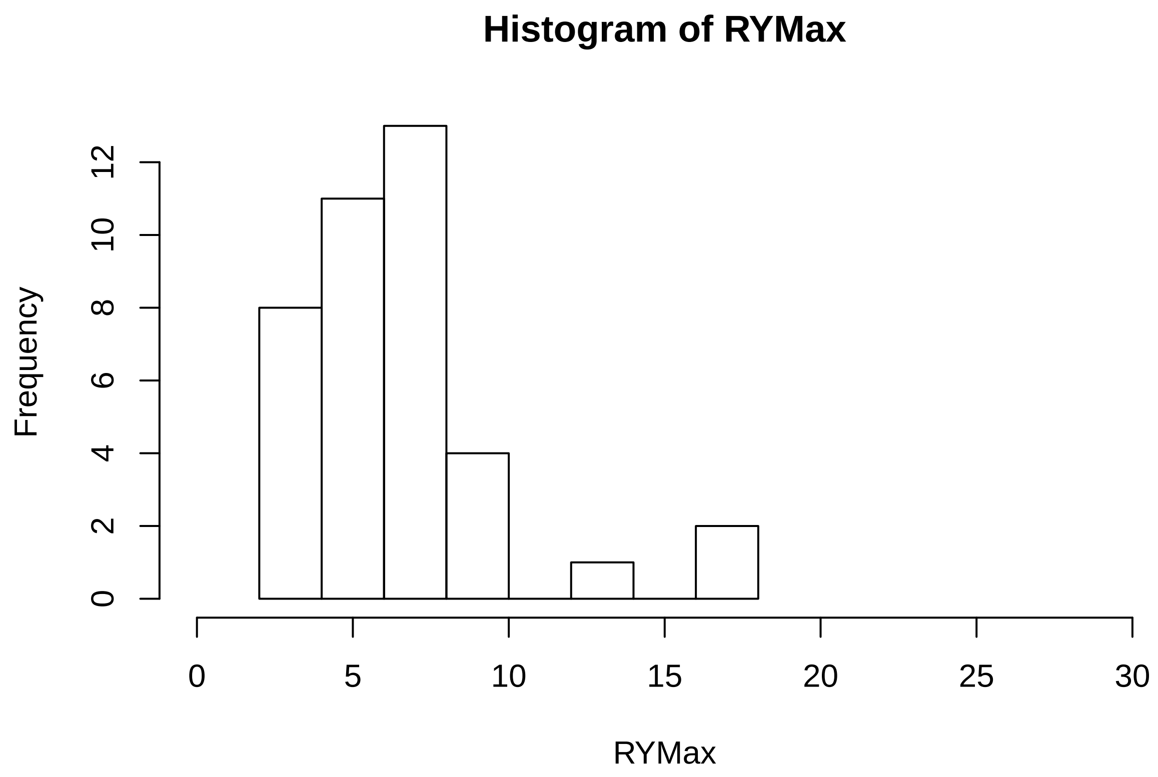
RY.R

Alice Li

Tue Jul 04 11:31:40 2017

```r
rm(list=ls())

#a)
#RY
RY<-read.csv("RY.csv")
x_array <- rep(0,nrow(RY)-1)

for (i in 1:nrow(RY)-1){
  x_array[i] = -100*log(RY$Adj.Close[i+1]/RY$Adj.Close[i])
}


blocksize <- 26
blocks <- floor(length(x_array)/blocksize)

max_array <- rep(0,blocks)

for (i in 1:blocks){
  start_ind <- (i - 1)*26
  end_ind  <- start_ind + 25
  max_array[i] = max(x_array[start_ind:end_ind])
}

hist(max_array,
     main="Histogram of RYMax",
     xlab="RYMax",
     xlim=c(0,30)
)




#b)
library(qrmdata) # for the S&P 500 data
```



```r
library(xts) # for functions around time series objects
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(QRM) # for fit.GEV() and fit.GPD()
```

```
## Loading required package: gsl

## Loading required package: Matrix

## Loading required package: mvtnorm

## Loading required package: numDeriv

## Loading required package: timeSeries

## Loading required package: timeDate

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-
```

```
##
## Attaching package: 'QRM'

## The following object is masked from 'package:base':
##
##     lbeta
```

```r
library(qrmtools) # for returns() <- be RYre to load "qrmtools" last
```

```
##
## Attaching package: 'qrmtools'

## The following objects are masked from 'package:QRM':
##
##     dGEV, dGPD, pGEV, pGPD, qGEV, qGPD, rGEV, rGPD

## The following object is masked from 'package:timeSeries':
##
##     returns
```

```r
fit.week <- fit.GEV(max_array)
(xi.week <- fit.week$par.ests[["xi"]])
```

```
## [1] 0.1122853
```

```r
(mu.week <- fit.week$par.ests[["mu"]])
```

```
## [1] 4.956501
```

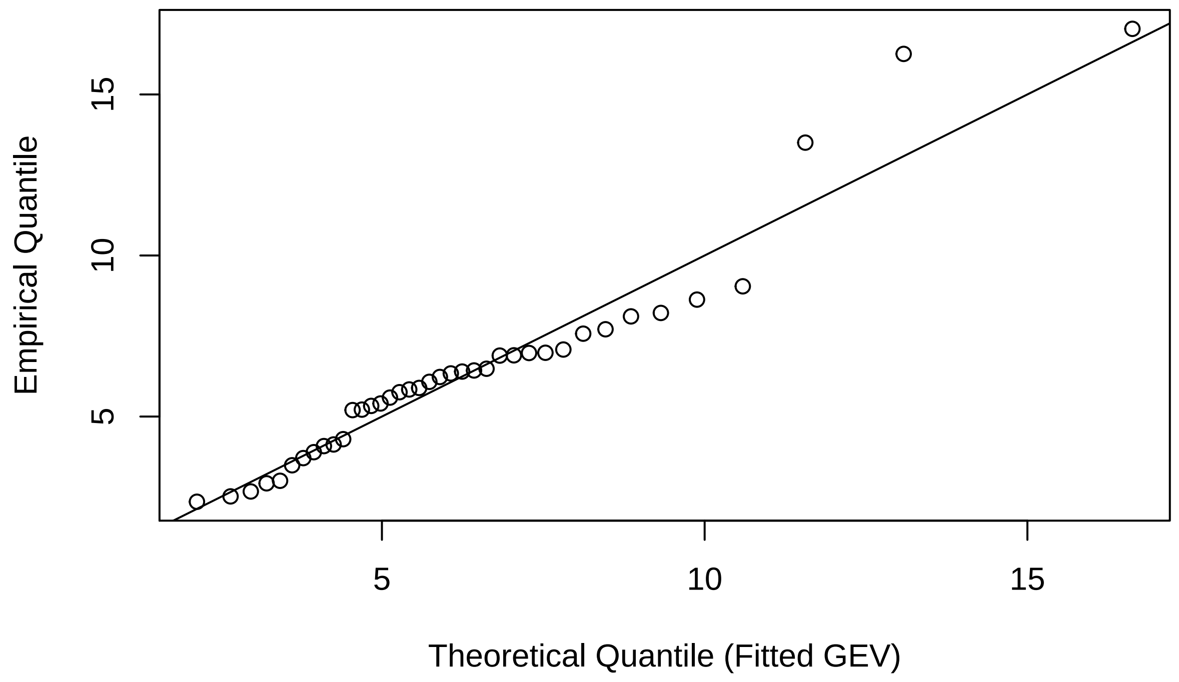
```r
(sig.week <- fit.week$par.ests[["sigma"]])
```

```
## [1] 2.080714
```

```r
fit.week
```

```
## $par.ests
##        xi        mu     sigma
## 0.1122853 4.9565007 2.0807136
##
## $par.ses
##        xi        mu     sigma
## 0.1212678 0.3763374 0.2836944
##
## $varcov
##              [,1]        [,2]         [,3]
## [1,]  0.014705867 -0.01487860 -0.005311442
## [2,] -0.014878596  0.14162988  0.052472686
## [3,] -0.005311442  0.05247269  0.080482486
##
## $converged
## [1] TRUE
##
## $llmax
## [1] -92.52517
```

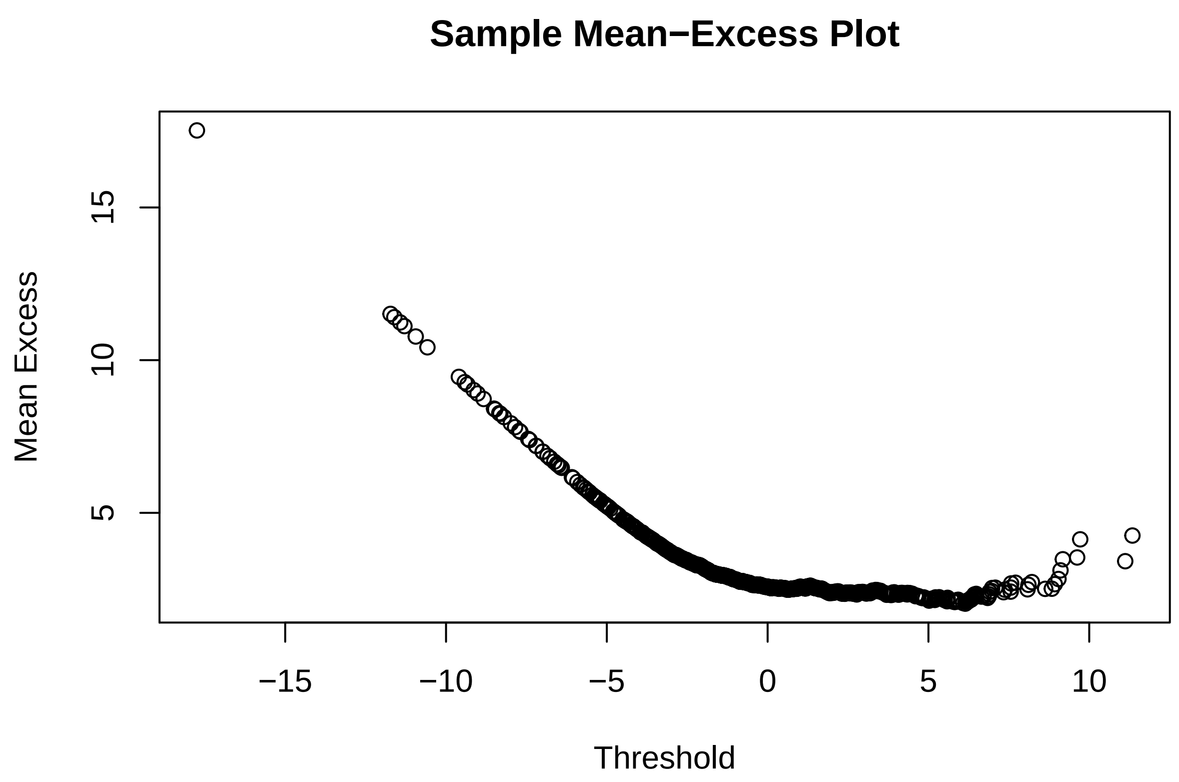
```r
#c)
q.week <- qGEV(ppoints(length(max_array)), xi.week, mu.week, sig.week) # contruct theoretical quantiles
qqplot(q.week, as.vector(max_array), xlab = "Theoretical Quantile (Fitted GEV)",
       ylab = "Empirical Quantile") # compare weekly maxima with theoretical quantiles
abline(0,1) # 45 degree line
```



```
#d)
llmax1 <- fit.week$llmax
h.func<-function(M){1/sig.week*exp(-(M-mu.week)/sig.week - exp(-(M-mu.week)/sig.week))}
llmax0 <- sum(log(h.func(max_array)))
D<- 2*(llmax1 - llmax0)
1-pchisq(D,1)
```

```
## [1] 0.2536716
```

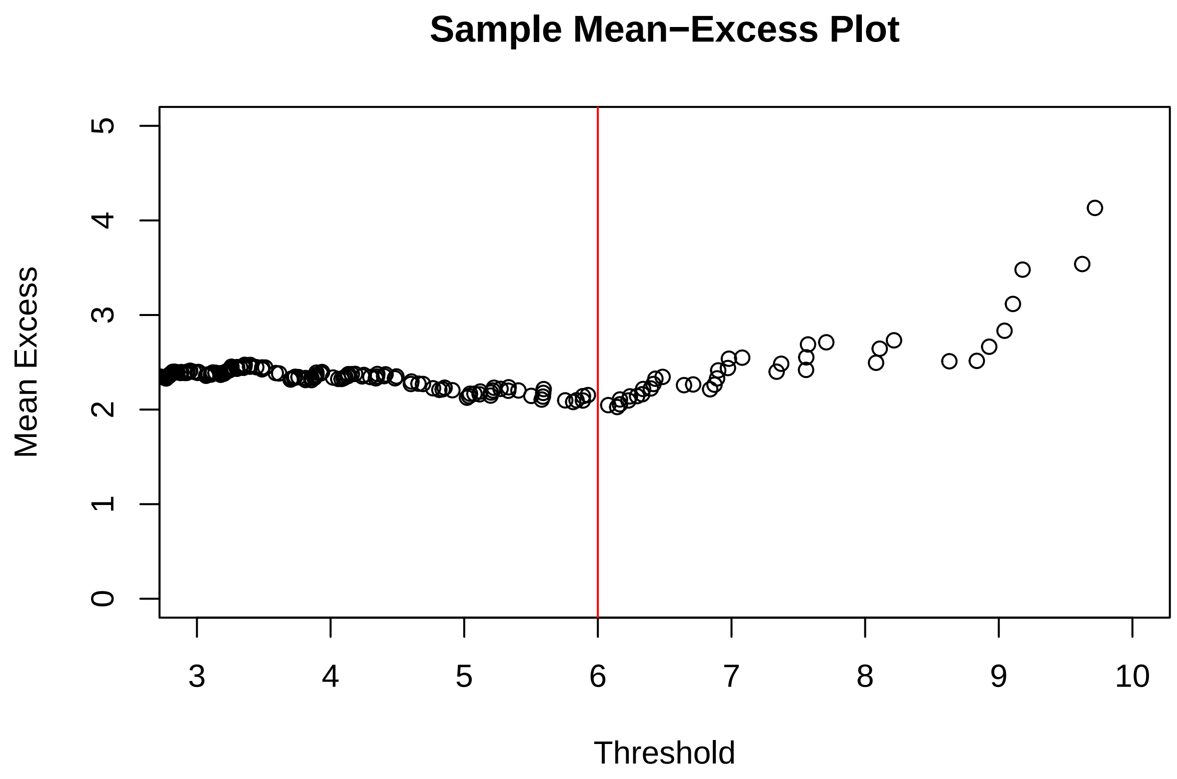
```
#e)
MEplot(x_array, main = "Sample Mean-Excess Plot")
```



```r
quantile(x_array,0.9)
```

```
##      90%
## 3.880546
```

```r
MEplot(x_array, main = "Sample Mean-Excess Plot", xlim = c(3,10), ylim = c(0,5)) #zoomin
#pick d1 = 6
d1 <- 6
abline(v = d1, col = 2)
```



```
#f)
fitGPD <- fit.GPD(x_array, threshold = d1)

xi2 <- fitGPD$par.ests[["xi"]]
beta2 <- fitGPD$par.ests[["beta"]]
lmax1 <- fitGPD$ll.max

#g)
g.func<-function(X){1/beta2 * exp(-X/beta2)}
lmax0 <- sum(log(g.func(x_array[x_array>d1] - d1)))
lmax1
```

```
## [1] -75.49648
```

```
lmax0
```

```
## [1] -77.19716
```

```
D2<- 2*(lmax1 - lmax0)
1-pchisq(D2,1)
```

```
## [1] 0.06514294
```

```
#h)
Sxd <- length(x_array[x_array>d1])/length(x_array)
Q99<- d1 + beta2/xi2 * ((Sxd / (1 - 0.99))^xi2 - 1)
Q995<- d1 + beta2/xi2 * ((Sxd / (1 - 0.995))^xi2 - 1)
Q99
```

```
## [1] 8.854395
```

```
Q995
```

```
## [1] 10.51019
```

```
CTE99 <- (Q99 + beta2 - xi2*d1)/(1-xi2)
CTE995 <- (Q995 + beta2 - xi2*d1)/(1-xi2)
CTE99
```

```
## [1] 11.61486
```

```
CTE995
```

```
## [1] 13.66035
```

```
#empirical
Q99e <- quantile(x_array,0.99)
Q995e <- quantile(x_array,0.995)
Q99e
```

```
##      99%
## 8.912565
```

```
Q995e
```

```
##    99.5%
## 9.711564
```

```
CTE99e <-mean(x_array[x_array > Q99e])
CTE995e <- mean(x_array[x_array > Q995e])

CTE99e
```

```
## [1] 11.35086
```

```
CTE995e
```

```
## [1] 13.16329
```