Modeling extreme values with a GEV mixture probability distributions

Standard normal distribution

Pascal Alain Dkengne Sielenou

2023-10-12

```
# Load useful functions
```

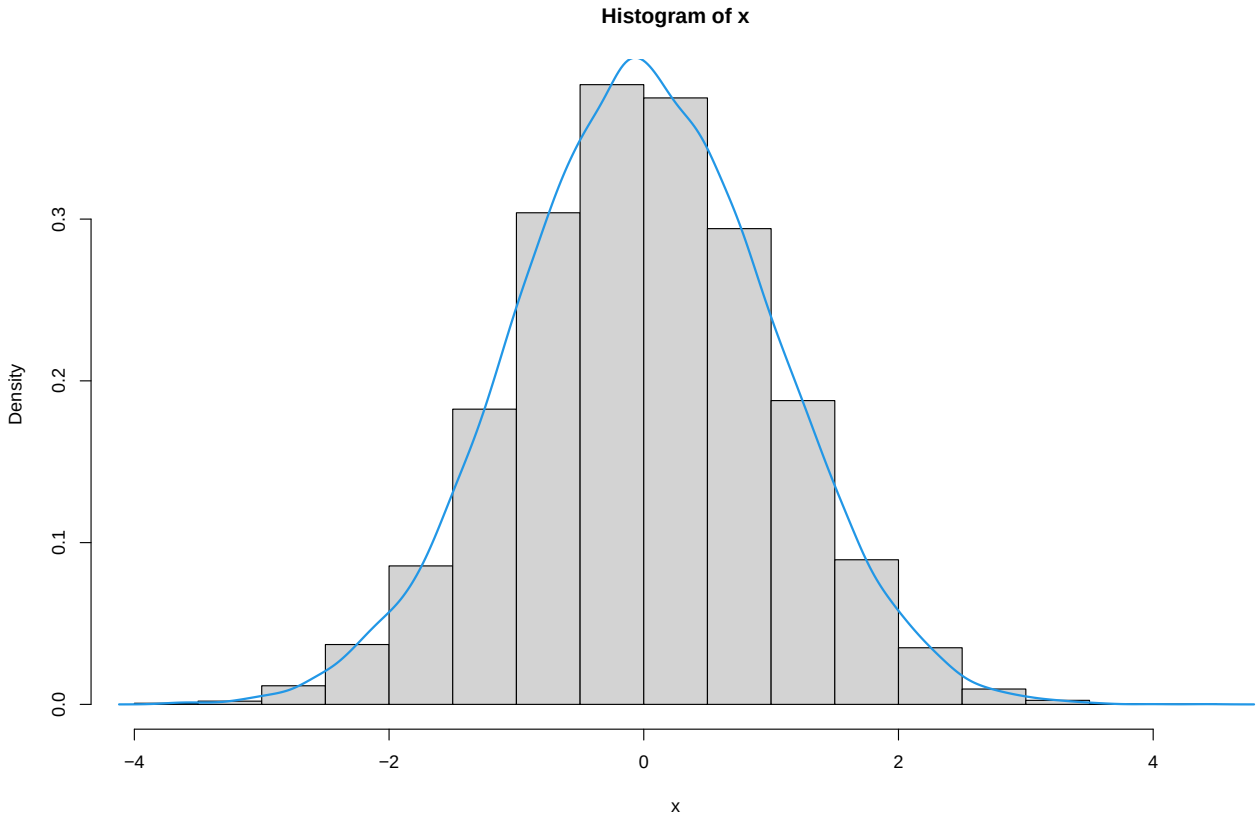
```
path <- ".."
```

```
xfun::in_dir(dir = path, expr = source("./src/generate_gev_sample.R"))
xfun::in_dir(dir = path, expr = source("./src/calculate_gev_inverse_cdf.R"))
xfun::in_dir(dir = path, expr = source("./src/estimate_gev_mixture_model_parameters.R"))
xfun::in_dir(dir = path, expr = source("./src/plot_gev_mixture_model_pdf.R"))
xfun::in_dir(dir = path, expr = source("./src/plot_gev_mixture_model_cdf.R"))
xfun::in_dir(dir = path, expr = source("./src/estimate_gev_mixture_model_quantile.R"))


# Generate a random sample
```

```
n <- 20000
```

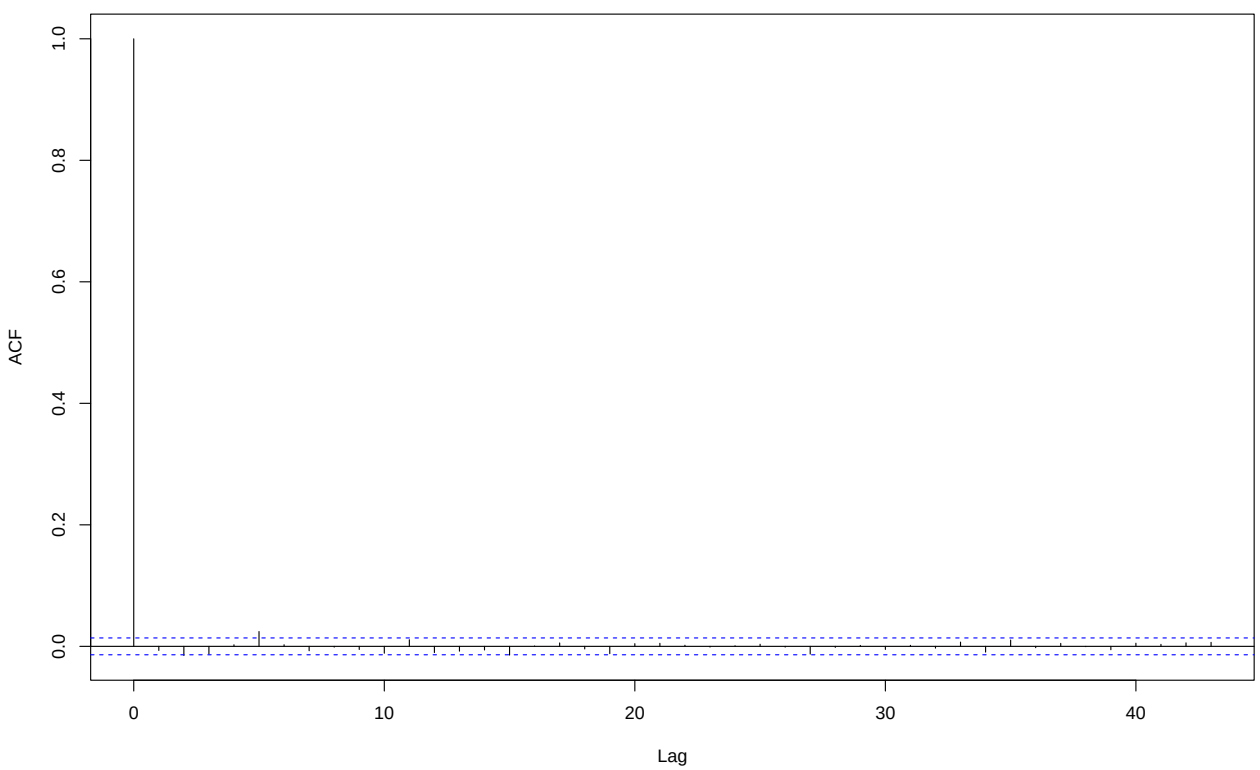
```
set.seed(1122)
x <- rnorm(n = n)

# Histogram of all data
```

```
hist(x, prob = TRUE)
lines(density(x),
      lwd = 2,
      col = 4)
```



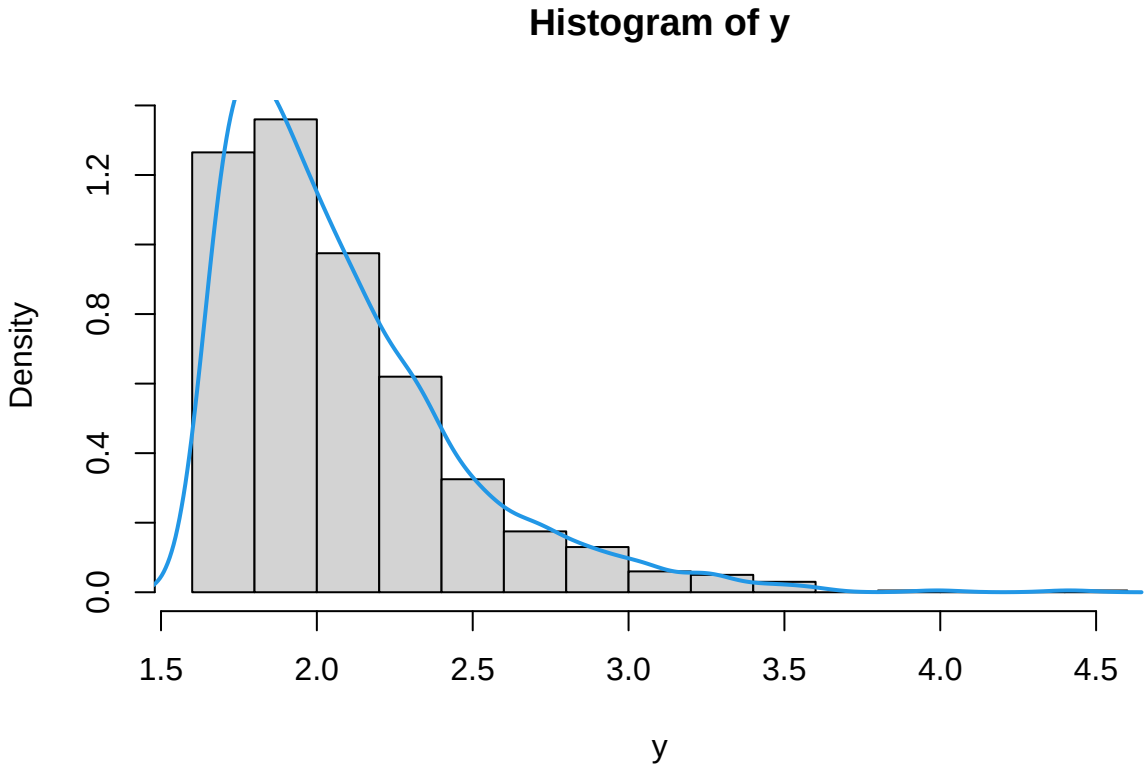

```
# Autocorrelation function of all data
```

```
acf(x)
```



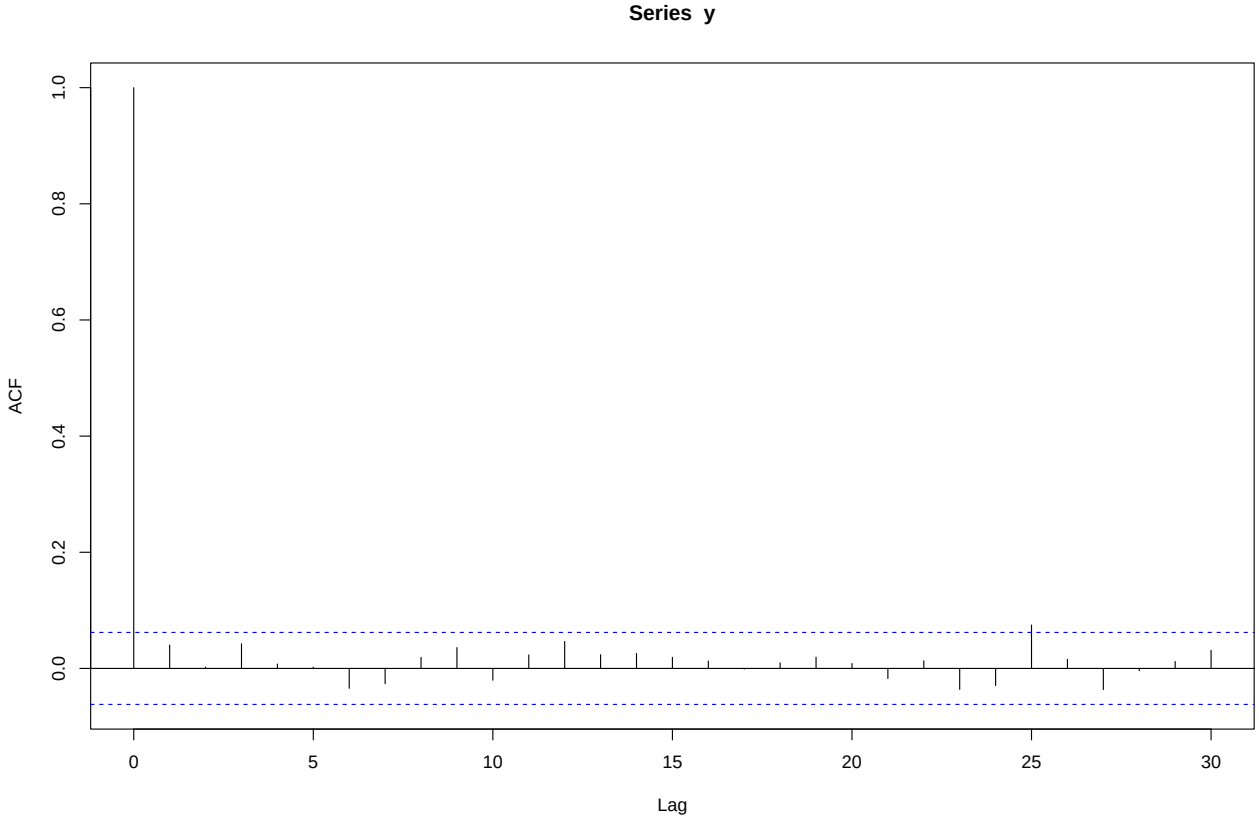

```
# Histogram of the largest data
```

```
nlargest <- 1000
y <- extract_nlargest_sample(x, n = nlargest)
hist(y, prob = TRUE)
lines(density(y),
      lwd = 2,
      col = 4)
```




```
# Autocorrelation function of the largest data
```

```
acf(y)
```



```r
# Estimation of gev mixture models
```

```r
gev_mixture_model <- estimate_gev_mixture_model_parameters(x = x,
                                                           block_sizes = NULL,
                                                           minimum_nblocks = 50,
                                                           threshold = NULL,
                                                           nlargest = nlargest,
                                                           confidence_level = 0.95,
                                                           use_extremal_index = TRUE,
                                                           use_lower_threshold = FALSE,
                                                           maximum_iterations = 1500,
                                                           log_mv = TRUE,
                                                           log_pw = TRUE,
                                                           trace = FALSE,
                                                           method = "MLE")
```

```
##   Successful convergence.
##   Successful convergence.
```

```r
gev_mixture_model$extremal_indexes
```

```
##            9           10           11           12           13           14
## 1.0000000000 1.0000000000 1.0000000000 1.0000000000 1.0000000000 1.0000000000
##           15           16           17           18           19           20
## 1.0000000000 0.9032437806 1.0000000000 1.0000000000 1.0000000000 1.0000000000
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##       loc_star   scale_star      shape_star
## 9  1.845624454 0.3206957588 -0.001081924160
## 10 1.837570294 0.3314580093 -0.006273752669
## 11 1.808753181 0.3463504607 -0.020892002529
## 12 1.722189698 0.4176441446 -0.074421281774
## 13 1.959028538 0.2712030901  0.033484042256
## 14 1.746248247 0.4105063672 -0.073446866095
## 15 1.907913412 0.3149986754 -0.012050541600
## 16 1.802977523 0.3803127987 -0.056975381879
## 17 1.936650498 0.2965818747  0.005617539725
## 18 1.909277773 0.3171843131 -0.014759142835
## 19 1.885721374 0.3016521075  0.004652974857
## 20 1.875880313 0.3330886098 -0.025548538004
```

```r
gev_mixture_model$full_normalized_gev_parameters_object
```

```
##       loc_star   scale_star      shape_star
## 9  1.845624454 0.3206957588 -0.001081924160
## 10 1.837570294 0.3314580093 -0.006273752669
## 11 1.808753181 0.3463504607 -0.020892002529
## 12 1.722189698 0.4176441446 -0.074421281774
## 13 1.959028538 0.2712030901  0.033484042256
## 14 1.746248247 0.4105063672 -0.073446866095
## 15 1.907913412 0.3149986754 -0.012050541600
## 16 1.764163417 0.3825242472 -0.056975381879
## 17 1.936650498 0.2965818747  0.005617539725
## 18 1.909277773 0.3171843131 -0.014759142835
## 19 1.885721374 0.3016521075  0.004652974857
```

```
## 20 1.875880313 0.3330886098 -0.025548538004
```

```
gev_mixture_model$automatic_weights_pw_shape
```

```
##             9            10            11            12            13
## 0.02312738997 0.03101488536 0.06393609169 0.24538489234 0.02469499368
##            14            15            16            17            18
## 0.24146363215 0.04246145036 0.17519622188 0.01352638255 0.04904039558
##            19            20
## 0.01490808996 0.07524557447
```

```
gev_mixture_model$automatic_weights_pw_scale
```

```
##              9             10             11             12             13
## 0.003020800873 0.017799514752 0.070145251071 0.331166021885 0.027312523108
##             14             15             16             17             18
## 0.304646499014 0.004165298900 0.201284418977 0.007857102566 0.003726407078
##             19             20
## 0.006841910823 0.022034250953
```

```
gev_mixture_model$automatic_weights_pw_loc
```

```
##               9              10              11              12              13
## 0.000000000e+00 0.000000000e+00 3.022861923e-05 4.914970946e-01 0.000000000e+00
##              14              15              16              17              18
## 3.175584283e-01 0.000000000e+00 1.909142485e-01 0.000000000e+00 0.000000000e+00
##              19              20
## 0.000000000e+00 0.000000000e+00
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object[3, ]
```

```
##                      loc_star   scale_star     shape_star
## automatic_weights 1.745255856 0.3927180792 -0.04971948363
```

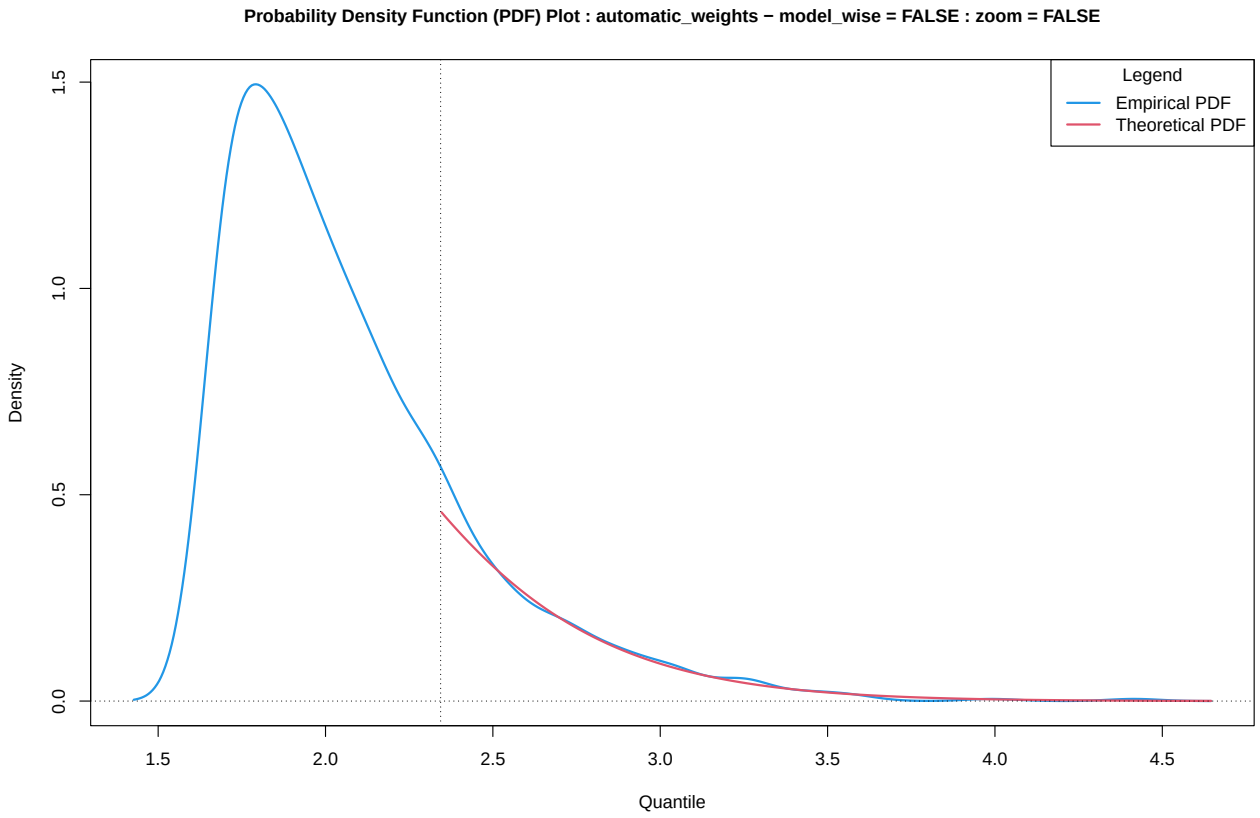
```
gev_mixture_model$automatic_weights_mw
```

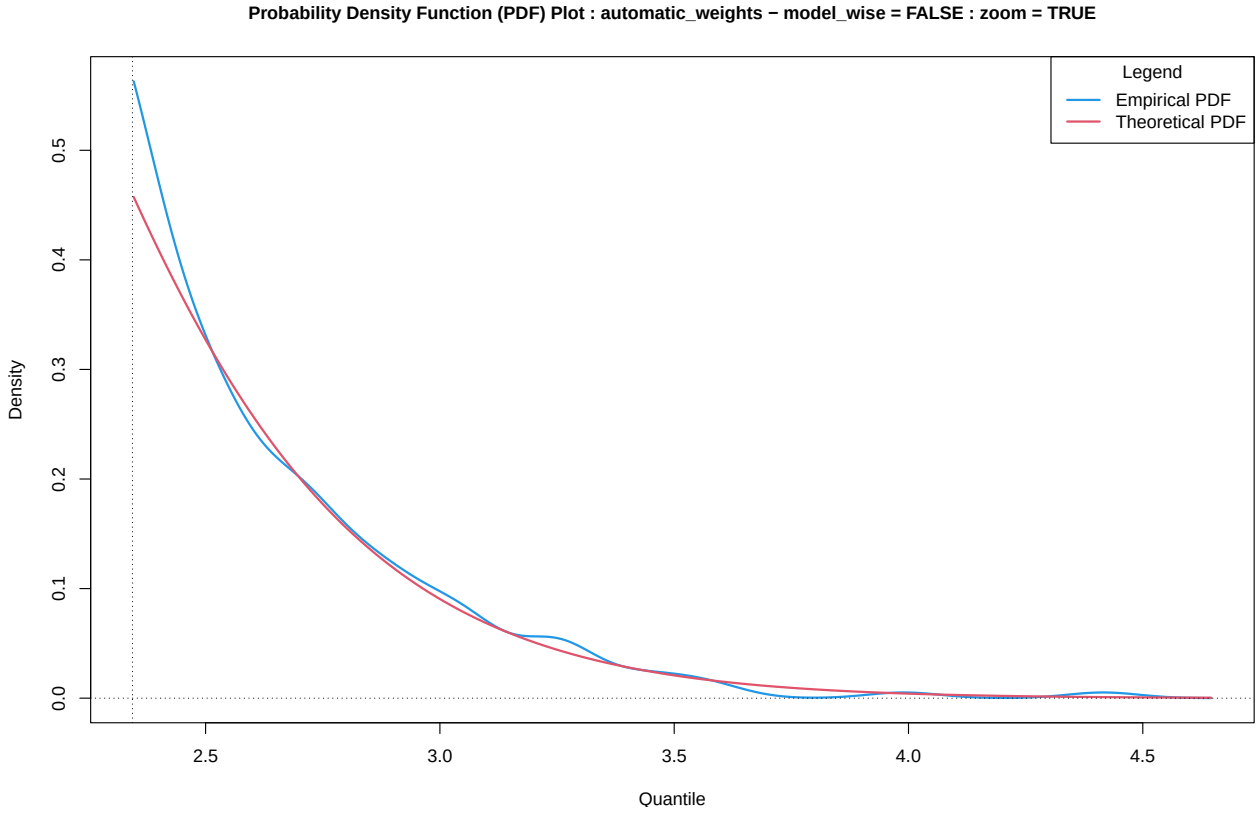
```
##           9          10          11          12          13          14
## 0.000000000 0.000000000 0.000000000 0.152294128 0.000000000 0.000000000
##          15          16          17          18          19          20
## 0.000000000 0.847705872 0.000000000 0.000000000 0.000000000 0.000000000
```

```
# Model diagnostics

## GEV mixture model with respect to parameters
```

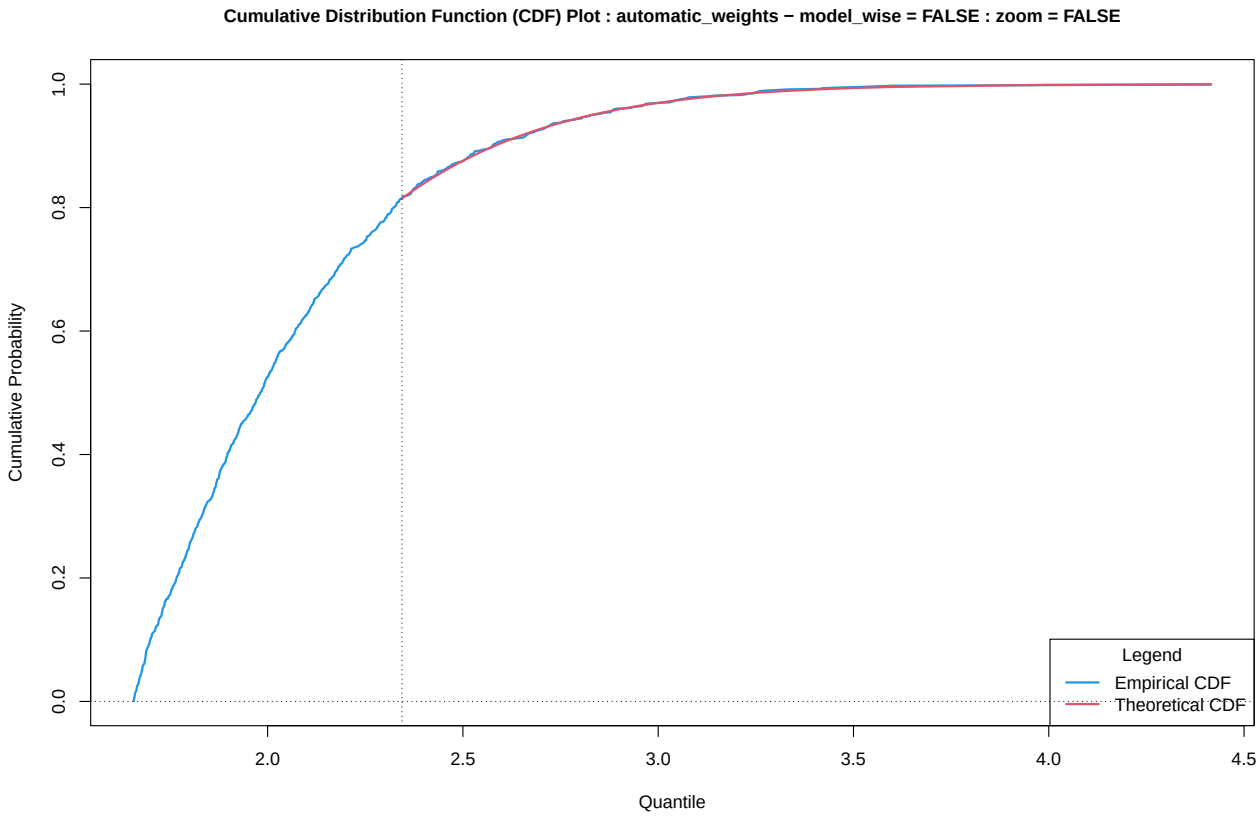
```
par(mfrow = c(2, 1))
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           main = "Probability Density Function (PDF) Plot")

plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
```

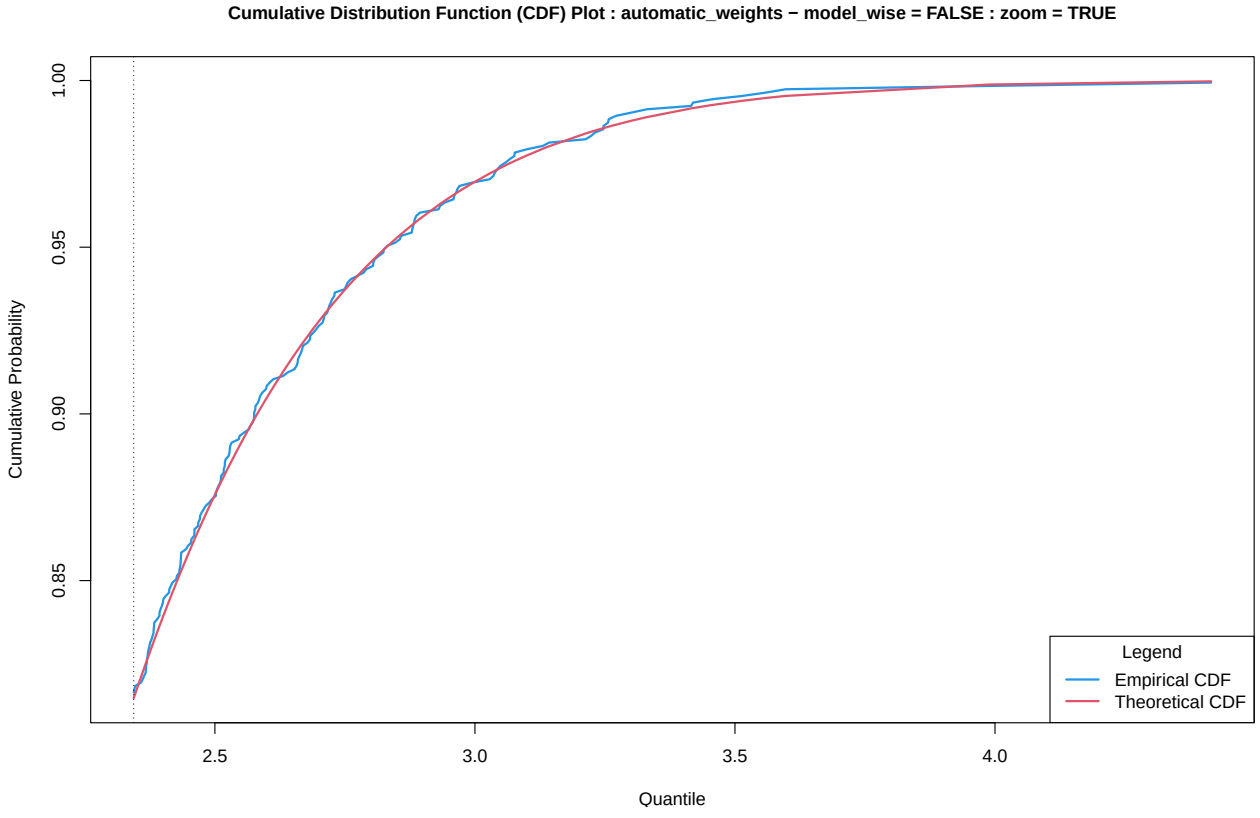
```
                        xlab = "Quantile",
                        ylab = "Density",
                        main = "Probability Density Function (PDF) Plot")
```


Probability Density Function (PDF) Plot : automatic_weights − model_wise = FALSE : zoom = FALSE
Legend
Empirical PDF
Theoretical PDF
Density
0.0
0.5
1.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Quantile


Probability Density Function (PDF) Plot : automatic_weights − model_wise = FALSE : zoom = TRUE
Legend
Empirical PDF
Theoretical PDF
Density
0.0
0.1
0.2
0.3
0.4
0.5
2.5
3.0
3.5
4.0
4.5
Quantile

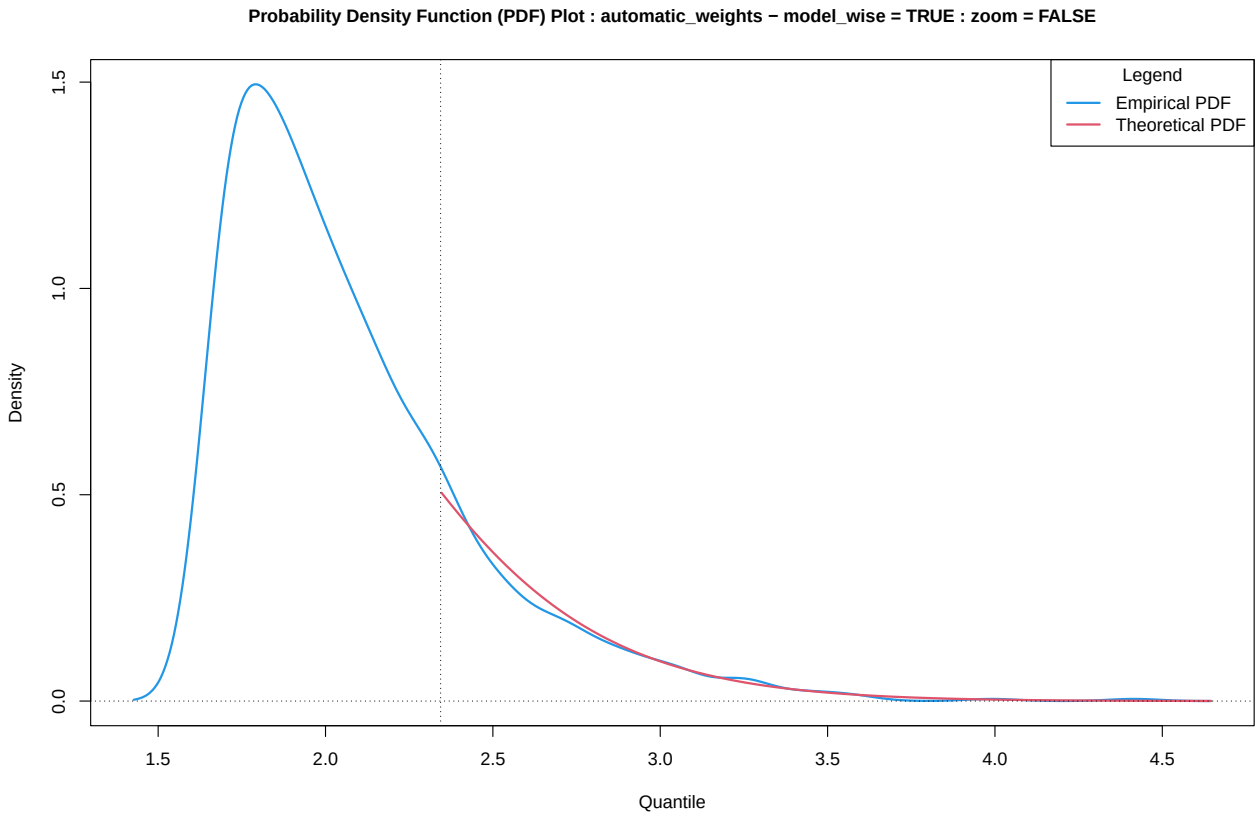
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Cumulative Probability",
                           main = "Cumulative Distribution Function (CDF) Plot")

plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Cumulative Probability",
                           main = "Cumulative Distribution Function (CDF) Plot")
```

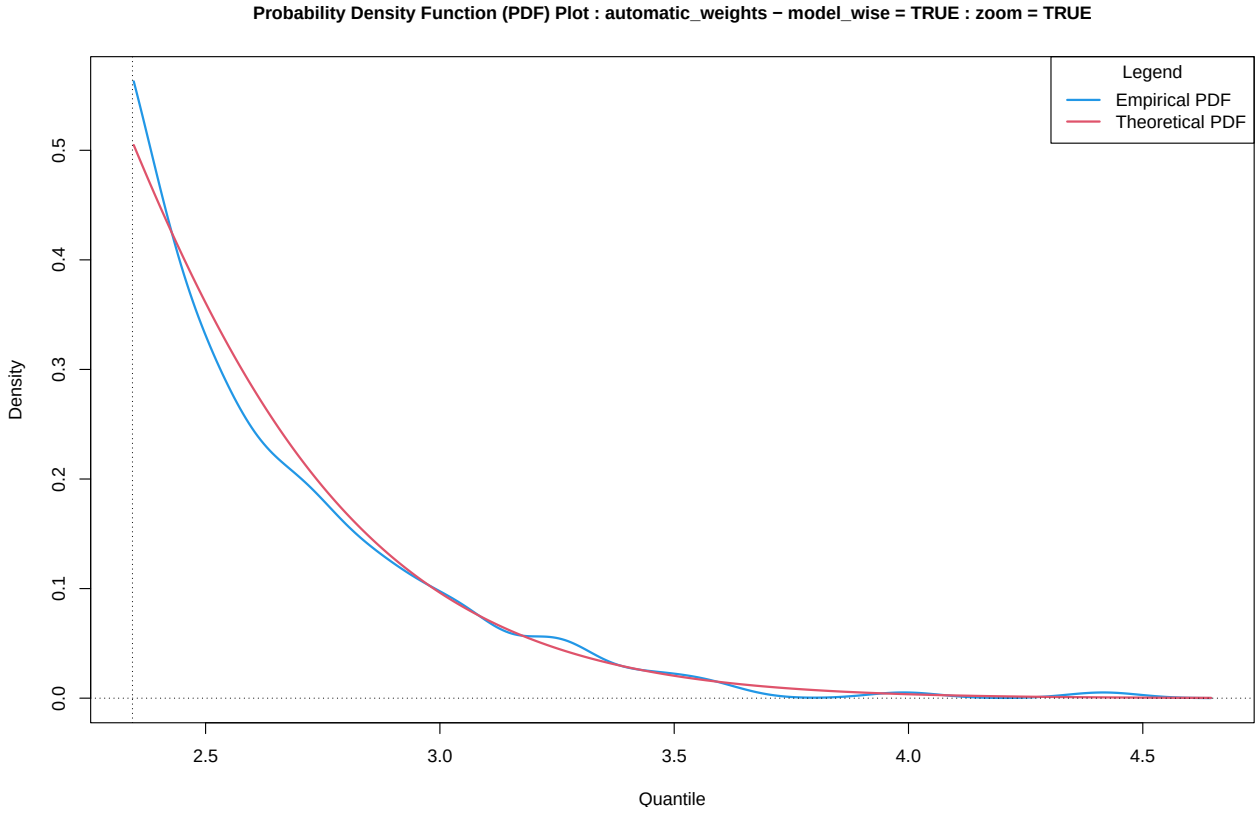

Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = FALSE : zoom = FALSE
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
2.0
2.5
3.0
3.5
4.0
4.5
Quantile
Legend
Empirical CDF
Theoretical CDF


Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = FALSE : zoom = TRUE
Cumulative Probability
0.85
0.90
0.95
1.00
2.5
3.0
3.5
4.0
Quantile
Legend
Empirical CDF
Theoretical CDF

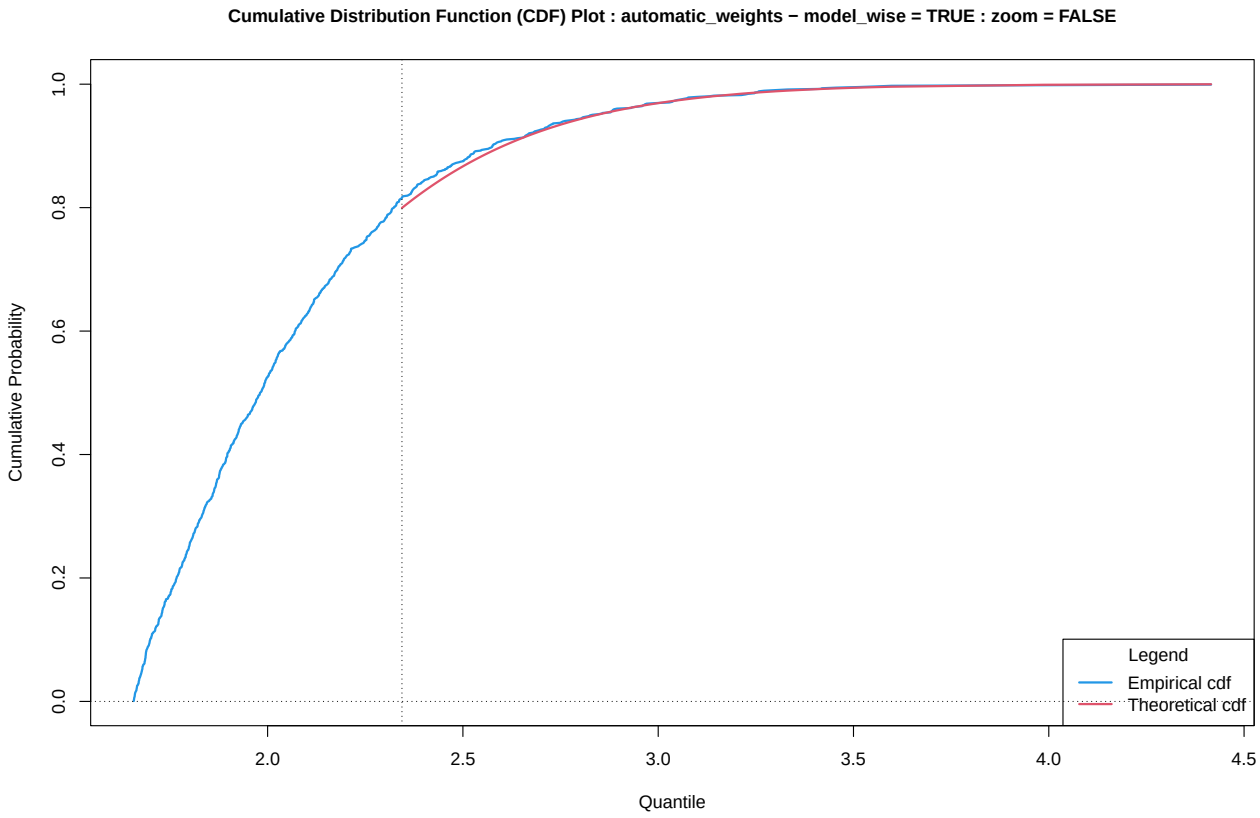
```r
## GEV mixture model with respect to distribution functions
```

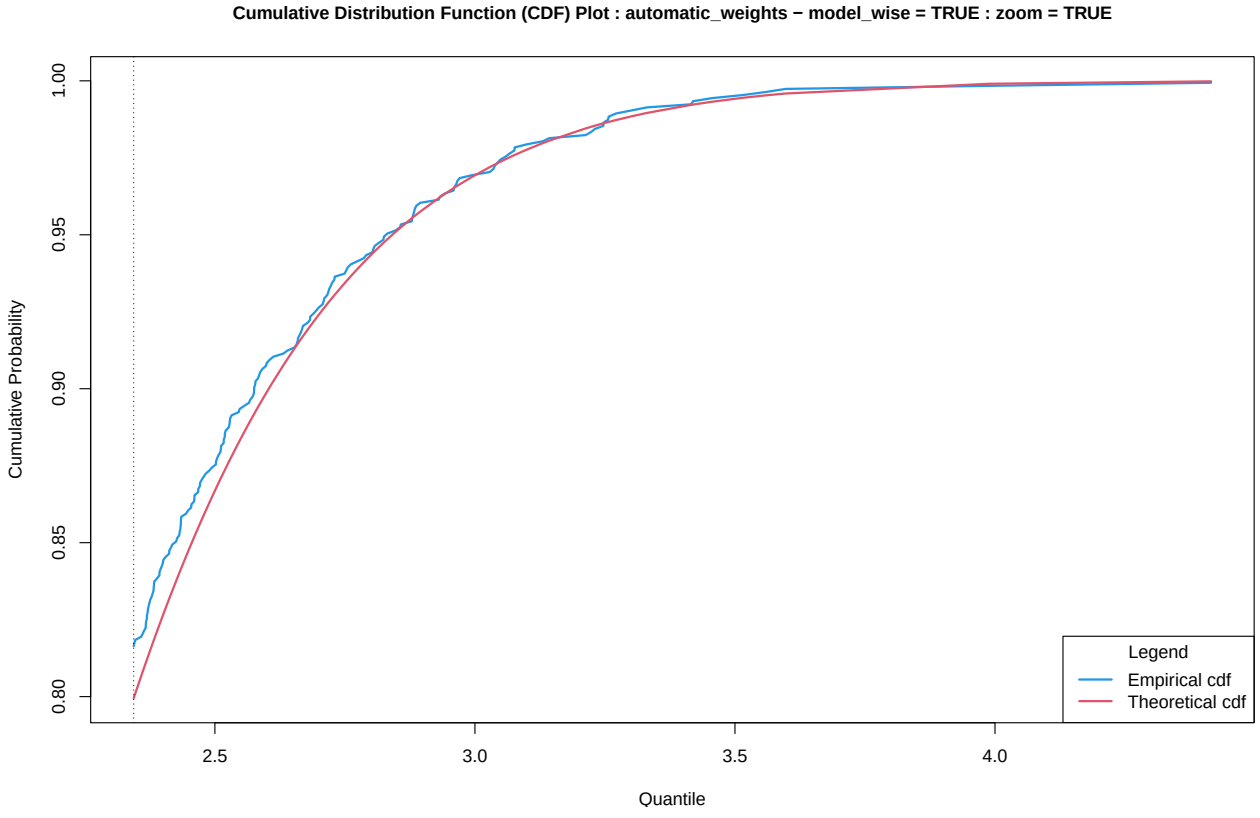
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           main = "Probability Density Function (PDF) Plot")

plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                           main = "Probability Density Function (PDF) Plot")
```


Probability Density Function (PDF) Plot : automatic_weights − model_wise = TRUE : zoom = FALSE
Legend
Empirical PDF
Theoretical PDF
Density
0.0
0.5
1.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Quantile


Probability Density Function (PDF) Plot : automatic_weights − model_wise = TRUE : zoom = TRUE
Legend
Empirical PDF
Theoretical PDF
Density
0.0
0.1
0.2
0.3
0.4
0.5
2.5
3.0
3.5
4.0
4.5
Quantile

```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Cumulative Probability",
                           main = "Cumulative Distribution Function (CDF) Plot")

plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Cumulative Probability",
                           main = "Cumulative Distribution Function (CDF) Plot")
```


Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = TRUE : zoom = FALSE
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
2.0
2.5
3.0
3.5
4.0
4.5
Quantile
Legend
Empirical cdf
Theoretical cdf


Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = TRUE : zoom = TRUE
Cumulative Probability
0.80
0.85
0.90
0.95
1.00
2.5
3.0
3.5
4.0
Quantile
Legend
Empirical cdf
Theoretical cdf

```r
# Estimation of an extreme quantile
```

```r
estimator_types <- c("automatic_weights_mw",
                     "pessimistic_weights_mw",
                     "identic_weights_mw",
                     "automatic_weights_pw",
                     "pessimistic_weights_pw",
                     "identic_weights_pw",
                     "empirical",
                     "confidence_interval_mw",
                     "confidence_interval_pw")
```

```r
alpha <- 10^(-14)


## Quantile from the true distribution
```

```r
true_rl <- qnorm(p = 1 - alpha)
true_rl
```

```
## [1] 7.650730905
```

```r
## Quantile from GEV mixture model with respect to parameters
```

```r
rl_pw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                             alpha = alpha,
                                             confidence_level = 0.95,
                                             do.ci = TRUE,
                                             estimator_type = estimator_types[4])

rl_pw[2]
```

```
##      estimate
## 1 7.798224505
```

```r
## Quantile from GEV mixture model with respect to distribution functions
```

```r
rl_mw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                             alpha = alpha,
                                             confidence_level = 0.95,
                                             do.ci = TRUE,
                                             estimator_type = estimator_types[1])

rl_mw[2]
```

```
##      estimate
## 1 7.204458308
```

```r
## Quantiles from equivalent estimated GEV models
```

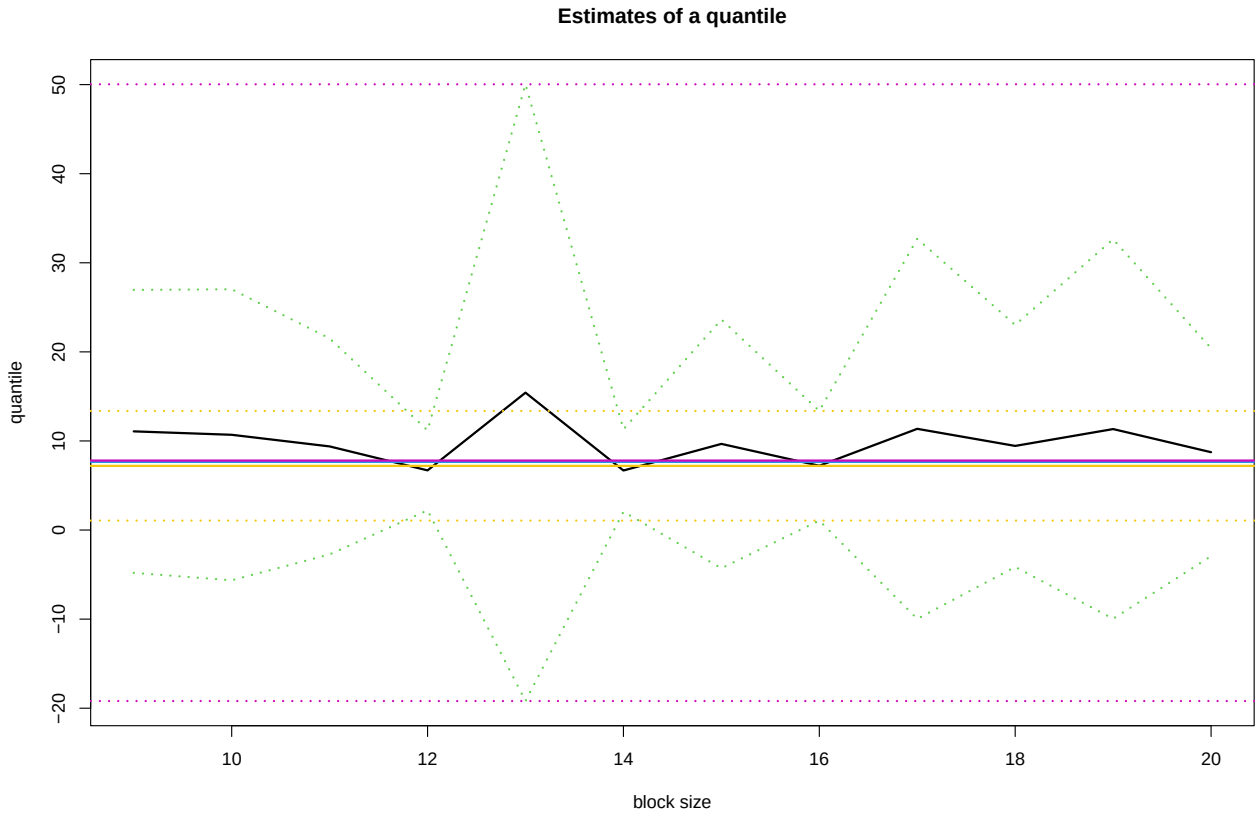
```r
est_rl_pw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                 alpha = alpha,
                                                 confidence_level = 0.95,
                                                 do.ci = TRUE,
                                                 estimator_type = estimator_types[9])

est_rl_pw
```

```
##           lower    estimate       upper
## 9  -4.803730588 11.07613695 26.95600449
## 10  -5.63856595 10.69253235 27.02363064
## 11 -2.738283589 9.387172096 21.51262778
## 12  2.214177861 6.697233859 11.18028986
## 13 -19.19482103 15.42058183 50.03598468
## 14  2.058566143 6.682809638 11.30705313
## 15 -4.289433657 9.670785408 23.63100447
## 16  1.061798493 7.216405174 13.37101186
## 17 -9.971684116  11.3617438 32.69517172
## 18 -4.139372125 9.441930519 23.02323316
## 19 -9.961066483 11.33436841  32.6298033
## 20 -2.933361236 8.736702841 20.40676692
```

```
## Comparison of estimated quantiles
```

```
est_rl_pw_range <- range(as.matrix(est_rl_pw))

est_rl_mw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                 alpha = alpha,
                                                 confidence_level = 0.95,
                                                 do.ci = TRUE,
                                                 estimator_type = estimator_types[8])

est_rl_mw_range <- range(as.matrix(est_rl_mw))
```

```
matplot(x = rownames(est_rl_pw),
        y = est_rl_pw,
        xlab = "block size",
        ylab = "quantile",
        main = "Estimates of a quantile",
        ylim = range(c(est_rl_pw_range, true_rl)),
        cex = 1,
        cex.lab = 1,
        cex.axis = 1,
        type = "l",
        lty = c("dotted", "solid", "dotted"),
        lwd = c(2,2,2),
        col = c(3, 1, 3))

abline(h = true_rl, col = 4, lwd = 2)
abline(h = rl_mw[2], col = 7, lwd = 2)
abline(h = rl_pw[2], col = 6, lwd = 2)
abline(h = est_rl_pw_range, col = 6, lty = "dotted", lwd = 2)
abline(h = est_rl_mw_range, col = 7, lty = "dotted", lwd = 2)
```




```
# Legend:
# blue: Quantile from the true distribution
# yellow: Quantile from GEV mixture model with respect to distribution functions
# pink: Quantile from GEV mixture model with respect to parameters
```