Risk Management: First Assignment

Lirida Papallazi

11/2/2021

Let’s import the libraries that we will need for this project.

Now we want to import our dataset:

```
usa = read.csv("C:/Users/liry9/Desktop/Risk Management/First Assignment/Dataset/us_accidents.cs
v", header=TRUE)
```

We keep only the columns corresponding to the ID, the County and the City and build a dataset denominated “usacall” that contains the City where the car accident happened, and the count of accidents per City. We then order the new dataset based on the count of accidents in descending order, so that the City with the highest number of accidents will be in first position.

```
## We keep only the id, the city and the county
usa2 <- usa[c("ID","City","County")]


usacall <- sqldf("SELECT  City, COUNT(*) AS Count
              FROM usa2
              GROUP BY City
              ORDER BY Count DESC")
```

We add a new id and rename the three columns with easier names.

```
usacall <- usacall %>% mutate(id = row_number())

## We rename the columns with easier names
colnames(usacall)[1] <- "city"
colnames(usacall)[2] <- "total"
colnames(usacall)[3] <- "rank"
```

Then we remove the original dataset and the intermediary ones used, since they are no longer useful.

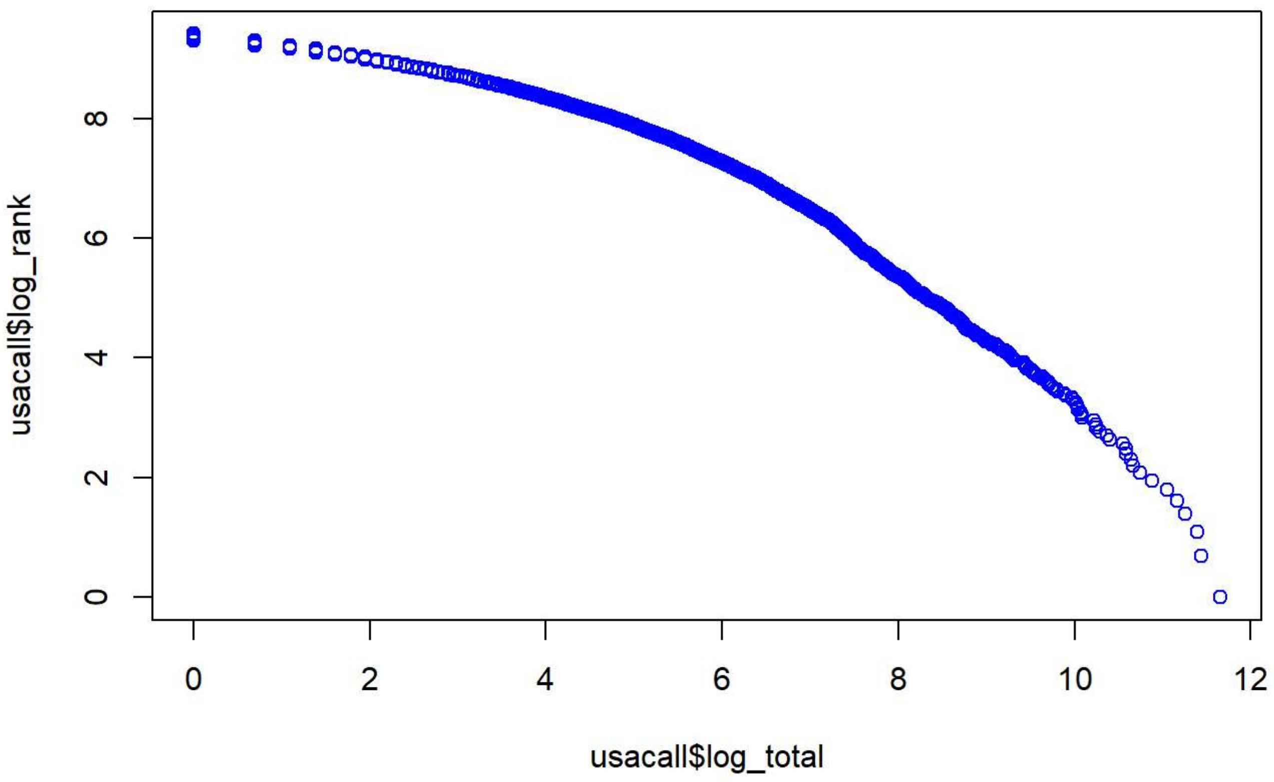
```
rm(usa)
rm(usa2)
```

We add a column called “cum_sum” with the cumulative sums and two columns with the logs of ranks and of the totals

```
usacall$cum_sum <- cumsum(usacall$total)

### Log-rank and Log-totals
usacall$log_rank <- log(usacall$rank)
usacall$log_total <- log(usacall$total)
```

We also plot our data using the log-log scale which is useful to prove again if our distribution behaves like a Power-Law:

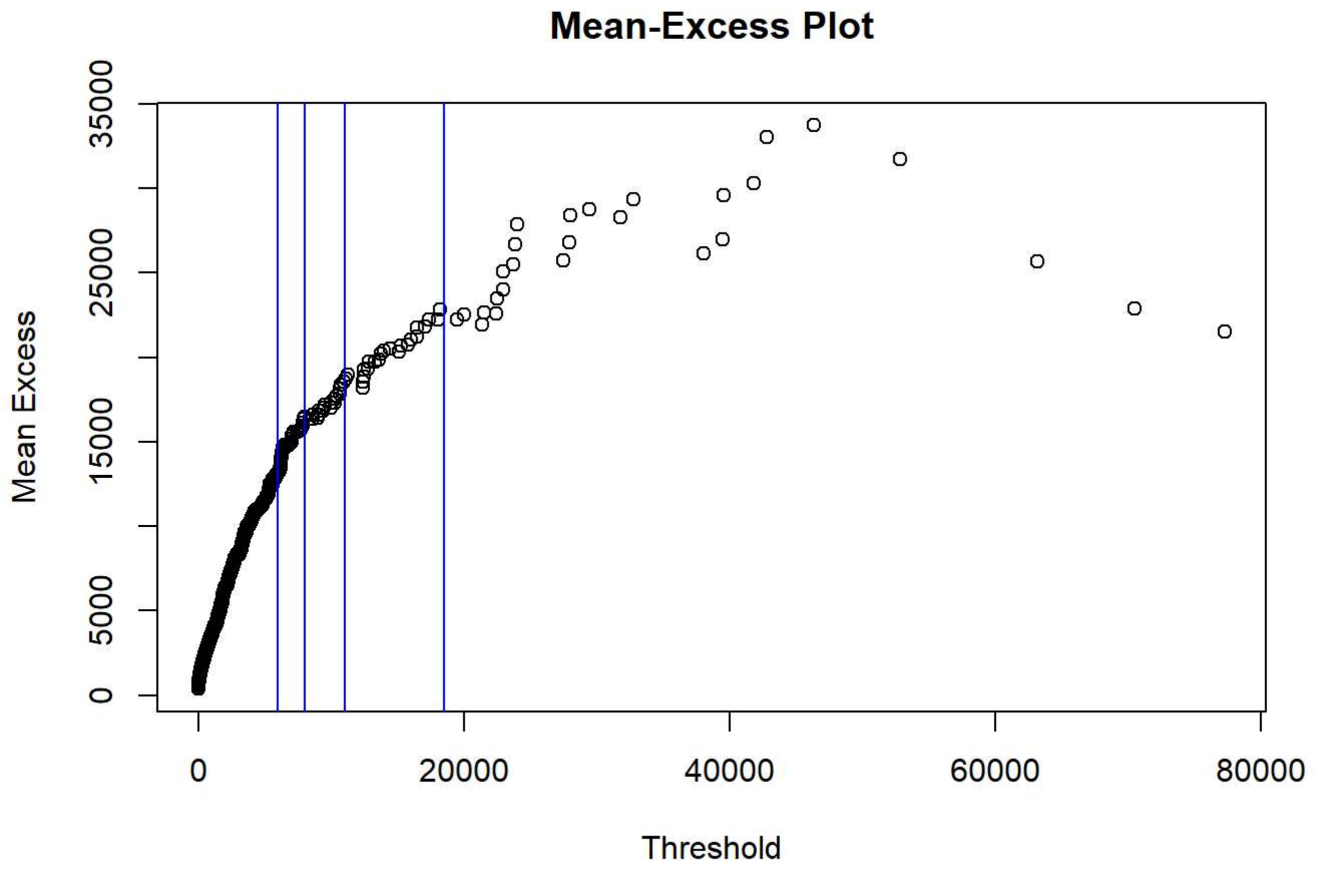
```
plot(x=usacall$log_total, y=usacall$log_rank, col="blue")
```



Mean excess function

We start off by plotting the Mean Excess Function (ME), which is useful to determine if our distribution is heavy-tailed. In our case we can note that it is heavy-tailed. Based on the ME plot we could set 3 to 4 thresholds at 6.000, 8.000, 11.000 and 18.500.

```
# Mean excess function plot
MEplot(usacall$total)

u_6000 <- 6000
u_8000 <- 8000
u_11000 <- 11000
u_18500 <- 18500

abline(v= c(u_6000,u_8000,u_11000,u_18500), col="blue")
```



GDP model

We fit now with the fit.GPD function a generalized pareto distribution to our data. This is done via maximum likelihood estimation (MLE).

```
### GPD model
mod_u6000=fit.GPD(usacall$total, threshold=u_6000)
mod_u8000=fit.GPD(usacall$total, threshold=u_8000)
mod_u11000=fit.GPD(usacall$total, threshold=u_11000)
mod_u18500=fit.GPD(usacall$total, threshold=u_18500)
```

We want to check now which one among the selected thresholds is the best one

```
## Threshold at 6.000
xi_u6000=as.numeric(mod_u6000$par.ests[1])
beta_u6000=as.numeric(mod_u6000$par.ests[2])

## Threshold at 8.000
xi_u8000=as.numeric(mod_u8000$par.ests[1])
beta_u8000=as.numeric(mod_u8000$par.ests[2])

## Threshold at 11.000
xi_u11000=as.numeric(mod_u11000$par.ests[1])
beta_u11000=as.numeric(mod_u11000$par.ests[2])

## Threshold at 18.500
xi_u18500=as.numeric(mod_u18500$par.ests[1])
beta_u18500=as.numeric(mod_u18500$par.ests[2])
```

We build also a small table gathering the results for each threshold.

```
xi_and_beta <- matrix(c(xi_u6000,beta_u6000,xi_u8000,beta_u8000,xi_u11000,beta_u11000,xi_u18500,
beta_u18500), ncol=4)
colnames(xi_and_beta) <- c('u6000', 'u8000', 'u11000', 'u18500')
rownames(xi_and_beta) <- c('xi', 'beta')
xi_and_beta_table <- as.table(xi_and_beta)
xi_and_beta_table
```

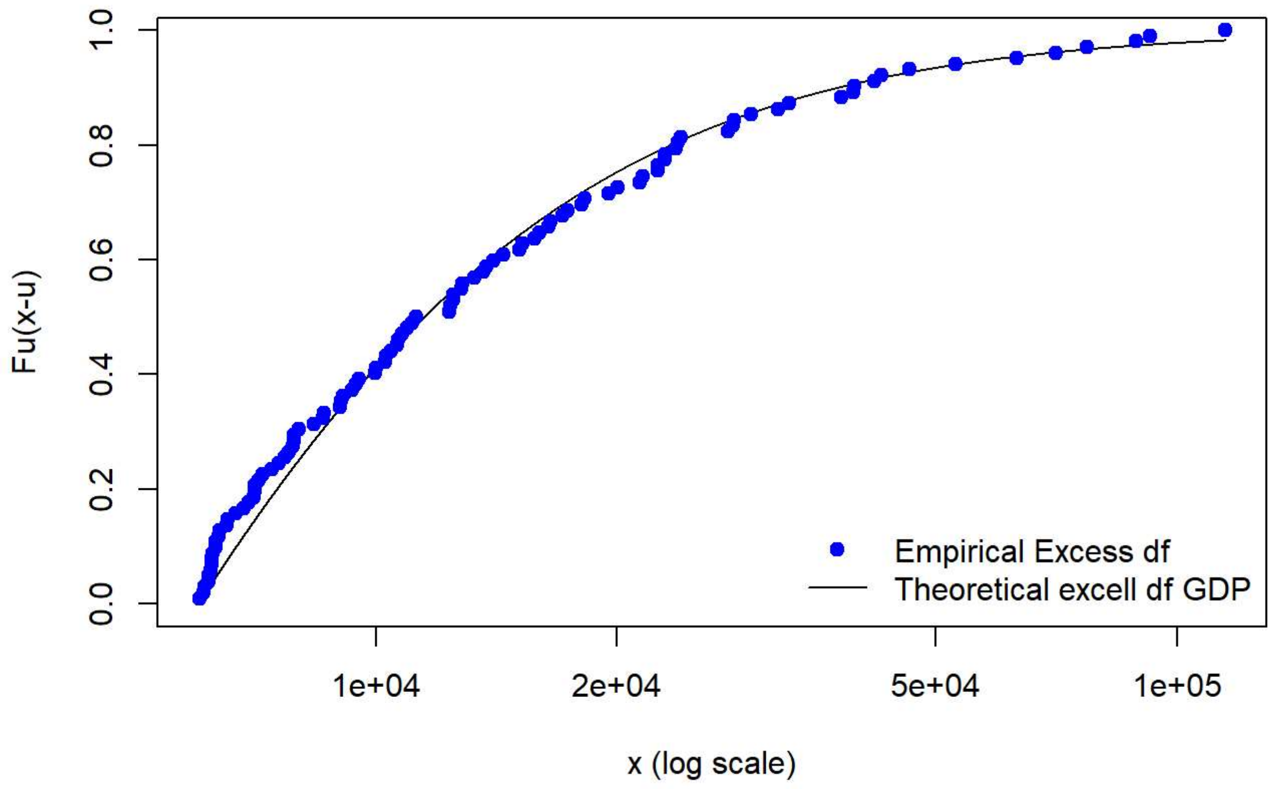
```
##             u6000        u8000       u11000       u18500
## xi   6.035809e-01 3.610107e-01 2.955752e-01 1.530367e-01
## beta 6.370884e+03 1.076801e+04 1.334396e+04 1.920729e+04
```

Fitted GDP vs Empirical GDP

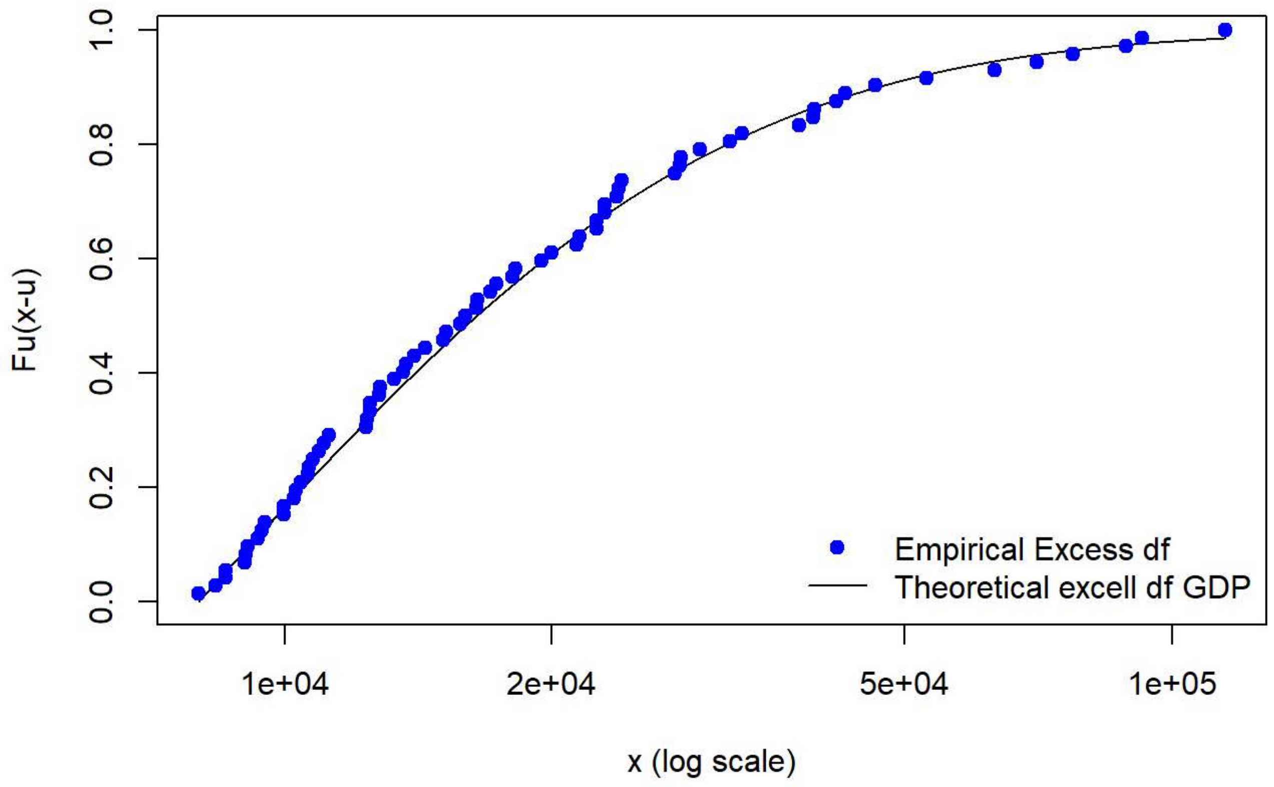
We chose thresholds equal to 6000 and 8000, beacuse they are respectively the first and second kink of our Mean Excess Function.

We also note that we have positive Xis, therefore our distribution behaves like a Pareto. And we can also control the values of the Betas.

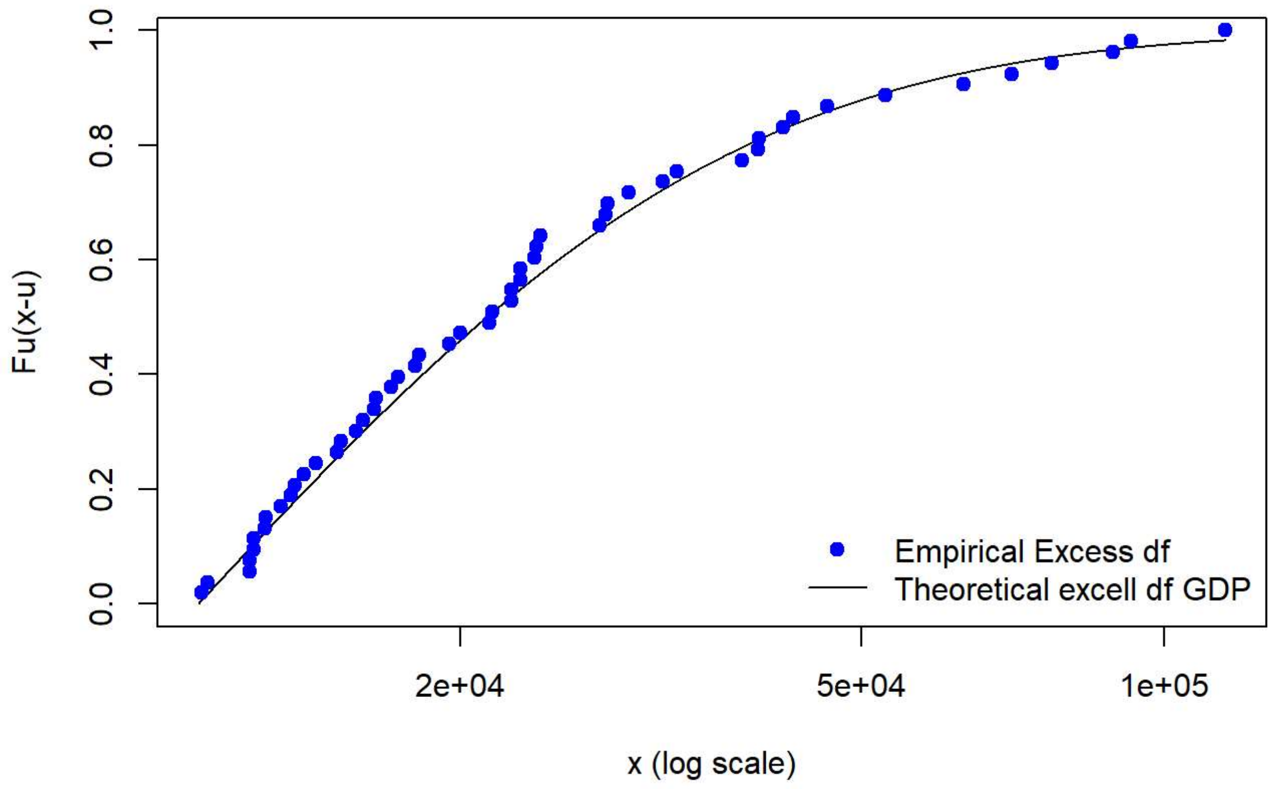
```
## Threshold u=6.000
plotFittedGPDvsEmpiricalExcesses(usacall$total, threshold = u_6000)
legend("bottomright", bty = "n", lty = 0:1, pch = c(19, NA),
       col = c("blue","black"), legend= c('Empirical Excess df','Theoretical excell df GDP'))
```



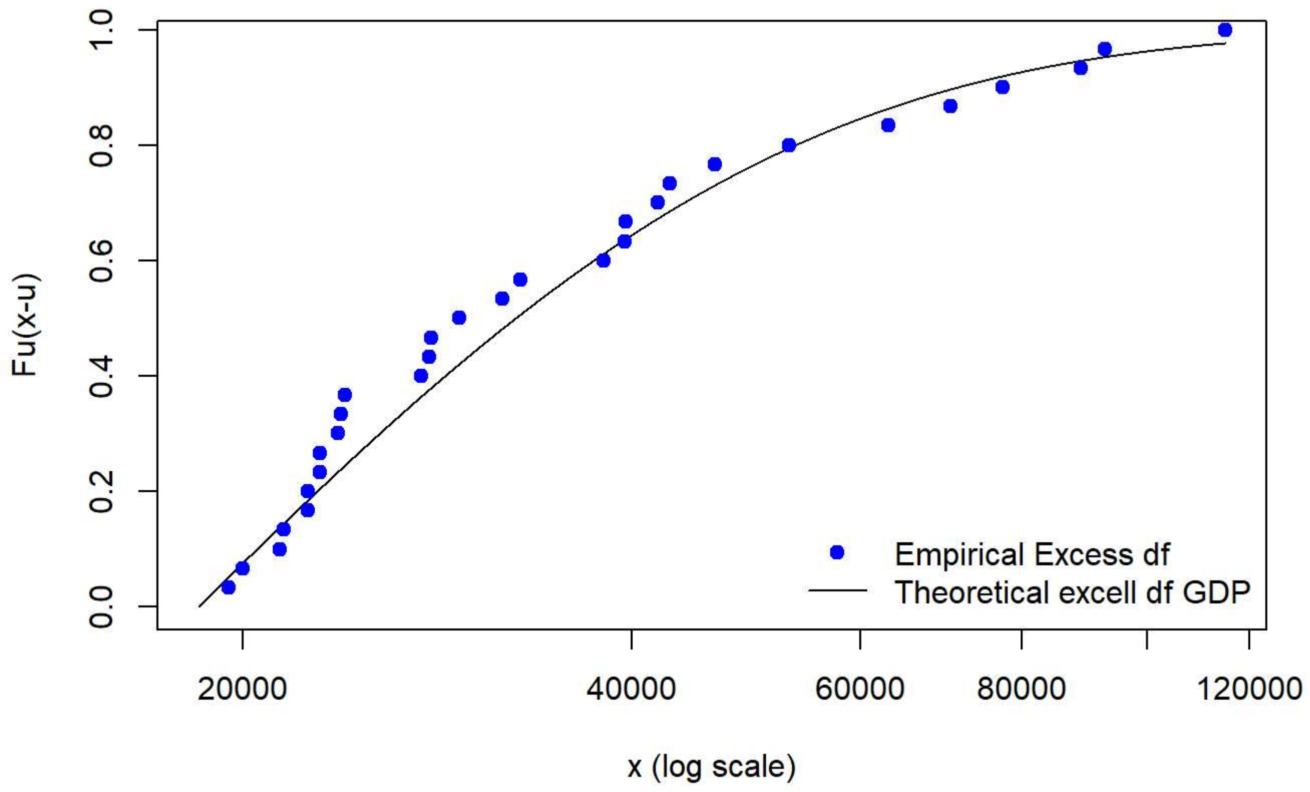
```
## Threshold u=8.000
plotFittedGPDvsEmpiricalExcesses(usacall$total, threshold = u_8000)
legend("bottomright", bty = "n", lty = 0:1, pch = c(19, NA),
       col = c("blue","black"), legend= c('Empirical Excess df','Theoretical excell df GDP'))
```



```
## Threshold u=11.000
plotFittedGPDvsEmpiricalExcesses(usacall$total, threshold = u_11000)
legend("bottomright", bty = "n", lty = 0:1, pch = c(19, NA),
       col = c("blue","black"), legend= c('Empirical Excess df','Theoretical excell df GDP'))
```

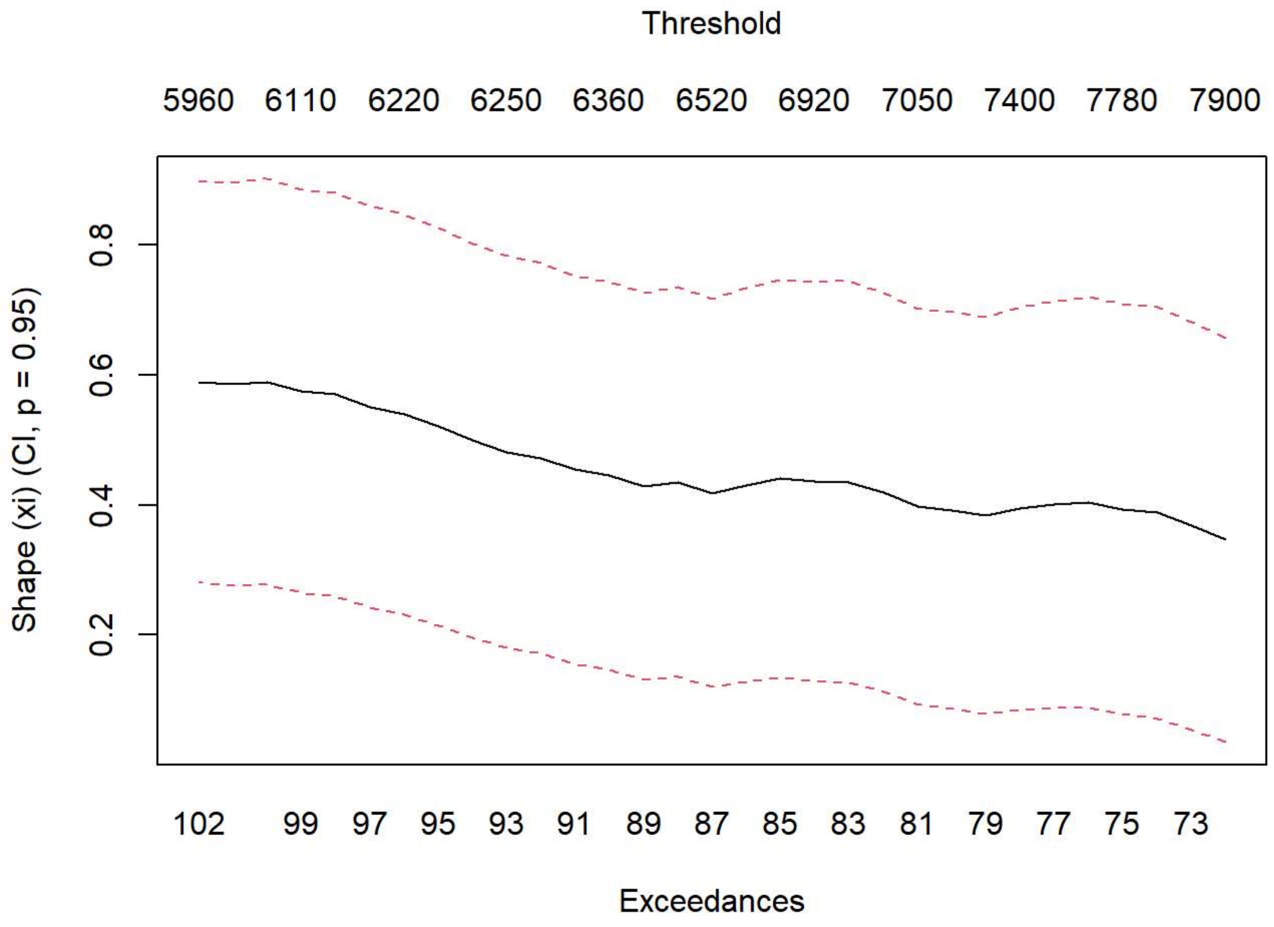


```
## Threshold u=18.500
plotFittedGPDvsEmpiricalExcesses(usacall$total, threshold = u_18500)
legend("bottomright", bty = "n", lty = 0:1, pch = c(19, NA),
       col = c("blue","black"), legend= c('Empirical Excess df','Theoretical excell df GDP'))
```



Now we perform a simultaneous comparison of different GPD models with a Confidence interval CI=0.95 (by default).

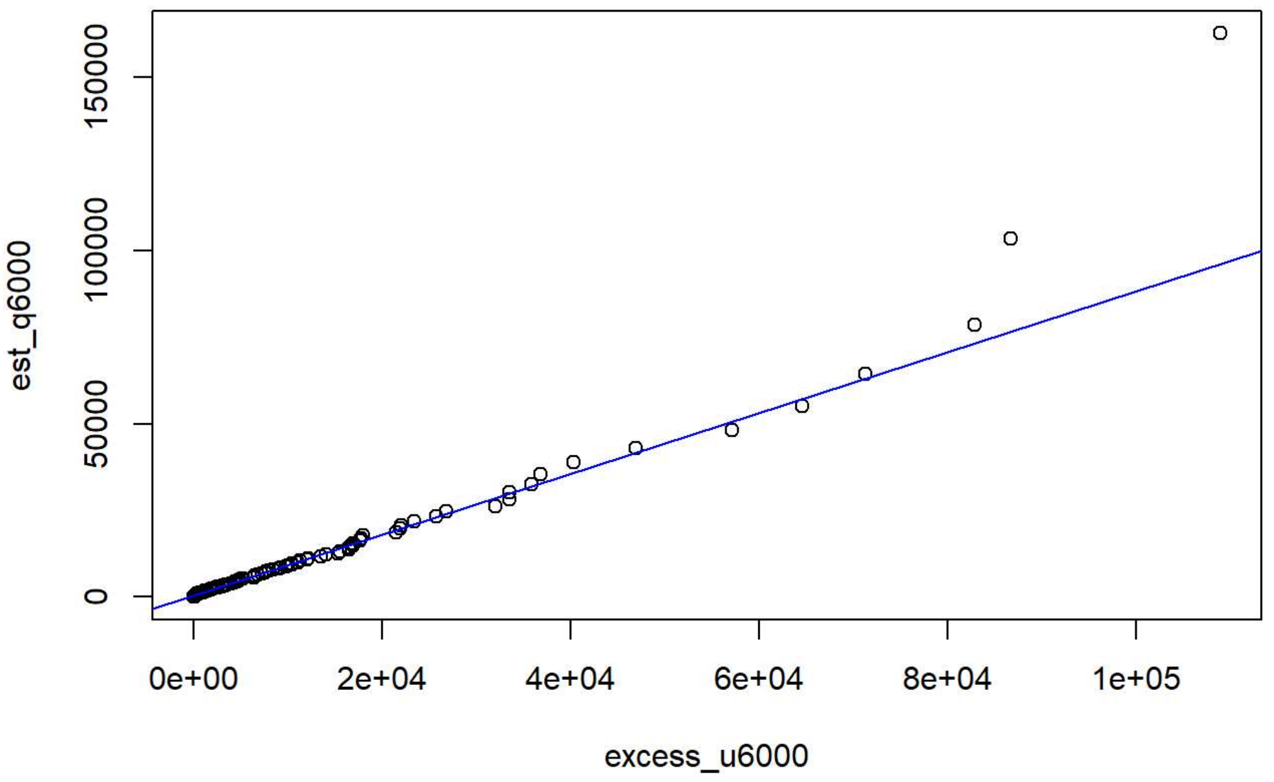
```
xiplot(usacall$total, models=50, start=mod_u6000$n.exceed, end=mod_u8000$n.exceed, reverse=TRUE)
```



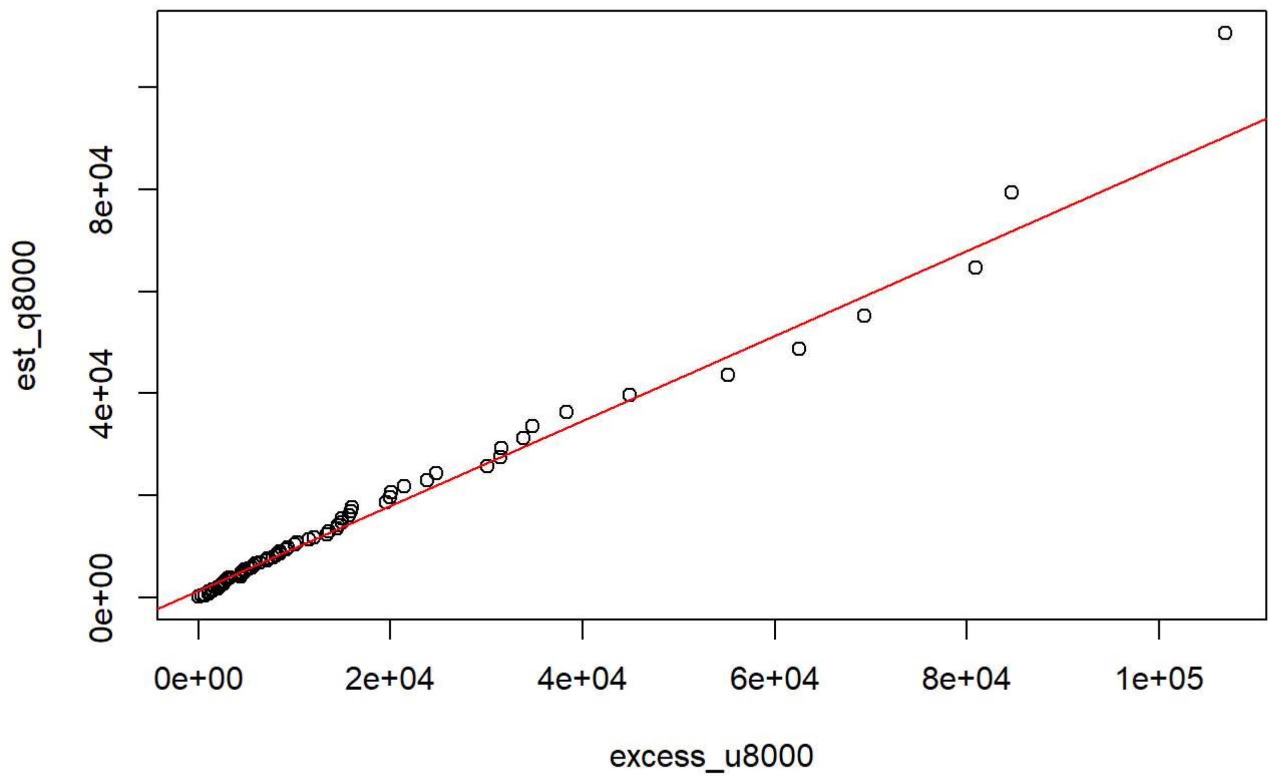
QQplots

In order to check now if the selected thresholds (at 6.000 and at 8.000) are the right ones we resolve to qqplots. Then we will plot the QQplot of quantile function of excedences u6000 compared to the hypotetical quantile function of GPD.

```
## QQplot of u=6.000
qf_u6000 <- function(p)
qGPD(p, xi_u6000, beta_u6000)

excess_u6000 <- sort(usacall$total[usacall$total > u_6000] - u_6000)
N_6000 = mod_u6000$n.exceed
est_q6000 <- qf_u6000(c(1:N_6000)/(N_6000+1))

plot(excess_u6000, est_q6000)
abline(lm(est_q6000[1:(N_6000-3)]~excess_u6000[1:(N_6000-3)]), col="blue")
```



```
## QQplot of quantile function of excedences of u8000 compared to the
# hypotetical quantile function of GPD
qf_u8000 <- function(p) # quantile function of df
qGPD(p, xi_u8000, beta_u8000)

excess_u8000 <- sort(usacall$total[usacall$total > u_8000] - u_8000)
N_8000 = mod_u8000$n.exceed
est_q8000 <- qf_u8000(c(1:N_8000)/(N_8000+1))

plot(excess_u8000, est_q8000)
abline(lm(est_q8000[1:(N_8000-3)]~excess_u8000[1:(N_8000-3)]), col="red")
```



Comparison between quantile levels

The riskmeasures function calculates Quantiles And Expected Shortfalls makes a rapid calculation of point estimates of prescribed quantiles and expected shortfalls using the output of the function gpd.

```
### Threshold at 6.000:
RiskMeasures(mod_u6000, p = c(0.99, 0.995))
```

```
##          p quantile    sfall
## [1,] 0.990 4894.956 19283.52
## [2,] 0.995 9804.126 31667.31
```

The VaR (0,990) of our ditribution considering u = 6.000, tells us that there is a 1% of probability that, in 3 years, there will be more than 1.934,511 car accidents in any City of the Database. The VaR (0,995) of our ditribution considering u = 6.000, tells us that there is a 0,05% of probability that, in 3 years, there will be more than 7.286,135 car accidents in any City of the Database.

```
### Threshold at 8.000
RiskMeasures(mod_u8000, p = c(0.99, 0.995))
```

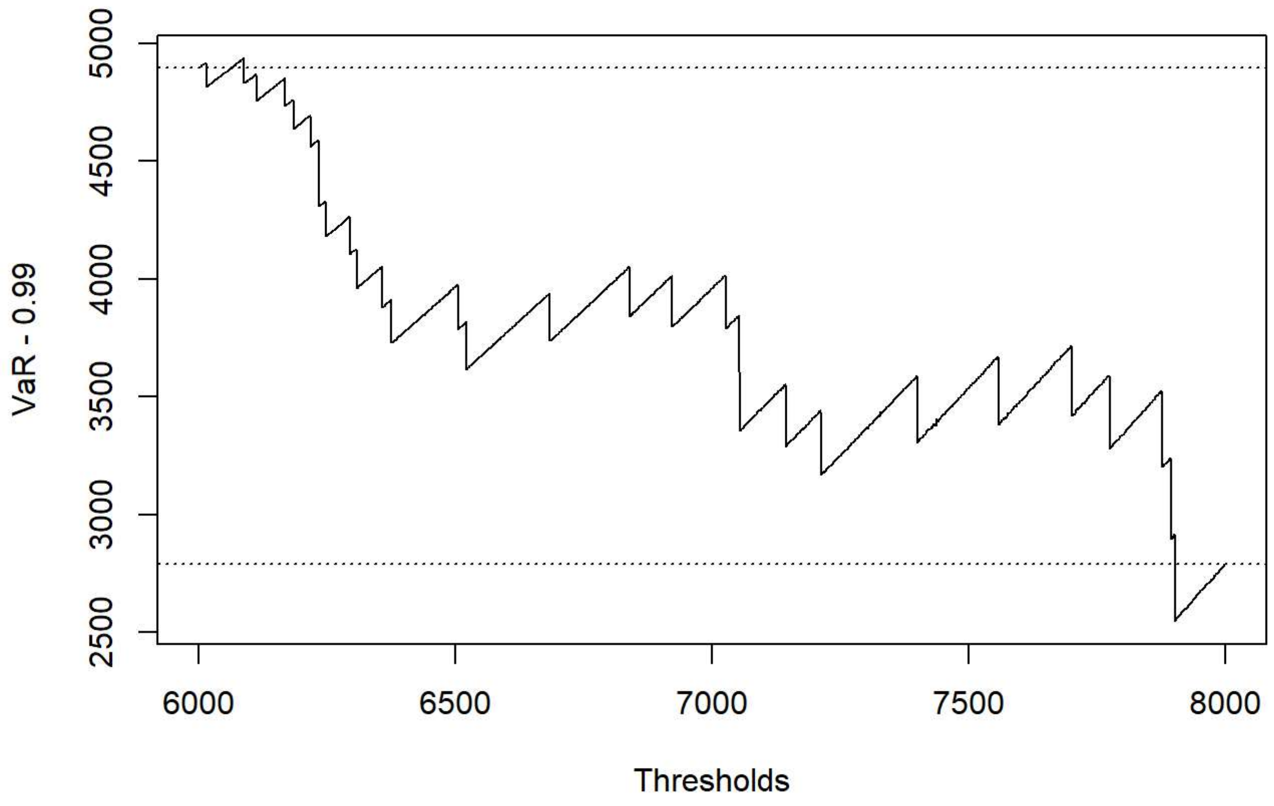
```
##          p quantile    sfall
## [1,] 0.990 2792.148 16701.48
## [2,] 0.995 9792.105 27656.21
```

The VaR (0,990) of our ditribution considering u8000, tells us that there is a 1% of probability that, in 3 years, there will be more than 1711,661 car accidents in any City of the Database. The VaR (0,995) of our ditribution considering u8000, tells us that there is a 0,05% of probability that, in 3 years, there will be more than 7194,097 car accidents in any City of the Database.

Simultaneous comparison of VaRs under different GPD models

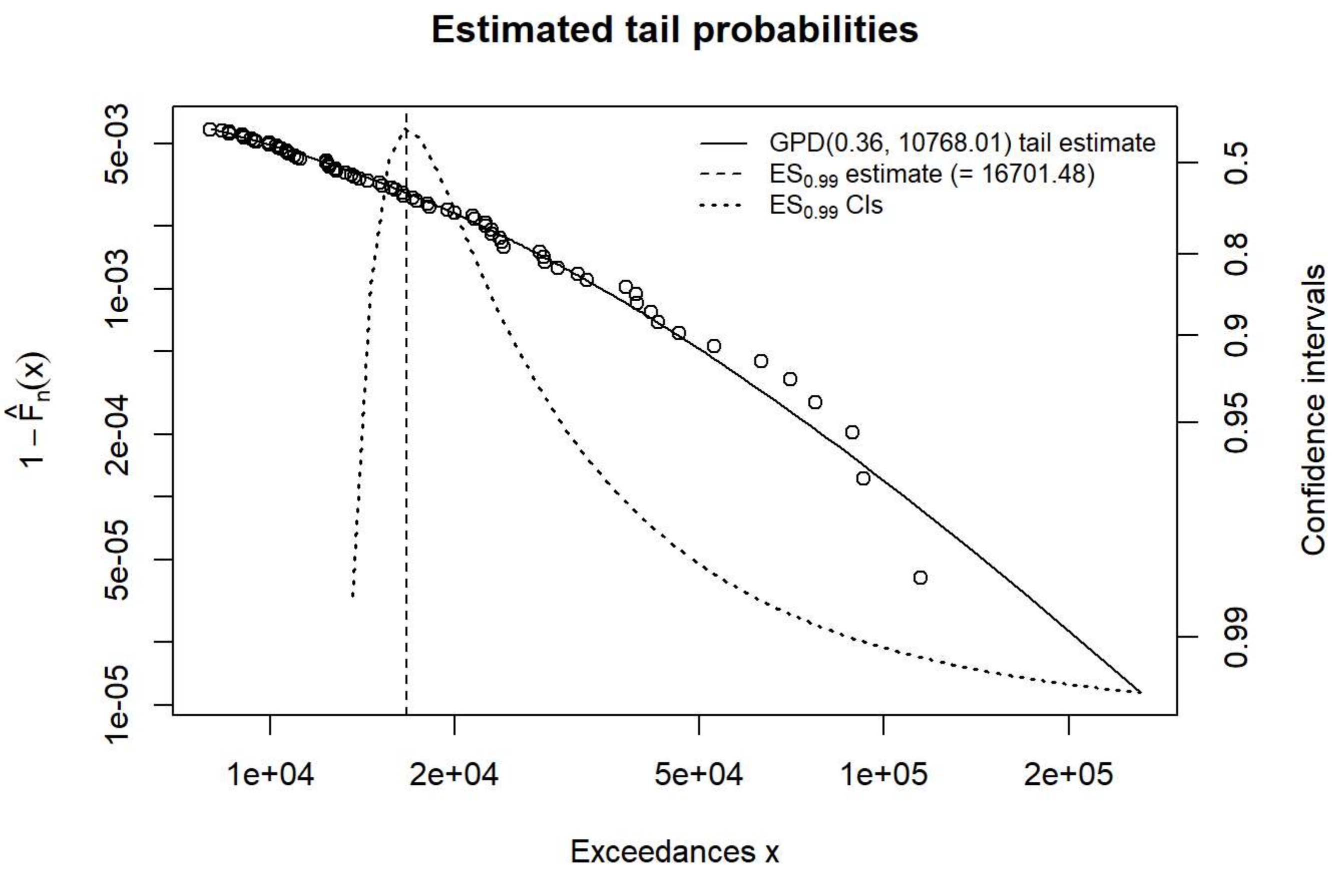
The following graph will tell us how much influence had the choice of the threshold in the VaR computation.

```
VaRplot<-function(data=usacall$total,ab=c(6000,8000))
{alpha=0.99
a=c(ab[1]:ab[2])
for(i in c(ab[1]:ab[2]))
{
  a[i-ab[1]+1]=RiskMeasures(fit.GPD(data,threshold=i), alpha)[[2]]
}
plot(c(ab[1]:ab[2]),a,type="l", xlab="Thresholds",ylab=paste("VaR -",alpha))
}
VaRplot()
abline(h = 4894.956, lty = 3) #Var estimate under u6000
abline(h = 2792.148, lty = 3) #Var estimate under u8000
```



Lastly we can plot the confidence intervals.

```
# Estimated Tail Probabilities
#showRM(mod_u6000, alpha=0.99, RM="VaR" , method="BFGS")

showRM(mod_u8000, alpha=0.99, RM="ES", method="BFGS")
```


Estimated tail probabilities
GPD(0.36, 10768.01) tail estimate
$ES_{0.99}$ estimate (= 16701.48)
$ES_{0.99}$ CIs
$1-\hat{F}_n(x)$
5e-03
1e-03
2e-04
5e-05
1e-05
Exceedances x
1e+04
2e+04
5e+04
1e+05
2e+05
Confidence intervals