CaseStudy_GiulioCattoni_MarioGarcia.R

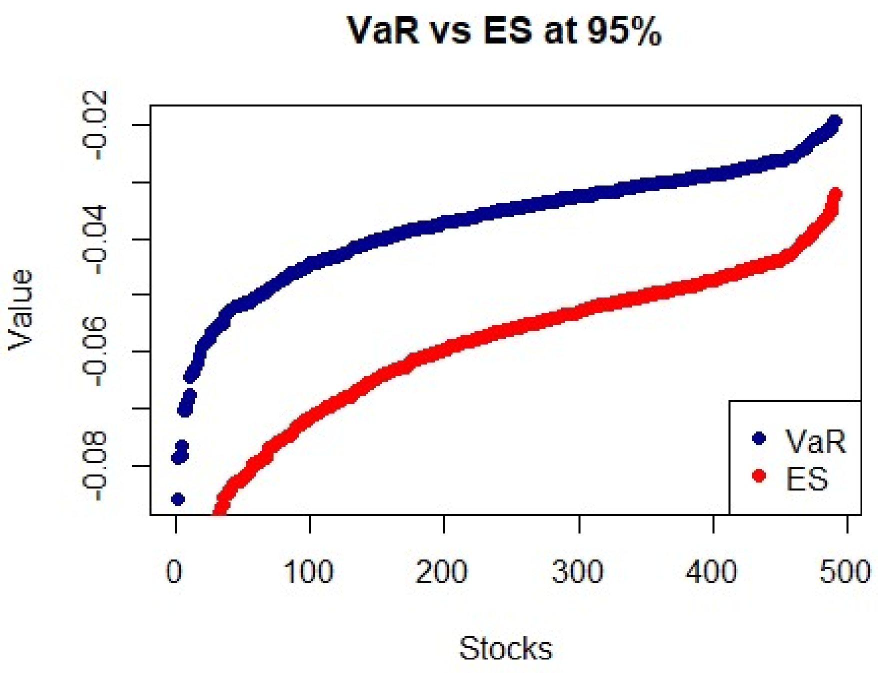
2023-12-17

```r
#                                   CASE STUDY 2

# Giulio Cattoni
# Mario Garcia

# INTRODUCTION:
# We present a solution for the case study n.2 for the period [2020-2022].
# Up to now the normal VaR distribution, is a method that is used in the financial
# markets to make previsions.
# All the statements that we made referred to the good-fitting of the normal VaR
# distribution have to be considered related on the dramatic scenario of Covid-19.

# In order to have a more complete analysis at the end of the case study, we
present
# a deepening, where we choose a different period [2015-17],to understand which
# is the behave of the normal VaR distribution for different levels of confidence.

################################################################################

library(zoo)

##
## Attaching package: 'zoo'

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

library(tseries)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(quantmod)

## Loading required package: xts

## Loading required package: TTR

library(moments)

# Functions
getPrices <- function(TickerSymbols,start,end,type){
  NumberOfStocks <- length(TickerSymbols)
  prices <- get.hist.quote(TickerSymbols[1],start=start,end=end,quote=type)
```

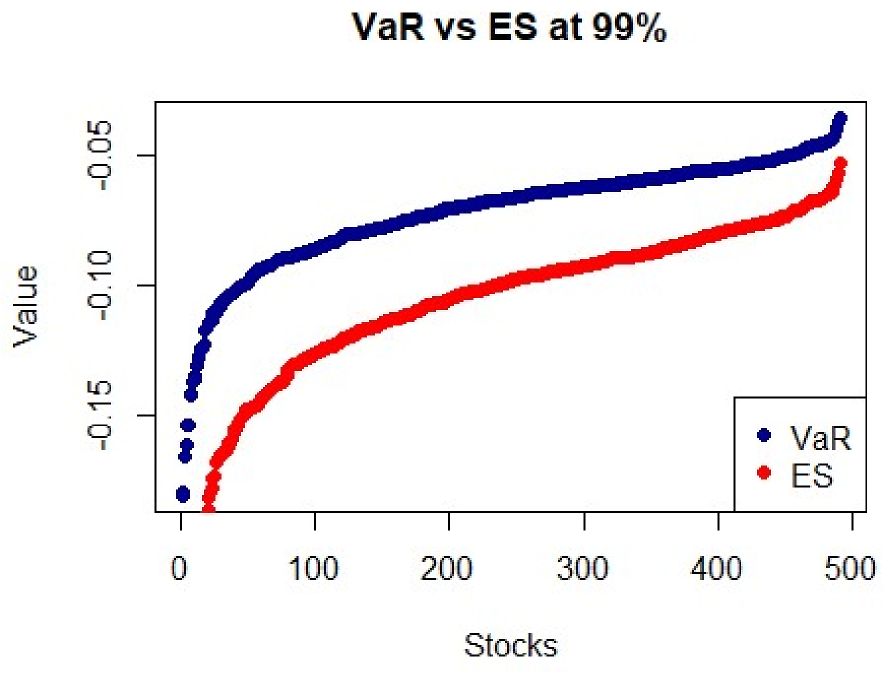
```r
  goodSymbols <- TickerSymbols[1]
  for (d in 2:NumberOfStocks) {
    tryCatch({
      P <- get.hist.quote(TickerSymbols[d],start=start,end=end,quote=type)
      prices <- cbind(prices,P)
      goodSymbols <- c(goodSymbols,TickerSymbols[d])
    }, error=function(err) { print(paste("Download ERROR: ", TickerSymbols[d])) }
)
  }
  prices <- data.frame(coredata(prices))
  colnames(prices) <- goodSymbols
  NumberOfGoodStocks <- dim(prices)[2]
  T <- dim(prices)[1]
  badSymbols <- rep(FALSE,NumberOfGoodStocks)
  for (d in 1:NumberOfGoodStocks) {
    if (is.na(prices[1,d]) || is.na(prices[T,d])) {
      badSymbols[d] <- TRUE
    } else {
      if ( sum(is.na(prices[,d]))>0) {
        print(paste(goodSymbols[d]," NAs filled: ", sum(is.na(prices[,d]))))
        prices[,d]<-na.approx(prices[,d])
      }
    }
  }
  if (sum(badSymbols)>0){
    prices <- prices[!badSymbols]
    print(paste("Removed due to NAs: ", goodSymbols[badSymbols]))
  }
  if ( sum(is.na(prices))==0 ) {
    if (sum(prices == 0) > 0) {print("Check Zeros!")}
  } else {print("Check NAs and Zeros")}

  prices

}

Expected_Shortfall_95 <- function(returns,alpha=0.95){
  N <- length(returns)
  VaR<-sort(coredata(returns))[ceiling(N*(1-alpha))]
  ES <- sum(returns[returns<=VaR])/length(returns[returns<=VaR])
  colnames(ES)<- colnames(returns)
  return(ES)
}
ValueAtRisk_95 <- function(returns,alpha=0.95){
  N = length(returns)
  sorted_return<- sort(returns)
  var<- sorted_return[ceiling(N*(1-alpha))]
  colnames(var)<- colnames(returns)
  return(var)

}
```

```r
Expected_Shortfall_99 <- function(returns,alpha=0.99){
  N <- length(returns)
  VaR<-sort(coredata(returns))[ceiling(N*(1-alpha))]
  ES <- sum(returns[returns<=VaR])/length(returns[returns<=VaR])
  colnames(ES)<- colnames(returns)
  return(ES)
}
ValueAtRisk_99 <- function(returns,alpha=0.99){
  N = length(returns)
  sorted_return<- sort(returns)
  var<- sorted_return[ceiling(N*(1-alpha))]
  colnames(var)<- colnames(returns)
  return(var)
}

NewVarFunction95 <- function(returns){
  var<- mean(returns)-1.645*sd(returns)
  return(var)
}

NewVarFunction99 <- function(returns){
  var<- mean(returns)-2.326*sd(returns)
  return(var)
}

getReturns <- function(prices) {
  NumberOfStocks <- dim(prices)[2]
  length <- dim(prices)[1]
  returns <- matrix(rep(0,NumberOfStocks*(length-1)), ncol=NumberOfStocks,
nrow=length-1)
  for (ind in 1:NumberOfStocks) {
    returns[,ind] <- diff(log(prices[,ind]))
  }
  colnames(returns)<- colnames(prices)
  returns
}
################################################################################

# Get symbols
SP500 <- read.table("C:/Users/catto/OneDrive/Desktop/HfWU/Geisinger - Financial
Analytics/Theory/SP500Ticker.csv",sep=";",header=TRUE)
Symbols <- SP500[,1]


# Get closing prices
Close <- getPrices(Symbols,start = "2020-01-01",end = "2022-12-31","Close")

# Calculate returns
Returns <- getReturns(Close)
names(Returns) <- names(Close)
```

```r
# Calculate VaR and ES at 95%
VAR95 <- apply(Returns,2,ValueAtRisk_95)
ES95 <- apply(Returns,2,Expected_Shortfall_95)

# Apply is a command that for a matrix "Returns" through
# the columns "2", enforces the function "ValueAtRisk_95".


# Plotting sorted VaR and ES, we can confirm that ES is always lower than the VaR
# compared to the same quantile.
plot(sort(VAR95),col="darkblue",type = "p",pch=16, main="VaR vs ES at 95%",
xlab="Stocks", ylab="Value")
points(sort(ES95),col="red",pch=16)
legend("bottomright", legend=c("VaR", "ES"), col=c("darkblue", "red"), pch=16)
```

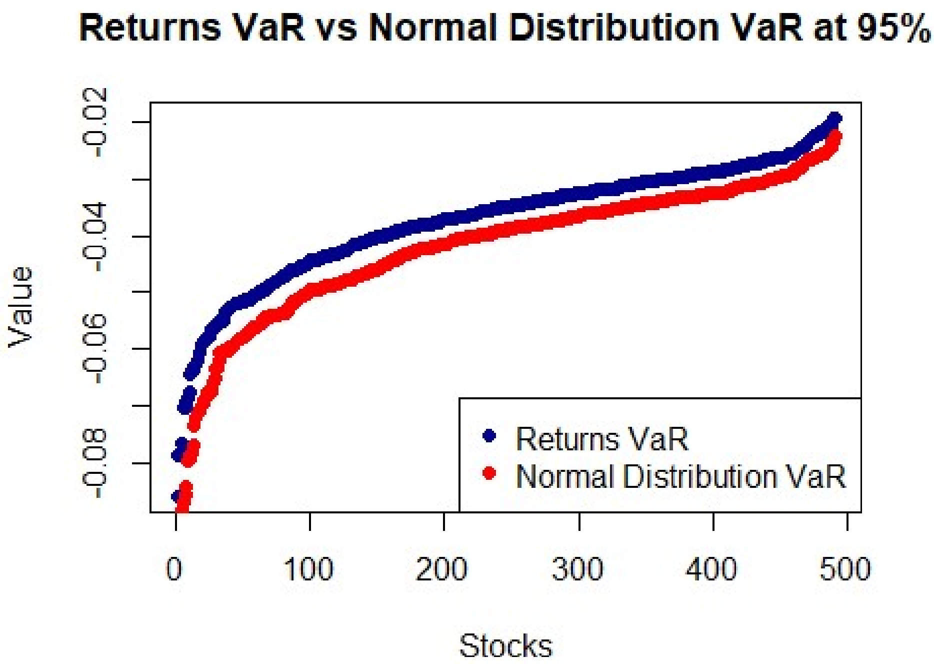



```r
# Calculate VaR and ES at 99%
VAR99 <- apply(Returns,2,ValueAtRisk_99)
ES99 <- apply(Returns,2,Expected_Shortfall_99)


# Plot VaR and ES at 99%; as the previous ES is always lower.
plot(sort(VAR99),col="darkblue",type = "p",pch=16, main="VaR vs ES at 99%",
xlab="Stocks", ylab="Value")
points(sort(ES99),col="red",pch=16)
legend("bottomright", legend=c("VaR", "ES"), col=c("darkblue", "red"),pch=16)
```

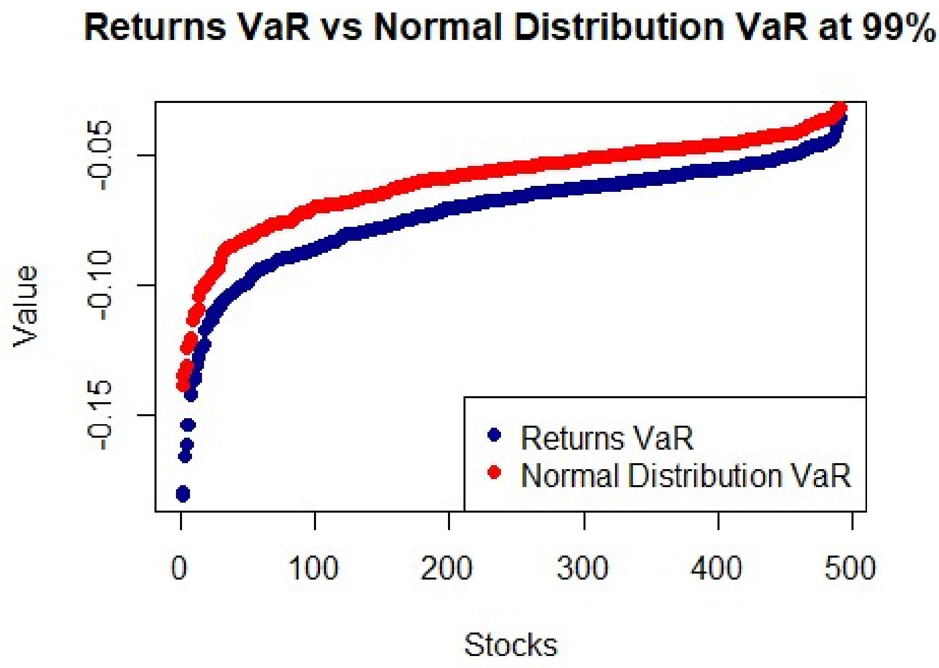



```r
# Calculate the mean and standard deviation of the returns
MeanRet<- apply(Returns,2,mean)
STDRet <- apply(Returns,2,sd)


# Generate a normal distribution with the same mean and standard deviation as
# the matrix of returns.
set.seed(1998)
NormDistr <- matrix(nrow = nrow(Returns), ncol = ncol(Returns))
for (i in 1:ncol(Returns)) {
  NormDistr[,i] <- rnorm(n = nrow(Returns), mean = MeanRet[i], sd = STDRet[i])
}

# Before running rnorm command, we need to set a seed. In this way can identify
# the same outliers, everytime we run the code.


# Calculate the VaR for the normal distribution
NormVar95 <- apply(NormDistr,2,NewVarFunction95)
NormVar99 <- apply(NormDistr,2,NewVarFunction99)


# Plot VaR for the returns and the normal distribution.
plot(sort(VAR95),col="darkblue",type = "p",pch=16, main="Returns VaR vs Normal
Distribution VaR at 95%", xlab="Stocks", ylab="Value")
points(sort(NormVar95),col="red",pch=16)
```

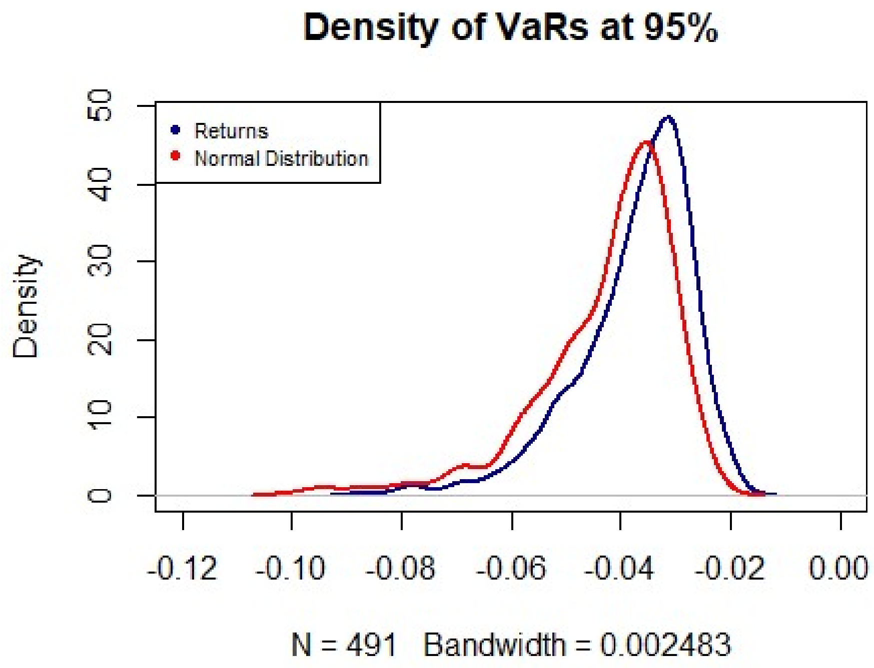
```r
legend("bottomright", legend=c("Returns VaR", "Normal Distribution VaR"),
col=c("darkblue", "red"), pch=16)
```



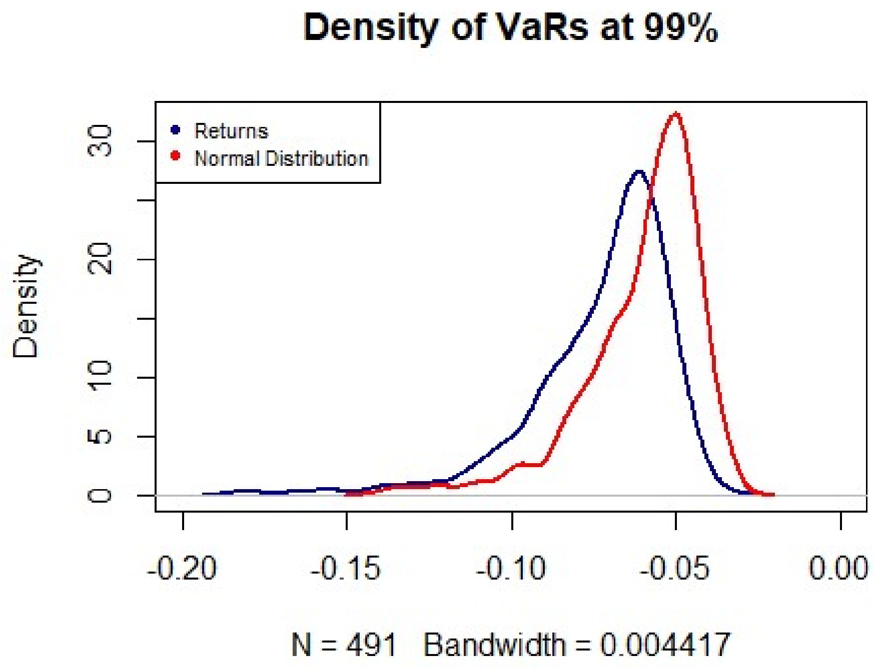

```r
plot(sort(VAR99),col="darkblue",type = "p",pch=16, main="Returns VaR vs Normal
Distribution VaR at 99%", xlab="Stocks", ylab="Value")
points(sort(NormVar99),col="red",pch=16)
legend("bottomright", legend=c("Returns VaR", "Normal Distribution VaR"),
col=c("darkblue", "red"), pch=16)
```



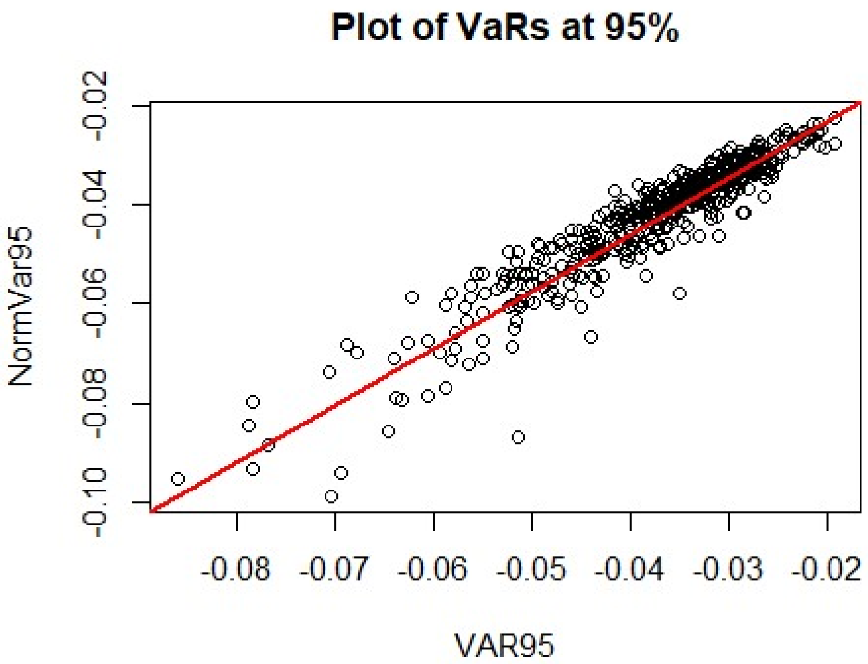

```
# We observe that we are plotting in the negative ordinate, the NormVaR is lower
# than the Statistical VaR calculated from the real returns with alpha = 0.95,
# that suggest us a conservative way to forecast it.
# At alpha=99%, the NormVar is higher indicating an underestimation of the risk,
# and this is because the real returns have heavier tails than the normal
distribution.
# We are expecting that the mean kurtosis of the outliers and of the real returns
# is greater than the normal distribution's kurtosis.
# Extreme events are more common (as in this pandemic case) than predicted
# by the normal model, which becomes evident at higher confidence
# levels.
# In order to have also a graphical answer we could plot the density of both
# distribution:

plot(density(VAR95),main = "Density of VaRs at 95%",col="darkblue",xlim=c(-
0.12,0),lwd=2)
lines(density(NormVar95),col="red",lwd=2)
legend("topleft",legend=c("Returns", "Normal Distribution"),cex=0.7,
col=c("darkblue", "red"),pch=16)
```



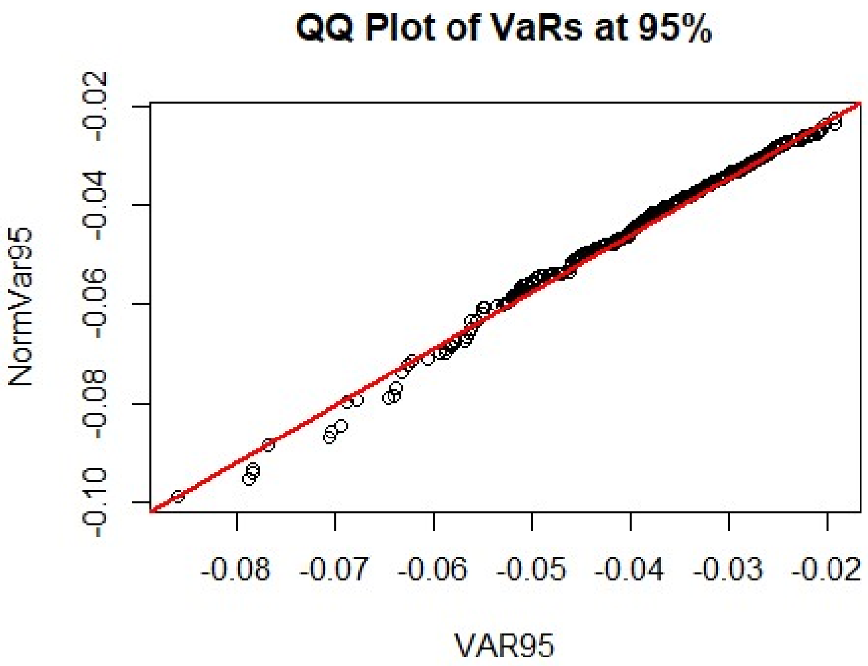

```r
# Red line left of blue suggests that VAR95 data may not fit a normal distribution
# well.

plot(density(VAR99),main = "Density of VaRs at 99%",col="darkblue",xlim=c(-
0.20,0),ylim=c(0,32),lwd=2)
lines(density(NormVar99),col="red",lwd=2)
legend("topleft",legend=c("Returns", "Normal Distribution"),cex=0.7,
col=c("darkblue", "red"),pch=16)
```




```
# The tails of statistical VaR distribution are really heavier.


# IDENTIFY OUTLIERS
# We use the command qqplot to compare quantile per quantile the
# distribution of the real returns with its normal distribution.
# If there are points that are far from the abline, those are outliers.
plot(VAR95,NormVar95, main="Plot of VaRs at 95%",xlab="VAR95",ylab="NormVar95")
abline(a=0,b=1.15,lwd=2,col="red") #y = bx + a
```



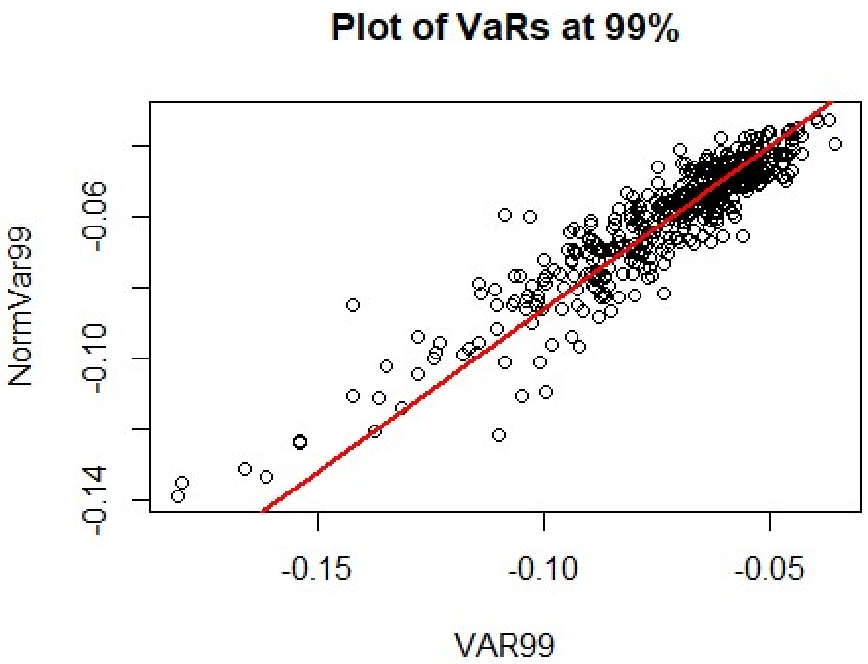

```r
qqplot(VAR95,NormVar95, main="QQ Plot of VaRs at
95%",xlab="VAR95",ylab="NormVar95")
abline(a=0,b=1.15,lwd=2,col="red") #y = bx + a
```



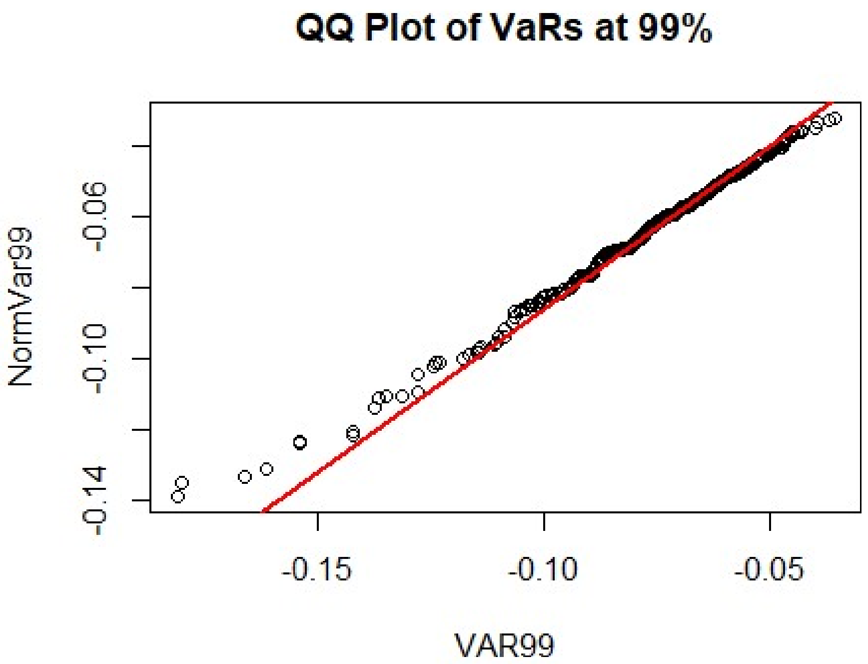
```r
# In order to applicate STD method to find the value of outliers, we have to know
# their distance from the abline.
# ax + by + c = 0
Distance95 <- abs((-1.15*VAR95+NormVar95)/sqrt(1+1.15^2))
meanDistance95 <- mean(Distance95)
stdDistance95 <- sd(Distance95)


# We choose a variable (2) to create the thresholds in order to identify the
outliers.

minDistance95 <- meanDistance95-(2*stdDistance95)
maxDistance95 <- meanDistance95+(2*stdDistance95)


#The outliers need to be outside the thresholds
Outliers95 <- Distance95[which(Distance95 < minDistance95 | Distance95 >
maxDistance95)]


# Transforming the names of outliers in a string vector
NamesOutliers95 <- names(Outliers95)
NamesOutliers95

##  [1] "AAL"  "APA"  "AXON" "BIO"  "BIIB" "BG"   "CZR"  "ENPH" "GEN"  "GNRC"
## [11] "LRCX" "MRO"  "OXY"  "OKE"  "RL"   "TRGP" "VTR"  "VFC"

# Get the returns of the outliers
OutReturns95 <- Returns[,NamesOutliers95]


# Calculate the kurtosis and the minimum of the returns of the outliers
KurtOut95 <- apply(OutReturns95,2,kurtosis)
MinOut95 <- apply(OutReturns95,2,min)


# Calculate the mean of the kurtosis and the min of the the outliers
MeanKurtOut95 <- mean(KurtOut95)
MeanMinOut95 <- mean(MinOut95)


# Calculate the kurtosis and the min of the normal VaR distribution
KurtNorm <- apply(NormDistr,2,kurtosis)
MinNorm <- apply(NormDistr,2,min)


# Calculate the mean of the kurtosis and the min of the normal distribution
MeanKurtNorm <- mean(KurtNorm)
MeanMinNorm <- mean(MinNorm)
```

```r
# Calculate the kurtosis and min of the returns
KurtReturns <- apply(Returns,2,kurtosis)
MinReturns <- apply(Returns,2,min)


# Calculate the mean of the kurtosis of the returns
MeanKurtReturns <- mean(KurtReturns)
MeanMinReturns <- mean(MinReturns)


# Excluding outliers to see how is their impact on the mean kurtosis and minimal
# values in the statistical distribution.
CloseWO95<- Close[, !names(Close)%in%NamesOutliers95]
ReturnsWO95<- getReturns(CloseWO95)
KurtReturnsWO95 <- apply(ReturnsWO95,2,kurtosis)
MinReturnsWO95 <- apply(ReturnsWO95,2,min)
MeanKurtReturnsWO95 <- mean(KurtReturnsWO95)
MeanMinReturnsWO95 <- mean(MinReturnsWO95)


# Creating data frame to visualize the statistical measures.
Comparison_Out95_Returns<-
data.frame(MeanKurtOut95,MeanKurtReturns,MeanMinOut95,MeanMinReturns)
Comparison_Out95_Norm<-
data.frame(MeanKurtOut95,MeanKurtNorm,MeanMinOut95,MeanMinNorm)
Comparison_Norm_Returns <-
data.frame(MeanKurtNorm,MeanKurtReturns,MeanMinNorm,MeanMinReturns)
Comparison_ReturnsWO95_Returns <-
data.frame(MeanKurtReturnsWO95,MeanKurtReturns,MeanMinReturnsWO95,MeanMinReturns)
Comparison_Norm_ReturnsWO95 <-
data.frame(MeanKurtNorm,MeanKurtReturnsWO95,MeanMinNorm,MeanMinReturnsWO95)

Comparison_Out95_Returns

##   MeanKurtOut95 MeanKurtReturns MeanMinOut95 MeanMinReturns
## 1      33.42339        13.88552   -0.3793521     -0.1859536

Comparison_Out95_Norm

##   MeanKurtOut95 MeanKurtNorm MeanMinOut95 MeanMinNorm
## 1      33.42339     2.998544   -0.3793521 -0.08104994

# Of course the outliers have a worse performance than the mean of both
distributions.

Comparison_Norm_Returns

##   MeanKurtNorm MeanKurtReturns MeanMinNorm MeanMinReturns
## 1     2.998544        13.88552 -0.08104994     -0.1859536
```

```
# We can see how far is the normal model to a correct approximation of the
measures
# of this period.

Comparison_ReturnsWO95_Returns

##   MeanKurtReturnsWO95 MeanKurtReturns MeanMinReturnsWO95 MeanMinReturns
## 1            13.14201        13.88552         -0.1785939     -0.1859536

Comparison_Norm_ReturnsWO95

##   MeanKurtNorm MeanKurtReturnsWO95 MeanMinNorm MeanMinReturnsWO95
## 1     2.998544            13.14201 -0.08104994         -0.1785939

# In this comparison analysis we can observe that there is a significant
# difference between the measures of the normal model and the real events.
# As the graphics said to us before, the normal model doesn't consider the extreme
# events.
# Exclusion of outliers appears to reduce kurtosis and minimal values,
# but these still remain significantly different from the normal distribution.
# This is enough to justify that a normal distribution isn't a good approximation
# for VaR of real returns.
# However it's really important to specify that SP500 is NOT a equally weighted
# portfolio.
# So, although the metrics may show significant differences from the mean,
# it's not clear if the mean of actual returns differs significantly from the
# mean of returns excluding outliers.


#We repeat the analysis for the confidence level of 99%.
plot(VAR99,NormVar99, main="Plot of VaRs at 99%",xlab="VAR99",ylab="NormVar99")
abline(a=0.006,b=0.92,lwd=2,col="red") #y= bx + a
```



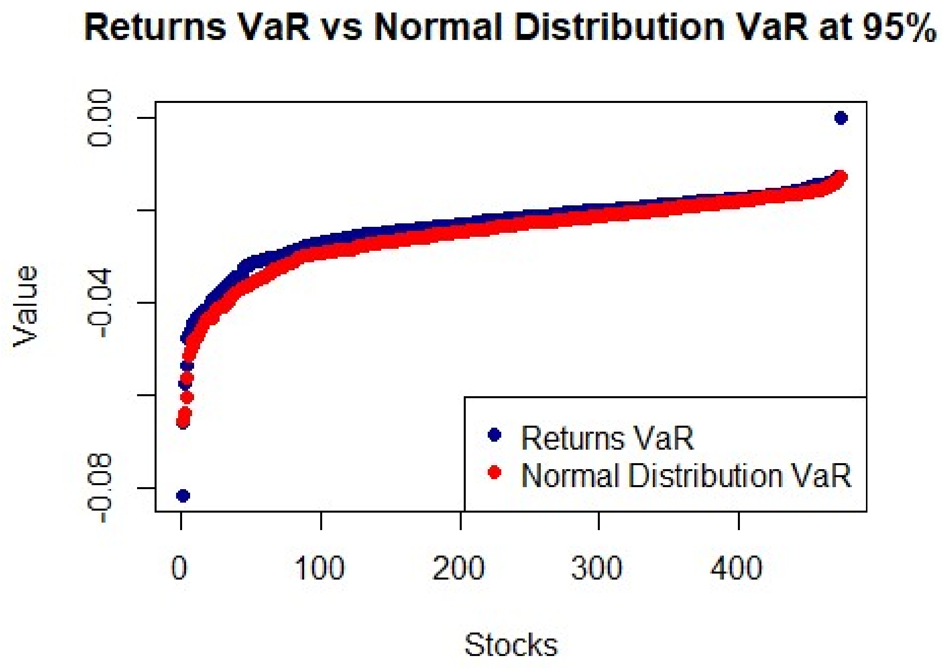

```
qqplot(VAR99,NormVar99, main="QQ Plot of VaRs at
99%",xlab="VAR99",ylab="NormVar99")
abline(a=0.006,b=0.92,lwd=2,col="red") #y= bx + a
```



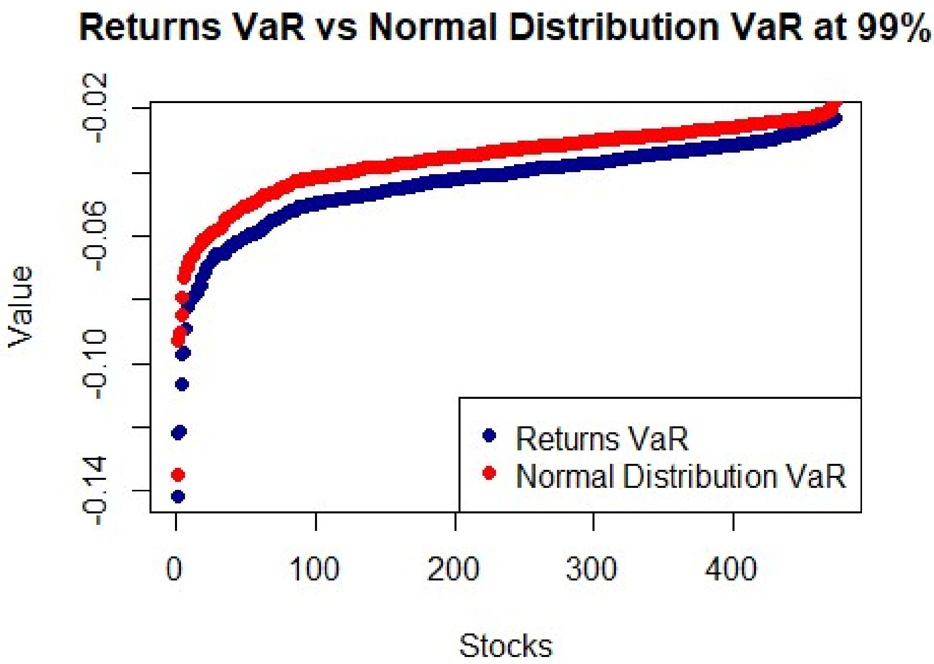
```r
# As we can expected from the comparison of the plot of VAR99 and NormVar99
# we can find outliers in a higher position from the abline, that means that the
# quantiles of statistical distribution are greater than the normal one.

Distance99 <- abs((-0.92*VAR99+NormVar99-0.006)/sqrt(1+0.92^2))
meanDistance99 <- mean(Distance99)
stdDistance99 <- sd(Distance99)
minDistance99 <- meanDistance99-(2*stdDistance99)
maxDistance99 <- meanDistance99+(2*stdDistance99)
Outliers99 <- Distance99[which(Distance99 < minDistance99 | Distance99 >
maxDistance99)]
NamesOutliers99 <- names(Outliers99)
NamesOutliers99

##  [1] "BIIB" "CZR"  "CINF" "DRI"  "FANG" "EPAM" "GEN"  "HPQ"  "LW"   "LYV"
## [11] "MRO"  "NCLH" "OXY"  "OKE"  "PPL"  "RHI"  "ROP"  "SPG"  "SEDG" "WRK"

# Get the returns of the outliers
OutReturns99 <- Returns[,NamesOutliers99]


# Calculate the kurtosis and the minimal returns of the outliers
KurtOut99 <- apply(OutReturns99,2,kurtosis)
MinOut99 <- apply(OutReturns99,2,min)


# Calculate the mean of the kurtosis and the minimal returns of the outliers
MeanKurtOut99 <- mean(KurtOut99)
MeanMinOut99 <- mean(MinOut99)


# Exclude outliers to see how is their impact on the mean kurtosis and minimal
# values in the statistical distribution.
CloseWO99<- Close[, !names(Close)%in%NamesOutliers99]
ReturnsWO99<- getReturns(CloseWO99)
KurtReturnsWO99 <- apply(ReturnsWO99,2,kurtosis)
MinReturnsWO99 <- apply(ReturnsWO99,2,min)
MeanKurtReturnsWO99 <- mean(KurtReturnsWO99)
MeanMinReturnsWO99 <- mean(MinReturnsWO99)


# Make comparisons between outliers
Comparison_Out99_Returns<-
data.frame(MeanKurtOut99,MeanKurtReturns,MeanMinOut99,MeanMinReturns)
Comparison_Out99_Norm<-
data.frame(MeanKurtOut99,MeanKurtNorm,MeanMinOut99,MeanMinNorm)
Comparison_Out95_Out99 <-
data.frame(MeanKurtOut95,MeanKurtOut99,MeanMinOut95,MeanMinOut99)
Comparison_ReturnsWO99_Returns <-
data.frame(MeanKurtReturnsWO99,MeanKurtReturns,MeanMinReturnsWO99,MeanMinReturns)
Comparison_Norm_ReturnsWO99 <-
```

```r
data.frame(MeanKurtNorm,MeanKurtReturnsWO99,MeanMinNorm,MeanMinReturnsWO99)

Comparison_Out95_Returns

##   MeanKurtOut95 MeanKurtReturns MeanMinOut95 MeanMinReturns
## 1      33.42339        13.88552   -0.3793521     -0.1859536

Comparison_Out99_Returns

##   MeanKurtOut99 MeanKurtReturns MeanMinOut99 MeanMinReturns
## 1       31.2417        13.88552   -0.3464359     -0.1859536

Comparison_Out99_Norm

##   MeanKurtOut99 MeanKurtNorm MeanMinOut99 MeanMinNorm
## 1       31.2417     2.998544   -0.3464359 -0.08104994

Comparison_Out95_Out99

##   MeanKurtOut95 MeanKurtOut99 MeanMinOut95 MeanMinOut99
## 1      33.42339       31.2417   -0.3793521   -0.3464359

Comparison_ReturnsWO99_Returns

##   MeanKurtReturnsWO99 MeanKurtReturns MeanMinReturnsWO99 MeanMinReturns
## 1            13.14853        13.88552         -0.1791391     -0.1859536

Comparison_Norm_ReturnsWO99

##   MeanKurtNorm MeanKurtReturnsWO99 MeanMinNorm MeanMinReturnsWO99
## 1     2.998544            13.14853 -0.08104994         -0.1791391

# In this comparison, we observe that increasing alpha level doesn't correspond
# to an increase of their measures of kurtosis and minimal values.
# The reasons behind this discrepancy are:
# - Different outliers for different confidence levels.
# - Kurtosis is more closely related to the shape of the tails of the
distribution.

# However, especially for a 99% confidence level, the difference between the
normal
# distribution and the observed distribution is too significant to assume that
# the normal distribution is a good-fitting model.

################################################################################
```

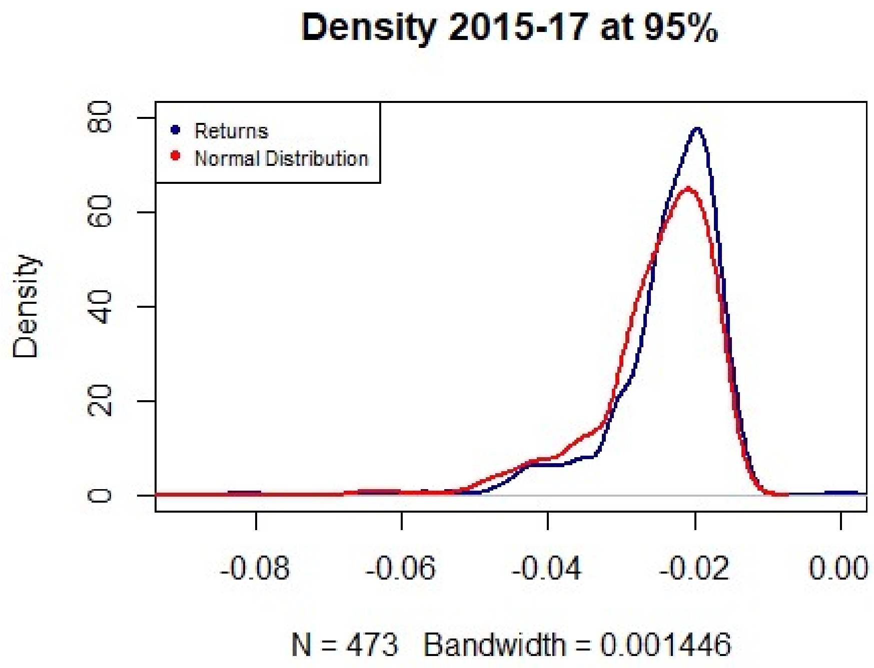
```r
# DEEPENING:
# In order to evaluate the normal VaR distribution in a more specific analysis, we
# should take a different panel of time. In this case we choose a good financial
# period [2015-2017]

# Using the previous analysis we plot the Statistical Vs Normal VaR distribution,
# we renaming the variables in order to exclude cases of overwriting.
SP5001 <- read.table("C:/Users/catto/OneDrive/Desktop/HfWU/Geisinger - Financial
Analytics/Theory/SP500Ticker.csv",sep=";",header=TRUE)
Symbols1 <- SP5001[,1]
Close1 <- getPrices(Symbols1, start = "2015-01-01", end = "2017-12-31", "Close")

Returns1 <- getReturns(Close1)
names(Returns1) <- names(Close1)
VAR951 <- apply(Returns1, 2, ValueAtRisk_95)
ES951 <- apply(Returns1, 2, Expected_Shortfall_95)
VAR991 <- apply(Returns1, 2, ValueAtRisk_99)
ES991 <- apply(Returns1, 2, Expected_Shortfall_99)
MeanRet1 <- apply(Returns1, 2, mean)
STDRet1 <- apply(Returns1, 2, sd)

# We'll use the same seed to obtain the same normal distribution in order to get
# more accuracy. Of course the values are different because they are calculated
# with a different mean and standard deviation of the period [2020-22].

set.seed(1998)
NormDistr1 <- matrix(nrow = nrow(Returns1), ncol = ncol(Returns1))
for (i in 1:ncol(Returns1)) {
  NormDistr1[, i] <- rnorm(n = nrow(Returns1), mean = MeanRet1[i], sd =
STDRet1[i])
}
NormVar951 <- apply(NormDistr1, 2, NewVarFunction95)
NormVar991 <- apply(NormDistr1, 2, NewVarFunction99)

plot(sort(VAR951), col="darkblue", type = "p", pch=16, main="Returns VaR vs Normal
Distribution VaR at 95%", xlab="Stocks", ylab="Value")
points(sort(NormVar951), col="red", pch=16)
legend("bottomright", legend=c("Returns VaR", "Normal Distribution VaR"),
col=c("darkblue", "red"), pch=16)
```



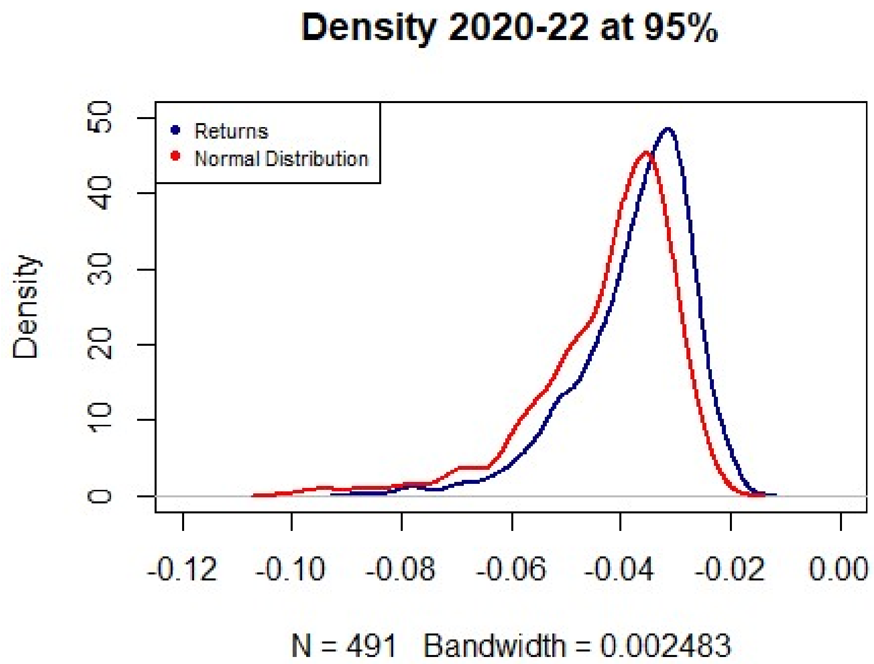

```r
# We observe that in good financial period, for a confidence level of 95%
# the VaR of a normal distribution roughly fit with the Statistical one.
# We also have to say that in the external part of the graphic, there are
# blue points that are really far from the normal distribution. We are expecting
# that they represent the substantially difference between normal and real
returns'
# kurtosis.

plot(sort(VAR991), col="darkblue", type = "p", pch=16, main="Returns VaR vs Normal
Distribution VaR at 99%", xlab="Stocks", ylab="Value")
points(sort(NormVar991), col="red", pch=16)
legend("bottomright", legend=c("Returns VaR", "Normal Distribution VaR"),
col=c("darkblue", "red"), pch=16)
```



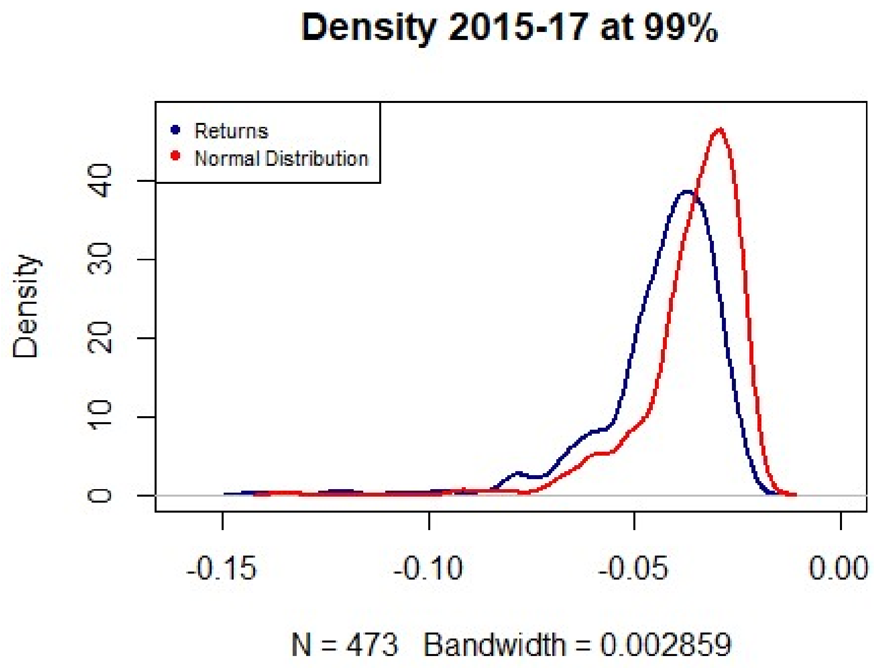

```
# For a confidence level of 99%, despite the period is thriving, the normal model,
# underestimates the probability of extreme events.

plot(density(VAR951),xlim=c(-0.09,0),col="darkblue",lwd=2,main="Density 2015-17 at
95%",ylim=c(0,80))
lines(density(NormVar951),col="red",lwd=2)
legend("topleft",legend=c("Returns", "Normal Distribution"),cex=0.7,
col=c("darkblue", "red"),pch=16)
```



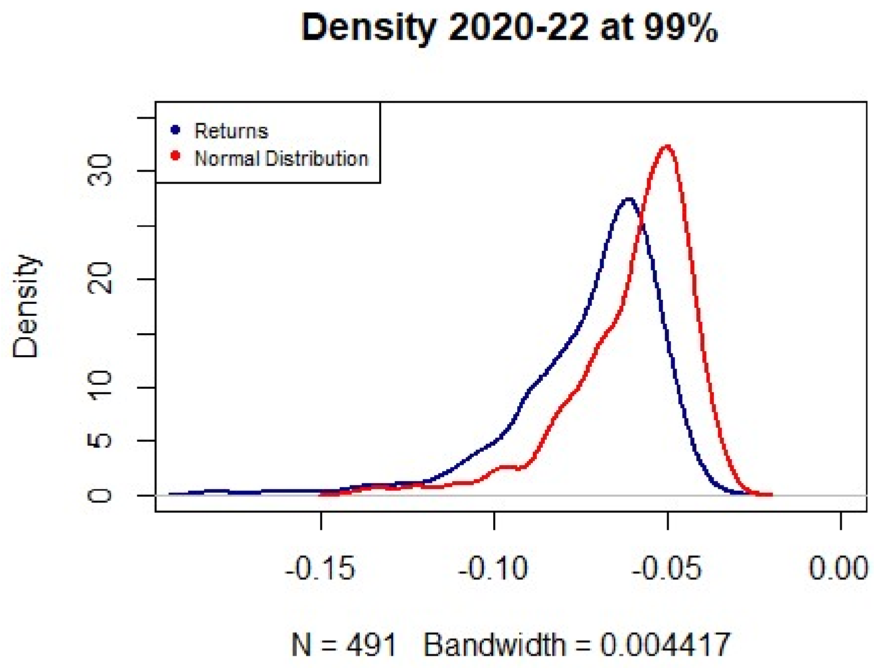

```
plot(density(VAR95),main="Density 2020-22 at 95%",col="darkblue",xlim=c(-
0.12,0),ylim=c(0,50),lwd=2)
lines(density(NormVar95),col="red",lwd=2)
legend("topleft",legend=c("Returns", "Normal Distribution"),cex=0.7,
col=c("darkblue", "red"),pch=16)
```



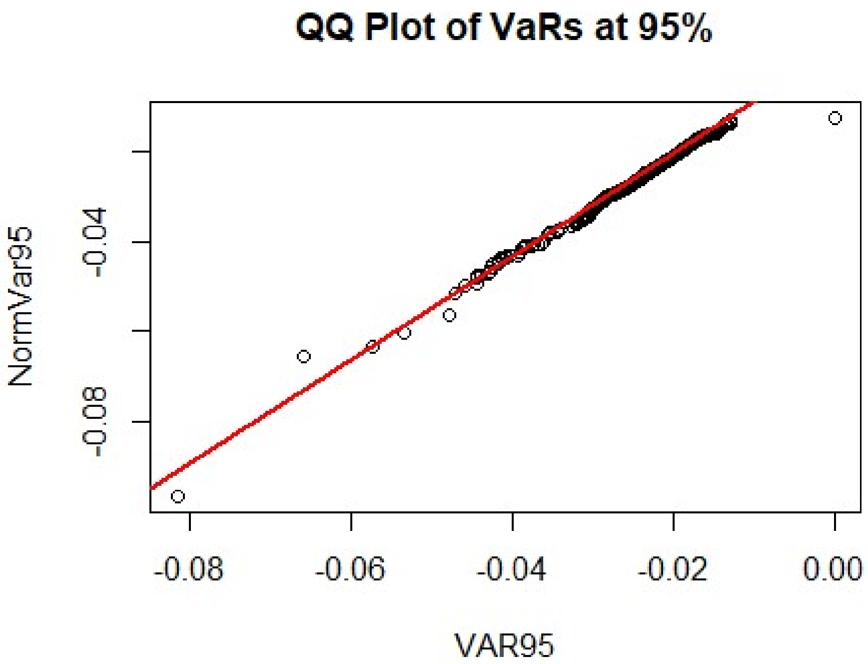

```r
# Here we can see that if we consider a dependence level of 95% normal model
# represent a good-fit-model in order to forecast VaR.
# In a worse case scenario as it is Covid-19, the distribution are more asymmetric
# and not valid to forecast VaR.

plot(density(VAR991),xlim=c(-0.16,0),col="darkblue",lwd=2,main="Density 2015-17 at
99%",ylim=c(0,48))
lines(density(NormVar991),col="red",lwd=2)
legend("topleft",legend=c("Returns", "Normal Distribution"),cex=0.7,
col=c("darkblue", "red"),pch=16)
```




```
plot(density(VAR99),main="Density 2020-22 at 99%",col="darkblue",xlim=c(-
0.19,0),ylim=c(0,35),lwd=2)
lines(density(NormVar99),col="red",lwd=2)
legend("topleft",legend=c("Returns", "Normal Distribution"),cex=0.7,
col=c("darkblue", "red"),pch=16)
```

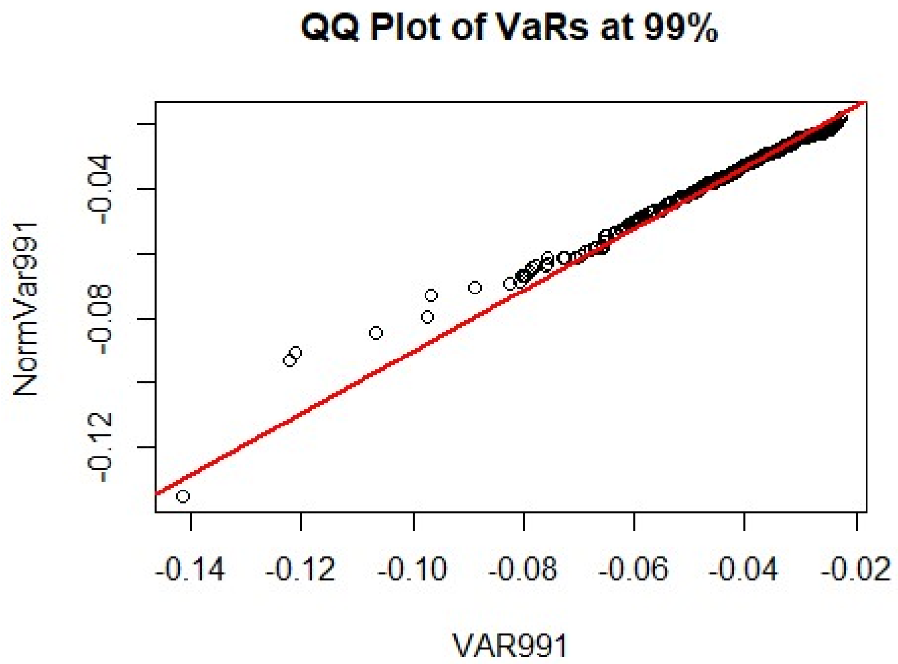



```
# At 99% the normal distribution is totally disaligned with the statistical one
# in all scenarios. We can assume that for an higher level of confidence
# the normal distribution of VaRs isn't useful because doesn't consider the
# more extreme event.

# IDENTIFY OUTLIERS
qqplot(VAR951,NormVar951, main="QQ Plot of VaRs at 95%", xlab="VAR95",
ylab="NormVar95")
abline(a=0.0025, b=1.15, lwd=2, col="red") #y = bx + a
```




```r
# This qqplot is the best in terms of outliers that we will see in this analysis
Distance951 <- abs((-1.15 * sort(VAR951) + sort(NormVar951)-0.0025) / sqrt(1 +
1.15^2))
meanDistance951 <- mean(Distance951)
stdDistance951 <- sd(Distance951)
minDistance951 <- meanDistance951 - (2 * stdDistance951)
maxDistance951 <- meanDistance951 + (2 * stdDistance951)
Outliers951 <- Distance951[which(Distance951 < minDistance951 | Distance951 >
maxDistance951)]
NamesOutliers951 <- names(Outliers951)
NamesOutliers951

## [1] "ENPH" "FCX"  "MU"   "AMCR"

# There are only 4 outliers in a 491 stocks.
OutReturns951 <- Returns1[,NamesOutliers951]
KurtOut951 <- apply(OutReturns951, 2, kurtosis)
MinOut951 <- apply(OutReturns951, 2, min)
MeanKurtOut951 <- mean(KurtOut951)
MeanMinOut951 <- mean(MinOut951)
KurtNorm1 <- apply(NormDistr1, 2, kurtosis)
MinNorm1 <- apply(NormDistr1, 2, min)
MeanKurtNorm1 <- mean(KurtNorm1)
MeanMinNorm1 <- mean(MinNorm1)
KurtReturns1 <- apply(Returns1,2,kurtosis)
MinReturns1 <- apply(Returns1,2,min)
MeanKurtReturns1 <- mean(KurtReturns1)
MeanMinReturns1 <- mean(MinReturns1)
```

```r
CloseWO951<- Close1[, !names(Close1)%in%NamesOutliers951]
ReturnsWO951<- getReturns(CloseWO951)
KurtReturnsWO951 <- apply(ReturnsWO951,2,kurtosis)
MinReturnsWO951 <- apply(ReturnsWO951,2,min)
MeanKurtReturnsWO951 <- mean(KurtReturnsWO951)
MeanMinReturnsWO951 <- mean(MinReturnsWO951)
Comparison_Out951_Returns1<-
data.frame(MeanKurtOut951,MeanKurtReturns1,MeanMinOut951,MeanMinReturns1)
Comparison_Norm1_Returns1 <-
data.frame(MeanKurtNorm1,MeanKurtReturns1,MeanMinNorm1,MeanMinReturns1)
Comparison_ReturnsWO951_Returns1 <-
data.frame(MeanKurtReturnsWO951,MeanKurtReturns1,MeanMinReturnsWO951,MeanMinReturn
s1)
Comparison_Norm1_ReturnsWO951 <-
data.frame(MeanKurtNorm1,MeanKurtReturnsWO951,MeanMinNorm1,MeanMinReturnsWO951)
Comparison_Out951_Returns1

##   MeanKurtOut951 MeanKurtReturns1 MeanMinOut951 MeanMinReturns1
## 1       33.95679         12.09101    -0.2467254       -0.100548

Comparison_Norm1_Returns1

##   MeanKurtNorm1 MeanKurtReturns1 MeanMinNorm1 MeanMinReturns1
## 1      2.998906         12.09101  -0.04894817       -0.100548

Comparison_ReturnsWO951_Returns1

##   MeanKurtReturnsWO951 MeanKurtReturns1 MeanMinReturnsWO951 MeanMinReturns1
## 1             11.90452         12.09101         -0.09930125       -0.100548

Comparison_Norm1_ReturnsWO951

##   MeanKurtNorm1 MeanKurtReturnsWO951 MeanMinNorm1 MeanMinReturnsWO951
## 1      2.998906             11.90452  -0.04894817         -0.09930125

# We repeat the analysis for a level of confidence of 99%
qqplot(VAR991,NormVar991, main="QQ Plot of VaRs at 99%", xlab="VAR991",
ylab="NormVar991")
abline(a=0.0045, b=0.95, lwd=2, col="red") # y = bx + a
```




```
Distance991 <- abs((-0.95 * sort(VAR991) + sort(NormVar991) - 0.0045) / sqrt(1 +
0.95^2))
meanDistance991 <- mean(Distance991)
stdDistance991 <- sd(Distance991)
minDistance991 <- meanDistance991 - (2 * stdDistance991)
maxDistance991 <- meanDistance991 + (2 * stdDistance991)
Outliers991 <- Distance991[which(Distance991 < minDistance991 | Distance991 >
maxDistance991)]
NamesOutliers991 <- names(Outliers991)
NamesOutliers991

##  [1] "ENPH" "WMB"  "TRGP" "FCX"  "NRG"  "AMD"  "MRO"  "NEM"  "STX"  "MU"
## [11] "NOW"

OutReturns991 <- Returns[,NamesOutliers991]
KurtOut991 <- apply(OutReturns991, 2, kurtosis)
MinOut991 <- apply(OutReturns991, 2, min)
MeanKurtOut991 <- mean(KurtOut991)
MeanMinOut991 <- mean(MinOut991)
CloseWO991<- Close1[, !names(Close1)%in%NamesOutliers991]
ReturnsWO991<- getReturns(CloseWO991)
KurtReturnsWO991 <- apply(ReturnsWO991,2,kurtosis)
MinReturnsWO991 <- apply(ReturnsWO991,2,min)
MeanKurtReturnsWO991 <- mean(KurtReturnsWO991)
MeanMinReturnsWO991 <- mean(MinReturnsWO991)
Comparison_Out991_Returns1<-
data.frame(MeanKurtOut991,MeanKurtReturns1,MeanMinOut991,MeanMinReturns1)
Comparison_Out951_Out991 <-
```

```
data.frame(MeanKurtOut951,MeanKurtOut991,MeanMinOut951,MeanMinOut991)
Comparison_ReturnsWO991_Returns1 <-
data.frame(MeanKurtReturnsWO991,MeanKurtReturns1,MeanMinReturnsWO991,MeanMinReturn
s1)
Comparison_Norm1_ReturnsWO991 <-
data.frame(MeanKurtNorm1,MeanKurtReturnsWO991,MeanMinNorm1,MeanMinReturnsWO991)
Comparison_Out951_Returns1

##   MeanKurtOut951 MeanKurtReturns1 MeanMinOut951 MeanMinReturns1
## 1       33.95679         12.09101    -0.2467254       -0.100548

Comparison_Out991_Returns1

##   MeanKurtOut991 MeanKurtReturns1 MeanMinOut991 MeanMinReturns1
## 1       19.42016         12.09101    -0.2885686       -0.100548

Comparison_Out951_Out991

##   MeanKurtOut951 MeanKurtOut991 MeanMinOut951 MeanMinOut991
## 1       33.95679       19.42016    -0.2467254    -0.2885686

Comparison_ReturnsWO991_Returns1

##   MeanKurtReturnsWO991 MeanKurtReturns1 MeanMinReturnsWO991 MeanMinReturns1
## 1             12.03641         12.09101         -0.09712043       -0.100548

Comparison_Norm1_ReturnsWO991

##   MeanKurtNorm1 MeanKurtReturnsWO991 MeanMinNorm1 MeanMinReturnsWO991
## 1      2.998906             12.03641  -0.04894817         -0.09712043

# From the data, we observe an increase of kurtosis for both outliers and returns
# from 2015-2017 to 2020-2022. This suggests a "heavier" return distribution over
time,
# indicating more extreme values.
# That is made as a confirm of 2015-17 is a better case scenary. There were also
# geopolitical events like Brexit or american election, that could affect
# the outliers of this distribution, but they had a less relevant impact on the
# returns despite of the pandemic scenary.

################################################################################

# CONCLUSION:
# In conclusion, our analysis revealed that, in extreme market conditions,
# it might not be the optimal strategy to predict Value at Risk (VaR) for both
# 95% and 99% confidence levels using a normal distribution.
# In other conditions, as we have seen in the deepening, the normal VaR
distribution
# appeared to work efficiently if compared with a better case scenario.
# There was a acceptable fit as seen by the Normal VaR values, which matched
# approximatively  with the VaR calculated for real returns.
# But when we analyise the 99% confidence level there were some notable variations
# between the VaR distribution based on real returns and the VaR at 99% normal
```

```
# distribution. The statistical VaR distribution's tails were heavier indicating
# that at this confidence level, the normal model can't get the quantity and
# impact of the extreme events.
# From the analysis of outliers, we registered a significantly difference on
# the statistical measures of kurtosis and minimal values from what we expected
# by the normal model, and what represent the reality.
# In addition, the density plots showed that, in comparison to the normal
distribution,
# the statistical VaR distribution presented a larger tail spread.
# This illustrates how hard it was for the standard model to accurately represent
# the risk of extremely abnormal movements in markets, especially in times of
# crisis like the one we examined.

################################################################################
```