# Group 6 - MA471 Project 2
# Modelling Insurance Claims using Extreme Value Theory
# Mentored by - Dr. Arabin Kumar Dey

Group members : Harit Gupta (170123020), Sakshi Sharma (170123044), Tanvi Ohri (170123051), Tejasvee Panwar (170123053), Kartik Sethi (170123057)

## Disclaimer

The content of this report and the product made is only meant for learning process as part of the course. This is not for use of making publication or making commercialisation without mentor's consent. Our contribution won't demand any claim in future for further progress of mentor's development and innovation along the direction unless there is a special continuous involvement.

## What is an extreme value?

An extreme value is either very small or very large values in a probability distribution. These extreme values are found in the tails of a probability distribution (i.e. the distribution's extremities).

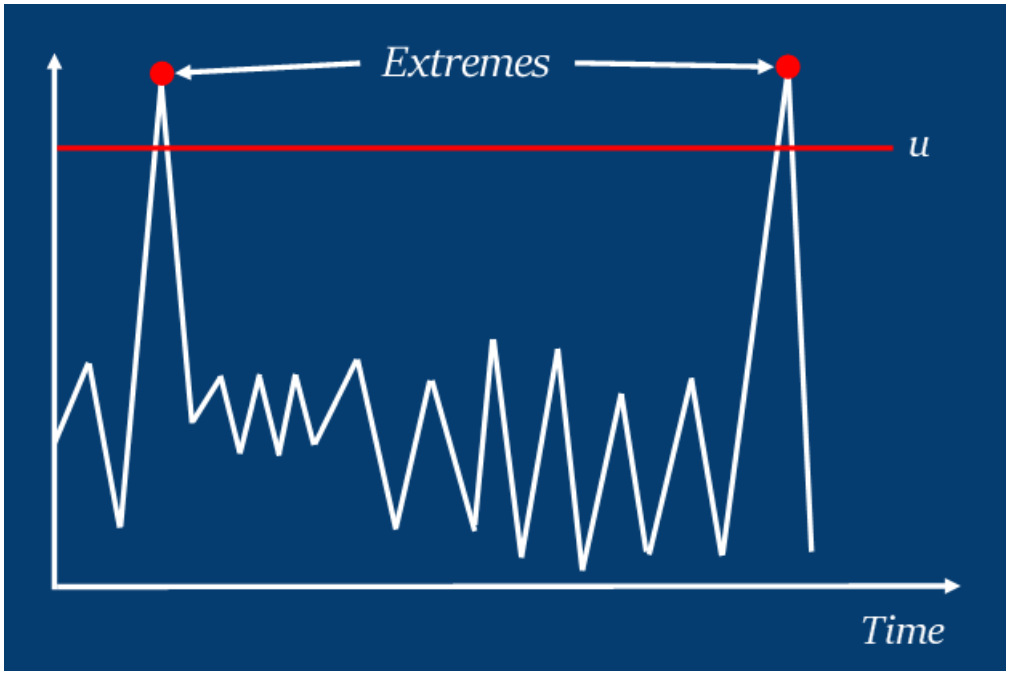



Figure 1: Example of extreme values

## Introduction to Extreme Value Theory

Extreme events ranging from travel disruptions to natural disasters have been a reason for huge losses since a long time, despite their rarity. Extreme Value Theory (EVT) is a branch of Statistical Mathematics that provides insights on their inherent scarcity and stark magnitude. In other words, it's a model for a process that has some kind of randomness. EVT aims to predict probabilities for rare events greater (or smaller) than previous recorded events. It was initially used to model the occurrences of records (say for example in athletic events) or quantify the probability of floods with magnitude greater

than what has been observed in the past, i.e it allows us extrapolate beyond the range of available data!

## Application of EVT in measuring financial risk

Assessing the probability of rare and extreme events is an important issue in the risk management of financial portfolios. Extreme value theory provides the solid fundamentals needed for the statistical modelling of such events and the computation of extreme risk measures.

## Financial risk measures

### 5.1 Value at Risk

Value-at-Risk is generally defined as the capital sufficient to cover, in most instances, losses from a portfolio over a holding period of a fixed number of days. Suppose a random variable X with continuous distribution function F models losses or negative returns on a certain financial instrument over a certain time horizon. $VaR_p$ can then be defined as the p-th quantile of the distribution F

$$VaR_p = F^{-1}(1-p)$$

where $F^{-1}$ is called the *quantile function*, defined as the inverse of the distribution function F.

### 5.2 Expected Shortfall

Another informative measure of risk is the expected shortfall (ES) or the tail conditional expectation which estimates the potential size of the loss exceeding VaR. The expected shortfall is defined as the expected size of a loss that exceeds $VaR_p$

$$ES_p = E(X \mid X > VaR_p)$$

## Methods of solving EVT problems

Basically, there are two ways of identifying extremes in real data. They can be explained briefly using an example. Let us consider a random variable representing daily losses or returns. The first approach considers the maximum the variable takes in successive periods, for example months or years. These selected observations constitute the extreme events, also called block (or per period) maxima. On left in Figure 2, the observations $X_2, X_5, X_7$ and $X_{11}$ represent the block maxima for four periods of three observations each. The second approach focuses on the values exceeding a given (high) threshold. The observations $X1, X2, X7, X8, X9$ and $X_{11}$ on the right in Figure 2, all exceed the threshold u and constitute extreme events.

Now, let us discuss the two in detail.

### 6.1 Peak Over Threshold method

First of all, let us give the definition of GPD cumulative distribution function which is as follows:

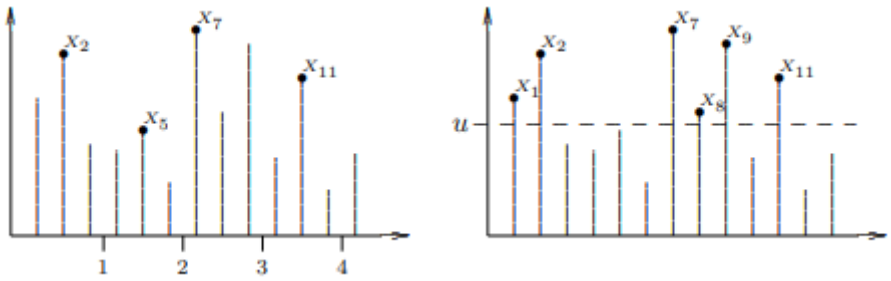



Figure 2: Block-maxima (left) and excesses over a threshold u (right)

$$GDP(x;\xi,\mu,\sigma) = \begin{cases} 1-(1+\xi(\frac{x-\mu}{\sigma}))^{-1/\xi} & \text{if } \xi \neq 0 \\ (1-\exp(-\frac{x-\mu}{\sigma})) & \text{if } \xi = 0 \end{cases}$$

In this case, a series of random variables $X_1, X_2, ..., X_n$ (i.i.d.) and certain threshold level u are considered. Assuming that right tail of a distribution is of interest, for all realisations $x_i$ above the threshold u the values of exceedances $y_1, y_2, .., y_n$ are calculated ($y_i = x_i u$). The distribution of exceedances above the u threshold is defined as:

$$F_u(x;u) = P(X = u + y | X > u) = \frac{F(y+u)-F(u)}{1-F(u)}$$

Assuming that for a certain threshold u the distribution of observations being above the threshold is the $GDP(x;\xi,\mu,\sigma)$, the tail of the distribution of the return rates above the assumed cut-off point can be written as follows:

$$F(x) = F(y+u) = [1-F(u)]GDP(x;\xi,\mu,\sigma) + F(u)$$

where F (.) is a cumulative distribution function, u is the cut-off threshold, y is a loss level above the cut-off threshold u and $G_{\xi,\sigma}(y)$ is the cumulative distribution function value of the GPD.
Value at Risk at the a level is calculated from the following formula:

$$VaR(\alpha) = u + \frac{\hat{\sigma}}{\hat{\xi}}[[\frac{n}{N_u}(1-\alpha)]^{-\hat{\xi}} - 1]$$

where VaR() is Value at Risk at the significance level a, u is the cut-off threshold, $\hat{\xi}, \hat{\sigma}$, are GPD parameters, n is the total number of the analysed return rates, $N_u$ is the number of return rates below the cut-off point u.

### 6.2 Block Maximization method

As we already saw, the block maximization method consists in dividing the set of data into $M(m = 1, 2, ..., M)$ time intervals of length n each. The values used for estimation are the minimums or maximums observed in subsequent M time intervals. In other words, if $X_1 m, X_2 m, ..., X_n m$ is a sequence of independent and identically distributed random variables from a time interval m, the maximum values can be defined as $M_m = max(X_1 m, ..., X_n m)$. The minimum values can be defined analogously by reversing their sign. To find a non-degenerated cumulative distribution function (cdf), the maximum

values $M_m$ are standardised by the variance $\sigma_m$ and the expected value $\mu_m(S_m = (M_m - \mu_m)/s_m)$. From the lectures, we know that a theorem given by Fisher, Tippet and Gnedenko states that if such a non-degenerate cdf exists, it must belong to one of the Gumbel, the Frechet or the Weibull distributions. These distributions can be interpreted as special cases of generalised extreme value (GEV) distribution. The cdf of this distribution is defined as follows (with a shape parameter x):

$$GEV(x;\xi,\mu,\sigma) = \begin{cases} \exp\left[-(1+\xi(\frac{x-\mu}{\sigma}))^{-1/\xi}\right] & \text{if } \xi \neq 0 \text{ and } \ (1+\xi(\frac{x-\mu}{\sigma})) > 0 \\ \exp\left[-\exp\left(-(\frac{x-\mu}{\sigma})\right)\right] & \text{if } \ \xi = 0 \end{cases}$$

The x sign determines which of the distributions has been selected. The Gumbel, Frechet or Weibull distribution is assumed for x = 0, x > 0 and x < 0, respectively.
From the GEV distribution, a VaR can be estimated as follows:

$$VaR(\alpha) = \begin{cases} \hat{\mu_n} - \frac{\hat{\sigma_n}}{\hat{\xi_n}}(1-(-nln(\alpha))^{-\hat{\xi_n}}) & \text{to } \xi > 0 \text{ (Fréchet)} \\ \hat{\mu_n} - \hat{\sigma_n}\ ln(-nln(\alpha)) & \text{to } \xi = 0 \text{ (Gumbel)} \end{cases}$$

where $VaR(\alpha)$ is the measure of the Value at Risk at the significance level $\alpha$, $\hat{\mu_n}$ is the estimated location parameter, $\hat{\sigma_n}$ is the estimated scale parameter and $\hat{\xi_n}$ is the estimated shape parameter.

## Experiment 1 : Modelling auto-claims using Peak Over Threshold Method

In this experiment, we are using following packages:

1. evir : Provides functions for extreme value theory
2. quantmod : Specify, build, trade, and analyse quantitative financial trading strategies
3. ineq : Inequality, concentration, and poverty measures
4. insuranceData : Insurance datasets, which are often used in claims severity and claims frequency modelling

We are using "AutoClaims" dataset. It contains claims experience from a large midwestern (US) property and casualty insurer for private passenger automobile insurance. The dependent variable is the amount paid on a closed claim, in (US) dollars (claims that were not closed by year end are handled separately). The first thing to do is to have a look at the data and at some basic statistics.

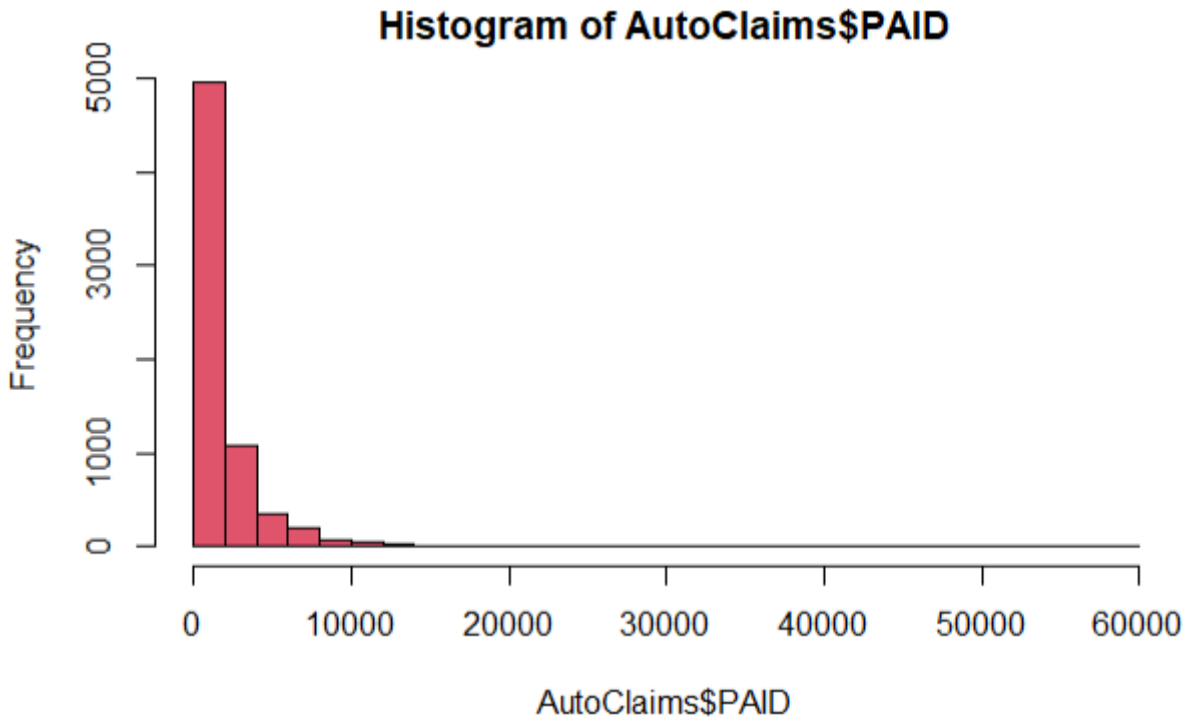



Figure 3: Histogram of auto claims dataset

```r
summary(AutoClaims$PAID)
sd(AutoClaims$PAID)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    9.5   523.7  1001.7  1853.0  2137.4 60000.0
[1] 2646.909
```

Figure 4: Summary of auto claims dataset

As we can see, the data is asymmetric and skewed-to-the-right or positively skewed (the mean is indeed larger than the median). Also, the standard deviation is quite large, compared to the mean, indicating a sensible variability.

With a simple exponential QQ-plot, we can try to understand if heavy tails are present or not. Given the things we have just seen, we would say yes. But let us verify.

The function ‘qplot‘ in the ‘evir‘ package allows us easily have the plot. The function is built on a GPD, hence the exponential is easily obtained by setting the parameter xi to 0.

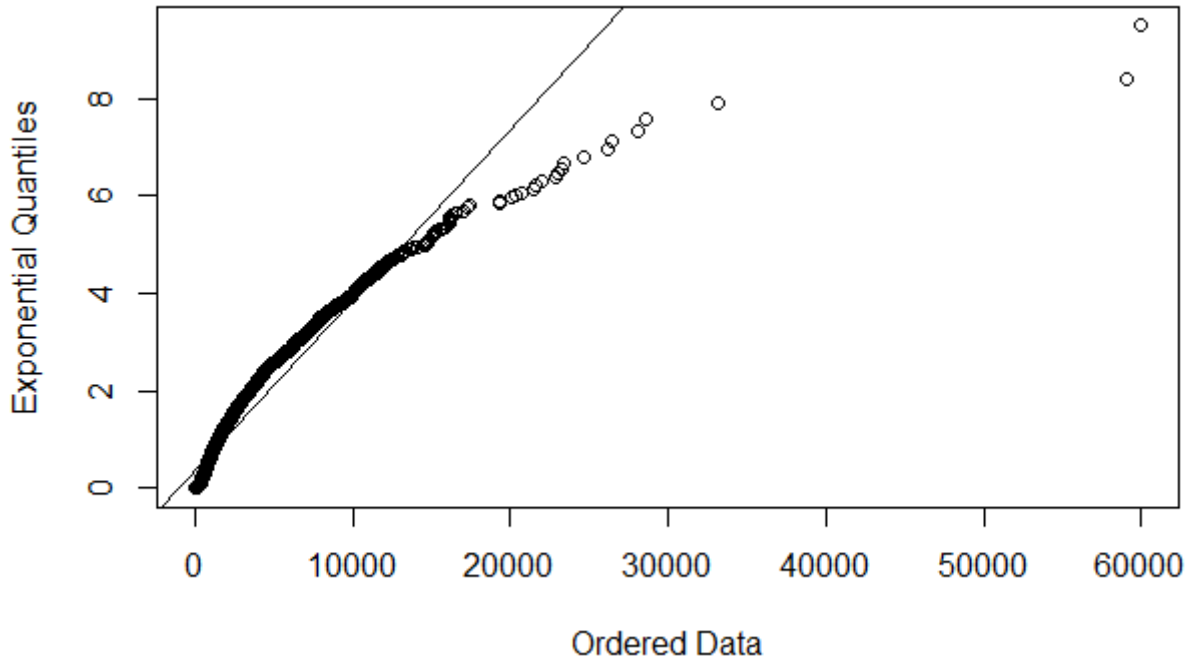



Figure 5: qplot of auto claims dataset

The clear concavity in the plot is a strong signal of the presence of heavy tails.

Now we will plot Zipf plot to look for the behavior of the survival function. It can be made using function ‘emplot‘ (empirical plot), with option "xy" to have a log-log representation.

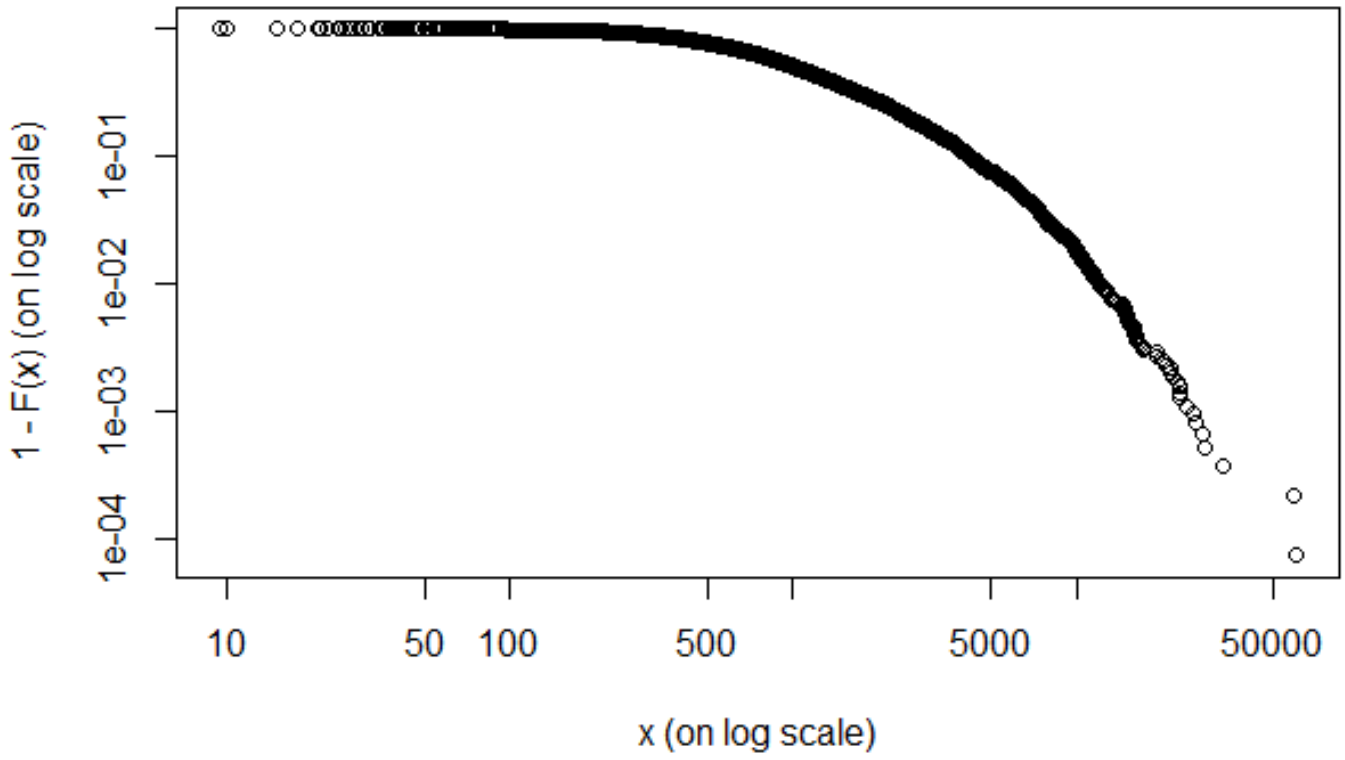



Figure 6: Zipf plot of auto claims dataset

We get a clear negative linear slope. This is a first signal of the fat tailed nature of the data. But,

a Zipf plot verifies a necessary yet not sufficient condition!

Looking at the range of the plot, the credibility of the plot seems okay. Given that linearity appears from the very beginning, we can think that our Danish claims actually follow a pure power law.

But a Zipf plot is not enough. We will now consider a Meplot, using the homonymous function ‘meplot‘.

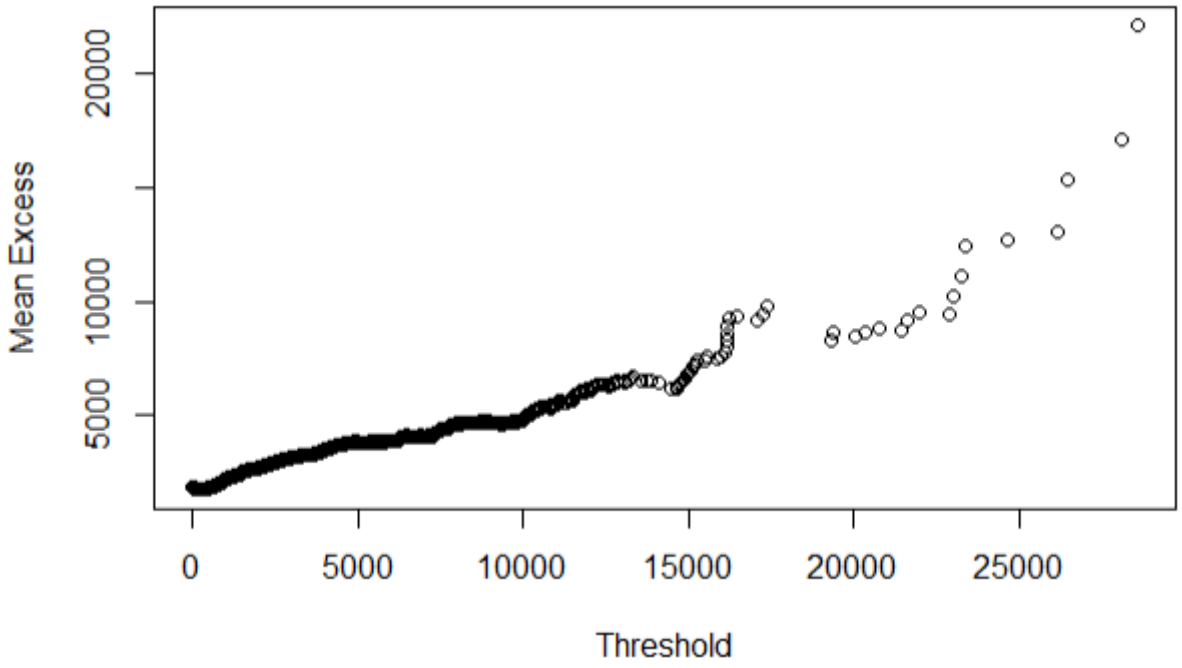



Figure 7: meplot plot of auto claims dataset

The plot is consistent with van der Wijk’s law. Another signal of the presence of a fat tail.

A concentration profile (CP) is another reliable tool to better understand the tail.

To build a CP, we can use the functions in the ‘ineq‘ library.

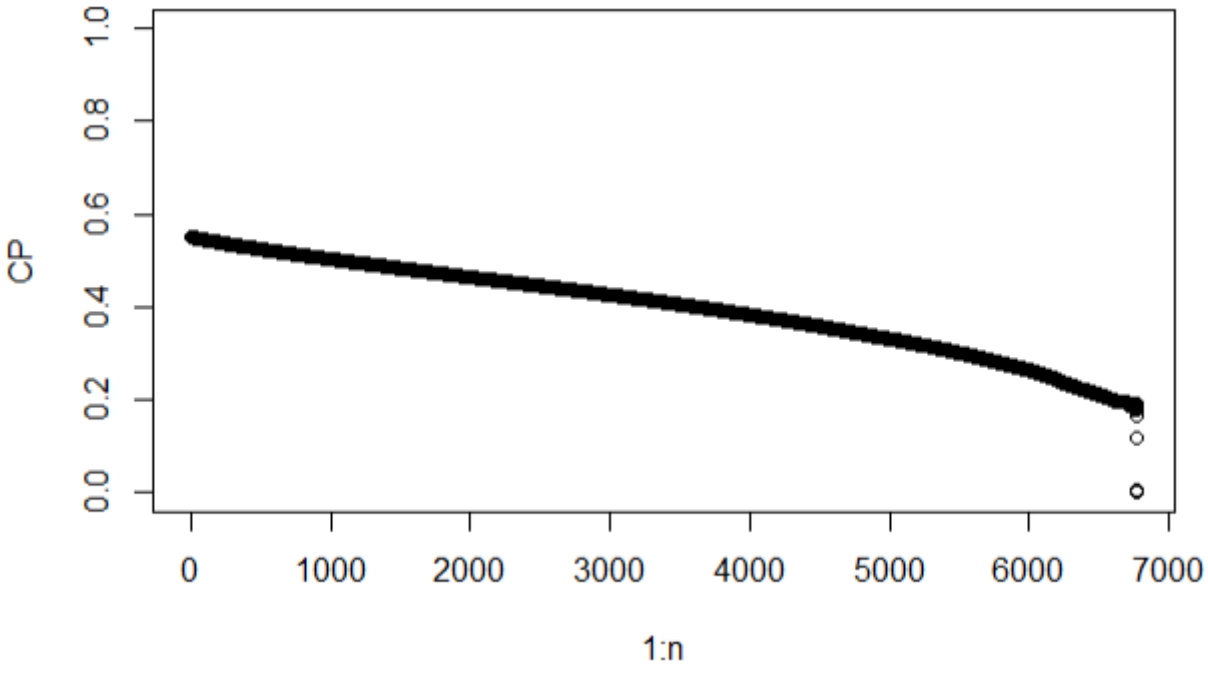



Figure 8: Concentration profile of auto claims dataset

The nearly horizontal behavior we observe (the last part of the plot is to be ignored for the limited amount of points considered) can be seen as a further signal of Paretianity.

Now we will analyse moments using the MS plot. We check for the first 4 moments.

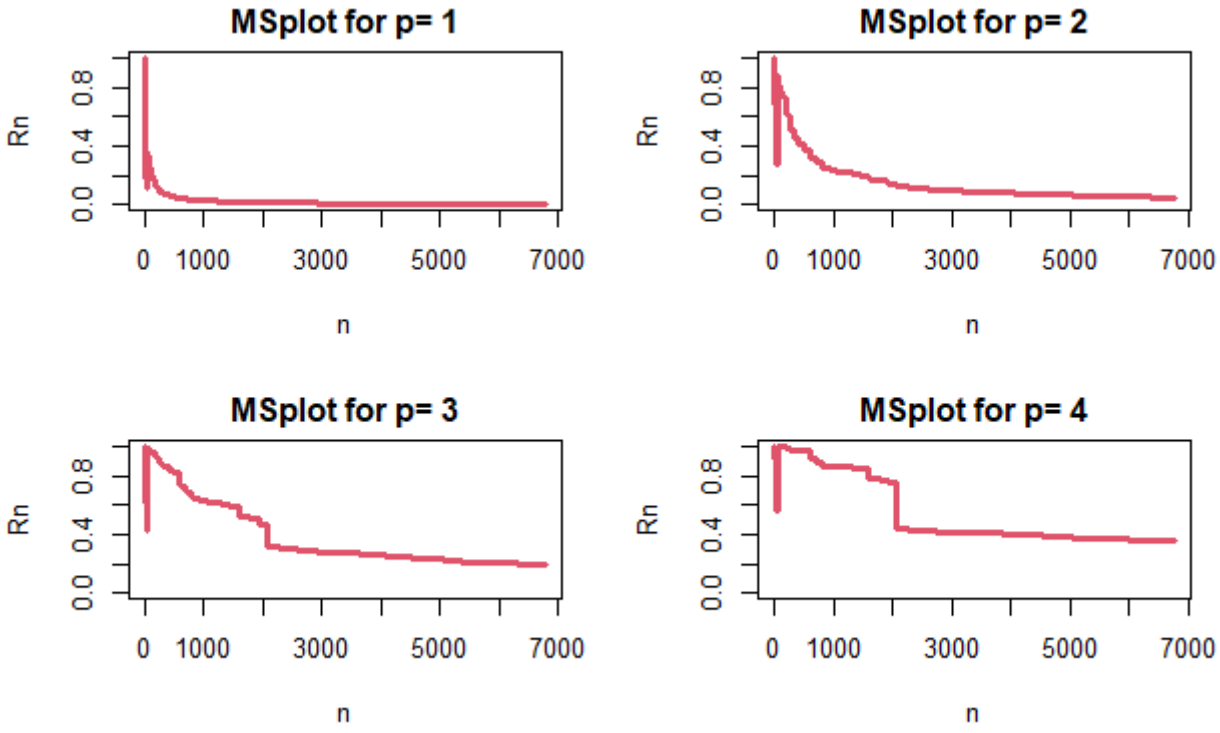



Figure 9: MSplot of auto claims dataset

We can see that for the first moment convergence is clear, while for the others (starting from the second) we can suspect that they are not defined. If confirmed, a similar finding would tell us that the standard deviation we have computed above is useless for inference.

A Hill plot is an excellent way to give extra substance to what we have just said about the moments.

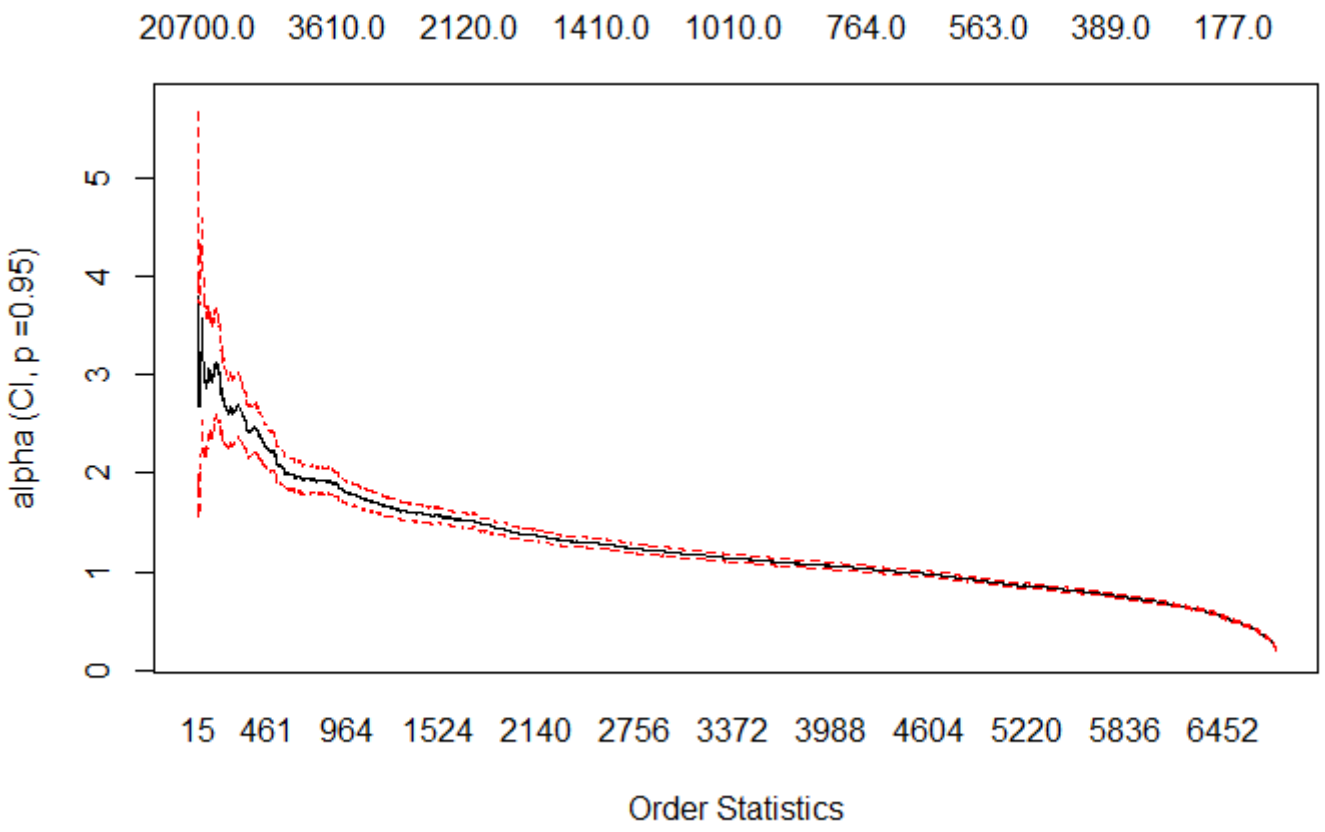



Figure 10: Hill plot of auto claims dataset

This is indeed the case. A value of alpha around 3 is highly plausible looking at the plot. The stability seems to kick in around a threshold of 3600 (look at the numbers on top). In terms if we expect xi to be 0.2 or so. In EVT, the suggestion is not to waste your time too much with second or third decimals. The first one is more than enough.

The 3600 threshold seems compatible with both the Zipf plot and the meplot. About 10% of all the claims lie above the threshold.

Now, let us fit a GPD above such a threshold. If the fit is reliable, the tail parameter should be stable under higher thresholds.

```r
fit=gpd(AutoClaims$PAID,3600)
tail(fit)
```

```
$par.ests
          xi         beta
   0.2398543 2493.8746673

$par.ses
          xi         beta
  0.04300746 134.62072069

$varcov
              [,1]          [,2]
[1,]  0.001849642    -3.761911
[2,] -3.761911301 18122.738439

$information
[1] "observed"

$converged
[1] 0

$nllh.final
[1] 7965.026
```

Figure 11: GPD fit of auto claims dataset

We get a xi=0.2 which is significant .For beta (or sigma in other parametrizations) we have 2493, which is also significant.

Now, let us verify the fitting of the tail. Using 'plot(fit)' and choosing options 1 to 4 from interactive menu, we show the following plots.

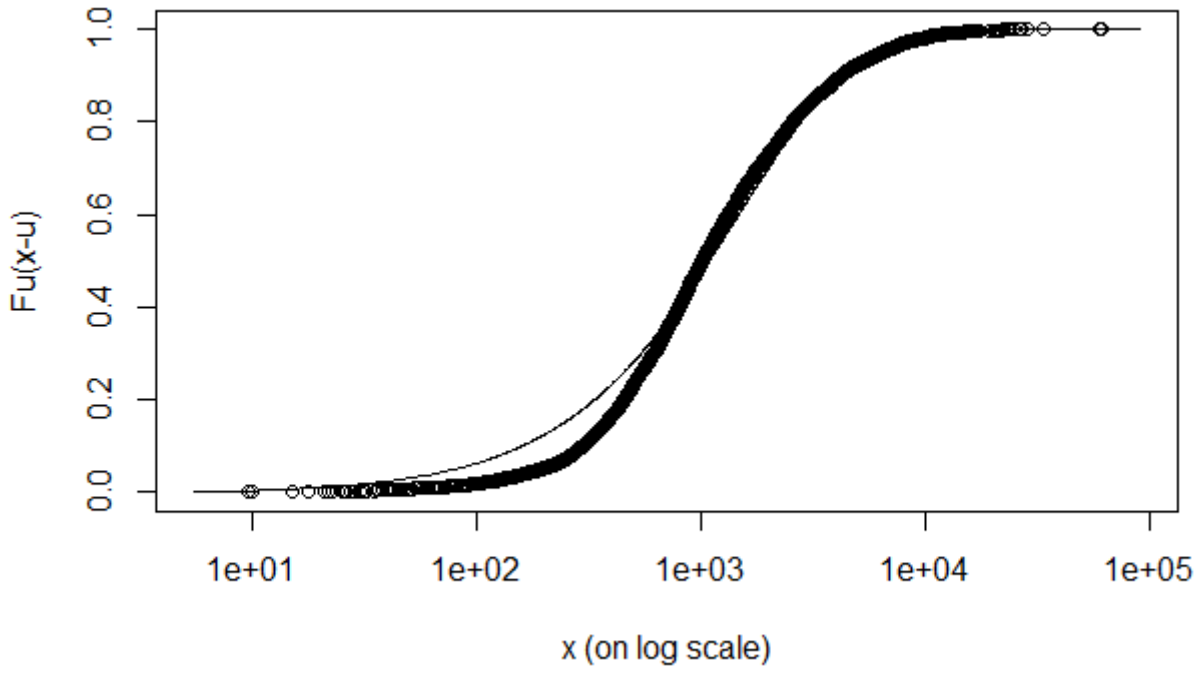



Figure 12: Excess Distribution

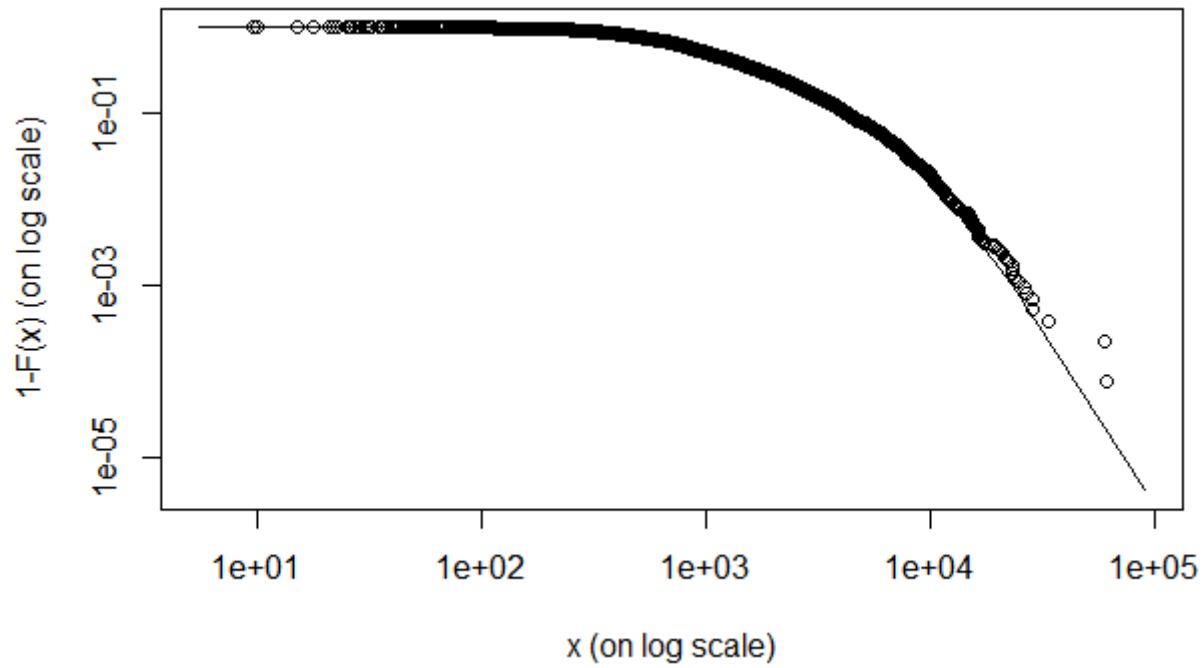



Figure 13: Tail of underlying distribution

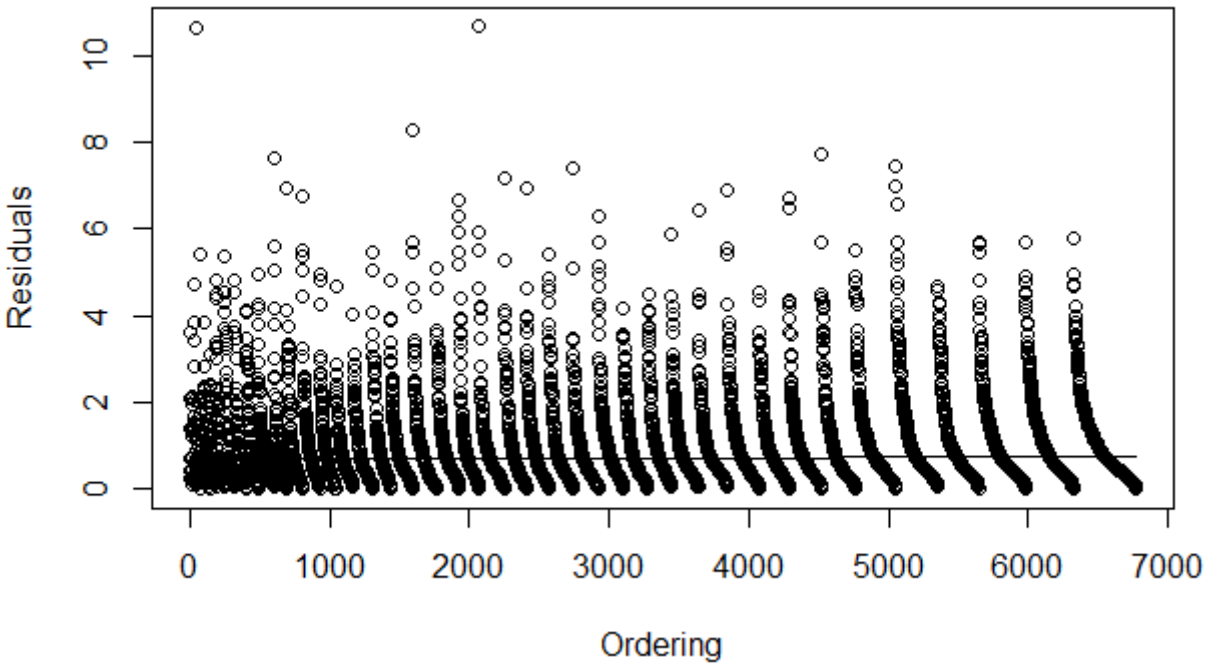



Figure 14: Scatterplot of residuals

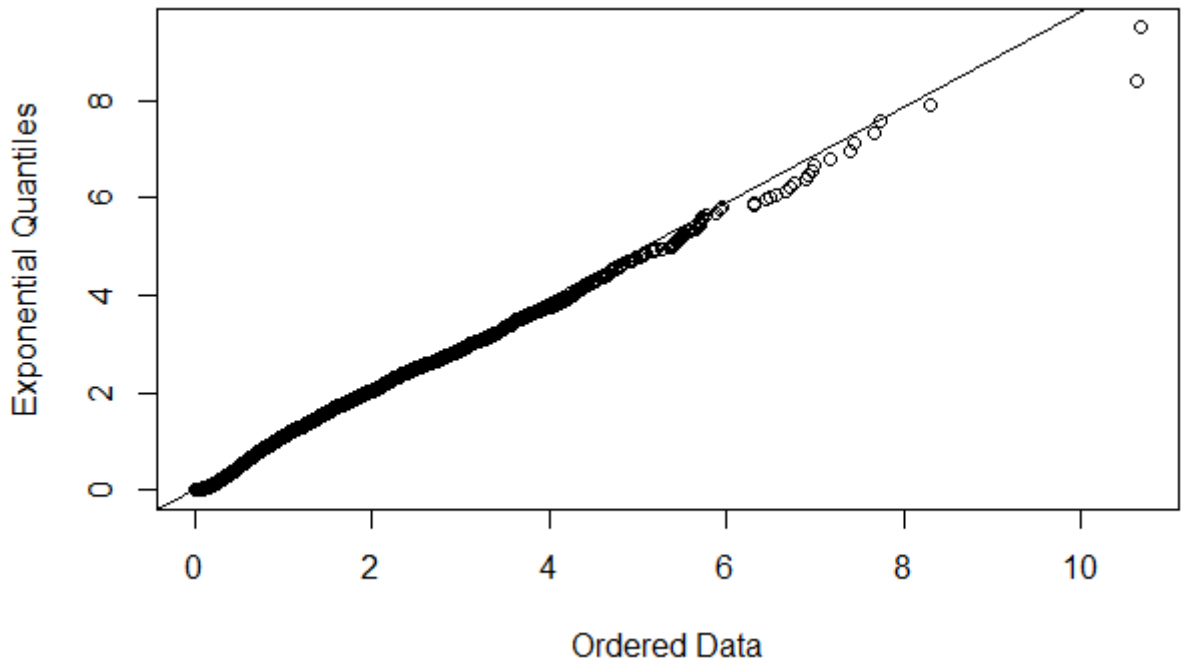



Figure 15: QQplot of residuals

The fitting is quite satisfactory.

We check if a higher threshold would change the value of xi.

```r
gpd(AutoClaims$PAID,20)$par.ests[1]
gpd(AutoClaims$PAID,20)$par.ses[1]
```

```
       xi
0.2244576
       xi
0.01341979
```

Figure 16: GPD fit for auto claims dataset with higher threshold

Qualitatively we would say no, a higher threshold preserves the xi. A value of 0.2 or so seems plausible and in line with our previous findings. Notice that it is confirmed that the second moment is not finite!

Given the GPD fit, we could be interested in estimating a very high quantile (VaR) and the corresponding ES. This approach is much more reliable than using empirical estimates, especially under fat tails.

We can rely on a useful function in the 'evir' package. The function requires a GPD fit in its arguments. We will start from a 99% confidence level.

```{r}
riskmeasures(fit,0.99)
quantile(AutoClaims$PAID,0.99) #99% Var
mean(AutoClaims$PAID[AutoClaims$PAID>=quantile(AutoClaims$PAID,0.99)]) #99% ES
```

```
        p quantile    sfall
[1,] 0.99 11328.45 16272.72
      99%
12052.29
[1] 18172.93
```

Figure 17: 99% confidence interval

While the VaR is comparable, the empirical ES seems to underestimate the tail risk.
Let us consider the so-called worst-case scenario, i.e. quantities at the 99.9% confidence level.

```{r}
riskmeasures(fit,0.999)
quantile(AutoClaims$PAID,0.999) #99.9% Var
mean(AutoClaims$PAID[AutoClaims$PAID>=quantile(AutoClaims$PAID,0.999)]) #99.9% ES
```

```
         p quantile   sfall
[1,] 0.999 26605.82 37145.8
    99.9%
24971.39
[1] 37359.2
```

Figure 18: 99.9% confidence interval

Notice that the empirical quantities, ignoring EVT, would make us underestimate the tail risk even more. In this case also the empirical VaR is less reliable. We will now see for 99.99. We are really zooming into the tail here. Empirically it is like we are considering less than 1 observation in the sample!

```{r}
riskmeasures(fit,0.9999)
quantile(AutoClaims$PAID,0.9999) #99.99% Var
mean(AutoClaims$PAID[AutoClaims$PAID>=quantile(AutoClaims$PAID,0.9999)]) #99.99% ES
```

```
          p quantile    sfall
[1,] 0.9999 51231.32 69541.56
   99.99%
59399.85
[1] 60000
```

Figure 19: 99.99% confidence interval

Hence, EVT definitely wins.

## Experiment 2 : Modelling auto-claims using Block Maximization method with MLE

In this experiment, we are using following packages:

1. ggplot2 : for creating graphics
2. evd : Provides functions for extreme value theory
3. evir : Provides functions for extreme value theory
4. nortest : Tests for Normality
5. insuranceData : Insurance datasets, which are often used in claims severity and claims frequency modelling
6. extRemes : Functions for performing extreme value analysis
7. VaRES : Computes Value at risk and expected shortfall, two most popular measures of financial risk
8. vars : VAR Modelling
9. PerformanceAnalytics : Collection of econometric functions for performance and risk analysis.

We are using "AutoClaims" dataset. It contains claims experience from a large midwestern (US) property and casualty insurer for private passenger automobile insurance. The dependent variable is the amount paid on a closed claim, in (US) dollars (claims that were not closed by year end are handled separately). We will first look at the data and at some basic statistics.

```r
data(AutoClaims)
dataset <- AutoClaims$PAID
summary(dataset)
```

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     9.5   523.7  1001.7  1853.0  2137.4 60000.0
```

```r
hist(dataset)
```

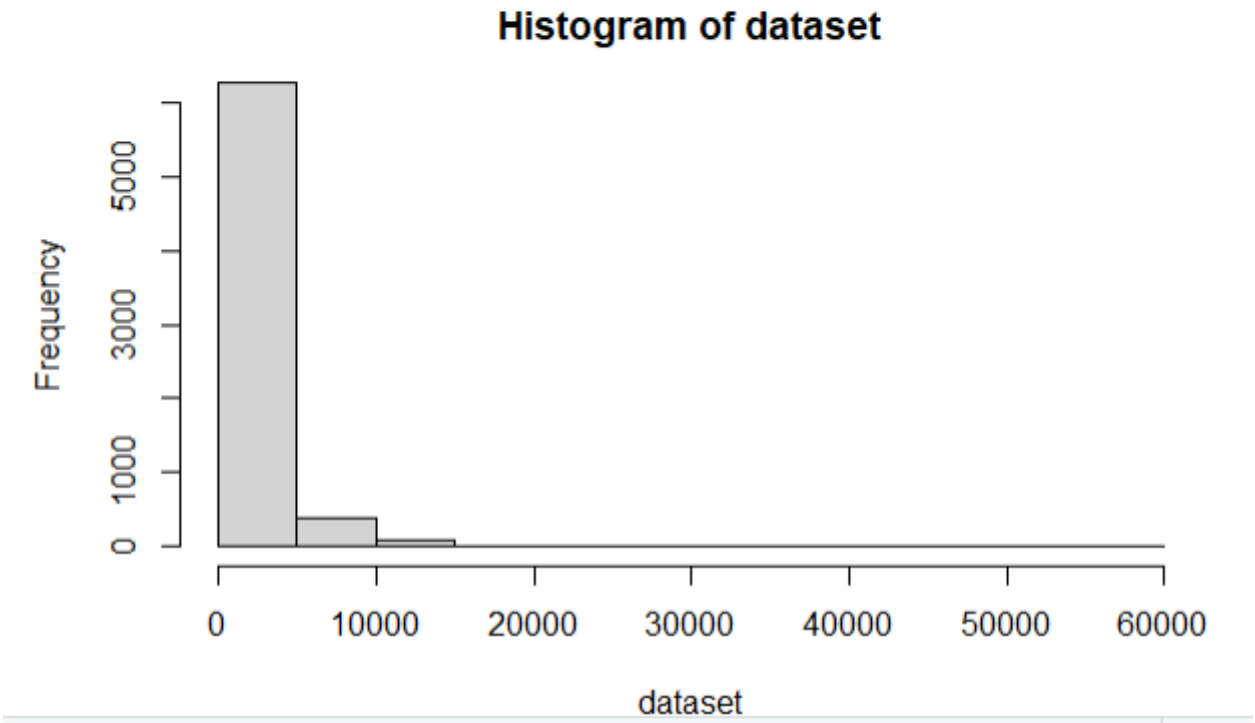



Figure 20: Basic statistics of dataset

Now, we will find the block maximum of the data set using "blockmaxxer" function. We take blen = 400 and span = 16. The histogram of blocks looks as follows:

Block Maxima

```r
blocks <- blockmaxxer.data.frame(data.frame(dataset), blen=400, span=16)
blocks
hist(blocks)
```

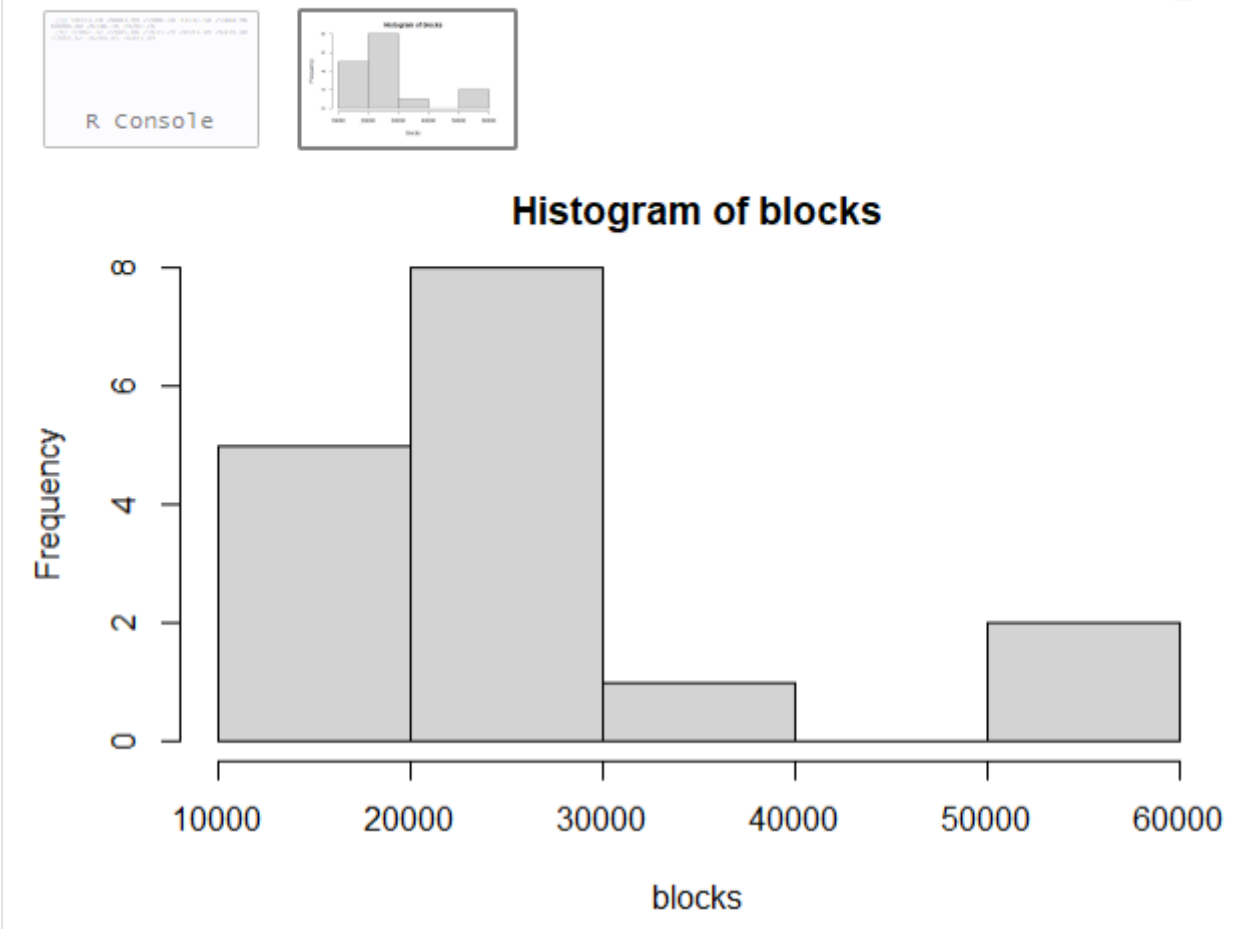



Figure 21: Histogram of blocks

We now perform the Anderson-Darling test for normality to detect all departures from normality.

```r
ad.test(blocks)
```

```
        Anderson-Darling normality test

data:  blocks
A = 2.8881, p-value = 2.434e-07
```

Figure 22: Anderson-Darling test for normality

Now we apply "gev" function to get maximum likelihood estimation of the 3-parameter generalized extreme value (GEV) distribution. Then we get the maximum-likelihood fitting for the generalized extreme value distribution, including linear modelling of the location parameter using "fgev" function, allowing any of the parameters to be held fixed if desired.

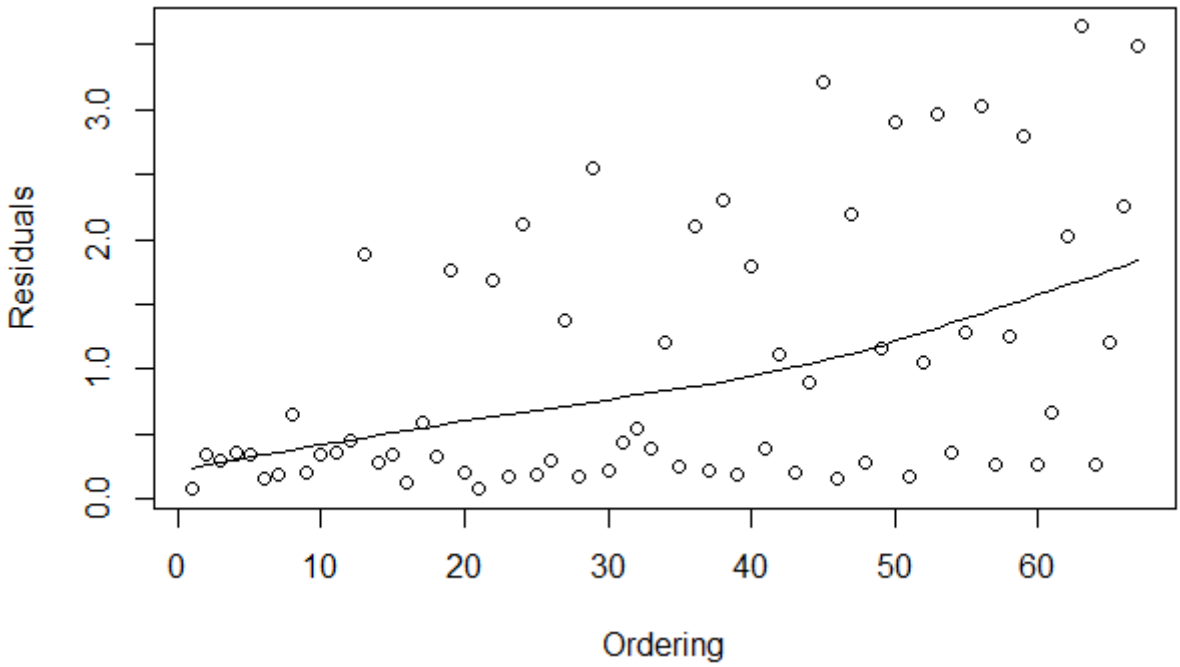



Figure 23: Scatterplot of residuals

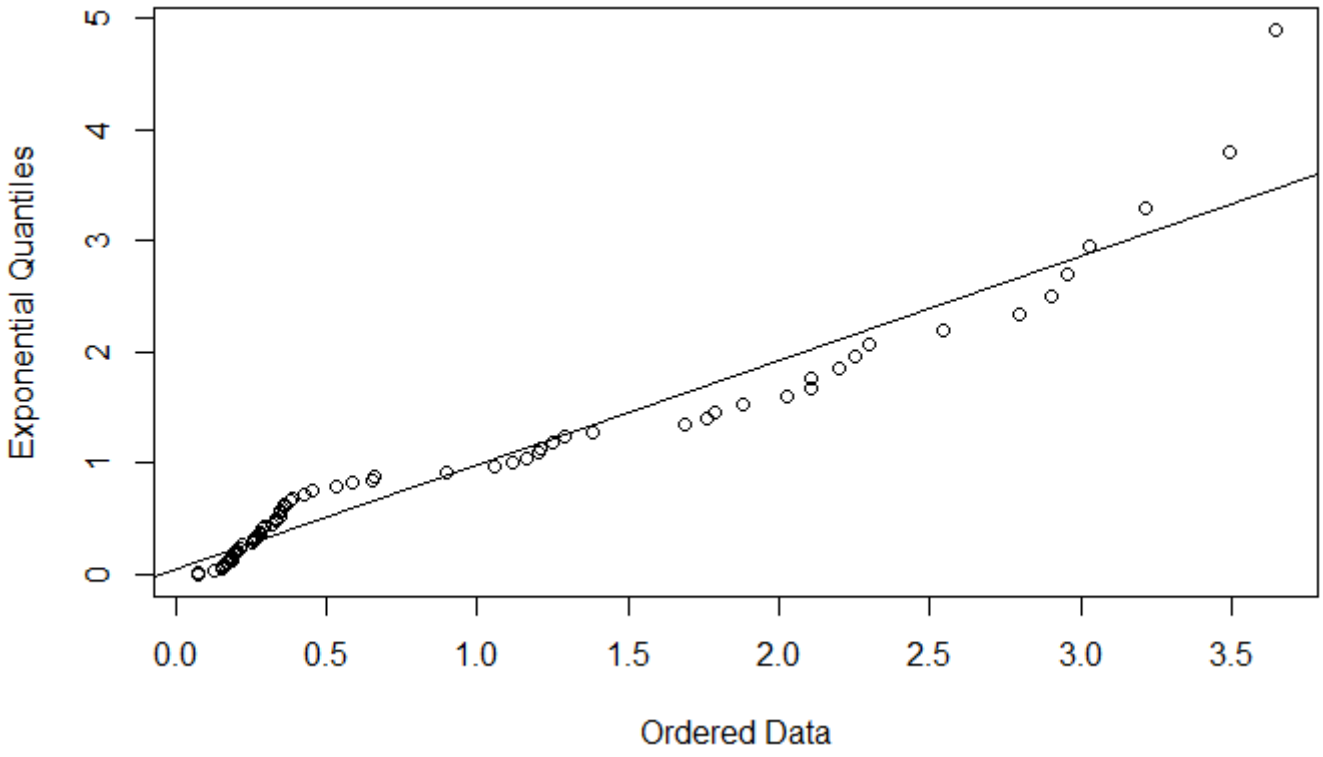



Figure 24: qqplot of residuals

```
Call: fgev(x = GEV$data, std.err = FALSE)
Deviance: 1416.823

Estimates
      loc      scale      shape
6585.520  9473.857      1.405
```

Figure 25: fgev

Since shape parameter > 0, the data follows a Fréchet distribution. We plot the Density Function of AutoClaims and Density Function of Fitted Frechet. We use "na.omit" to remove NaN values.

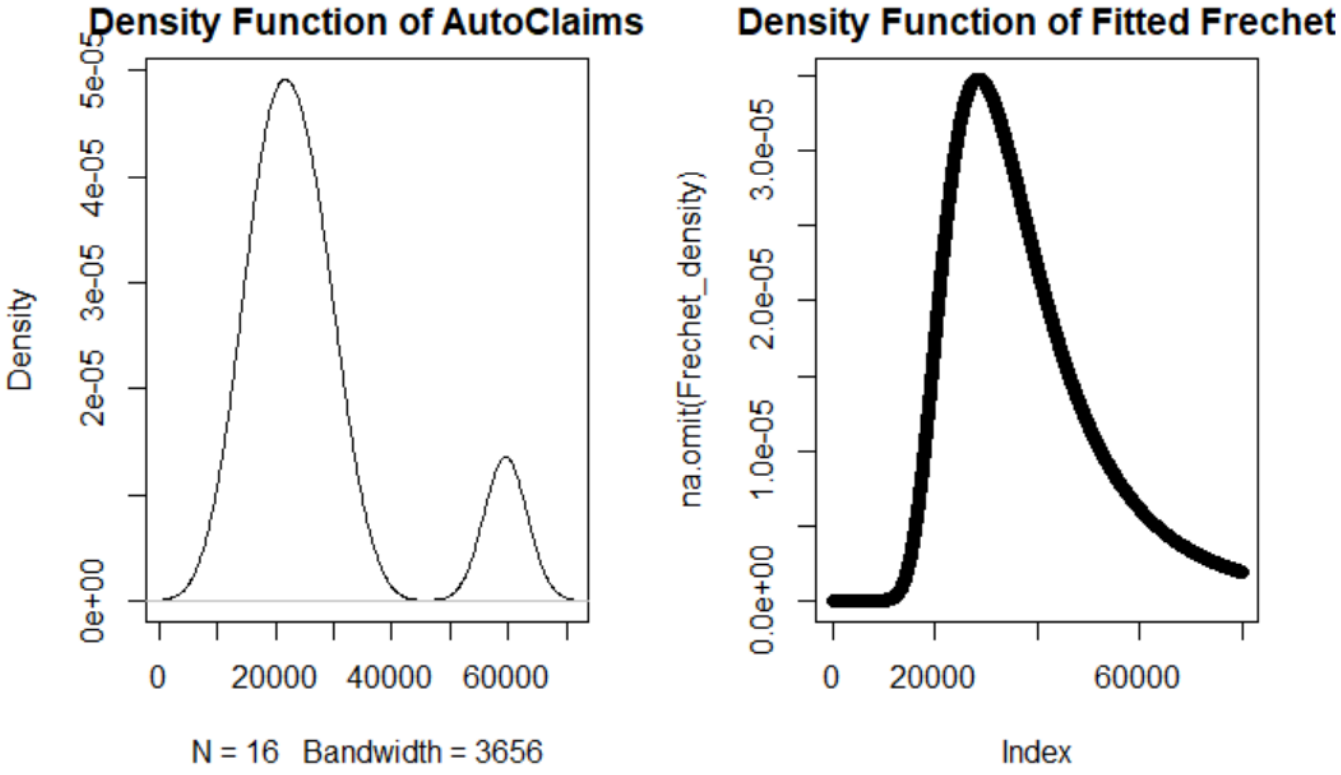



Figure 26: (i) Density Function of AutoClaims (ii) Density Function of Fitted Frechet

Finally, we calculate value at risk and expected shortfall for the generalized extreme value distribution due to Fisher and Tippett (1928), using "vargev" and "esgev" respectively and then compare.

```
75
76  Calculating the Expected Shortfall
77  ```{r}
78  p = 0.99
79  esgev(p, mu, sigma, epsilon)
80  ```

    [1] 28705.21

81  ```{r}
82  mean(blocks[blocks>=quantile(dataset,0.99)])
83  ```

    [1] 27898.64
```

Figure 27: Expected Shortfall

The expected value of ES is 27898.64 and the calculated value of ES is 28705.21, they agree reasonably well.

### Experiment 3 : Modelling auto-claims using Block Maximization method with Probability Weighted Moments

We will now model auto-claims dataset using Block Maximization method with Probability Weighted Moments, similar to experiment 2.

Packages used in addition to those used in experiment 2:

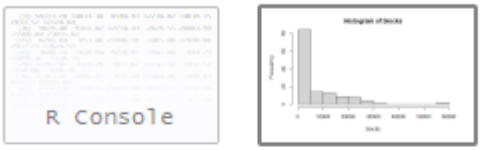
1. fExtremes : Provides functions for analysing and modelling extreme events in financial time series
2. RobExtremes : Optimally robust estimation for extreme value distributions

The basic statistics of the dataset have already been mentioned in the previous experiment. Now, we will find the block maximum of the data set using "blockmaxxer" function.

We take blen = 50 and span = 135. The histogram of blocks looks as follows:

Block Maxima

```r
blocks <- blockmaxxer.data.frame(data.frame(dataset), blen=50, span=135)
blocks
hist(blocks)
```



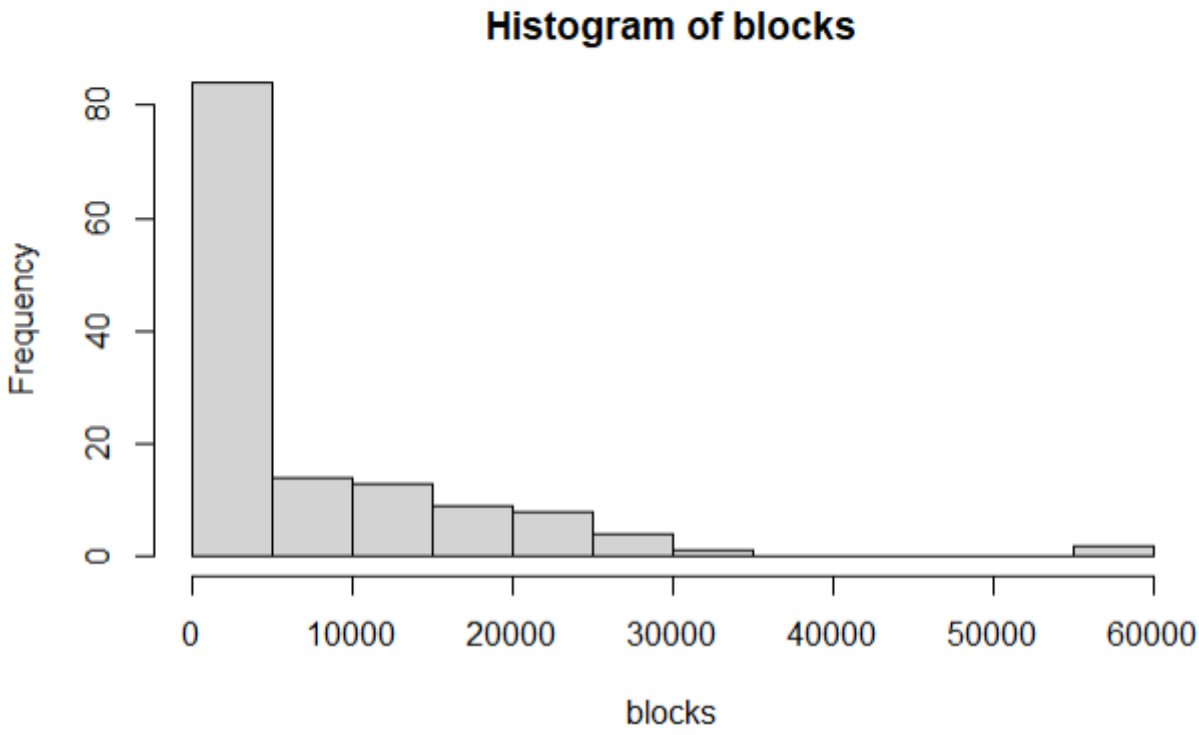




Figure 28: Histogram of blocks

To detect all departures from normality, we now perform the Anderson-Darling test for normality .

Anderson-Darling Test of robustness of the data

```r
ad.test(blocks)
```

```
        Anderson-Darling normality test

data:  blocks
A = 12.275, p-value < 2.2e-16
```

Figure 29: Anderson-Darling test for normality

We use "gevFit" to estimate the parameters either by the probability weighted moment method("pwm") or by maximum log likelihood estimation ("mle"). We have used PWM method here.

```
        xi
0.04568545

Title:
 GEV Parameter Estimation

Call:
 gevFit(x = dataset, block = 200, type = c("pwm"))

Estimation Type:
  gev pwm

Estimated Parameters:
           xi            mu          beta
4.568545e-02 1.355857e+04 9.167579e+03

Description
  Wed Nov 25 16:20:53 2020
```

Figure 30: Summary of fitted GEV

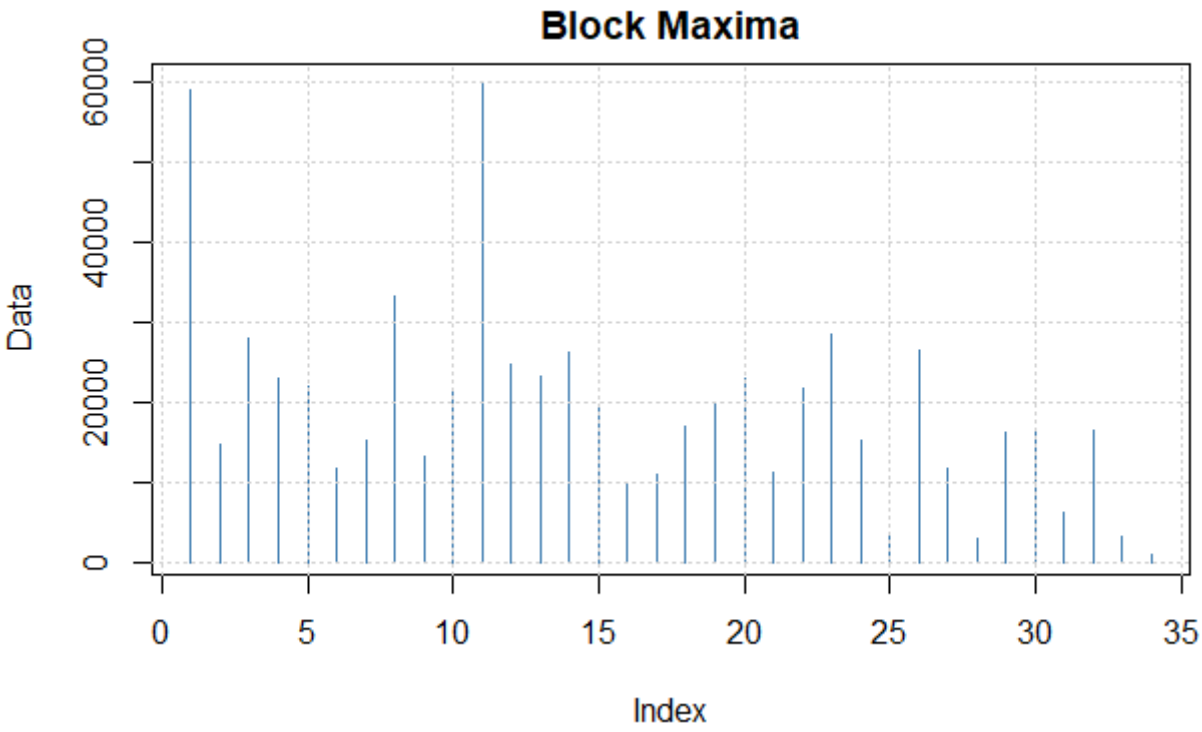



Figure 31: Block Maxima

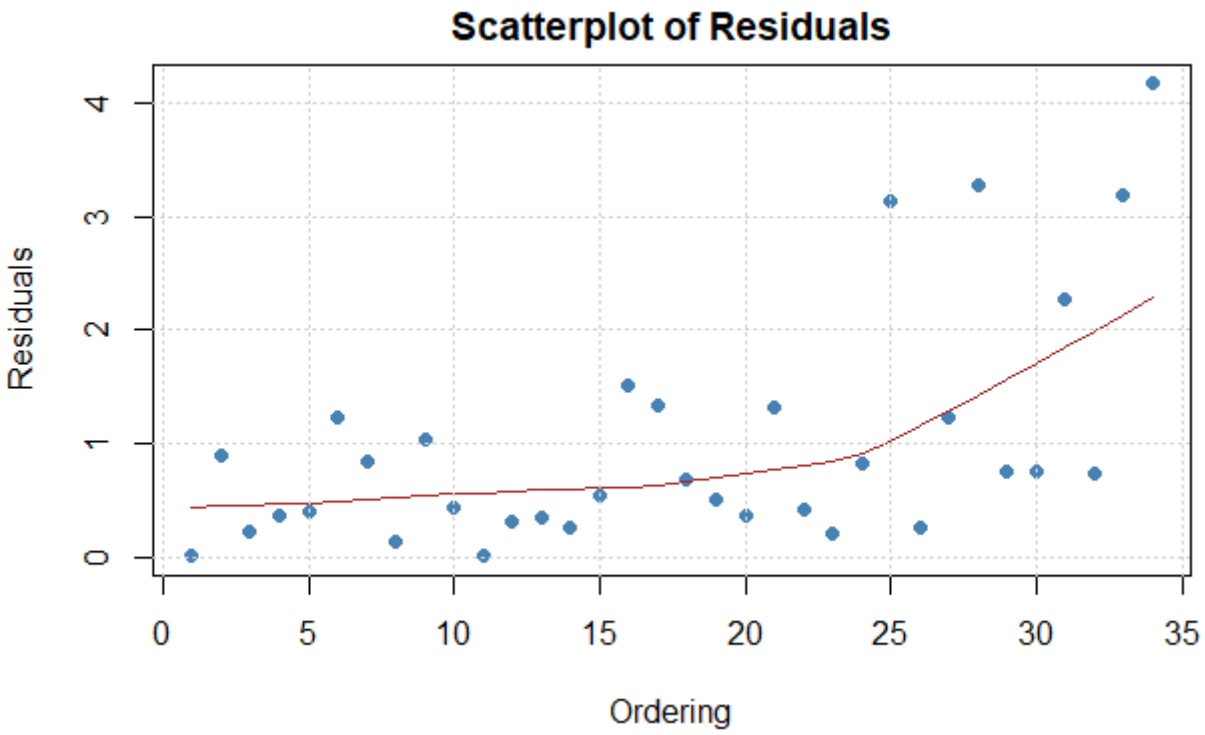



Figure 32: Scatterplot of residuals

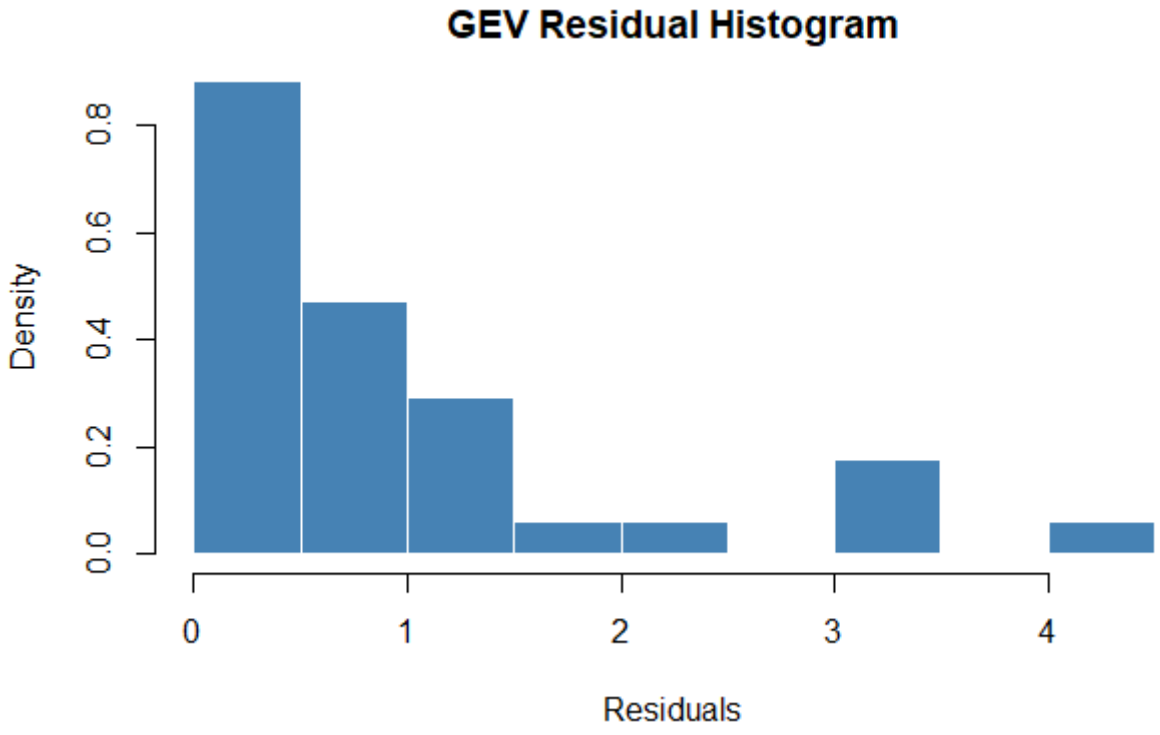



Figure 33: GEV Residual Histogram

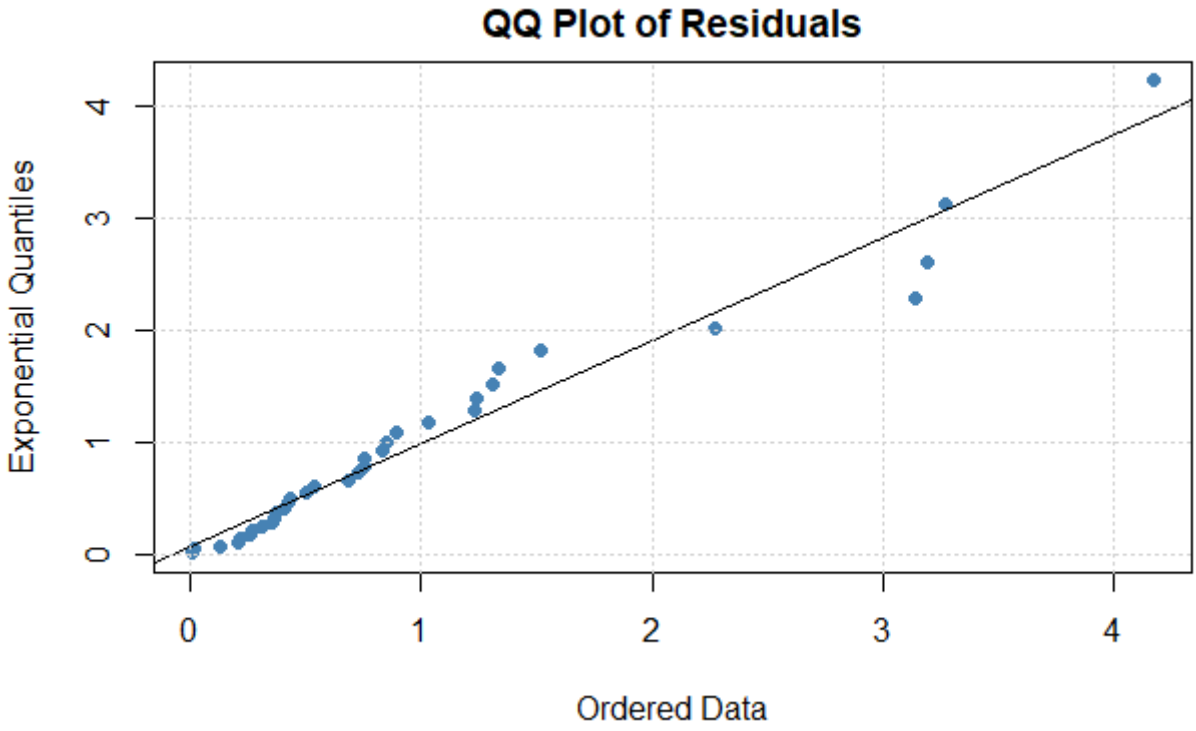



Figure 34: QQ plot of residuals

Since shape parameter > 0, the data follows a Fréchet distribution. We plot the Density Function of AutoClaims and Density Function of Fitted Frechet. We use "na.omit" to remove NaN values.

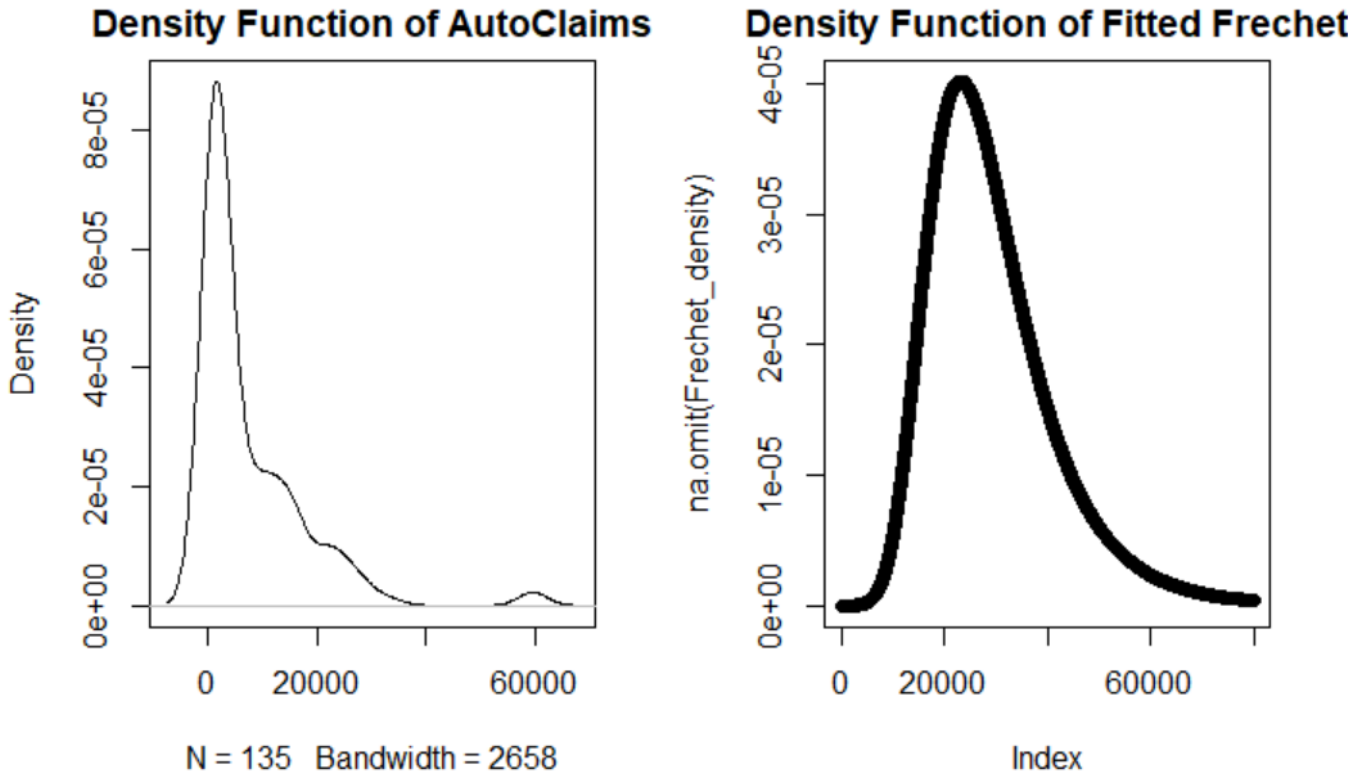



Figure 35: (i) Density Function of AutoClaims (ii) Density Function of Fitted Frechet

Finally, we calculate value at risk and expected shortfall for the generalized extreme value distribution due to Fisher and Tippett (1928), using "vargev" and "esgev" respectively and then compare.


Figure 36: Expected Shortfall

The expected value of ES is 21420.95 and the calculated value of ES is 18746.47, they agree reasonably well but not as well as in the case of MLE. MLE is, thus, a better method of estimation.

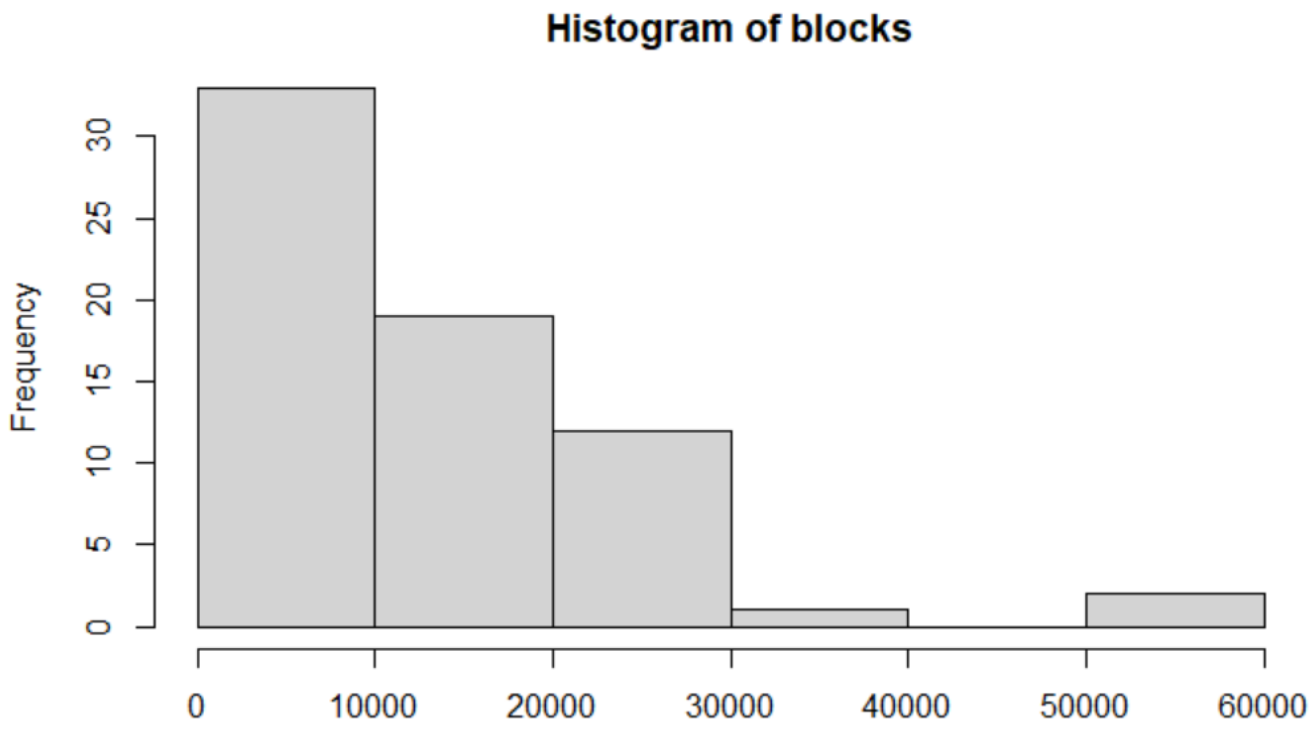
### 10.1 Estimation of GEV using MLE Method

While using the Block Maxima method, the key decision is the size of the blocks. We indeed saw this in practice. Before arriving at satisfactory fits and results for block size of 400, we had to try a number of different combinations of the block size. Here, we have added 2 cases of failures to demonstrate the catastrophic effects of changing the block size. We take two cases, one where we take a smaller block size than the successful block size, that is, 100, and one where we take a larger block size than the successful block size, that is, 600. Note that span has been adjusted accordingly because the size of the dataset is constant at 6773.

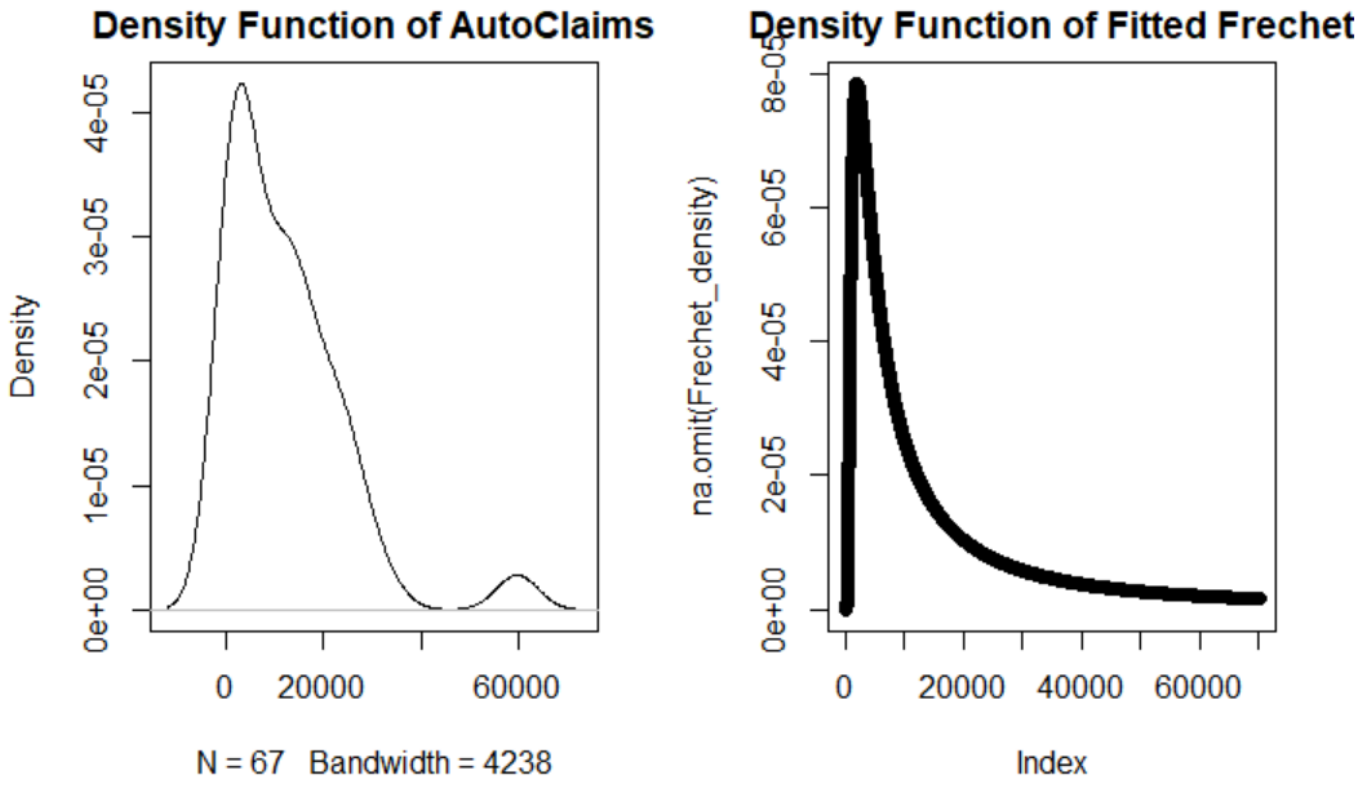
**Case I: blen= 100**

Block Maxima

```r
blocks <- blockmaxxer.data.frame(data.frame(dataset), blen=100, span=67)
blocks
hist(blocks)
```




The pictorial representation of the blocks doesn't tell us much. In fact, even the plots of the density functions agree quite well, as shown below.




This doesn't seem so bad. But this is misleading. We see that the calculated expected shortfall is far too removed from the actual expected shortfall value.

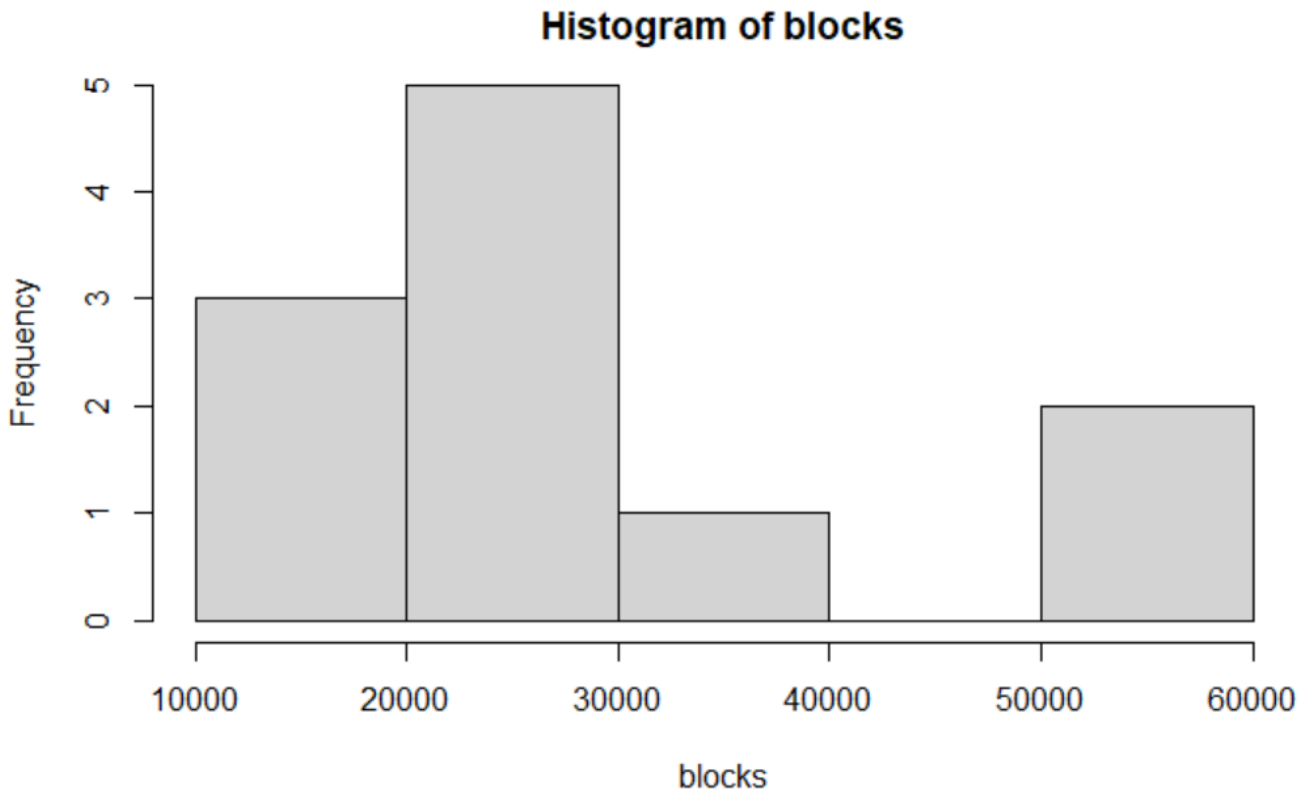
```
Calculating the Expected Shortfall
```{r}
p = 0.99
esgev(p, mu, sigma, epsilon)
```
[1] 83624.73
```{r}
mean(blocks[blocks>=quantile(dataset,0.99)])
```
[1] 22021.55
```

The actual ES value is 22021.55 while our fittedGEV predicts 83624.73 as the ES. We see that even though the fitting seems fine, compromise on the block lengths can have catastrophic effects. Now let's review the other side of the coin,

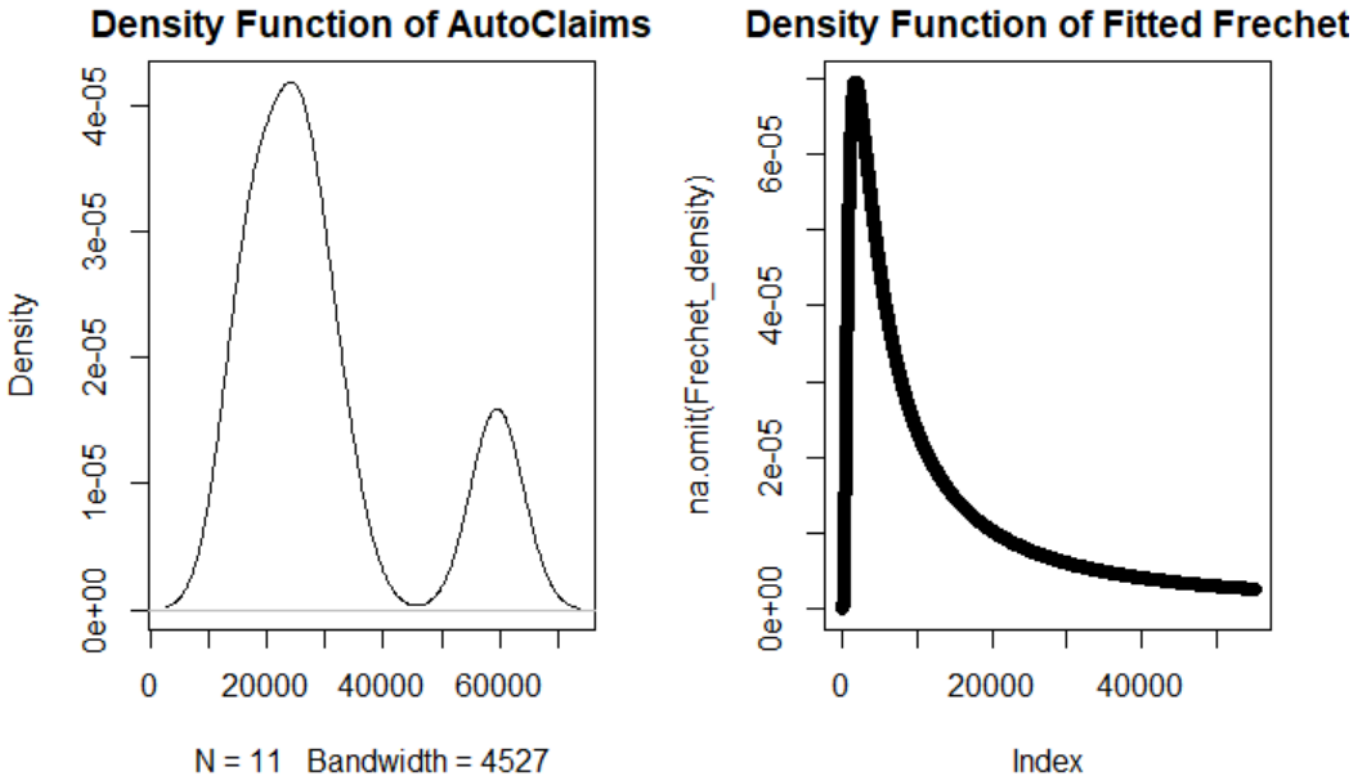
**Case II: blen= 600**

We see similar observations in this case as well.

```
Block Maxima
```{r}
blocks <- blockmaxxer.data.frame(data.frame(dataset), blen=600, span=11)
blocks
hist(blocks)
```
```




The pictorial representation of the blocks doesn't tell us much. In fact, even the plots of the density functions agree quite well, as shown below.




This doesn't seem so bad. But this is misleading. We see that the calculated expected shortfall is far too removed from the actual expected shortfall value.

```
Calculating the Expected Shortfall
```{r}
p = 0.99
esgev(p, mu, sigma, epsilon)
```
[1] 167555.5
```{r}
mean(blocks[blocks>=quantile(dataset,0.99)])
```
[1] 29918.29
```

The actual ES value is 29918.29 while our fittedGEV predicts 167555.5 as the ES. Once again we see that even though the fitting seems fine, compromise on the block lengths can have catastrophic effects.

### 10.2 Estimation of GEV using PWM Method

Since we are still using the Block Maxima method, the key decision is the size of the blocks. We indeed saw this in practice. Before arriving at the decision that block size of 50 is satisfactory, both in terms of results and computational time, we had to try a number of different combinations of the block size. Here, we have added 2 cases of failures to demonstrate our decision process. We take two cases, one where we take a smaller block size than the successful block size, that is, 10, and one where we

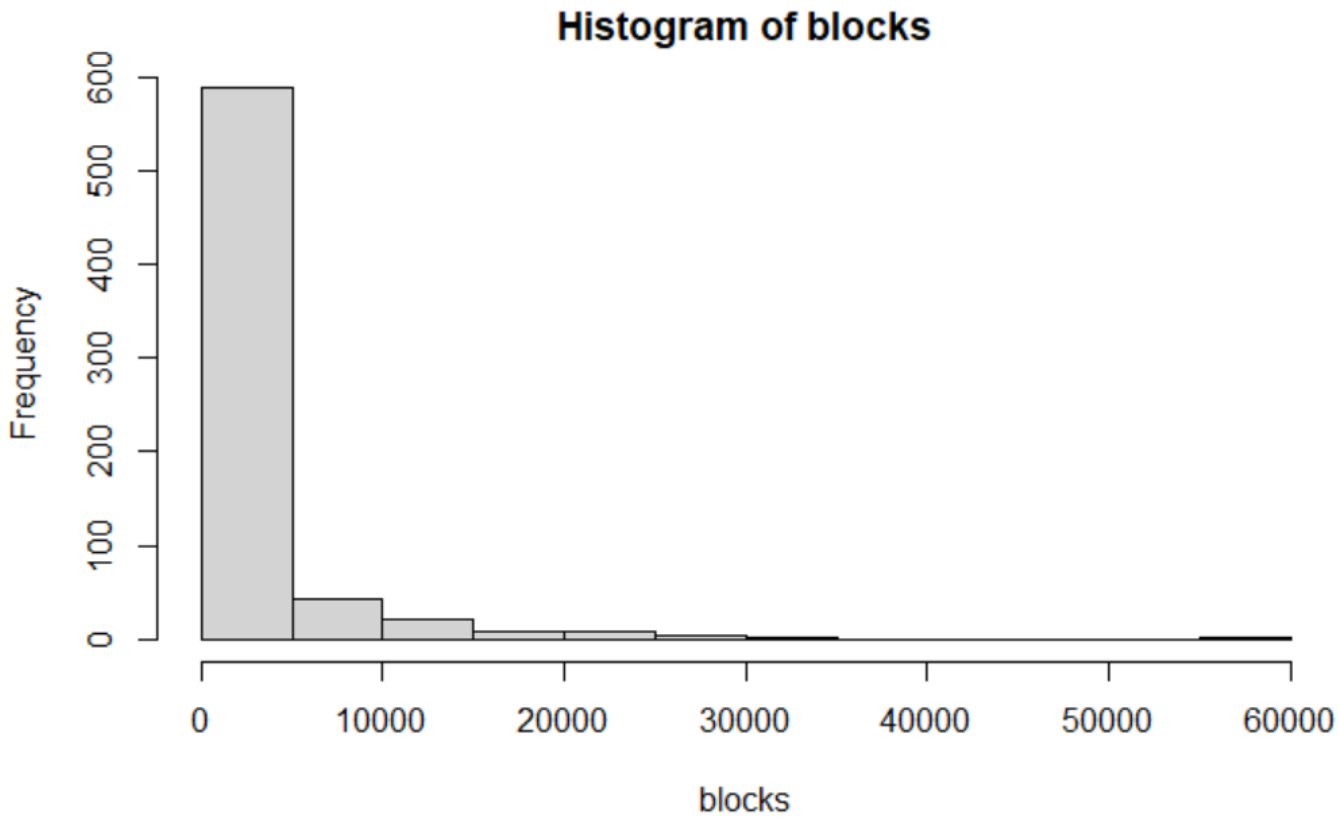
take a larger block size than the successful block size, that is, 200. Note that span has been adjusted accordingly because the size of the dataset is constant at 6773.
We will observe that the loss of accuracy is much lower in PWM method than the MLE method on changing the block size.

**Case I: blen= 10**

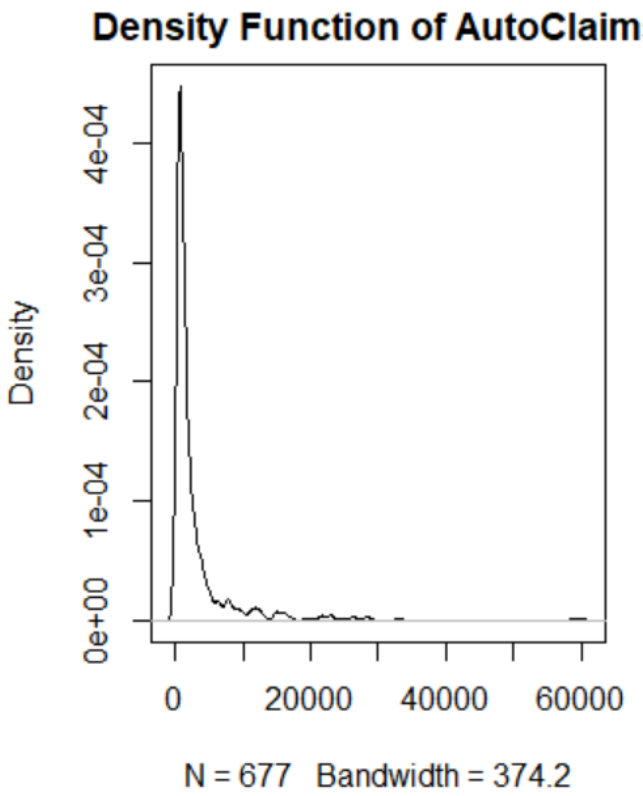
While trying different block sizes in PWM method, we noticed that the accuracy in calculation of ES kept improving as we decreased the block size, even when it was decreased to a value as low as 5.

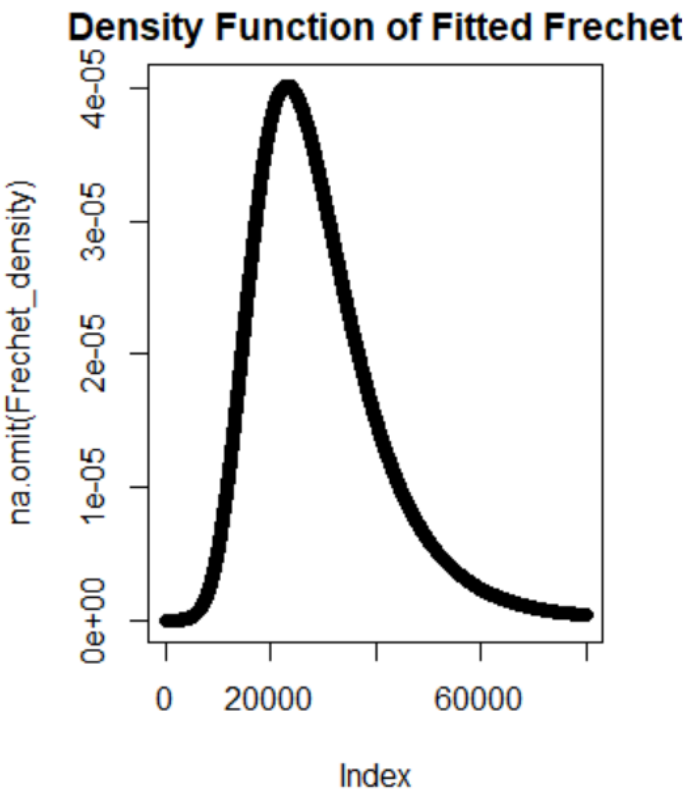
Block Maxima

```{r}
blocks <- blockmaxxer.data.frame(data.frame(dataset), blen=10, span=677)
blocks
hist(blocks)
```




The pictorial representation of the blocks doesn't tell us much.




We see that the fitting is slightly off. However, the expected shortfall in this case has better accuracy than when blen was 50.

```
Calculating the Expected Shortfall
```{r}
p = 0.99
esgev(p, mu, sigma, epsilon)
```
[1] 18746.47
```{r}
mean(blocks[blocks>=quantile(dataset,0.99)])
```
[1] 21160.99
```

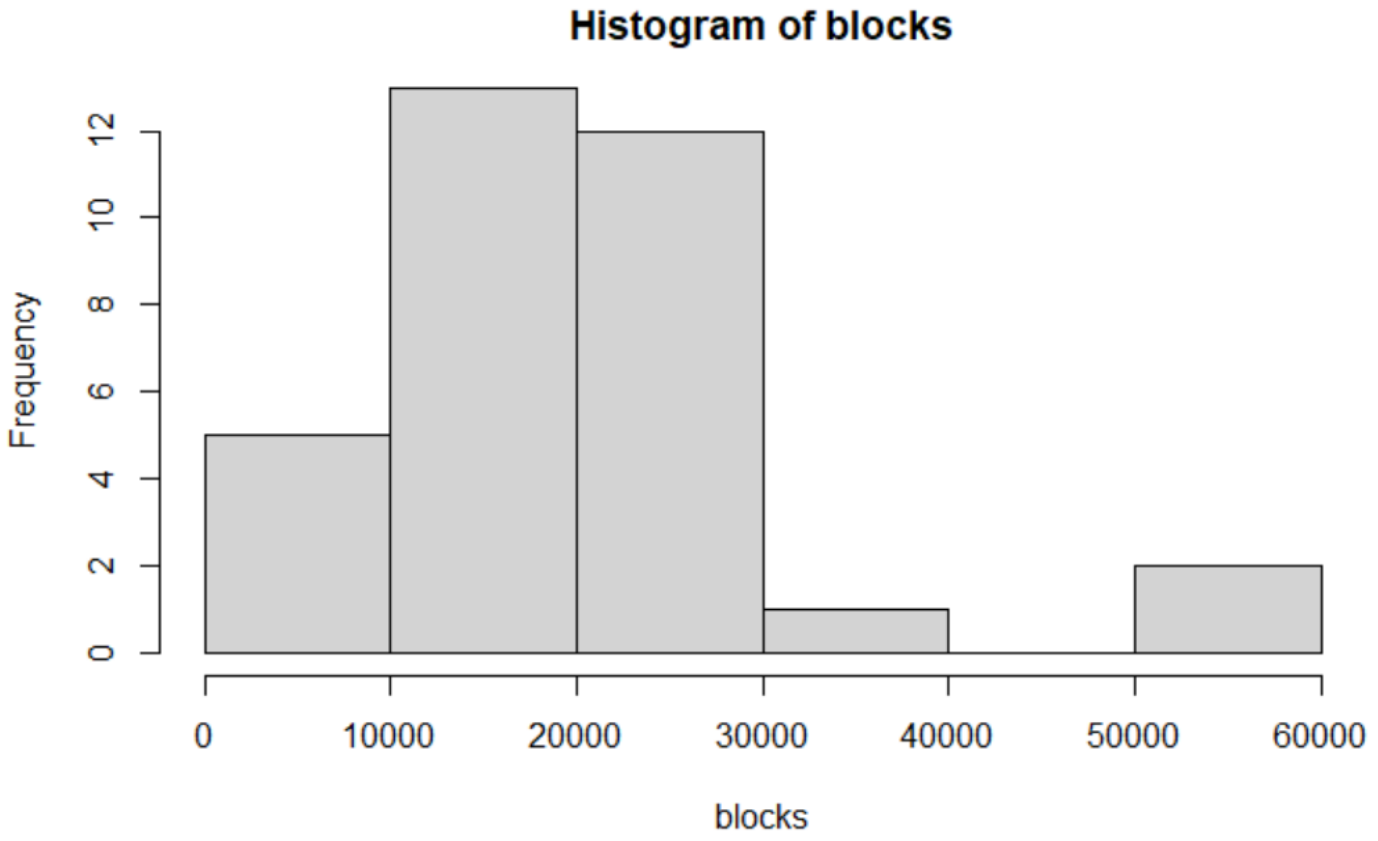
The actual ES value and the ES value that the fittedGEV predicts agree quite well, even better than when blen was 50. However the gap isn't significant enough to increase the computation time so much. Hence, we choose blen=50 as our value. Now let's review the other side of the coin,

**Case II: blen= 200**

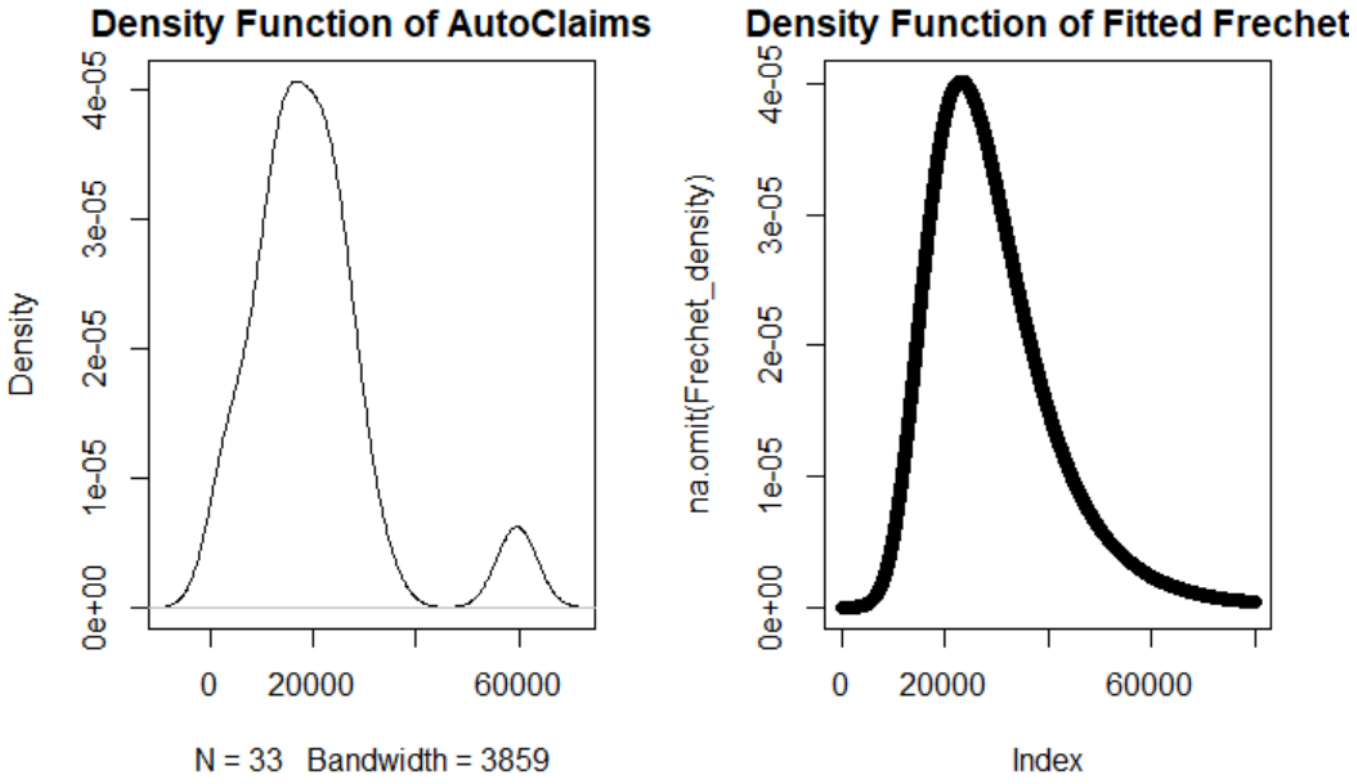
We see similar observations in this case as well.

Block Maxima

```{r}
blocks <- blockmaxxer.data.frame(data.frame(dataset), blen=200, span=33)
blocks
hist(blocks)
```




The pictorial representation of the blocks doesn't tell us much. In fact, even the plots of the density functions agree quite well, as shown below.




This is a good fit but we see that the gap between the calculated expected shortfall and the actual expected shortfall value is more than when block size was 50.

```
Calculating the Expected Shortfall
```{r}
p = 0.99
esgev(p, mu, sigma, epsilon)
```
```
[1] 18746.47
```
```{r}
mean(blocks[blocks>=quantile(dataset,0.99)])
```
```
[1] 24327.43
```

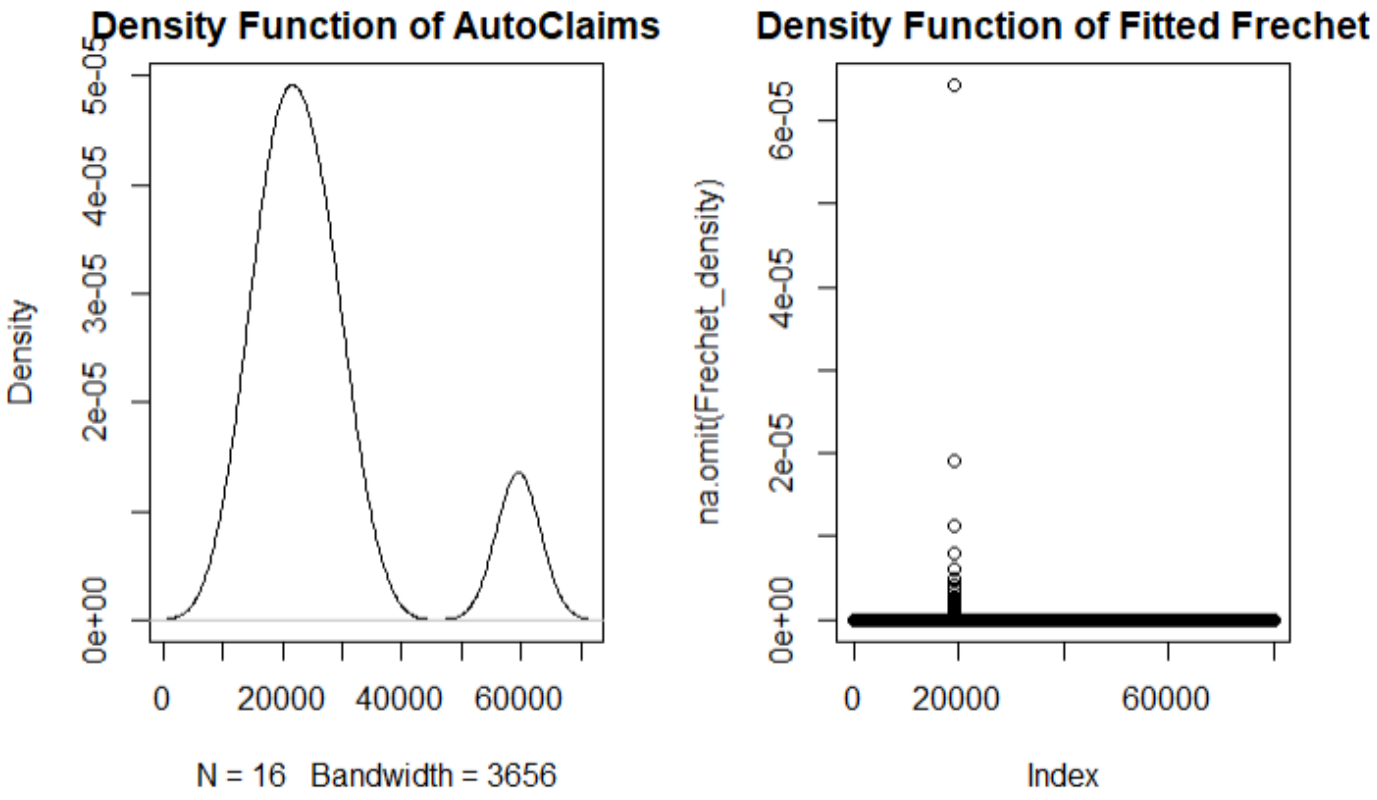
The actual ES value and the ES value that fittedGEV predicts are close but not as close as when blen=50. The difference is significant and hence we consider smaller block sizes.

### 10.3 Absurd Plot due to NaN values in dgev

Sometimes when the chunk shown below is run, the Density Function of Fitted Frechet is not what we would expect it to be.
```

```
Plot
```{r}
par(mfrow=c(1,2))
plot(density(blocks), main="Density Function of AutoClaims")
x <- seq(-10000, 70000, by=1)
Frechet_density <- dgev(x, mu, sigma, epsilon)
#Since NaNs are produced, we must used na.omit to remove these values
plot(na.omit(Frechet_density), main="Density Function of Fitted Frechet")
```
```

Instead, it is one with a lot of holes as shown below.




The reason for this is that dgev generates NaN values as well. These have to be omitted otherwise the plot function does not work.

Now, in most cases, even after ignoring the NaN values we are left with a good number of points to be plotted. But, sometimes, the randomly generated sequence of values 'x' maybe such that most of the dgev values turn out to be NaN. In such a case we get an absurd plot for the Density Function of Fitted Frechet.

Fortunately, this problem has a rather simple solution.

We can just restart R and run all the chunks. Usually, after 2-3 runs like this, the things get back to normal and we get a good plot.

### 10.4 Hill Estimator

A third way to estimate the parameter  of GEV is the Hill method, which is nonparametric. The Hill estimator is given by:

$$\hat{\xi}^{Hill} = [\tfrac{1}{k}\sum_{i=1}^{k} logX_{(i)} - logX_{(k)}]^{-1}$$

where, $X_{(1)} \geq X_{(2)} \geq X_{(3)} \geq ... \geq X_{(k)}$ are the largest, second largest and k-th largest values.
If data are IID and belong to $MDA(\xi)$, then the Hill estimator is consistent and asymptotically normal with asymptotic variance $\frac{\xi}{k}$. We can use this result to test whether shape parameter is significantly different from zero or not.
As shape parameter $\xi$ governs the tail behavior of the limiting distribution and $\alpha = \frac{-1}{\xi}$ is treated as a tail-index of the distribution, the hill estimator gives us the empirical tail behavior of the limiting distribution.
The weakness of the Hill estimator is the choice of k. One may plot the Hill estimator against k and find a proper k such that the estimate appears to be stable.

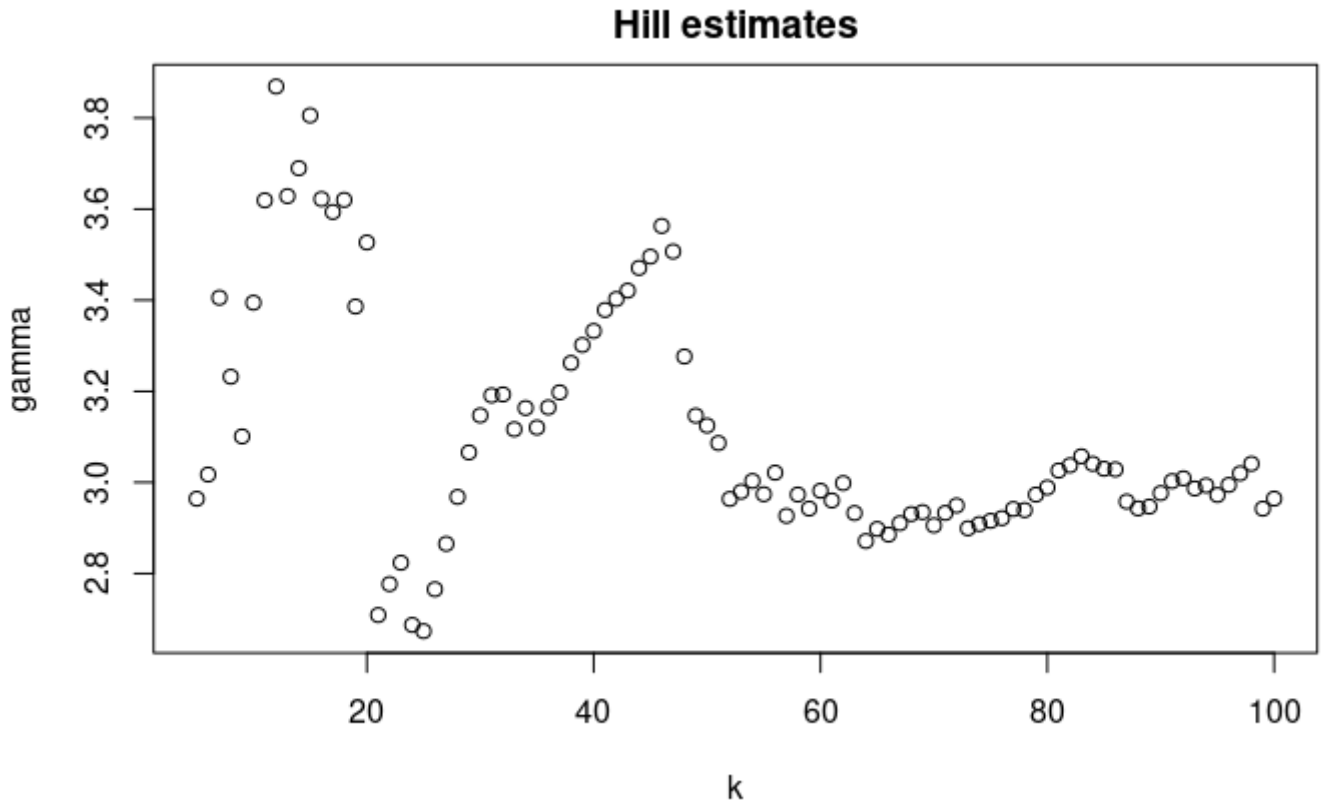



Figure 37: Hill estimate of $\xi$ for different k

As we can see from this plot for different values of $k = 5 \cdots 50$ the values of $\xi$ are random and unstable. It is only for $k > 50$ that we see some stabilisation and even then the value of parameter $\xi$

is far from the value estimated using other methods. We conclude that for our given dataset the Hill estimator is not a good choice for parameter estimation.

```r
mu<-mean(dataset)
sigma<-sqrt(var(dataset))
#print(mu)
#print(sigma)
#print(shape[70])
p = 0.99
vargev(p, mu, sigma, as.double(shape[70]))
esgev(p, mu, sigma, as.double(shape[70]))
```

```
[1] 588567790
[1] 3067777
```

```r
mean(dataset[dataset>=quantile(dataset,0.99)])
```

```
[1] 18172.93
```

Figure 38: Expected Shortfall using Hill Estimator

As we can see the estimate of Expected shortfall using Hill estimator is far greater than the actual value we get from our dataset.

## Bibliography


- R Documentation - Generalized Extreme Value Modelling (https://www.rdocumentation.org/packages/fExtremes/versions/3010.81/topics/GevMdaEstimation)
- fExtremes: Rmetrics - Modelling Extreme Events in Finance (https://rdrr.io/cran/fExtremes/man/GevModelling.html)
- Wikipedia - Generalized extreme value distribution (https://en.wikipedia.org/wiki/Generalized_extreme_value_distribution)
- R Documentation - GEV (https://www.rdocumentation.org/packages/evd/versions/2.3-3/topics/gev)
- R package - evir (https://cran.r-project.org/web/packages/evir/evir.pdf)

- R package - quantmod
  (https://cran.r-project.org/web/packages/quantmod/quantmod.pdf)
- R package - ineq
  (https://cran.r-project.org/web/packages/ineq/ineq.pdf)
- R package - insuranceData
  (https://cran.r-project.org/web/packages/insuranceData/insuranceData.pdf)
- R Documentation - Automobile Insurance Claims
  (https://www.rdocumentation.org/packages/insuranceData/versions/1.0/topics/AutoClaims)
- R package - ggplot2
  (https://cran.r-project.org/web/packages/ggplot2/index.html)
- R package - evd
  (https://cran.r-project.org/web/packages/evd/index.html)
- R package - nortest
  (https://cran.r-project.org/web/packages/nortest/index.html)
- R package - extRemes
  (https://cran.r-project.org/web/packages/extRemes/extRemes.pdf)
- R package - VaRES
  (https://cran.r-project.org/web/packages/VaRES/VaRES.pdf)
- R package - vars
  (https://cran.r-project.org/web/packages/vars/vars.pdf)
- R package - PerformanceAnalytics
  (https://cran.r-project.org/web/packages/PerformanceAnalytics/PerformanceAnalytics.pdf)
- R package - fExtremes
  (https://cran.r-project.org/web/packages/fExtremes/fExtremes.pdf)
- R package - RobExtremes
  (https://cran.r-project.org/web/packages/RobExtremes/index.html)