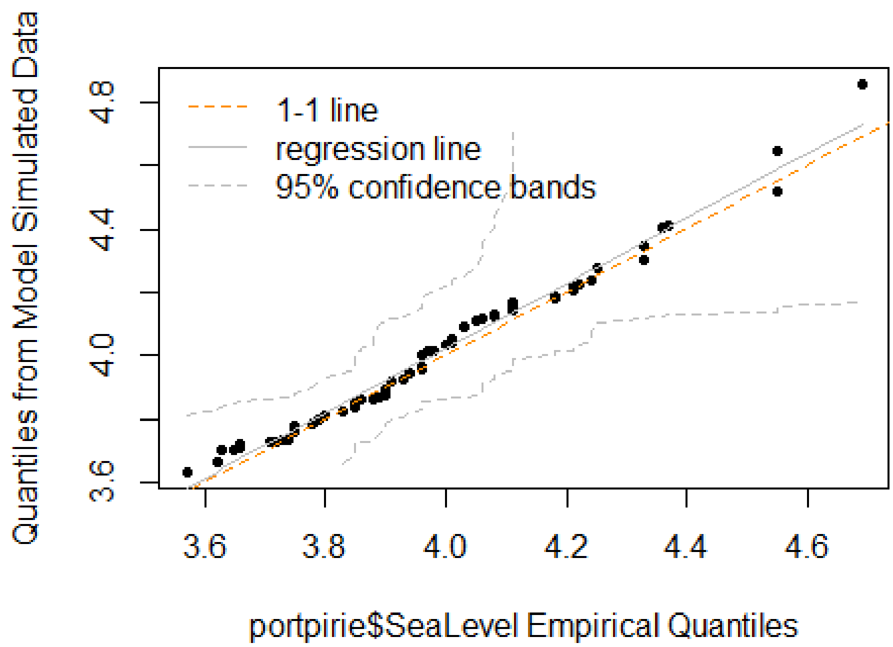
```
## fevd(x = portpirie$SeaLevel, type = "Gumbel", method = "MLE")
##
## [1] "Normal Approx."
##
## [1] "100-year return level: 4.766"
##
## [1] "95% Confidence Interval: (4.5742, 4.9578)"
```

```
plot(FitSeaGumbelMle,type="qq")
```



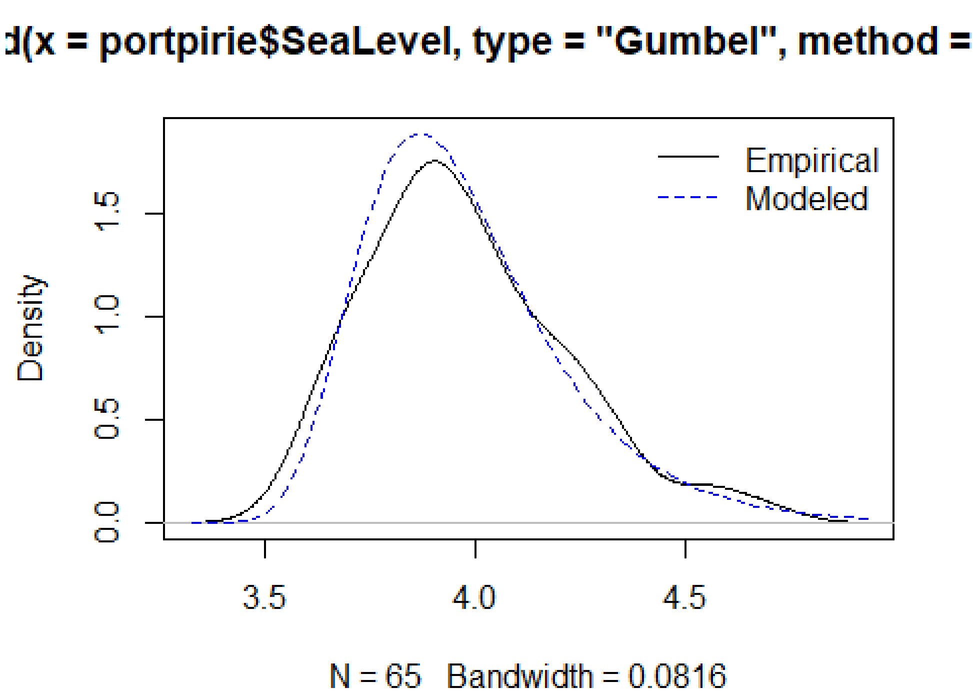
This plot is perfect, all the values are linear except one but we can ignore it.

```
plot(FitSeaGumbelMle,type="qq2")
```



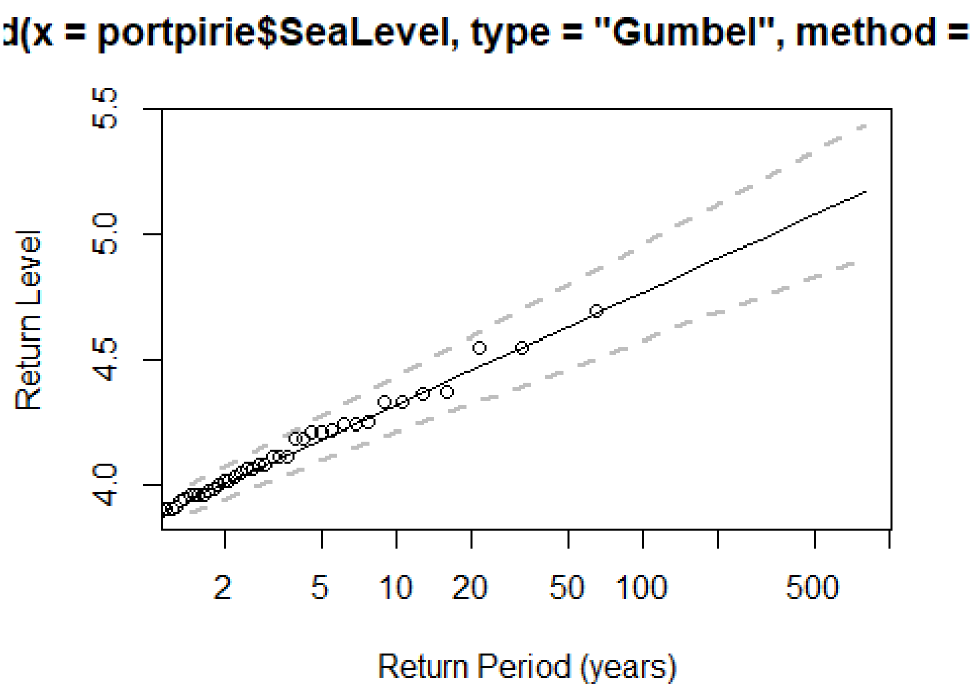
This plot is perfect, alal the values are in the confidence bands.

```
plot(FitSeaGumbelMle,type="density")
```



The density empirical is almost same as the modeled one, we can say that it is good.

```
plot(FitSeaGumbelMle,type="rl")
```



This plot is perfect, all the values are in the bands.

Comparaison:

```
lr.test(FitSeaMle,FitSeaGumbelMle)

##
##  Likelihood-ratio Test
##
## data:  portpirie$SeaLevelportpirie$SeaLevel
## Likelihood-ratio = 0.24275, chi-square critical value = 3.8415, alpha =
## 0.0500, Degrees of Freedom = 1.0000, p-value = 0.6222
## alternative hypothesis: greater
```

The p-value of the test is 0.6222 bigger than 0.05, we can conclude that the nested model provides a significantly better fit.

Probability Weighted Moment: PWM

```
FitSeaMoments=fevd(portpirie$SeaLevel,type="GEV",method="Lmoments")
FitSeaMoments

## fevd(x = portpirie$SeaLevel, type = "GEV", method = "Lmoments")
## [1] "GEV  Fitted to  portpirie$SeaLevel  using L-moments estimation."
```

```
##    location       scale       shape
##  3.87317236  0.20326758 -0.05147713
```

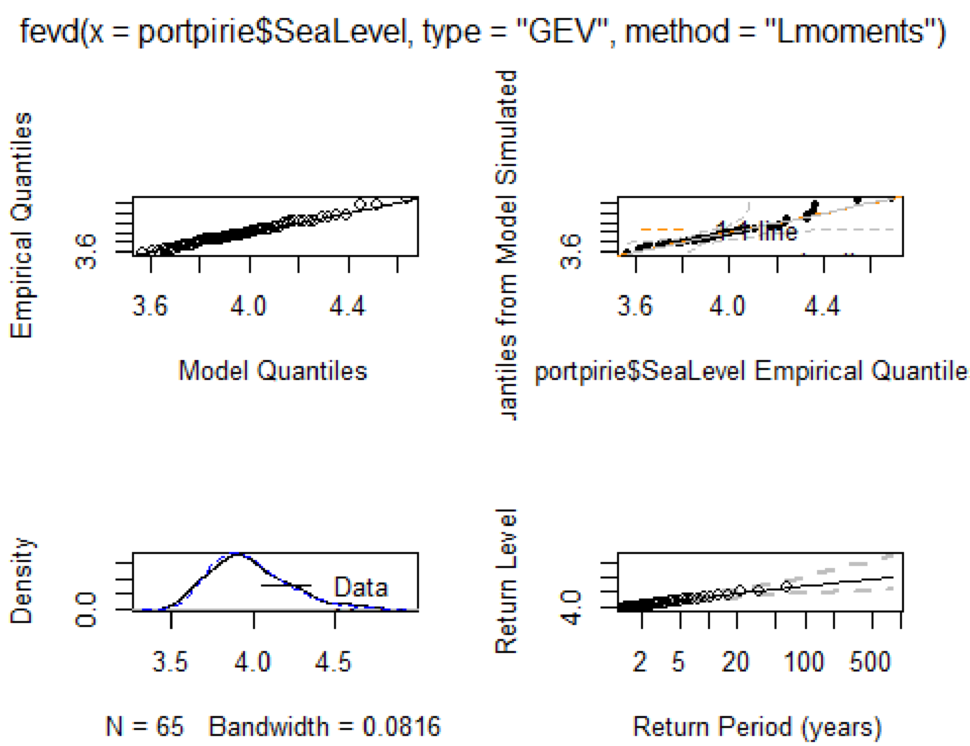
```
ci(FitSeaMoments,type="parameter")
```

```
## fevd(x = portpirie$SeaLevel, type = "GEV", method = "Lmoments")
##
## [1] "Parametric Bootstrap"
## 502  iterations
##
##                 2.5%    Estimate     97.5%
## location  3.8217557  3.87317236 3.9315773
## scale     0.1619381  0.20326758 0.2440845
## shape    -0.2378661 -0.05147713 0.1236184
```

Lambda here is negative too -0.05147713, and the interval contains 0.

Let’s take a look on the plots:

```
plot(FitSeaMoments)
```



All the plots are quite good; the density empirical is same as the modeled one. All the values are inside the bands.

Return Level for Fort GEV MLE and PWM:

```
return.level(FitSeaMle, return.period = c(100, 1000))

## fevd(x = portpirie$SeaLevel, type = "GEV", method = "MLE")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  GEV model fitted to  portpirie$SeaLevel
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  100-year level 1000-year level
##        4.688404        5.031059

return.level(FitSeaGumbelMle, return.period = c(100, 1000))

## fevd(x = portpirie$SeaLevel, type = "Gumbel", method = "MLE")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  Gumbel model fitted to  portpirie$SeaLevel
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  100-year level 1000-year level
##        4.765964        5.215595

return.level(FitSeaMoments, return.period = c( 100, 1000))

## fevd(x = portpirie$SeaLevel, type = "GEV", method = "Lmoments")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period =
return.period)
##
##  GEV model fitted to  portpirie$SeaLevel
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  100-year level 1000-year level
##        4.705766        5.054714

summary(portpirie$SeaLevel)

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

Data set	Method of estimation	Fréchet $\gamma > 0$	Gumbel $\gamma = 0$	Weibull $\gamma < 0$ (Endpoint)
Port pirie	MLE	X	4.764964	4.688404
	MOMENT	X	X	4.705766

Conclusion

It seems that this approach gives us a consistent return level which is not far from the max (4,690).

If I have to choose between the MLE or MOMENT, I prefer the MLE, it give us two values, with this values we can create an interval [4,68 ; 4,76].