Ex 4 :

Noha

2025-11-27

```r
# Construct yield spreads vs Germany
data <- data %>%
  drop_na()

# Convert to time series
GER <- xts(data$`Germany, Government Benchmarks, Macrobond, 10 Year, Yield`, order.by = data$Date...15)
AUT <- xts(data$`Austria, Government Benchmarks, Macrobond, 10 Year, Yield`, order.by = data$Date...7)
SPA <- xts(data$`Spain, Government Benchmarks, Macrobond, 10 Year, Yield`, order.by = data$Date...39 )

AUT_spread <- ts(AUT - GER)
SPA_spread <- ts(SPA - GER)

RAUT_spread = diff(AUT_spread, lag = 1)
RSPA_spread = diff(SPA_spread, lag = 1)

df = cbind(AUT_spread, SPA_spread)
return_df = cbind(RAUT_spread, RSPA_spread)
```

1°/ Preliminary analysis :

1.1°/ Statistic :

```r
basic_stats <- function(x) {
  c(
    Mean = mean(x),
    SD = sd(x),
    Skewness = skewness(x),
    Kurtosis = kurtosis(x),
    Min = min(x),
    Max = max(x)
  )
}

basic_stats(RAUT_spread)
```

```
##              Mean                SD Skewness.Series 1 Kurtosis.Series 1
##      2.588813e-05      4.798760e-02     -6.595501e-02      9.912310e+00
##               Min               Max
##     -3.800000e-01      3.100000e-01
```

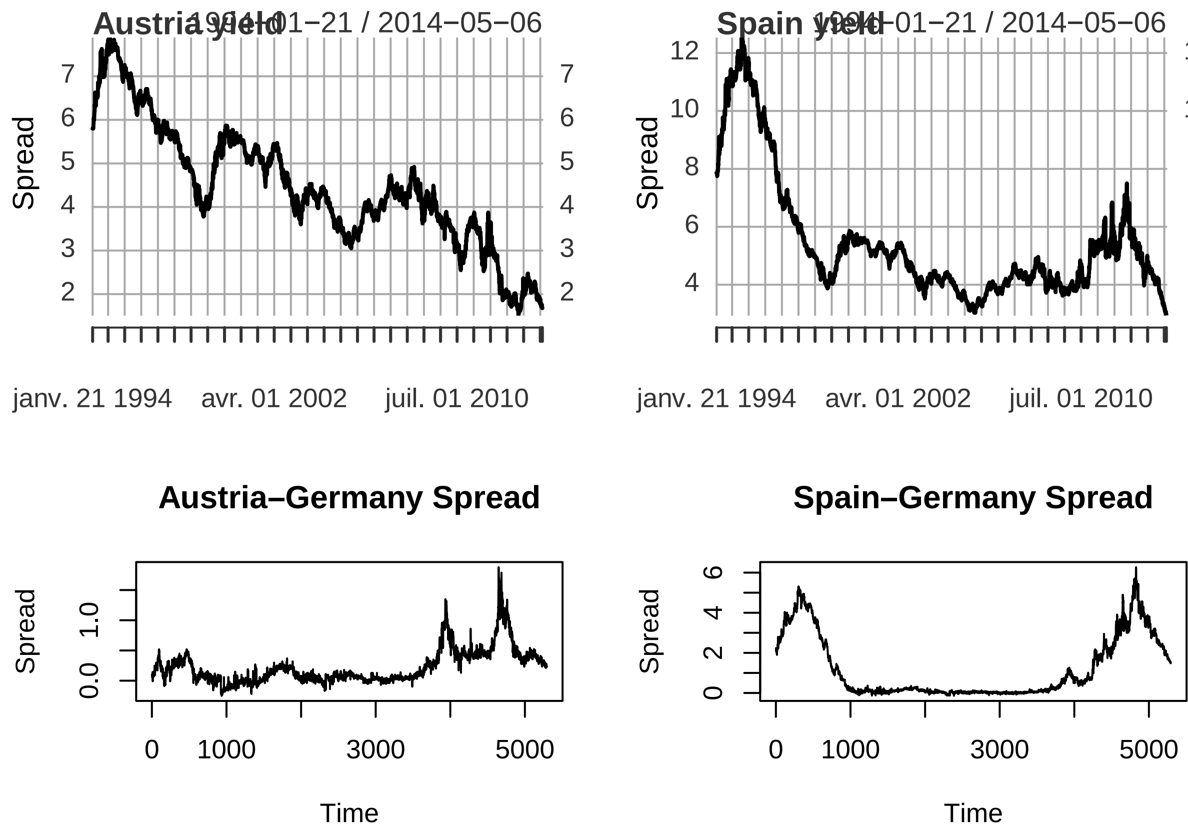
```r
basic_stats(RSPA_spread)
```

```
##              Mean                SD Skewness.Series 1 Kurtosis.Series 1
```

```
##     -0.000143613     0.074023461     -0.395133091     13.988979254
##              Min             Max
##     -0.870000000     0.560000000
```

1.2°/ Time series plot :

```r
par(mfrow = c(2,2))
plot(AUT, type="l", main="Austria yield", ylab="Spread")
plot(SPA, type="l", main="Spain yield", ylab="Spread")
plot(AUT_spread, type="l", main="Austria-Germany Spread", ylab="Spread")
plot(SPA_spread, type="l", main="Spain-Germany Spread", ylab="Spread")
```

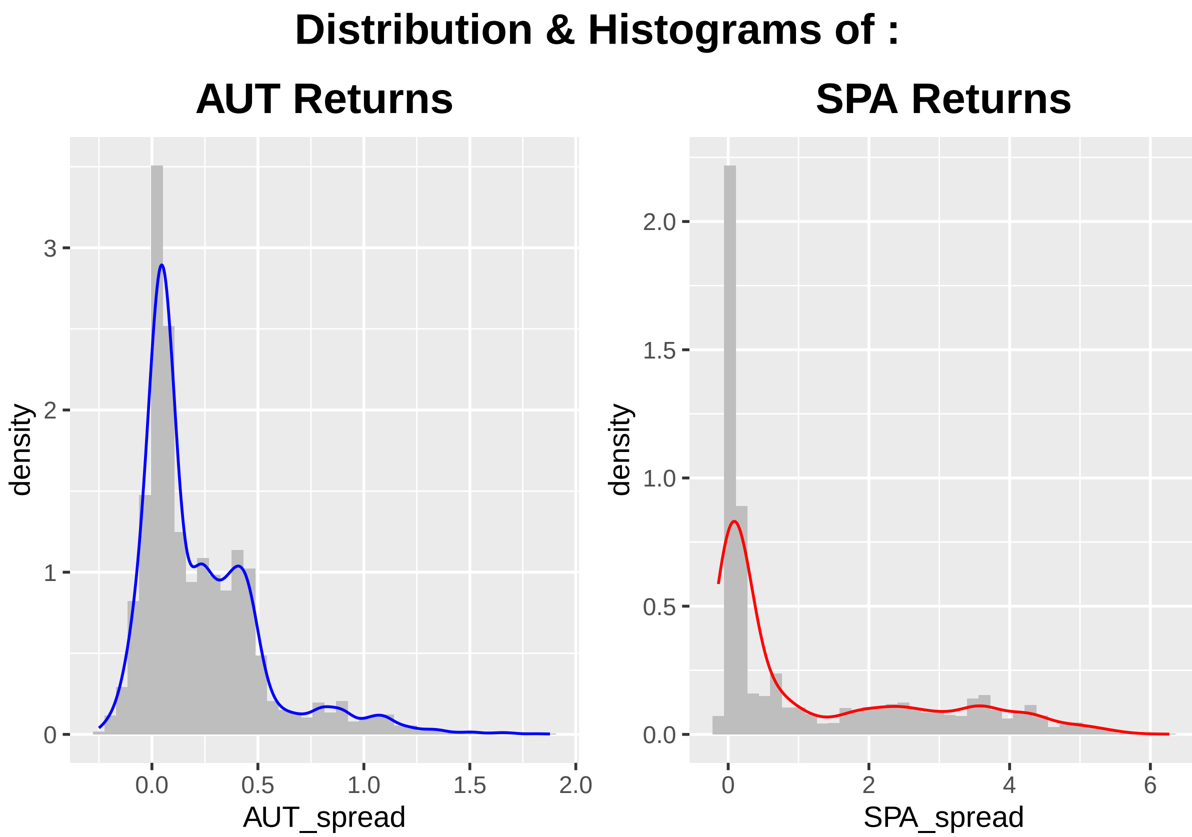


1.3°/ Distribution plots :

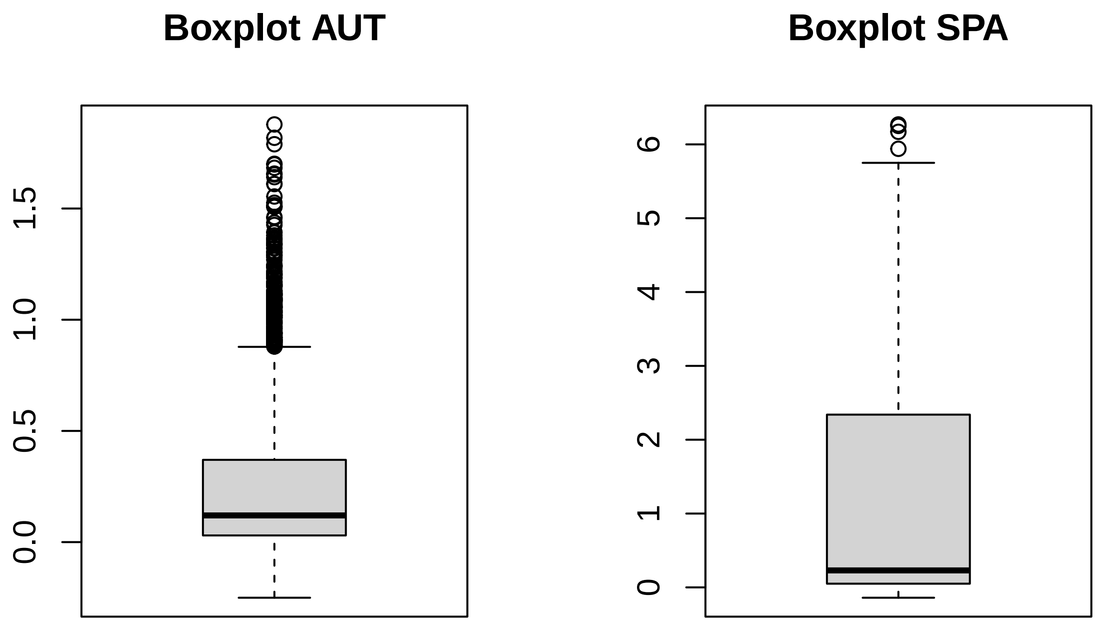
```r
p1 <- ggplot(df, aes(x = AUT_spread)) +
  geom_histogram(aes(y = ..density..), bins = 40, fill = "grey") +
  geom_density(color = "blue") +
  ggtitle("AUT Returns")

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

(p1 | p2) +
  plot_annotation(
```

```
    title = "Distribution & Histograms of : "
  ) &
  theme(
    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.
```

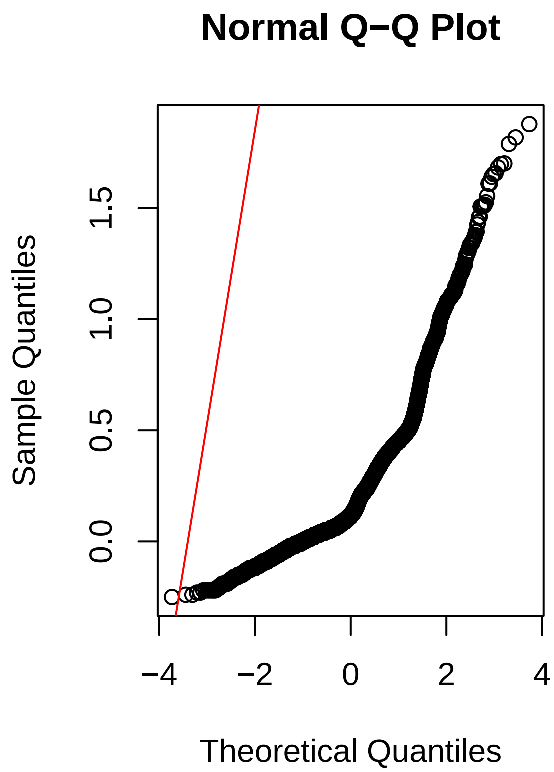


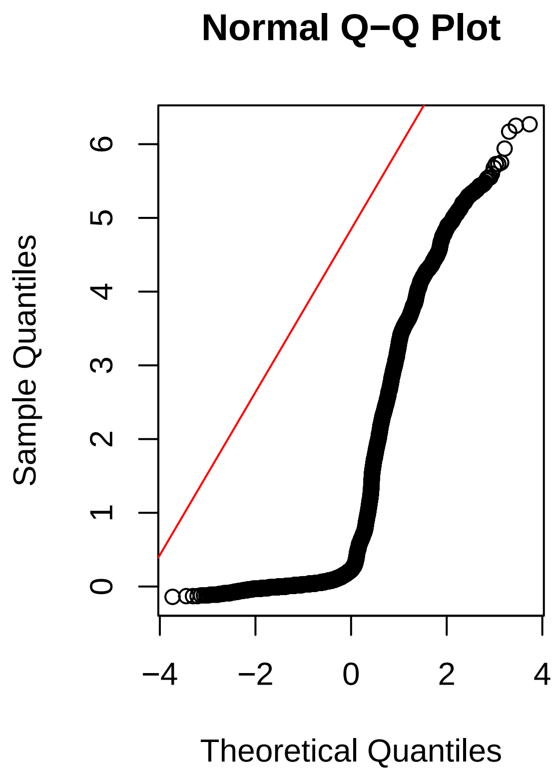

```
par(mfrow = c(1, 2))
boxplot(AUT_spread, main = "Boxplot AUT")
boxplot(SPA_spread, main = "Boxplot SPA")
```



1.4°/ QQ-plots :

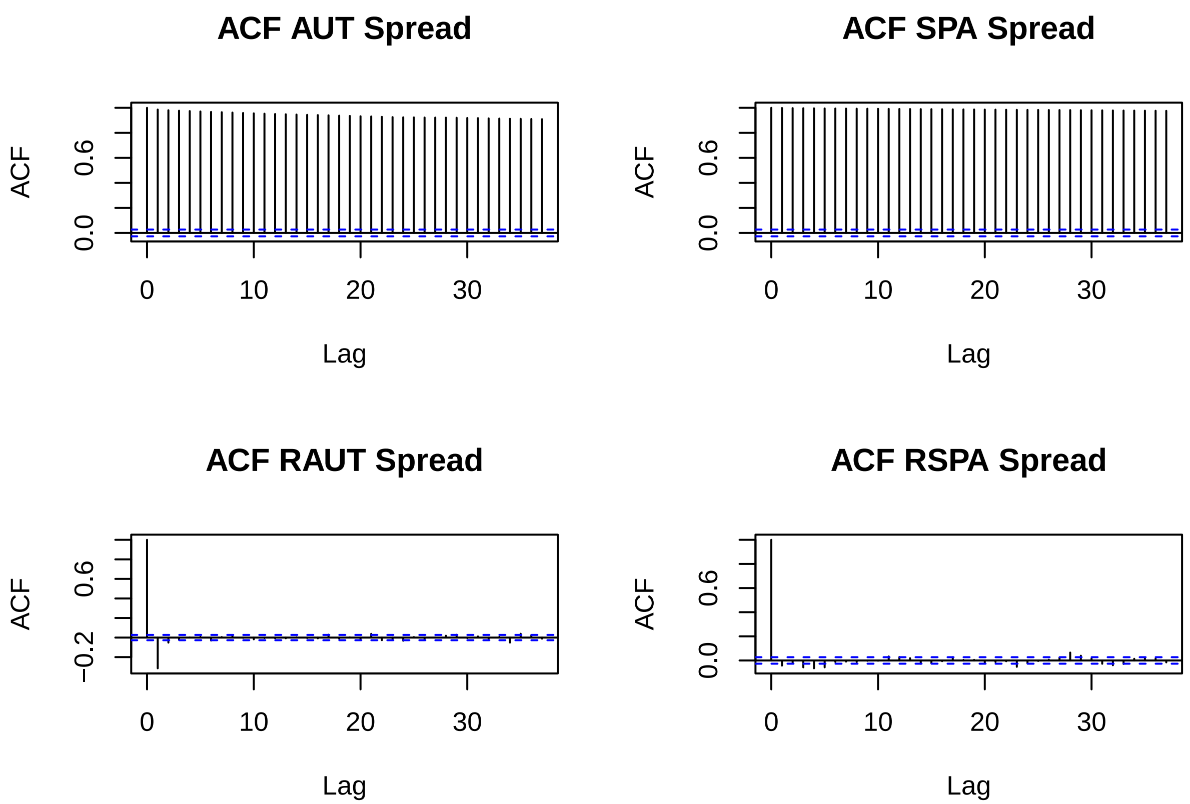
```
par(mfrow=c(1,2))
qqnorm(AUT_spread); qqline(AUT, col="red")
qqnorm(SPA_spread); qqline(SPA, col="red")
```





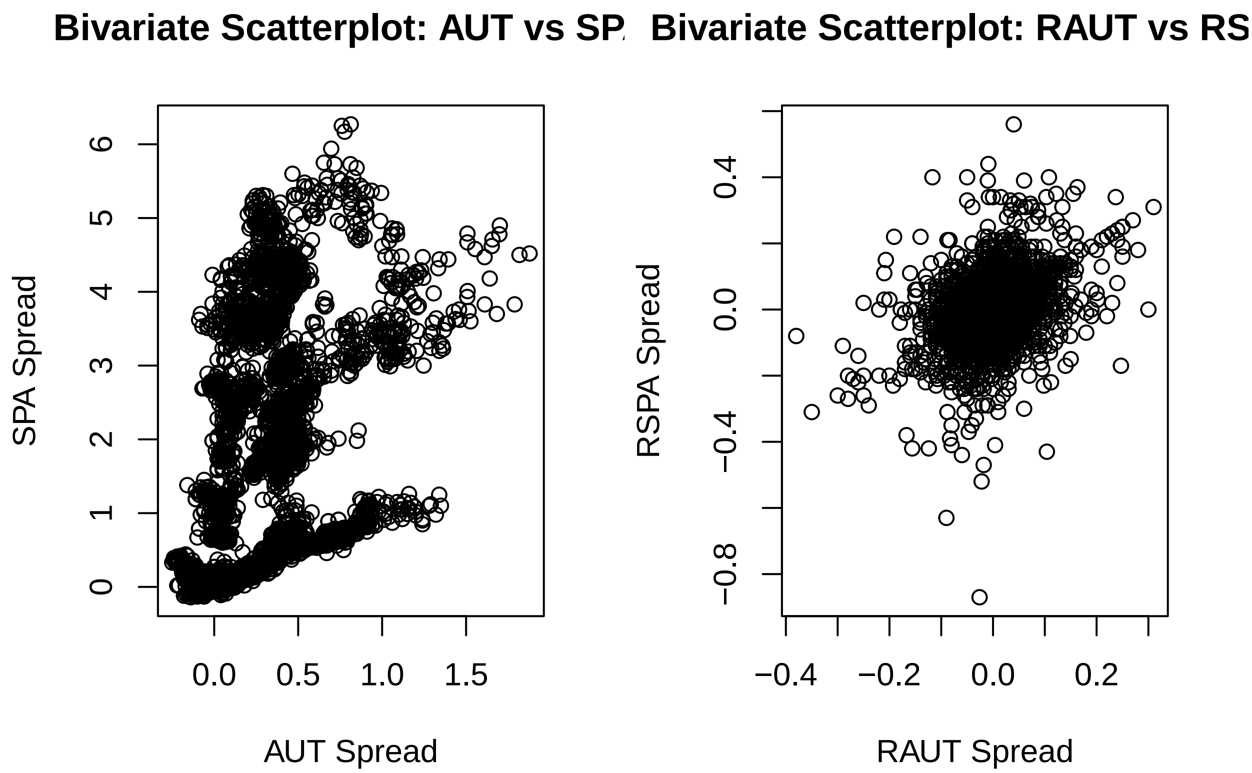
1.5°/ Autocorrelation functions :

```
par(mfrow = c(2, 2))
acf(AUT_spread, main="ACF AUT Spread")
acf(SPA_spread, main="ACF SPA Spread")
acf(RAUT_spread, main="ACF RAUT Spread")
acf(RSPA_spread, main="ACF RSPA Spread")
```



1.6°/ Bivariate scatterplot & PP-plot :

```
# Scatterplot
par(mfrow = c(1, 2))
plot(AUT_spread, SPA_spread,
     main="Bivariate Scatterplot: AUT vs SPA",
     xlab="AUT Spread", ylab="SPA Spread")

plot(RAUT_spread, RSPA_spread,
     main="Bivariate Scatterplot: RAUT vs RSPA",
     xlab="RAUT Spread", ylab="RSPA Spread")
```



2°/ Fit Weibull :

```
RAUT_numeric <- as.numeric(AUT_spread)
RSPA_numeric <- as.numeric(SPA_spread)

fit_AUT <- fitdist(RAUT_numeric - min(RAUT_numeric) + 1e-6, "weibull")
fit_AUT_logn <- fitdist(RAUT_numeric - min(RAUT_numeric) + 1e-6, "lnorm")
fit_AUT_gamma <- fitdist(RAUT_numeric - min(RAUT_numeric) + 1e-6, "gamma")

fit_SPA <- fitdist(RSPA_numeric - min(RSPA_numeric) + 1e-6, "weibull")
fit_SPA_logn <- fitdist(RSPA_numeric - min(RSPA_numeric) + 1e-6, "lnorm")
fit_SPA_gamma <- fitdist(RSPA_numeric - min(RSPA_numeric) + 1e-6, "gamma")
```

```
gofstat(list(fit_AUT, fit_AUT_logn, fit_AUT_gamma))
```

```
## Goodness-of-fit statistics
##                              1-mle-weibull 2-mle-lnorm 3-mle-gamma
## Kolmogorov-Smirnov statistic     0.1092154  0.07278392   0.1091223
## Cramer-von Mises statistic      17.6326602  6.19929029  11.7381610
## Anderson-Darling statistic     109.9383638 37.51374595  65.4471779
##
## Goodness-of-fit criteria
##                                1-mle-weibull 2-mle-lnorm 3-mle-gamma
## Akaike's Information Criterion      217.1982   -90.23496   -310.2139
## Bayesian Information Criterion      230.3465   -77.08668   -297.0656
```

```r
gofstat(list(fit_SPA, fit_SPA_logn, fit_SPA_gamma))
```

```
## Goodness-of-fit statistics
##                              1-mle-weibull 2-mle-lnorm 3-mle-gamma
## Kolmogorov-Smirnov statistic      0.183718   0.1544723   0.2009383
## Cramer-von Mises statistic       45.550895  38.4359686  52.3595017
## Anderson-Darling statistic      262.777013 227.8906118 292.7595180
##
## Goodness-of-fit criteria
##                                1-mle-weibull 2-mle-lnorm 3-mle-gamma
## Akaike's Information Criterion      13335.01    12767.88    13448.31
## Bayesian Information Criterion      13348.16    12781.03    13461.46
```

3°/ GEV distribution :

3.1°/ GEV estimation :

```r
library(data.table)
```

```
##
## Attachement du package : 'data.table'

## Les objets suivants sont masqués depuis 'package:lubridate':
##
##     hour, isoweek, isoyear, mday, minute, month, quarter, second, wday,
##     week, yday, year

## Les objets suivants sont masqués depuis 'package:dplyr':
##
##     between, first, last

## L'objet suivant est masqué depuis 'package:purrr':
##
##     transpose

## Les objets suivants sont masqués depuis 'package:xts':
##
##     first, last

## Les objets suivants sont masqués depuis 'package:zoo':
##
##     yearmon, yearqtr
```

```r
library(extRemes)
```

```
## Le chargement a nécessité le package : Lmoments

## Le chargement a nécessité le package : distillery

##
## Attachement du package : 'extRemes'

## L'objet suivant est masqué depuis 'package:evd':
##
##     mrlplot

## Les objets suivants sont masqués depuis 'package:stats':
##
##     qqnorm, qqplot
```

```r
data$Date = data$Date...15
data$mo<- strftime(data$Date,"%m")
data$yr<- strftime(data$Date,"%Y")

gev_df = data[-1,]

gev_df <- gev_df %>%
  arrange(Date)

gev_df$RAUT_spread = RAUT_spread
gev_df$RSPA_spread = RSPA_spread

minima <- setDT(gev_df)[, .(RAUT_down = -min(RAUT_spread)), by = .(yr, mo)]
```

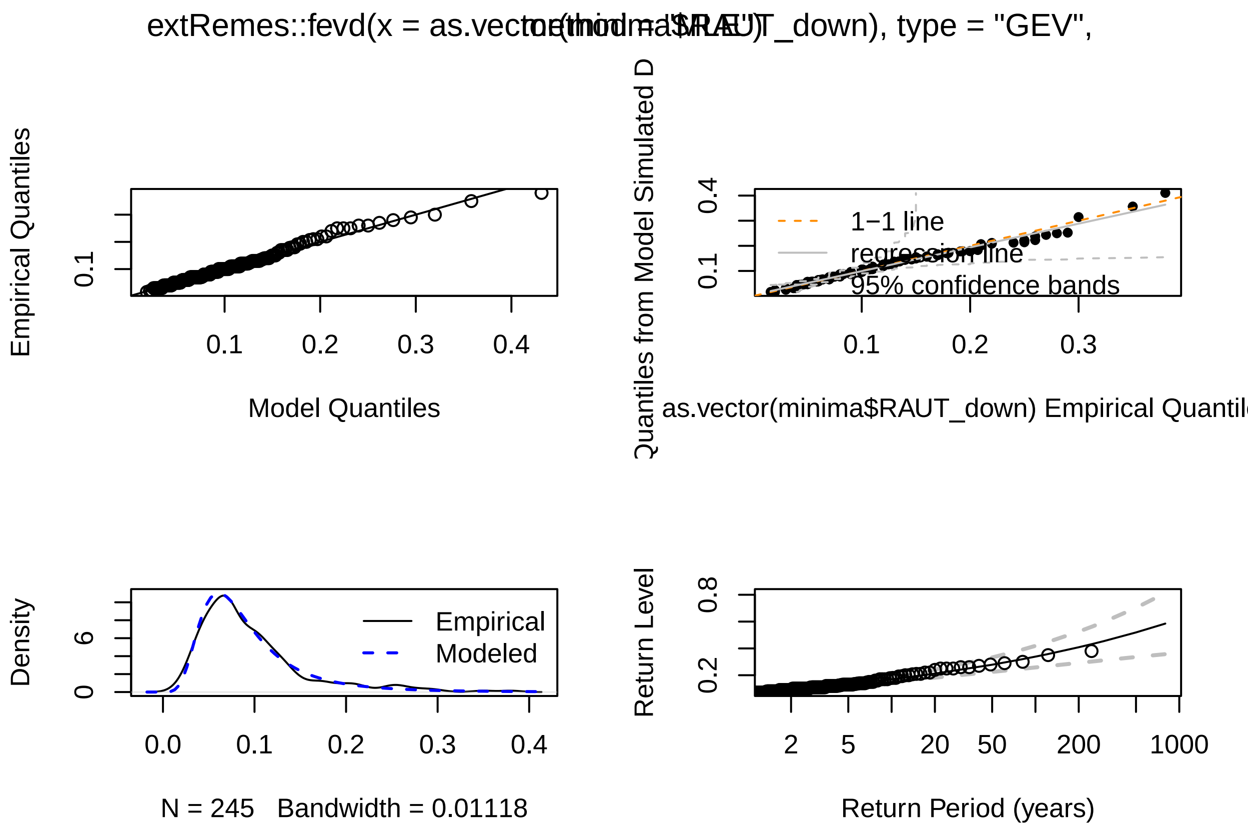
```
## Warning in setDT(gev_df): Certaines colonnes sont de type multi-colonnes (comme
## une colonne de matrice), par exemple la colonne 10. setDT conservera ces
## colonnes telles quelles, mais les opérations ultérieures telles que le
## regroupement et la jointure peuvent échouer. Pensez plutôt à utiliser
## as.data.table() qui créera une nouvelle colonne pour chaque colonne intégrée.
```

```r
maxima <- setDT(gev_df)[, .(RAUT_up = max(RAUT_spread)), by = .(yr, mo)]

fit_gev_RAUT_down <- extRemes::fevd(as.vector(minima$RAUT_down), method = "MLE", type = "GEV")
summary(fit_gev_RAUT_down, silent = FALSE) #to get the standard error of estimates
```

```
##
## extRemes::fevd(x = as.vector(minima$RAUT_down), type = "GEV",
##     method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  -409.9432
##
##
##  Estimated parameters:
##   location      scale      shape
## 0.06731343 0.03411650 0.21906172
##
##  Standard Error Estimates:
##    location       scale       shape
## 0.002475290 0.001982725 0.053539629
##
##  Estimated parameter covariance matrix.
##               location         scale         shape
## location  6.127061e-06  2.846686e-06 -4.073467e-05
## scale     2.846686e-06  3.931199e-06 -6.490147e-06
## shape    -4.073467e-05 -6.490147e-06  2.866492e-03
##
##  AIC = -813.8864
##
##  BIC = -803.3827
```

3.2°/ GEV plot :

```
plot(fit_gev_RAUT_down)
```



4°/ Elliptical copula :

```
U <- pobs(cbind(RAUT_spread, RSPA_spread))

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 5292 2-dimensional observations.
## Normal copula, dim. d = 2
##       Estimate Std. Error
## rho.1   0.3952      0.011
## The maximized loglikelihood is 446.5
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##        8        8
```

5°/ Archimedean Clayton Copula :

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

summary(fit_clayton)
```

```
## Call: fitCopula(clayton_cop, U, method = "ml")
## Fit based on "maximum likelihood" and 5292 2-dimensional observations.
## Clayton copula, dim. d = 2
##       Estimate Std. Error
## alpha   0.8128      0.026
## The maximized loglikelihood is 330.2
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##        3        3
```