Ex 1 :

Nohalito

2025-11-27

1°/ Data extraction :

```
# Basic stats
summary(df)
```

```
##   asset_excess      market_excess     
##  Min.   :-1.63907   Min.   :-1.667957  
##  1st Qu.:-0.08515   1st Qu.:-0.067744  
##  Median :-0.01463   Median :-0.000821  
##  Mean   : 0.02419   Mean   : 0.025333  
##  3rd Qu.: 0.09179   3rd Qu.: 0.087267  
##  Max.   : 1.90657   Max.   : 1.961963  
```

```
# Moments
apply(df, 2, skewness)
```

```
##  asset_excess market_excess 
##     0.5748387     0.5037051 
```

```
apply(df, 2, kurtosis)
```

```
##  asset_excess market_excess 
##      10.53346      11.57554 
```

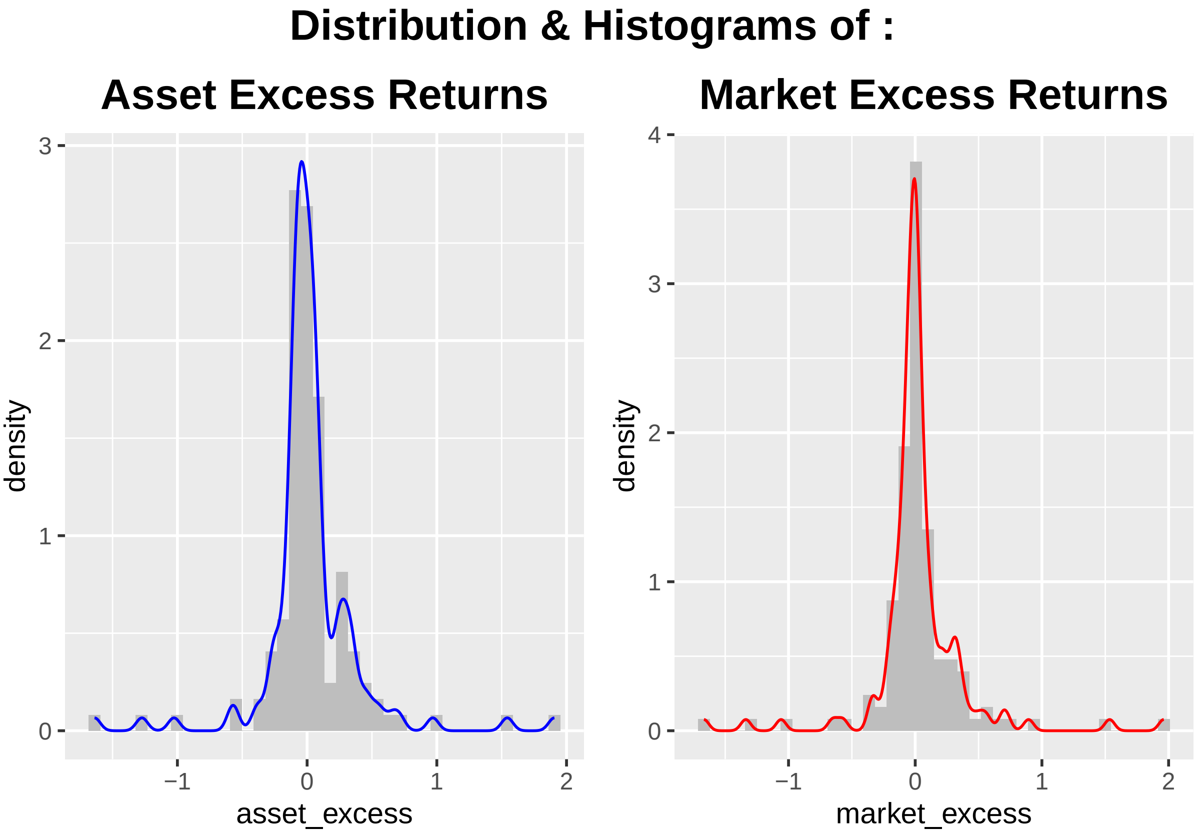
2°/ EDA :

2.1°/ Distribution plot :

```
p1 <- ggplot(df, aes(x = asset_excess)) +
  geom_histogram(aes(y = ..density..), bins = 40, fill = "grey") +
  geom_density(color = "blue") +
  ggtitle("Asset Excess Returns")

p2 <- ggplot(df, aes(x = market_excess)) +
  geom_histogram(aes(y = ..density..), bins = 40, fill = "grey") +
  geom_density(color = "red") +
  ggtitle("Market Excess Returns")

(p1 | p2) +
  plot_annotation(
    title = "Distribution & Histograms of : "
  ) &
  theme(
```

```r
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

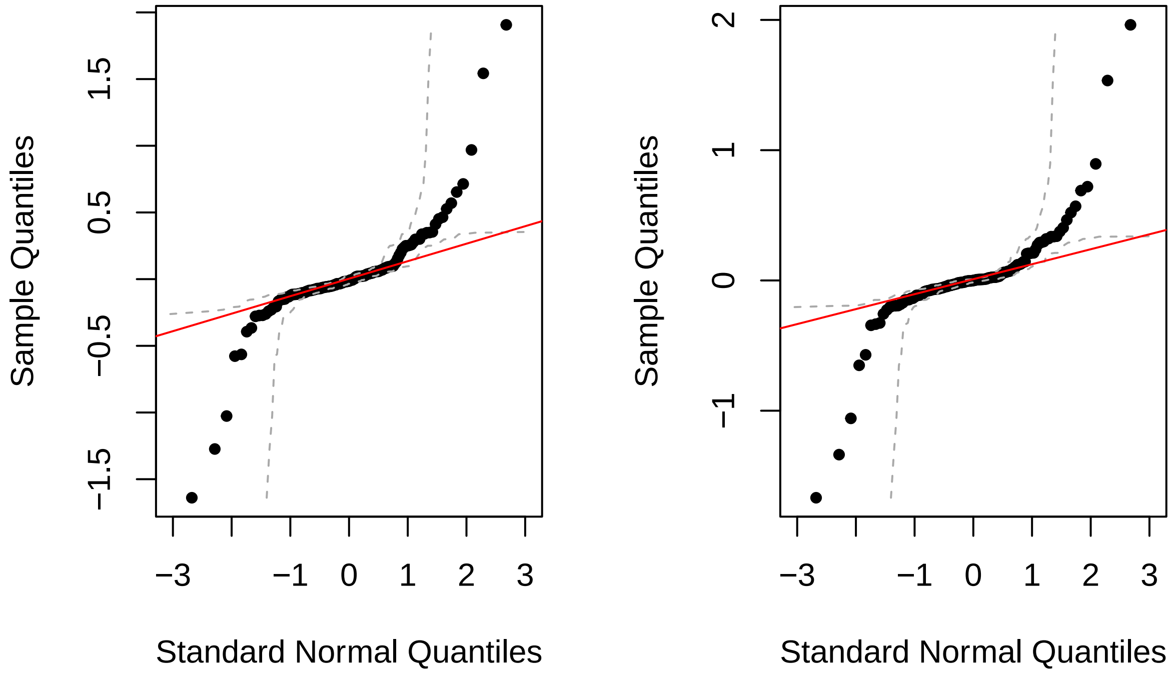


2.2°/ QQ-plot :

```r
# Touche moi le QQ-plot
par(mfrow = c(1, 2))

qqnorm(df$asset_excess) +
  qqline(df$asset_excess, col = "red")
```

```
## [1] lower upper qnorm data
## <0 lignes> (ou 'row.names' de longueur nulle)
```

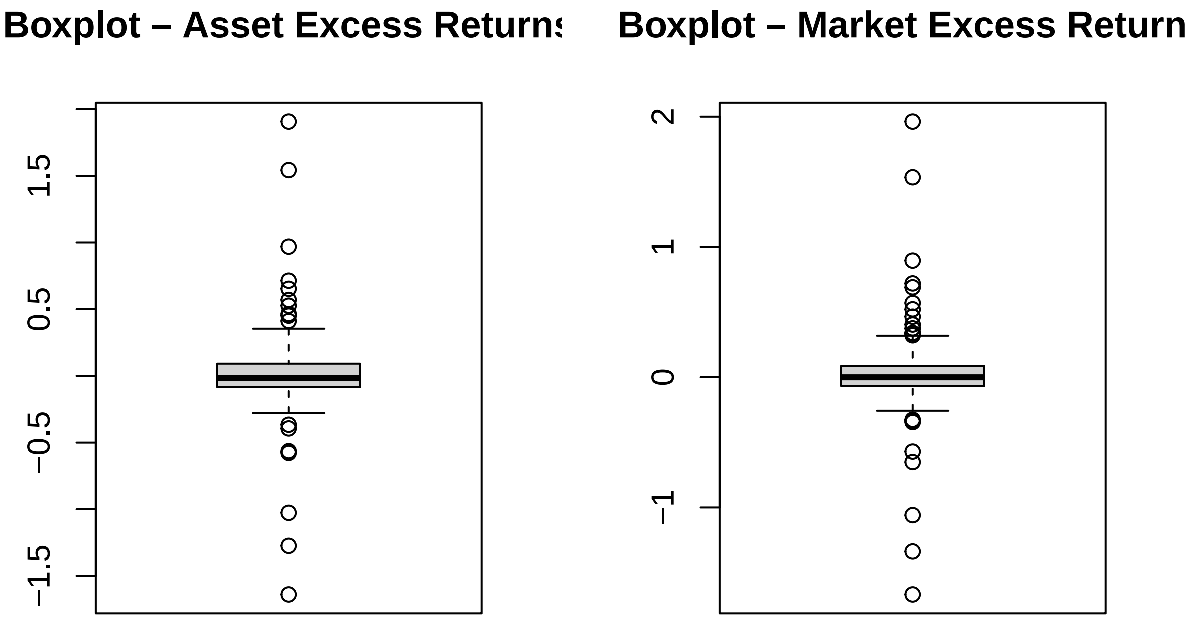
```r
qqnorm(df$market_excess) +
  qqline(df$market_excess, col = "red")
```



```
## [1] lower upper qnorm data
## <0 lignes> (ou 'row.names' de longueur nulle)
```

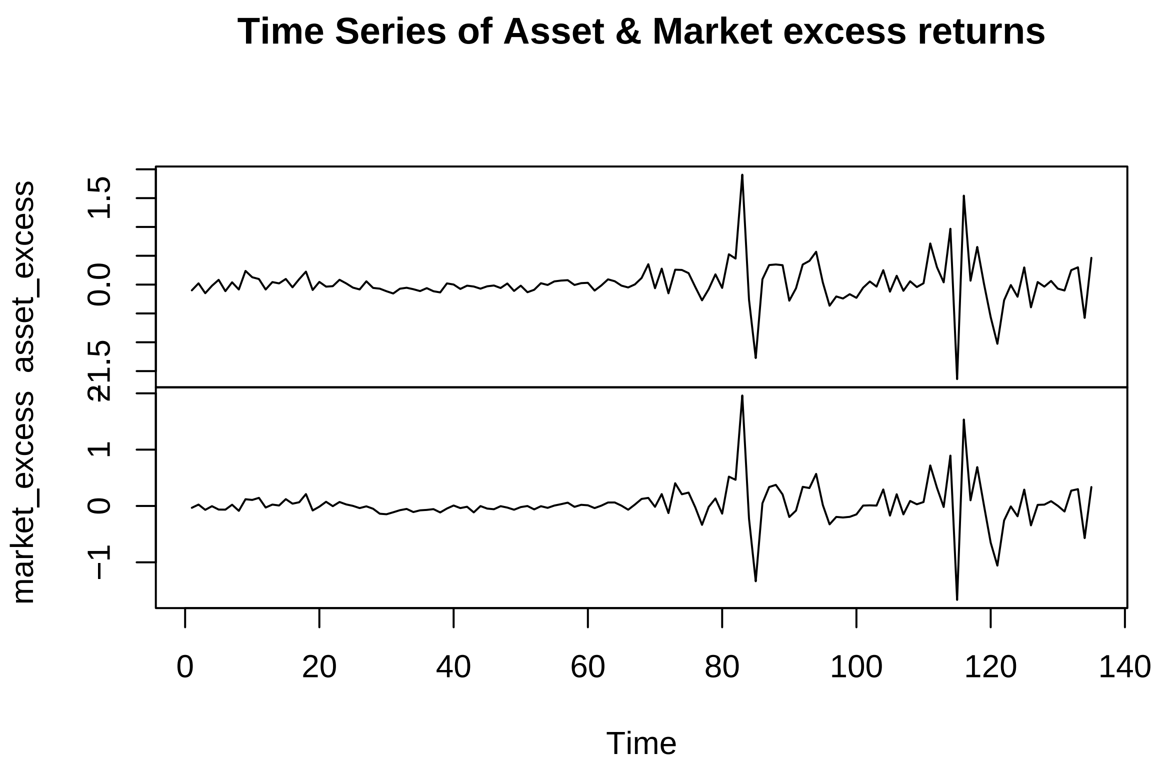
2.3°/ Box-plot :

```
par(mfrow = c(1, 2))

boxplot(df$asset_excess, main = "Boxplot - Asset Excess Returns")
boxplot(df$market_excess, main = "Boxplot - Market Excess Returns")
```



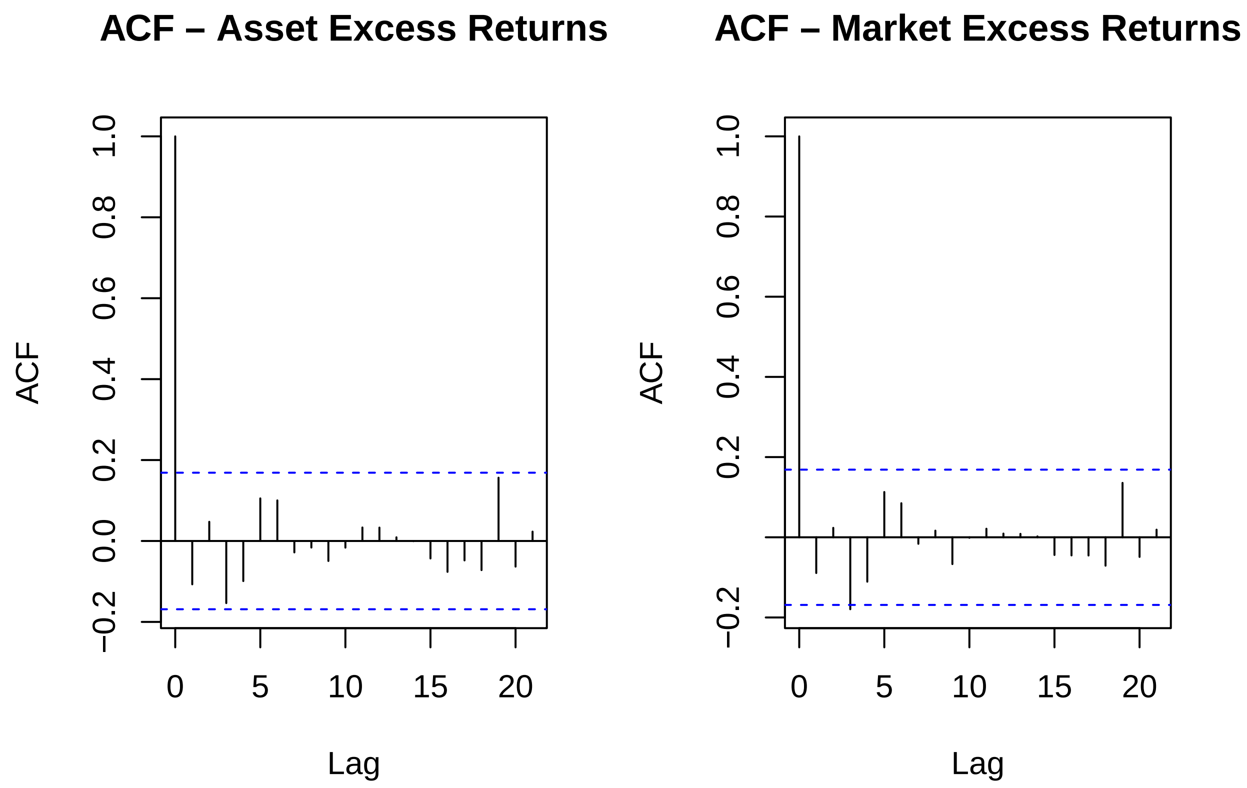
2.4°/ Time series plot :

```r
plot(ts(df), main = "Time Series of Asset & Market excess returns" )
```



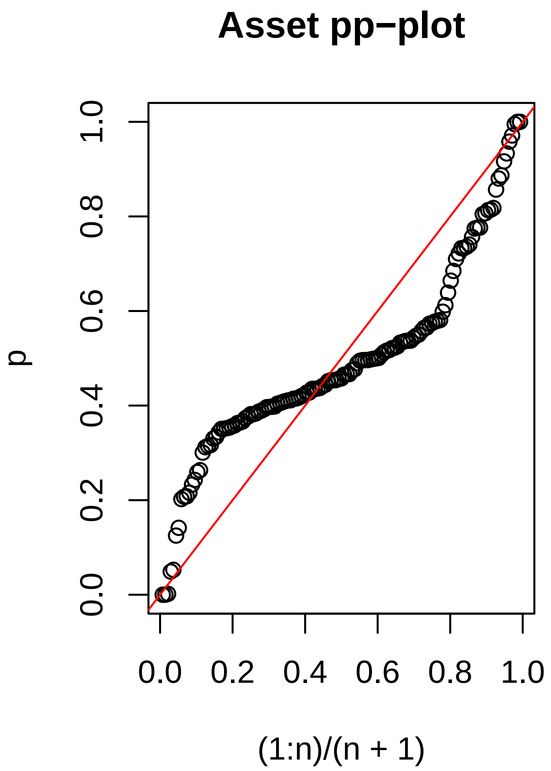
2.5°/ ACF :

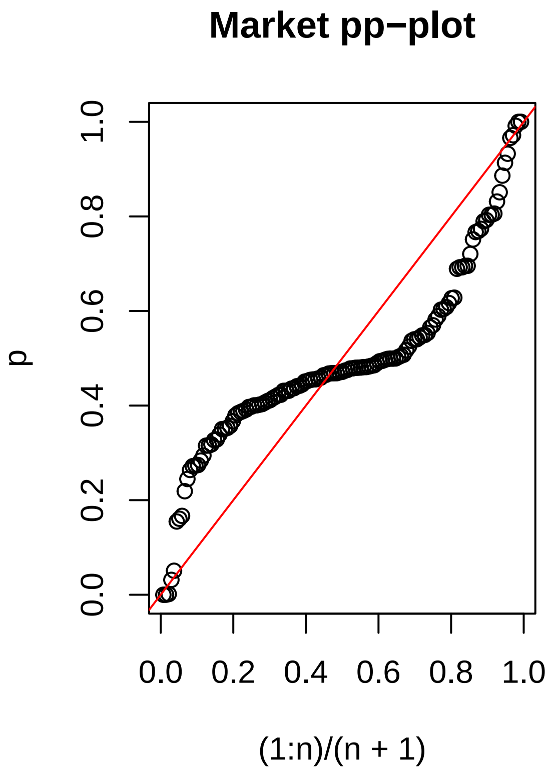
```r
par(mfrow = c(1, 2))
acf(df$asset_excess, main = "ACF - Asset Excess Returns")
acf(df$market_excess, main = "ACF - Market Excess Returns")
```



2.6°/ PP-plot :

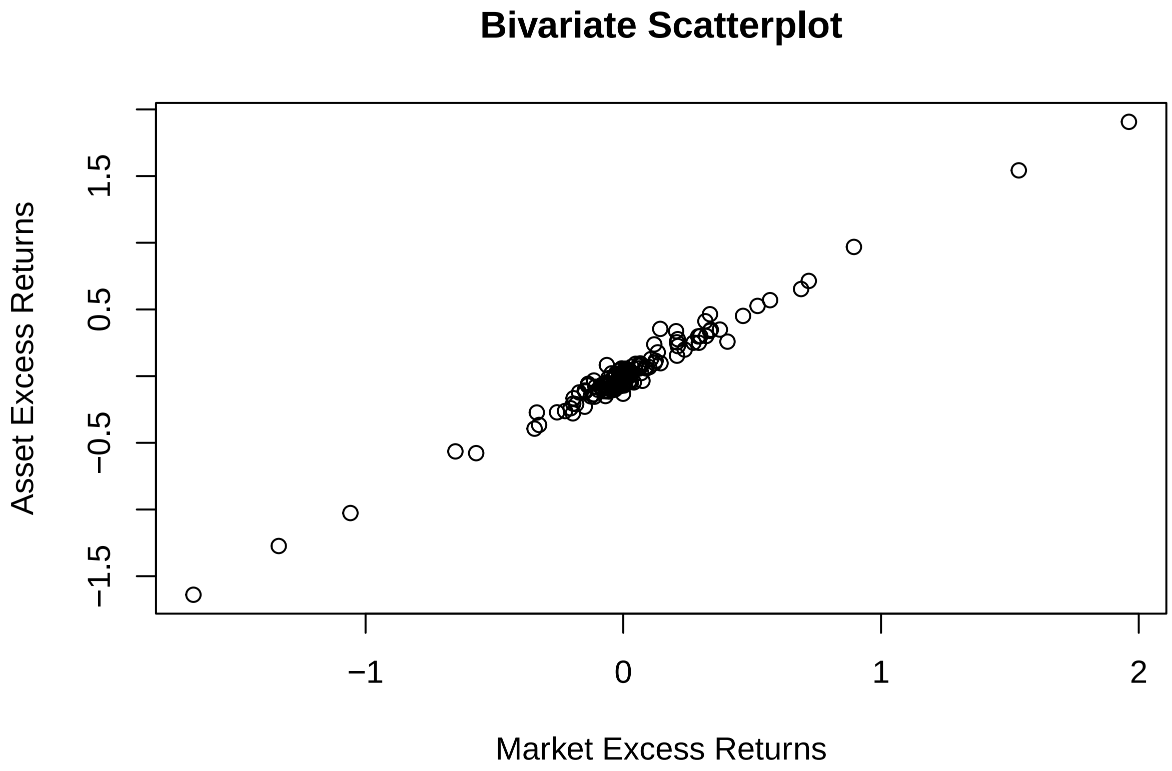
```
ppplot <- function(x, string) {
  n <- length(x)
  y <- sort(x)
  p <- pnorm(y, mean(x), sd(x))
  plot((1:n)/(n+1), p, main=string)
  abline(0,1,col="red")
}

par(mfrow = c(1, 2))
ppplot(df$asset_excess, "Asset pp-plot")
ppplot(df$market_excess, "Market pp-plot")
```





2.7°/ Bivariate scatterplot :

```
plot(df$market_excess, df$asset_excess,
     xlab = "Market Excess Returns",
     ylab = "Asset Excess Returns",
     main = "Bivariate Scatterplot")
```

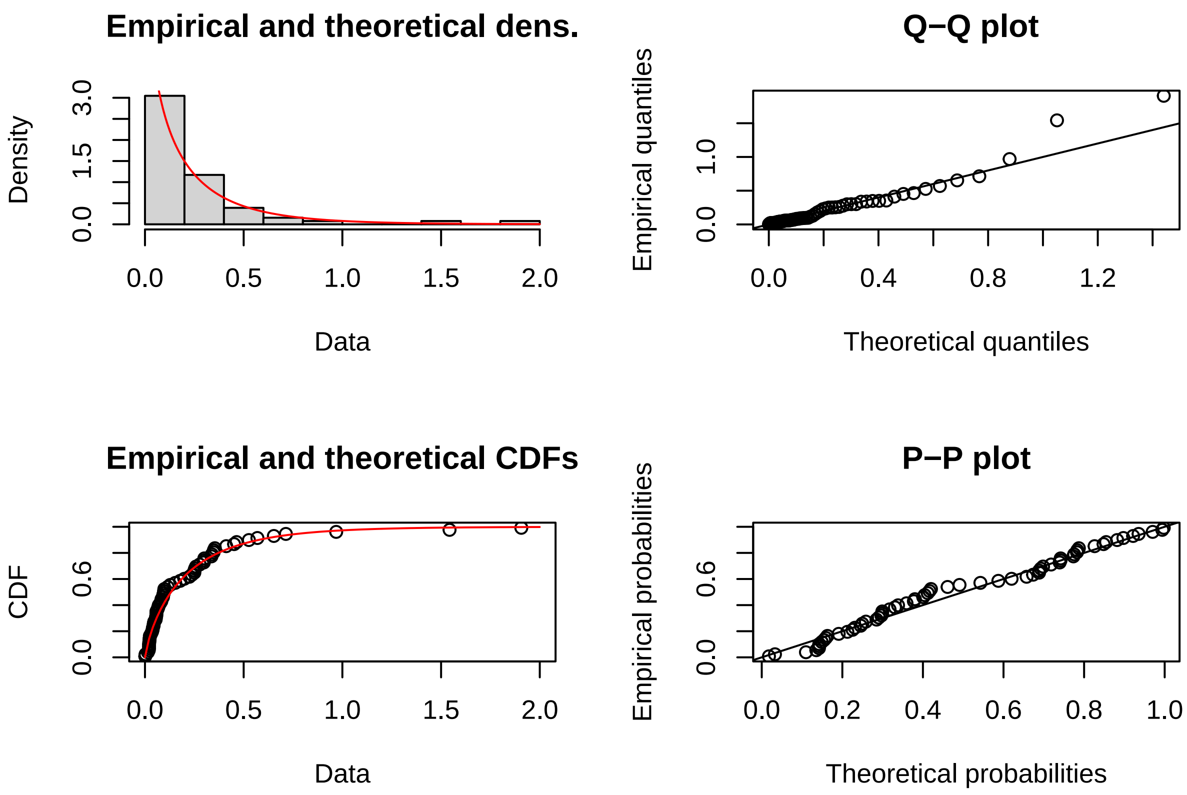


3°/ Fit an extreme distribution :

```
# Whatever the distribution, we need positive data => ofc I (don't) listen in class so I missed that pa
x <- df$asset_excess # Should I put '-' before it ?
x <- x[x > 0]   # focus on losses

# We ballin' on Redbull distribution :
fit_weibull <- fitdist(x, "weibull")
summary(fit_weibull)
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape 0.8145663 0.07591873
## scale 0.2072613 0.03365809
## Loglikelihood:  31.40099   AIC:  -58.80199   BIC:  -54.48422
## Correlation matrix:
##           shape     scale
## shape 1.0000000 0.3272248
## scale 0.3272248 1.0000000
```

```
plot(fit_weibull)
```



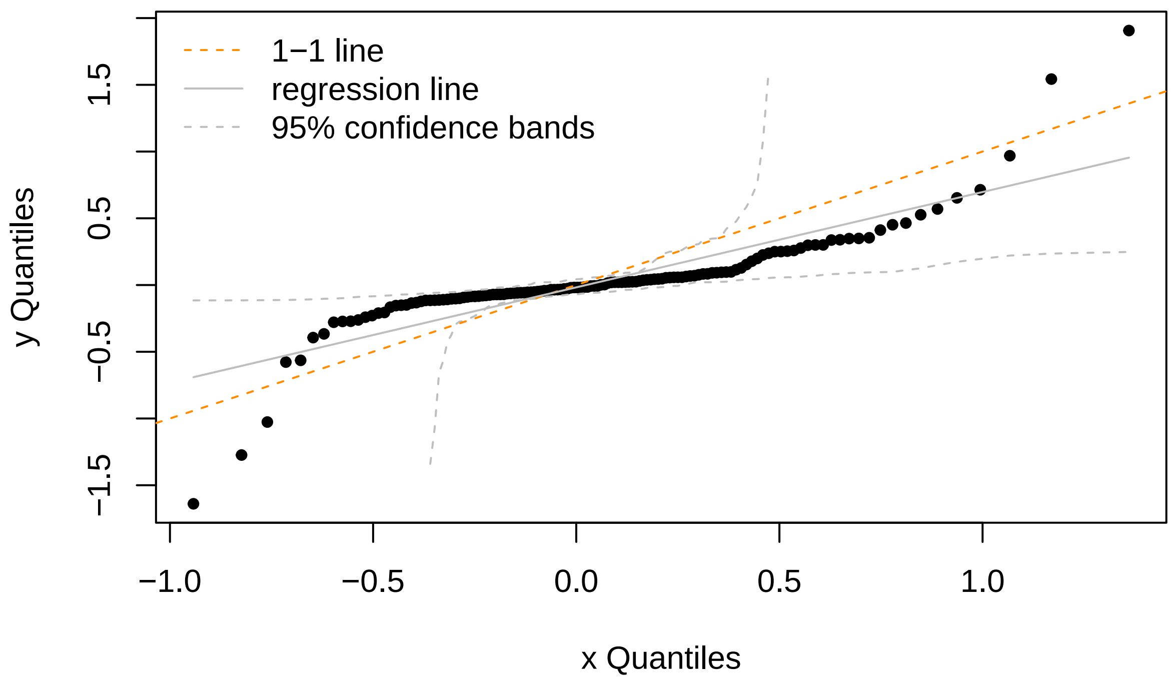
4°/ Fit the GEV distribution :

```
# Fuck it, we rollin' on evd library instead
gev_fit <- fgev(df$asset_excess)
summary(gev_fit)
```

```
##              Length Class  Mode
## estimate       3    -none- numeric
## std.err        3    -none- numeric
## fixed          0    -none- NULL
## param          3    -none- numeric
## deviance       1    -none- numeric
## corr           0    -none- NULL
## var.cov        9    -none- numeric
## convergence    1    -none- character
## counts         2    -none- numeric
## message        0    -none- NULL
## data         135    -none- numeric
## tdata        135    -none- numeric
## nsloc          0    -none- NULL
## n              1    -none- numeric
## prob           0    -none- NULL
## loc            1    -none- numeric
## call           2    -none- call
```

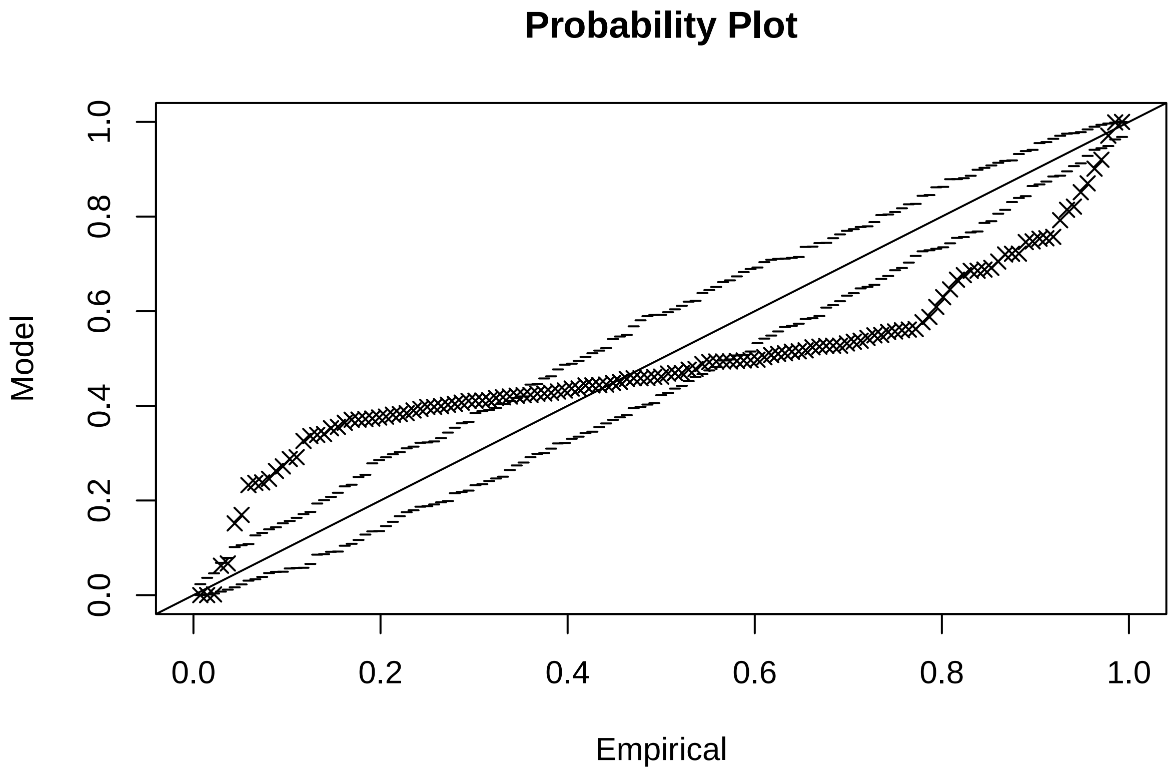
4.1°/ GEV QQ-plot :

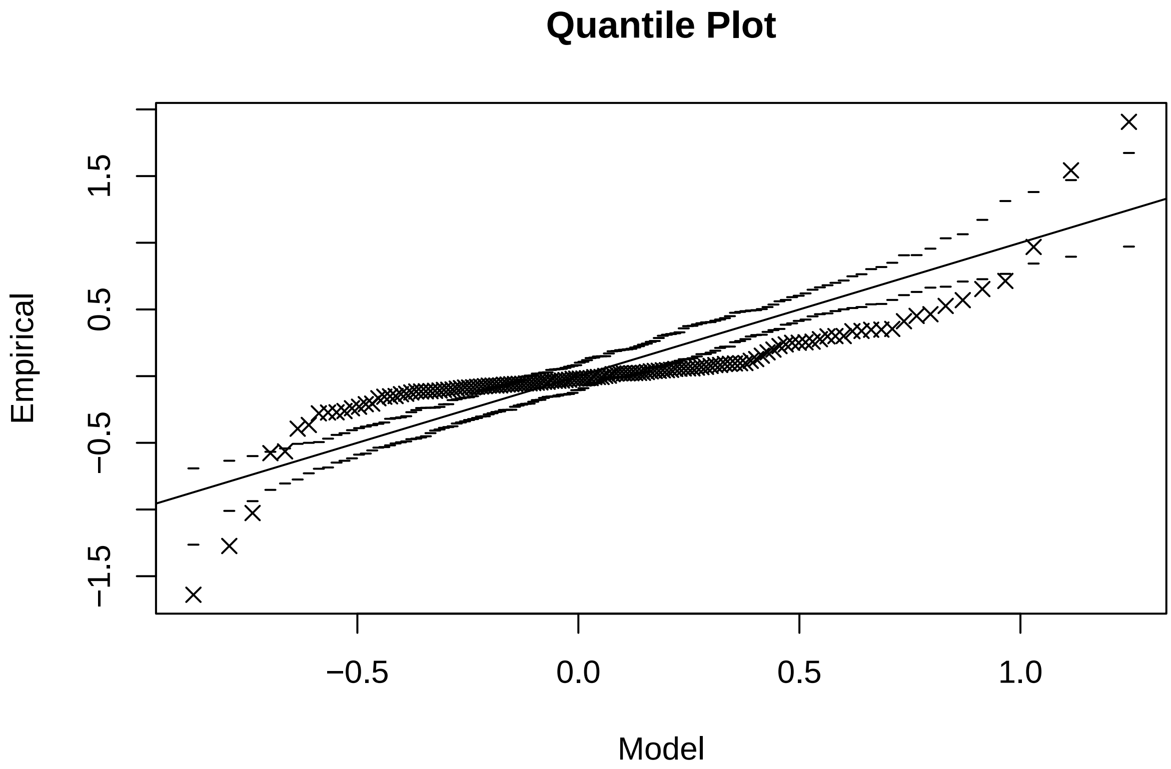
```r
# Extract fitted parameters
params <- gev_fit$estimate

# Compute the theoretical quantiles from the fitted GEV distribution
theoretical_quantiles <- qgev(ppoints(length(df$asset_excess)), loc = params[1], scale = params[2], sha

# Create the QQ plot manually
qqplot(theoretical_quantiles, df$asset_excess)
```

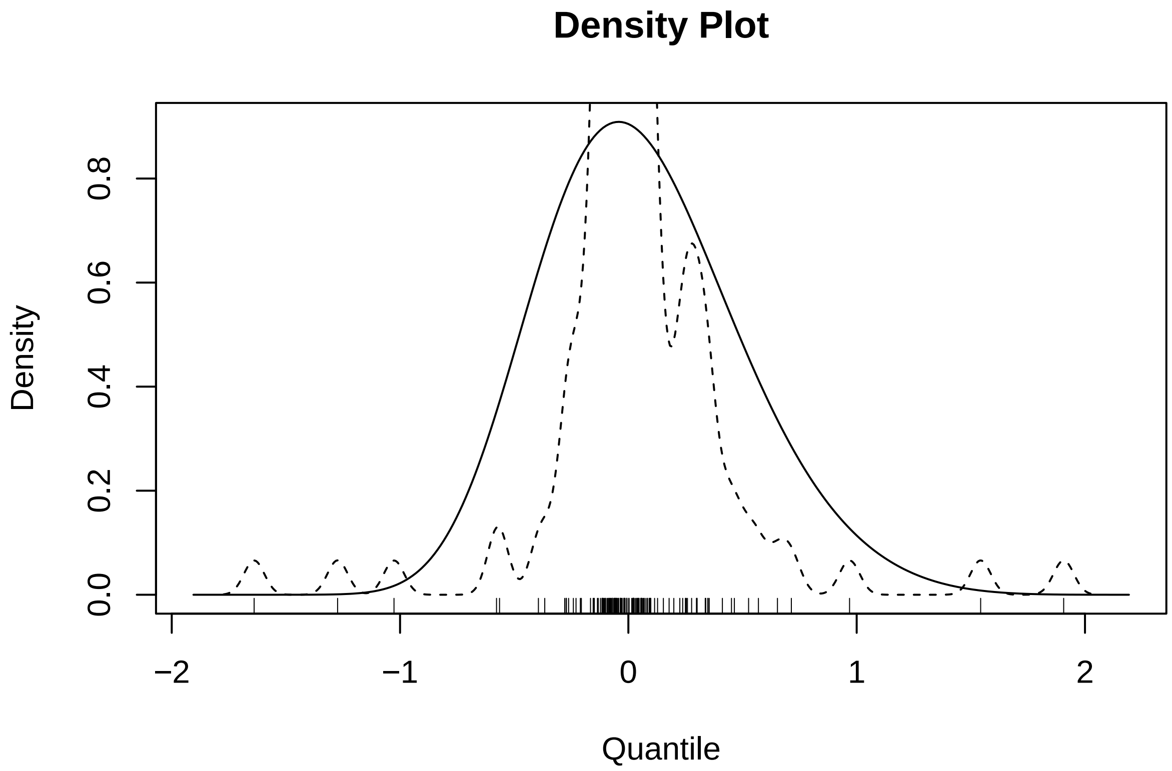


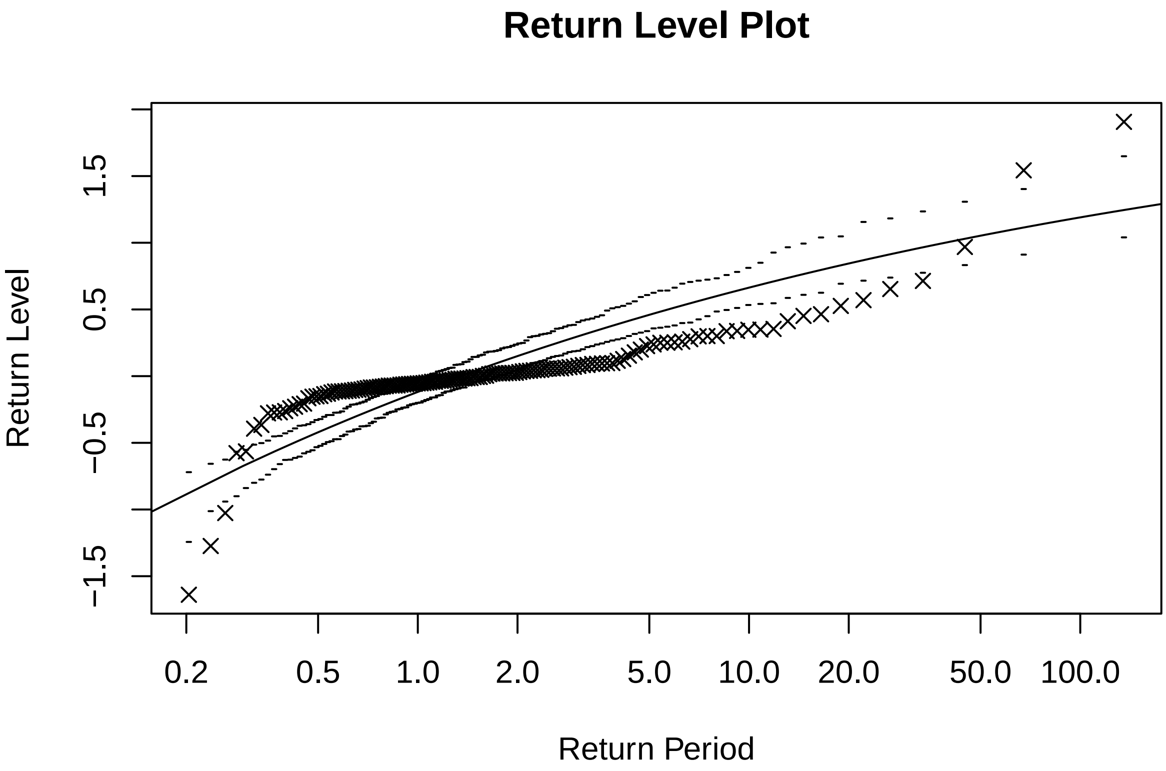
4.2°/ GEV Density :

```r
plot(gev_fit)
```


Probability Plot
Model
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Empirical

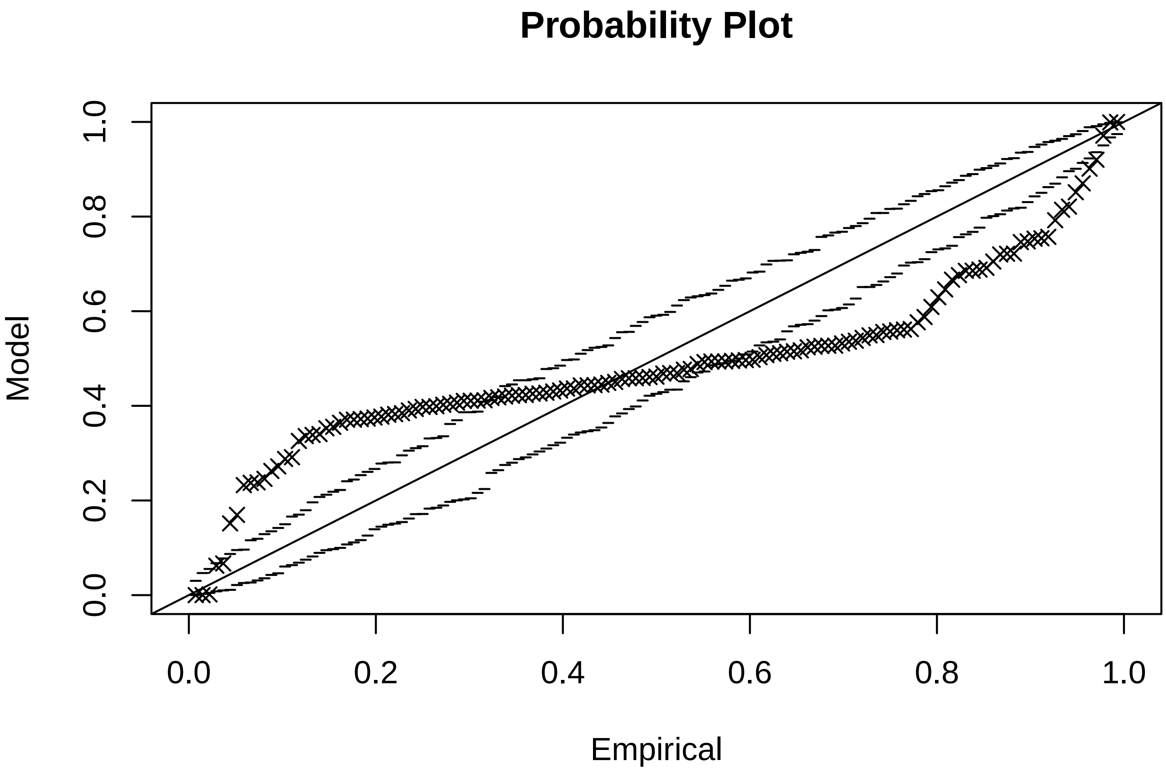

Quantile Plot
Empirical
Model
−1.5
−0.5
0.5
1.5
−0.5
0.0
0.5
1.0

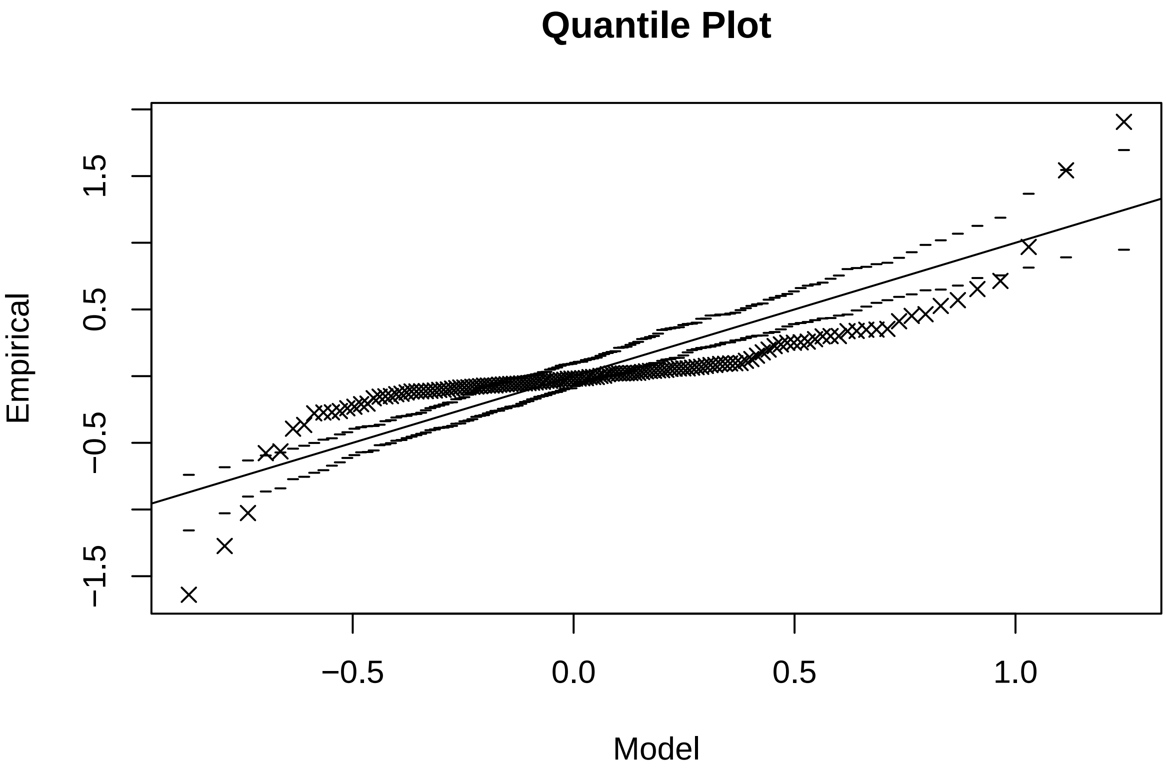

Density Plot
Density
0.0
0.2
0.4
0.6
0.8
−2
−1
0
1
2
Quantile

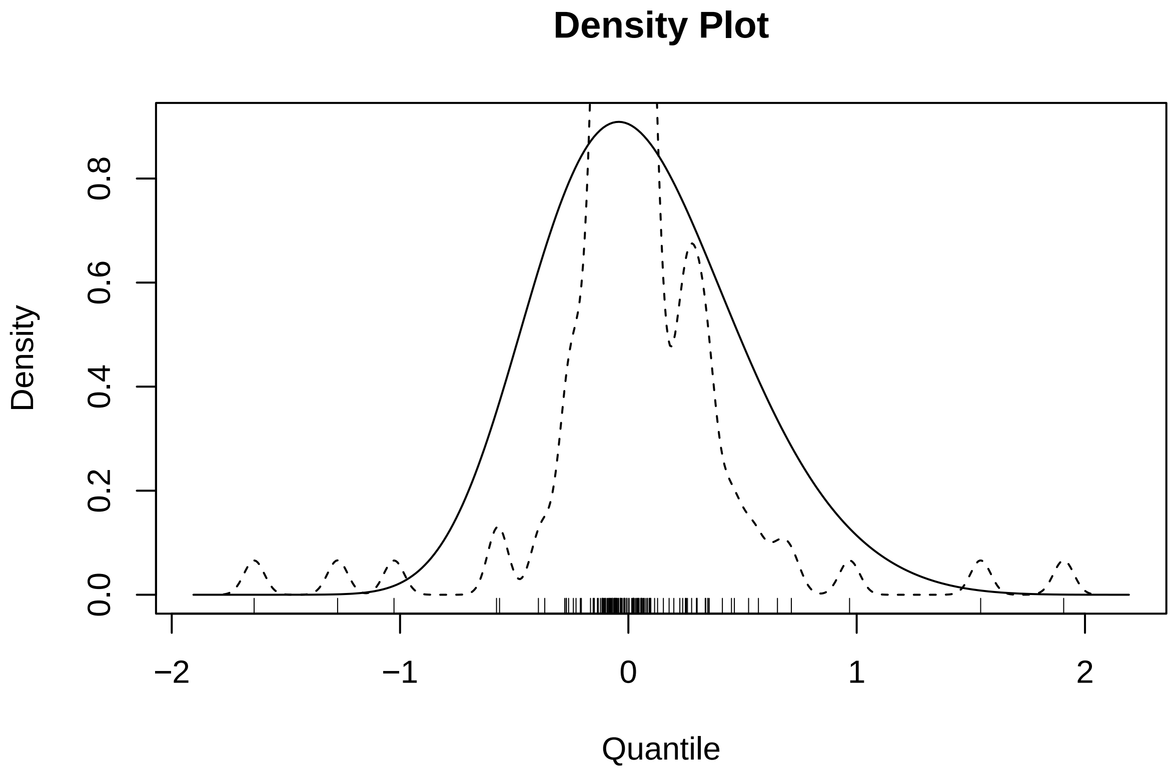


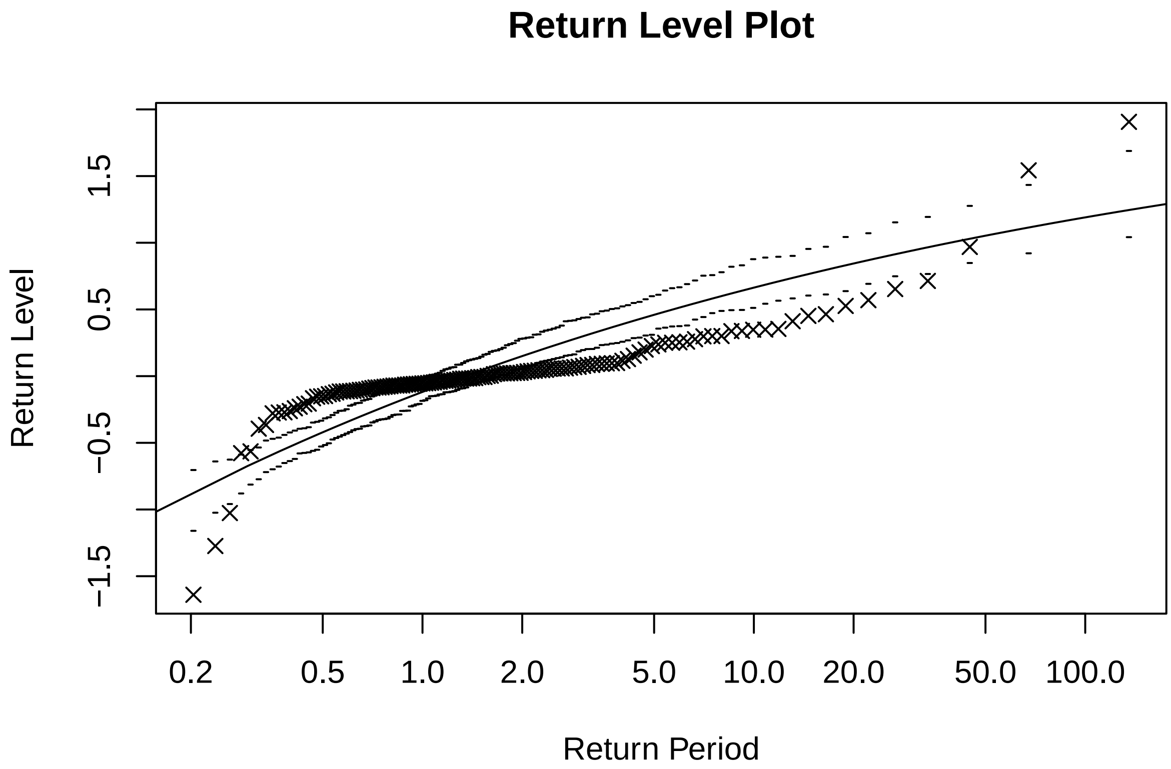
4.3°/ GEV Probability

```
plot(gev_fit)
```


Probability Plot
Model
Empirical
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Quantile Plot
Empirical
Model
−1.5
−0.5
0.5
1.5
−0.5
0.0
0.5
1.0


Density Plot
Density
0.0
0.2
0.4
0.6
0.8
−2
−1
0
1
2
Quantile



5 : Elliptical copulas :

```
u_asset  <- rank(df$asset_excess)/(nrow(df)+1)
u_market <- rank(df$market_excess)/(nrow(df)+1)

U <- cbind(u_asset, u_market)
```

```
gauss_cop <- normalCopula(dim = 2)
fit_gauss <- fitCopula(gauss_cop, U, method = "ml")
summary(fit_gauss)
```

```
## Call: fitCopula(gauss_cop, U, method = "ml")
## Fit based on "maximum likelihood" and 135 2-dimensional observations.
## Normal copula, dim. d = 2
##       Estimate Std. Error
## rho.1    0.953      0.006
## The maximized loglikelihood is 157.2
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##        8        8
```

6°/ Archimedean copulas : Clayton :

```r
clayton <- claytonCopula(dim = 2)
fit_clayton <- fitCopula(clayton, U, method = "ml")
summary(fit_clayton)
```

```
## Call: fitCopula(clayton, U, method = "ml")
## Fit based on "maximum likelihood" and 135 2-dimensional observations.
## Clayton copula, dim. d = 2
##       Estimate Std. Error
## alpha    5.384      0.482
## The maximized loglikelihood is 134.7
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##        6        6
```