QRM CW2

05/12/2021

```
library(readr)
df <- read_csv("QRM-2021-cw2-data.csv")
price <- df$TSLA
```

a)

Fit GARCH(1,1) to loss data

```
#install.packages("expm")
#install.packages("rugarch")
library(rugarch)
```

```
n <- length(price)
r <- log(price[2:n]/price[1:n-1])

spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                 garchOrder = c(1, 1)),
                   mean.model= list(armaOrder = c(0,0), include.mean = TRUE),
                    distribution.model = "norm")
garch <- ugarchfit(spec = spec, data = -r)
coef(garch)
```

```
##            mu         omega        alpha1         beta1
## -8.416092e-04  5.618270e-06  1.725557e-02  9.767228e-01
```

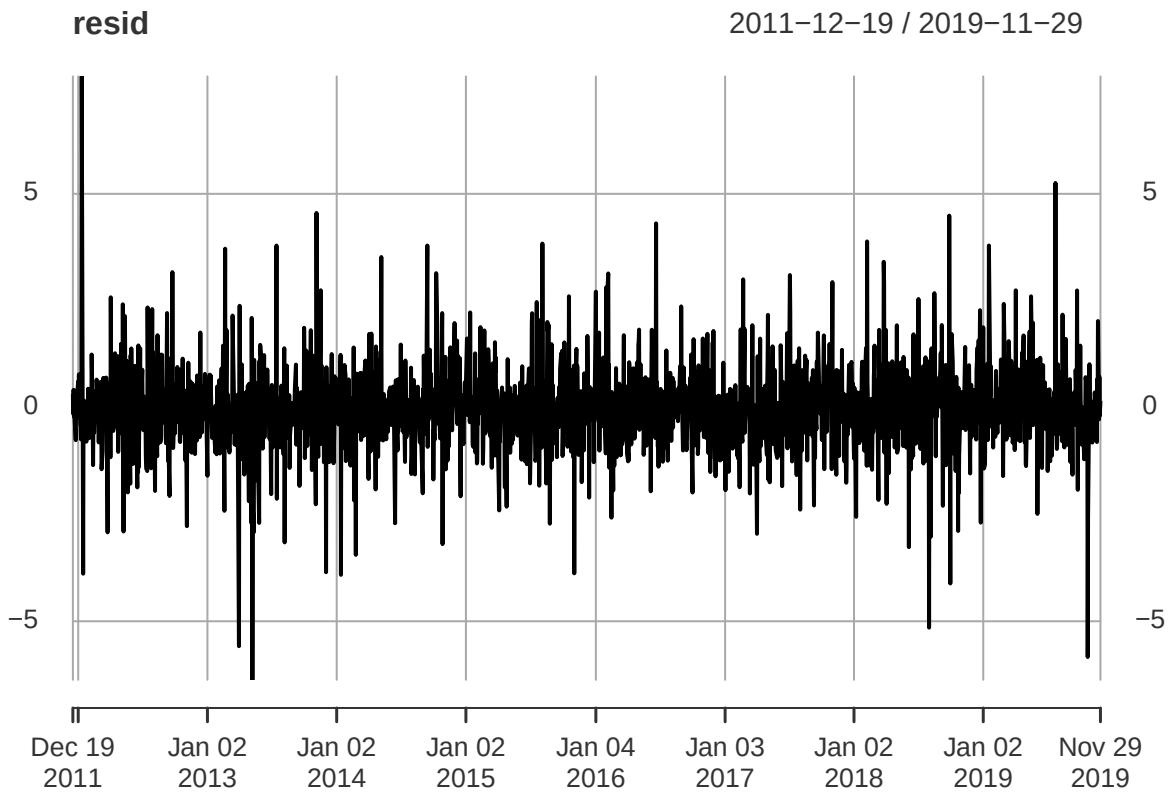
Standardized residuals

```
library(xts)
resid <- residuals(garch, standardize=TRUE)
index(resid) <- as.Date(df$Date[2:n])
write.csv(resid, "garch_resid.csv", row.names = FALSE)
garch_resid <- read.csv("garch_resid.csv")
```

```
head(resid)
```

```
##                  [,1]
## 2011-12-19  0.3140784
## 2011-12-20 -0.1468106
## 2011-12-21  0.4146921
## 2011-12-22 -0.2093894
## 2011-12-23 -0.1265794
## 2011-12-27 -0.7631684
```

```
plot(resid)
```

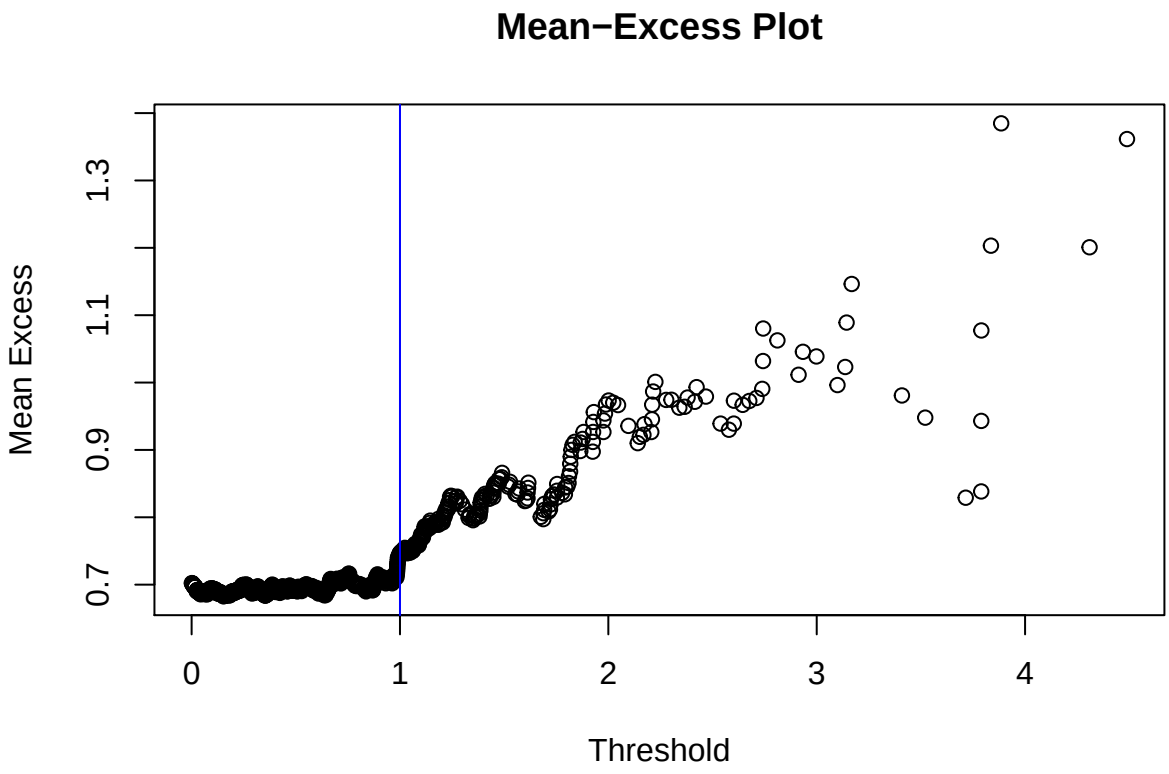


b)

Plot sample mean excess function

```
#install.packages("QRM")
require(QRM)
```

```
MEplot(garch_resid[,1][garch_resid[,1]>0], omit = 3., main = "Mean-Excess Plot", xlab = "Threshold", yla
u <- quantile(garch_resid[,1], probs=0.9, names=FALSE)
abline(v=1,col='blue')
```

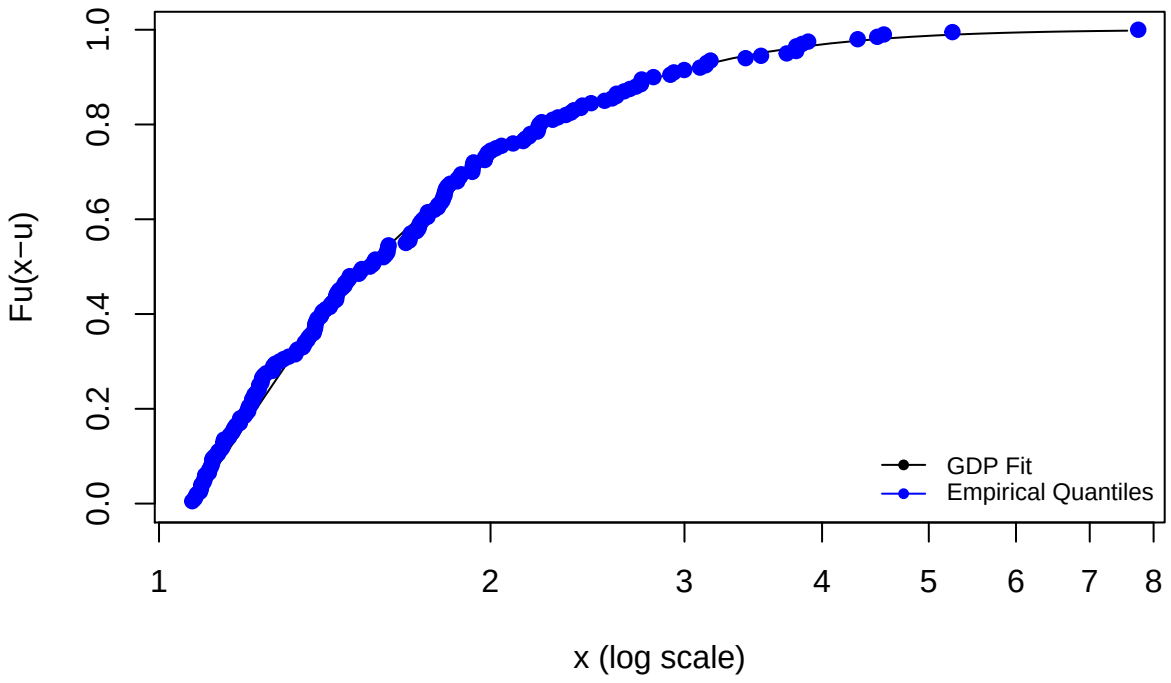


c)

Select a threshold u>0

The threshold is selected as the point at which the ME plot becomes linear, which is 1.

```
plotFittedGPDvsEmpiricalExcesses(garch_resid[,1], threshold = u)
legend("bottomright", inset=0, lty=c(1, 1), bty="n",
       col=c("black", "blue"),
       legend=as.expression(c("GDP Fit","Empirical Quantiles")),cex = 0.75,  pch = 16)
```

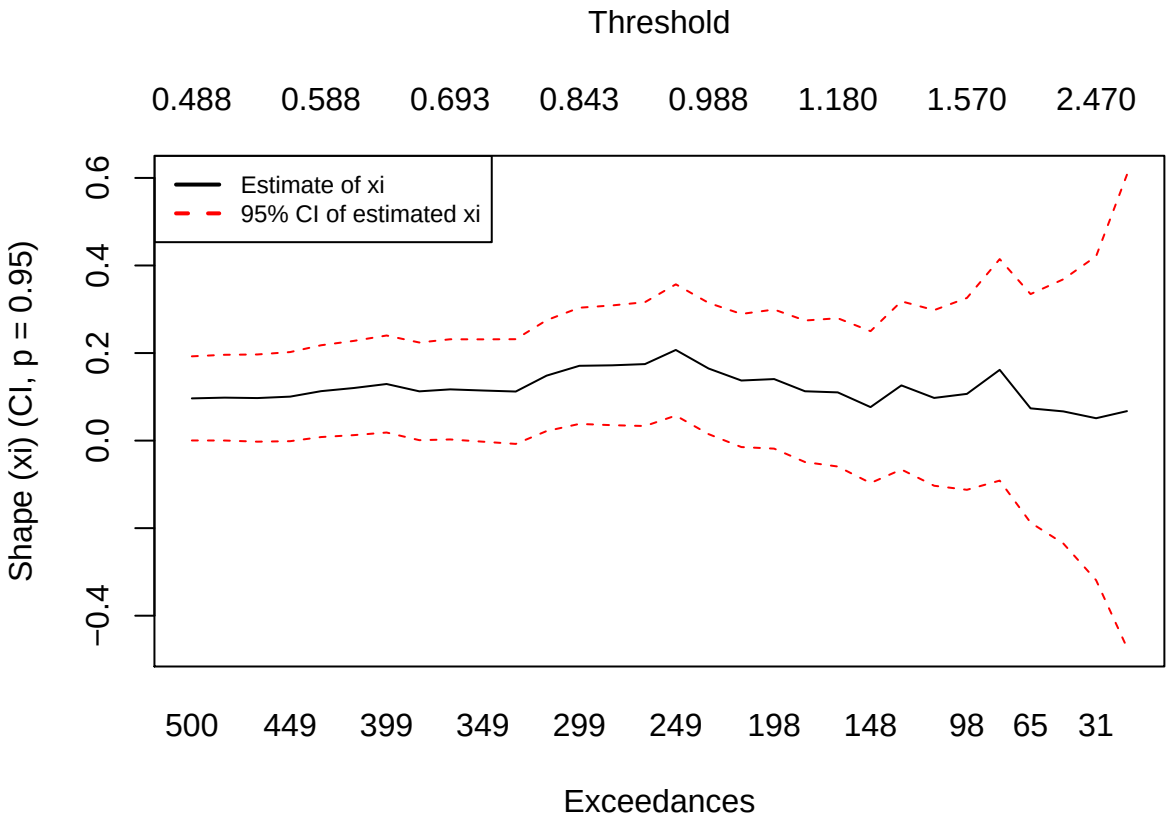


Fit GPD to standardized residuals exceeding the threshold

```
gpd <- fit.GPD(garch_resid[,1],threshold=1)
gpd_param <- as.vector(gpd$par.ests)
print(gpd$par.ests)
```

```
##        xi      beta
## 0.1446651 0.6370055
```

```
xiplot(garch_resid[,1])
legend("topleft", col=c("black", "red"),
       legend=as.expression(c("Estimate of xi","95% CI of estimated xi")),
       lty = c(1, 2), lwd = 2,cex = 0.75)
```



d) Day-ahead VaR abd ES forecasts at 95% and 99% CI

```
vol <- sigma(garch)
t <- length(garch_resid[,1])
garch_param <- as.vector(coef(garch))
mu_pred <- garch_param[1]
vol_pred <- sqrt(garch_param[2]+garch_param[3]*(-r[t]-mu_pred)^2
                 +garch_param[4]*as.numeric(vol[t])^2)

#install.packages("stats")
require(stats)
sorted_resid <- sort(garch_resid[,1])
F_hat <- ecdf(sorted_resid)

q_hat <-function(a){
  q <- gpd$threshold + (gpd_param[2]/gpd_param[1])*
    ((((1-a)/(1-F_hat(gpd$threshold)))^(-gpd_param[1]))-1)
  return(q)}

var_gpd <-function(a){
  v <- mu_pred + vol_pred*q_hat(a)
  return(v)}

es_gpd <-function(a){
  es <- mu_pred + vol_pred*((q_hat(a)+gpd_param[2]-gpd_param[1]*gpd$threshold)/
                            (1-gpd_param[1]))
  return(es)}
```

```r
result <- matrix(c(var_gpd(0.95), es_gpd(0.95), var_gpd(0.99), es_gpd(0.99)),
                 ncol=2, byrow=FALSE)
rownames(result) <- c('day-ahead VaR forecast', 'day-ahead ES forecast')
colnames(result) <- c('95%', '99%')
result
```

```
##                                95%        99%
## day-ahead VaR forecast 0.04796366 0.08894553
## day-ahead ES forecast  0.07440251 0.12231575
```