Modeling extreme values with a GEV mixture probability distributions

Standard Gumbel distribution

Pascal Alain Dkengne Sielenou

2023-10-12

```
# Load useful functions
```

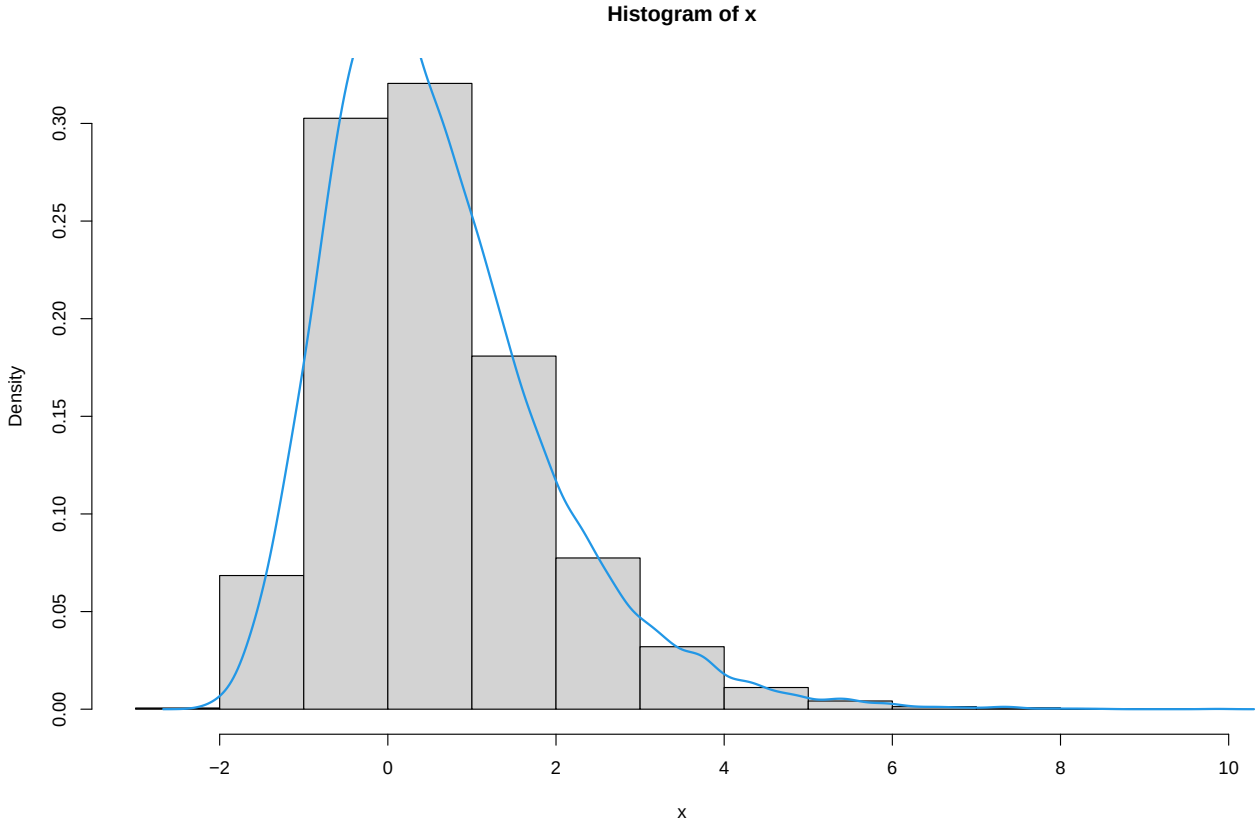
```
path <- ".."
```

```
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
```

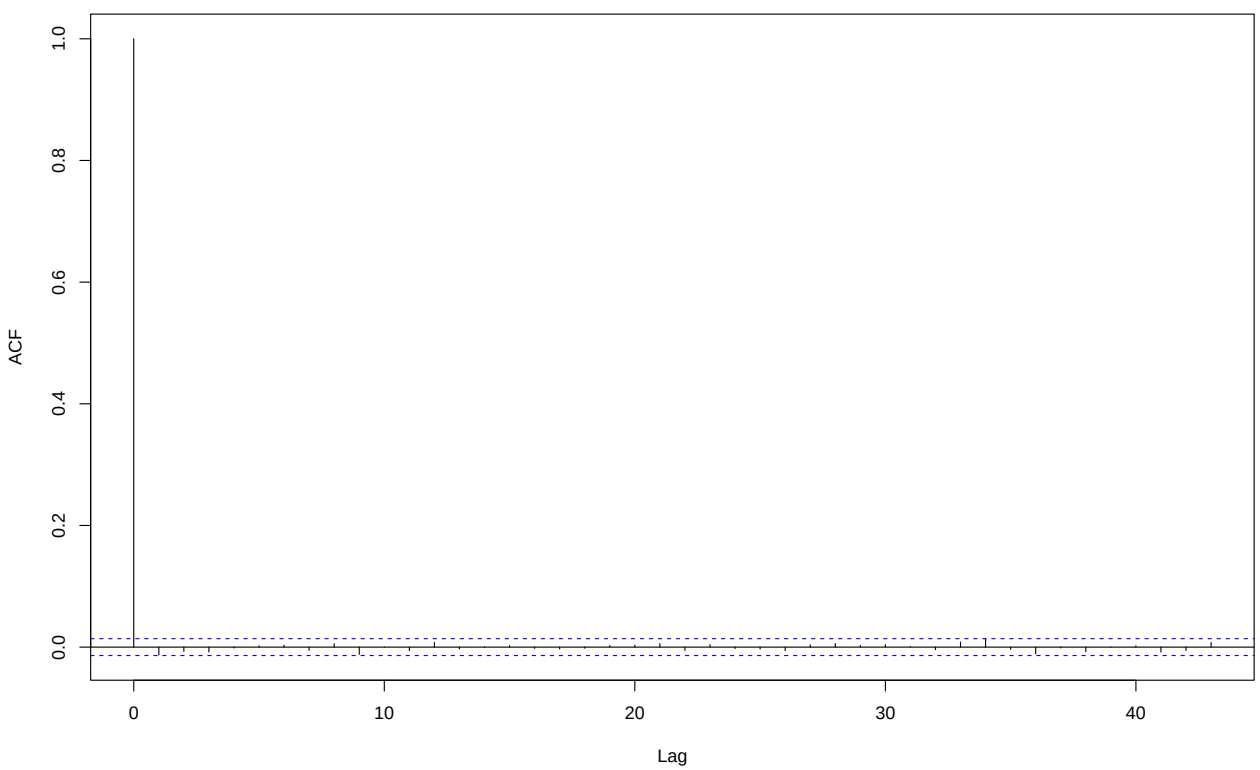
```
loc <- 0
scale <- 1
shape <- 0

set.seed(1122)
x <- generate_gev_sample(n = n, loc = loc, scale = scale, shape = shape)

# Histogram of all data
```

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



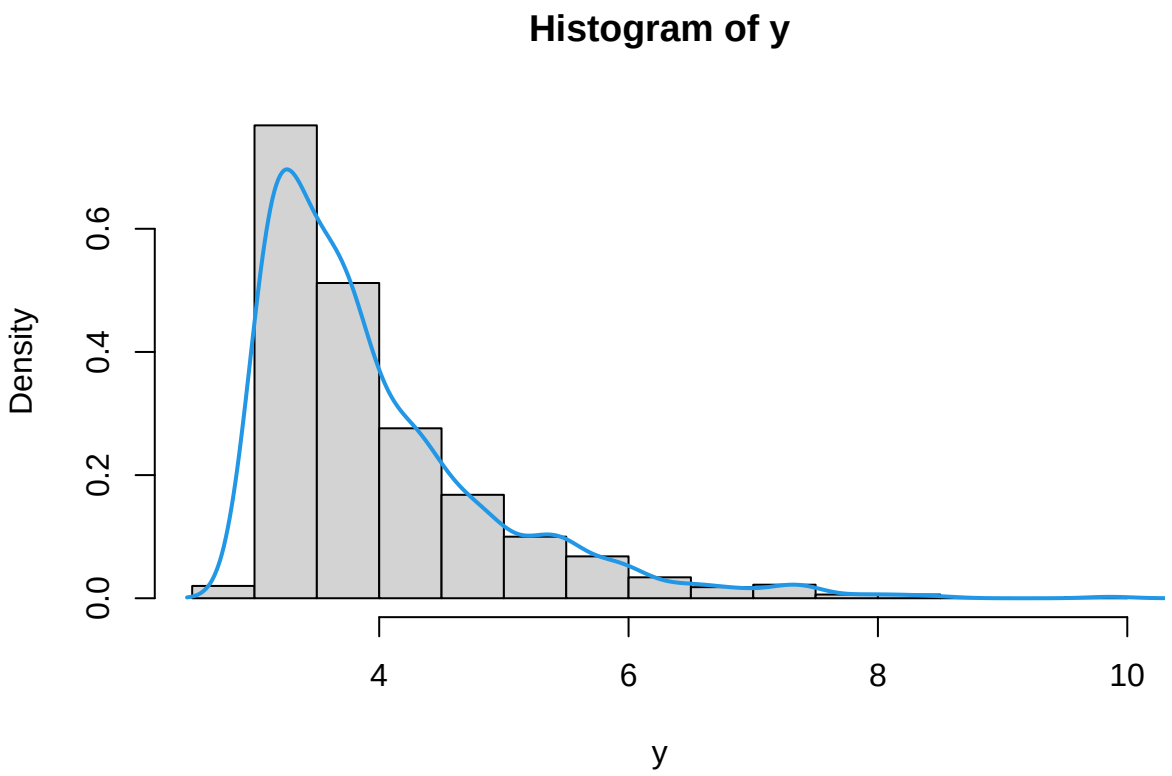

```
# Autocorrelation function of all data
```

```
acf(x)
```



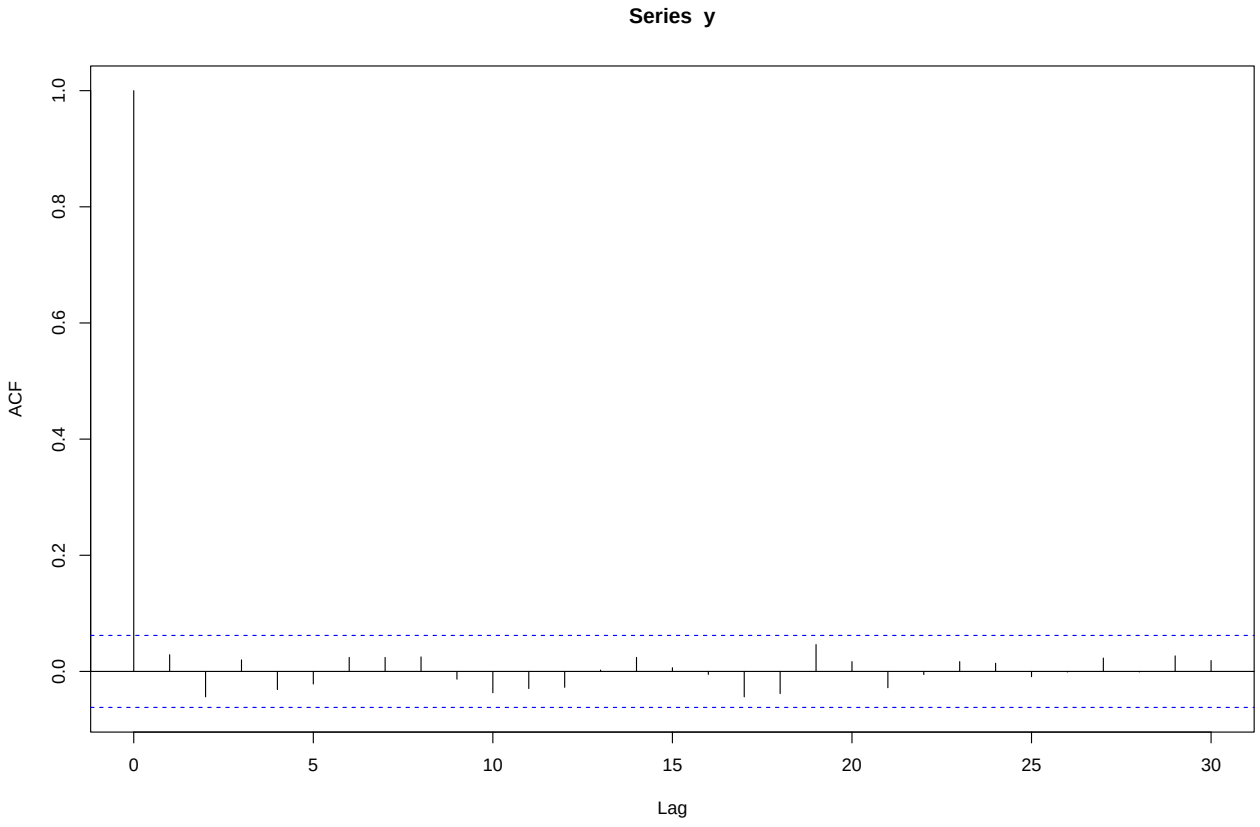

```
# Histogram of the largest data
```

```
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
```

```
##           13           14           15           16           17           18
## 0.8920296849 0.9412167819 0.9109104376 0.9663117674 1.0000000000 1.0000000000
##           19           20
## 0.9153737685 1.0000000000
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##       loc_star   scale_star     shape_star
## 13 2.738730574 1.2972204786 -0.11094214484
## 14 3.173500173 1.0537637674 -0.06604485185
## 15 2.962237100 1.1338401059 -0.07956577765
## 16 2.943030605 1.1978828474 -0.09487103167
## 17 2.640938537 1.3340390618 -0.11772435182
## 18 3.339054187 0.9284680982 -0.02948815858
## 19 2.200619920 1.5943004373 -0.15367097316
## 20 3.387462307 0.9167513605 -0.02821859821
```

```r
gev_mixture_model$full_normalized_gev_parameters_object
```

```
##       loc_star   scale_star     shape_star
## 13 2.589572170 1.3137684319 -0.11094214484
## 14 3.109533392 1.0579884440 -0.06604485185
## 15 2.856043969 1.1422894450 -0.07956577765
## 16 2.901913848 1.2017836365 -0.09487103167
## 17 2.640938537 1.3340390618 -0.11772435182
## 18 3.339054187 0.9284680982 -0.02948815858
## 19 2.058685279 1.6161116717 -0.15367097316
## 20 3.387462307 0.9167513605 -0.02821859821
```

```r
gev_mixture_model$automatic_weights_pw_shape
```

```
##           13           14           15           16           17           18
## 0.1340587202 0.1178963789 0.1227637377 0.1282733987 0.1365001850 0.1057314779
##           19           20
## 0.1494401683 0.1053359333
```

```r
gev_mixture_model$automatic_weights_pw_scale
```

```
##           13           14           15           16           17           18
## 0.1168023571 0.1315041849 0.1253806495 0.1224047623 0.1157881200 0.1427175822
##           19           20
## 0.1016704514 0.1437318925
```

```r
gev_mixture_model$automatic_weights_pw_loc
```

```
##           13           14           15           16           17           18
## 0.1161439230 0.1315013058 0.1229014897 0.1244577977 0.1169475723 0.1392863559
##           19           20
## 0.1078334539 0.1409281017
```

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

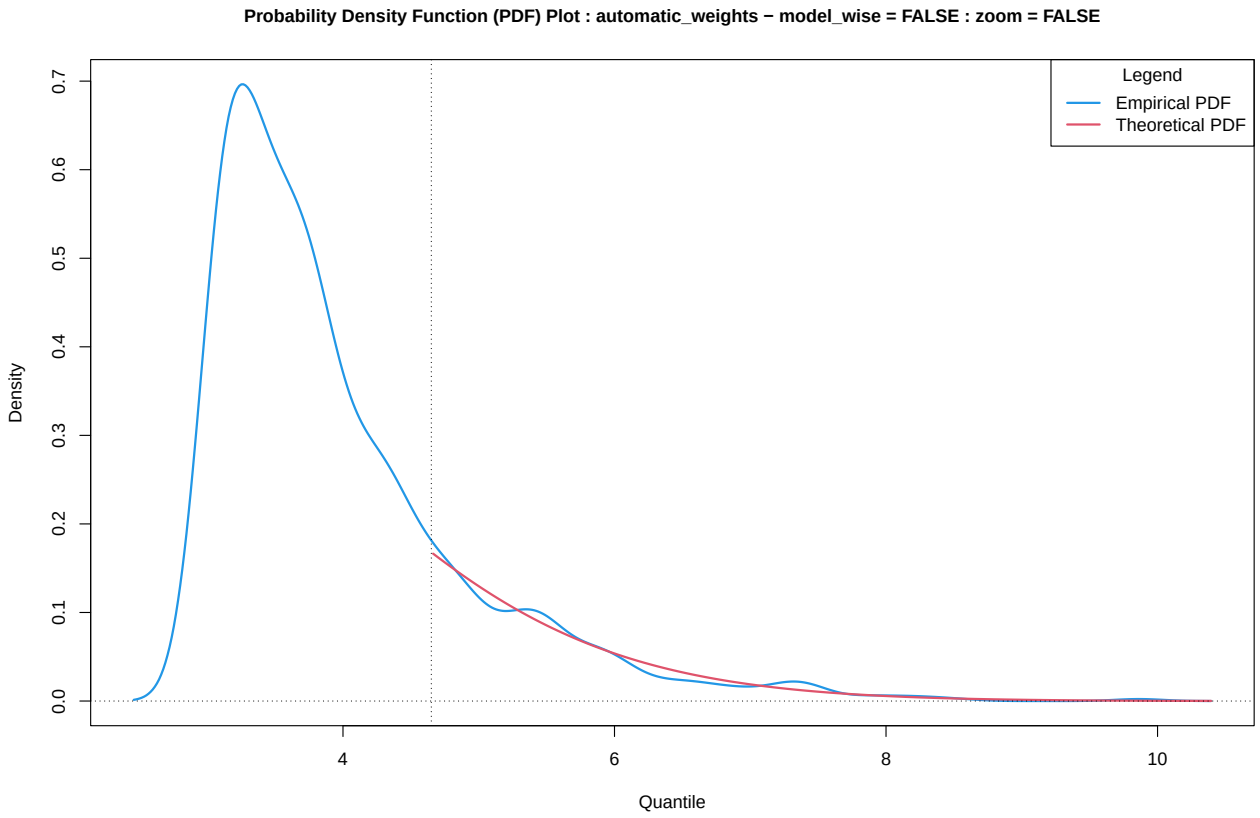
```
##                      loc_star  scale_star     shape_star
## automatic_weights 2.95437791 1.159715179 -0.08972070362
```

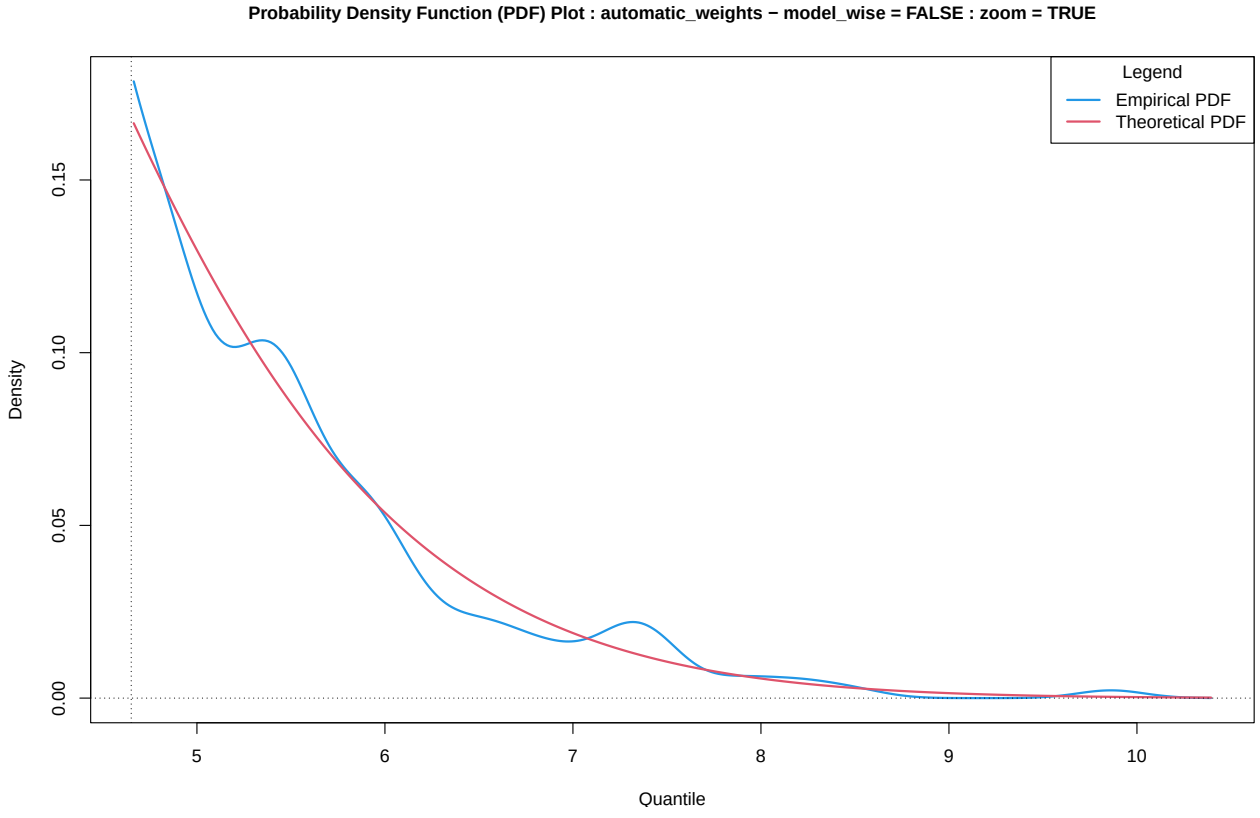
```r
gev_mixture_model$automatic_weights_mw
```

```
##            13            14            15            16            17
## 0.14900807916 0.30984800794 0.33393783784 0.16664609783 0.04055997724
##            18            19            20
## 0.00000000000 0.00000000000 0.00000000000
```

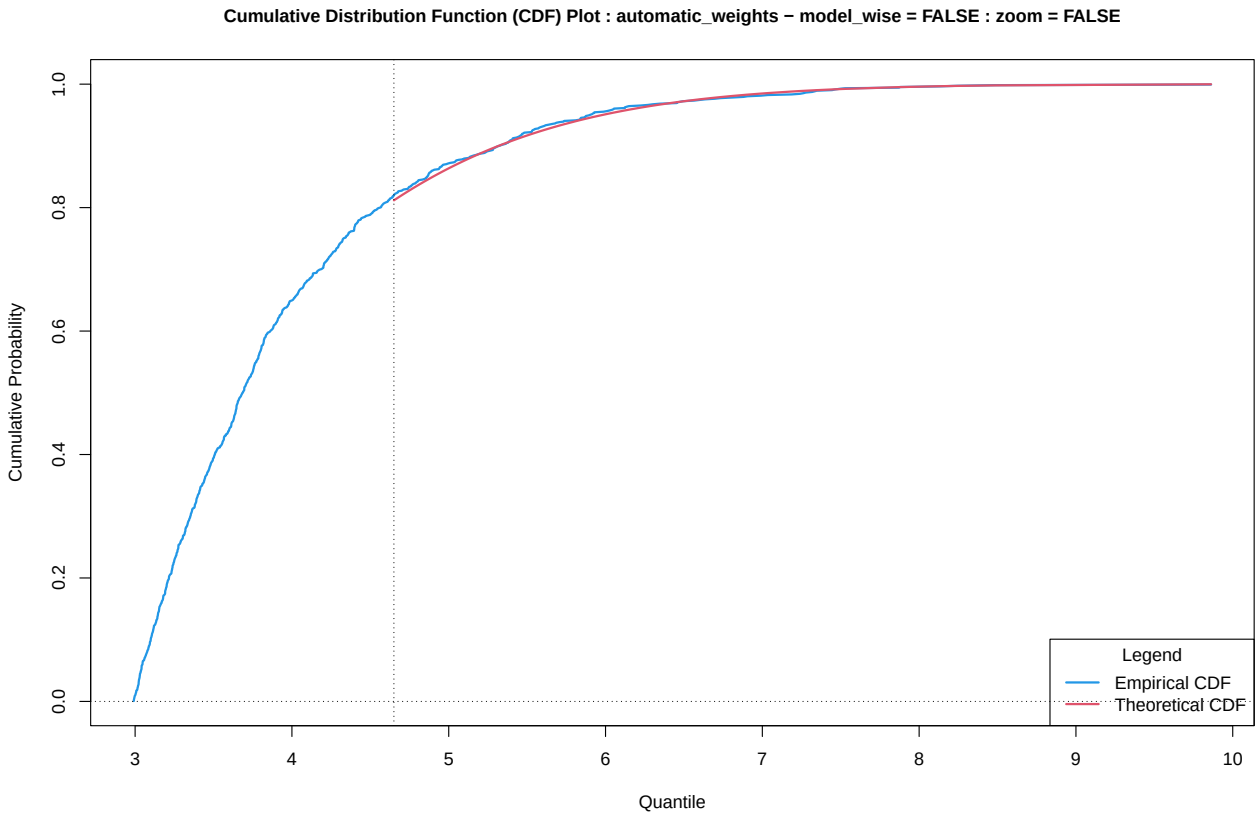
```r
# Model diagnostics

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

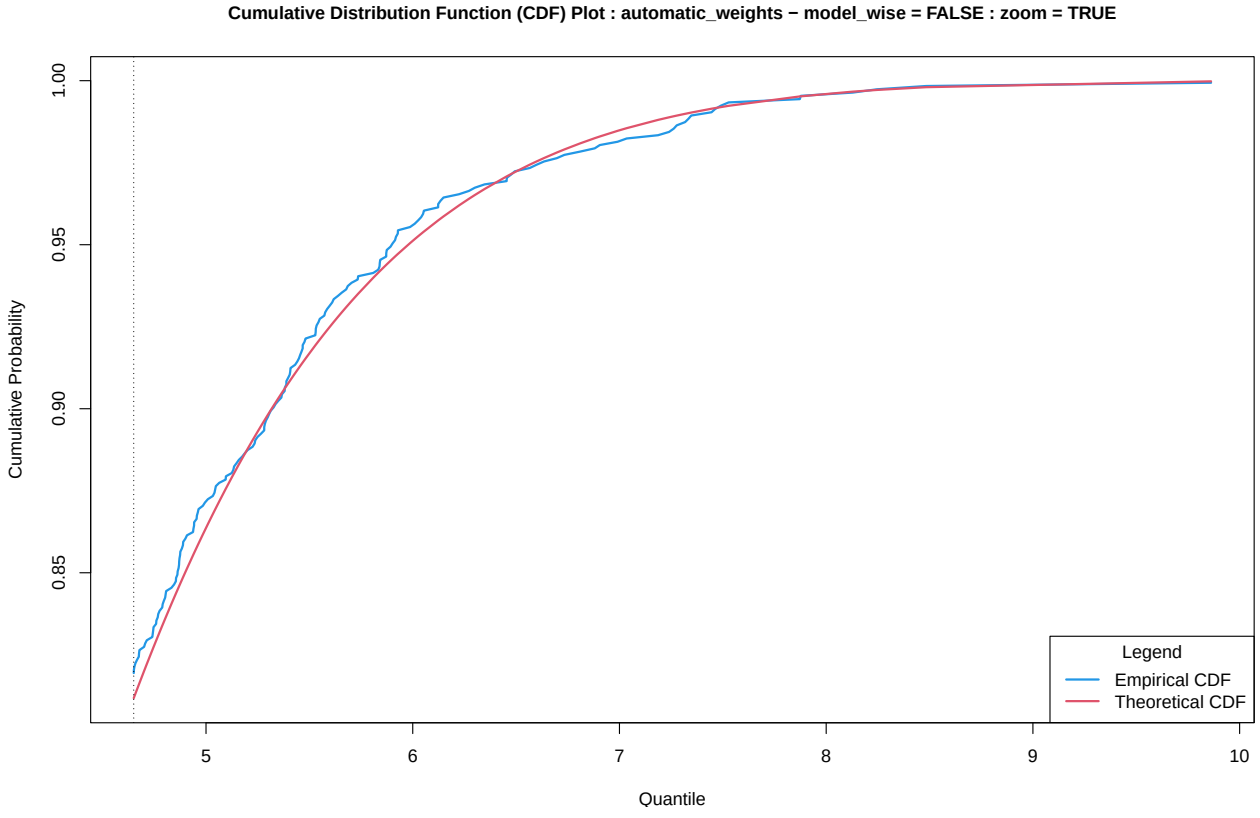
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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
0.1
0.2
0.3
0.4
0.5
0.6
0.7
4
6
8
10
Quantile


Probability Density Function (PDF) Plot : automatic_weights − model_wise = FALSE : zoom = TRUE
Legend
Empirical PDF
Theoretical PDF
Density
0.00
0.05
0.10
0.15
5
6
7
8
9
10
Quantile

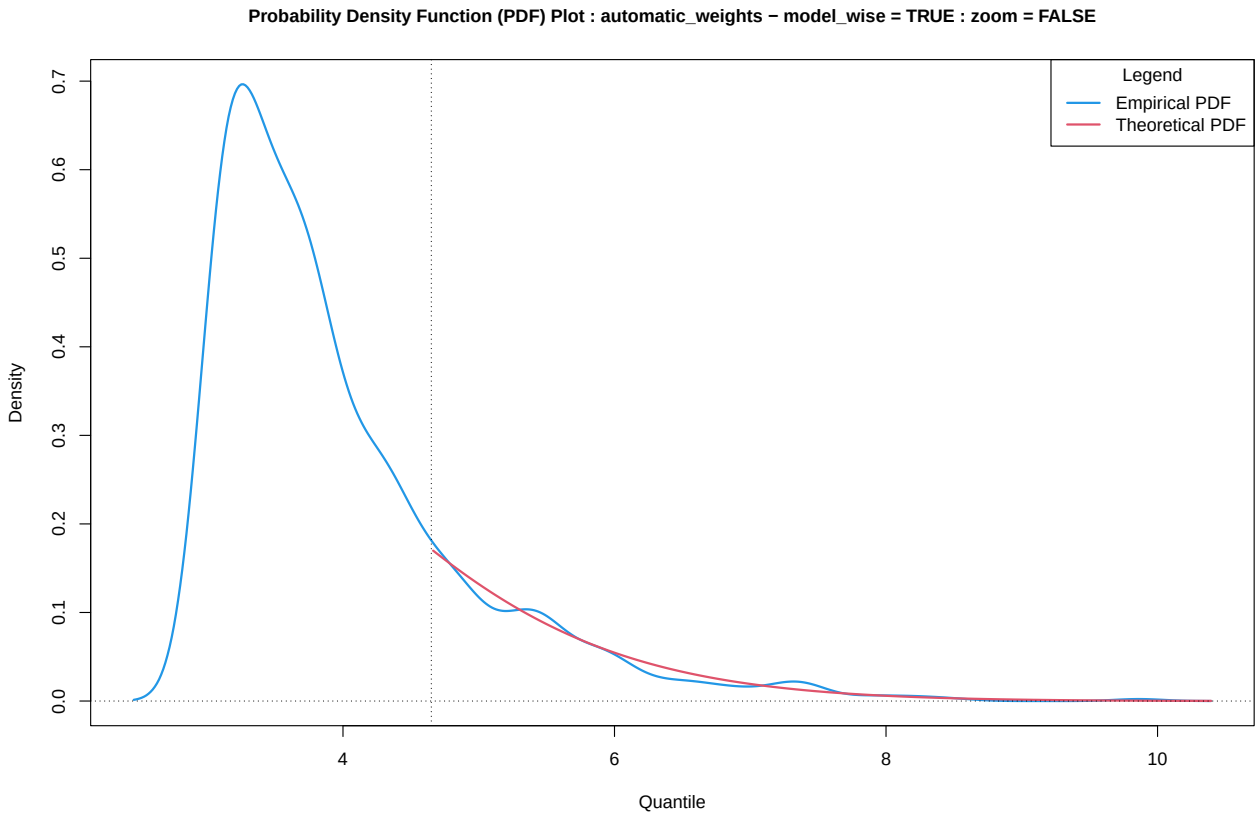
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Cumulative 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")
```

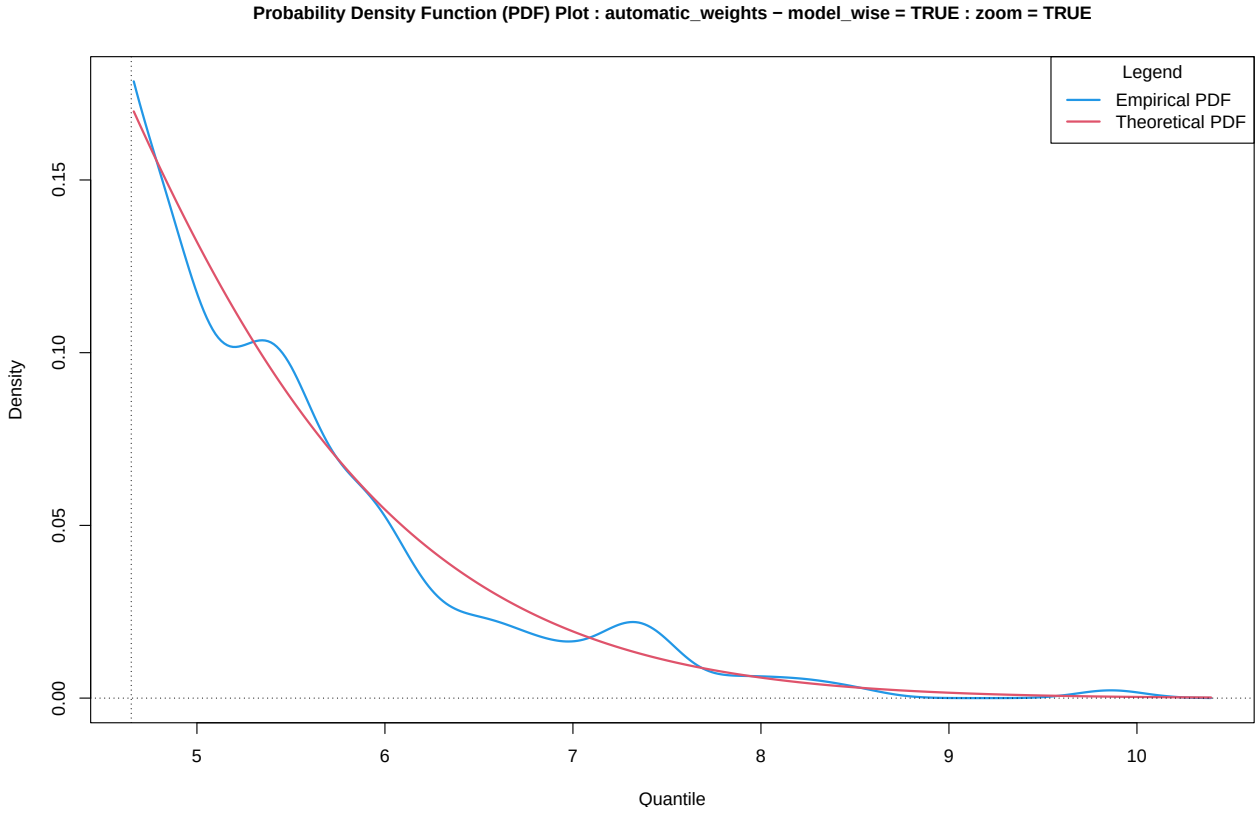

Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = FALSE : zoom = FALSE
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
3
4
5
6
7
8
9
10
Quantile
Legend
Empirical CDF
Theoretical CDF


Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = FALSE : zoom = TRUE
Cumulative Probability
0.85
0.90
0.95
1.00
5
6
7
8
9
10
Quantile
Legend
Empirical CDF
Theoretical CDF

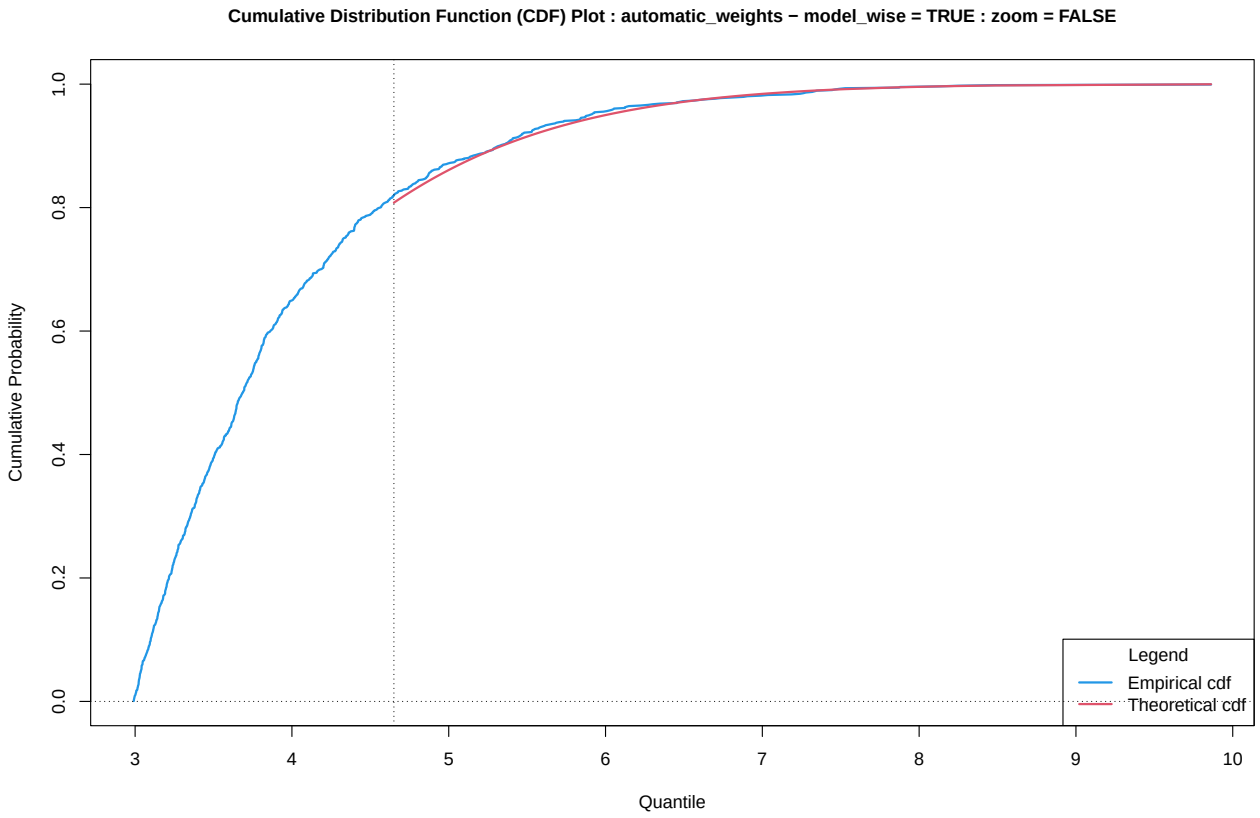
```r
## GEV mixture model with respect to distribution functions
```

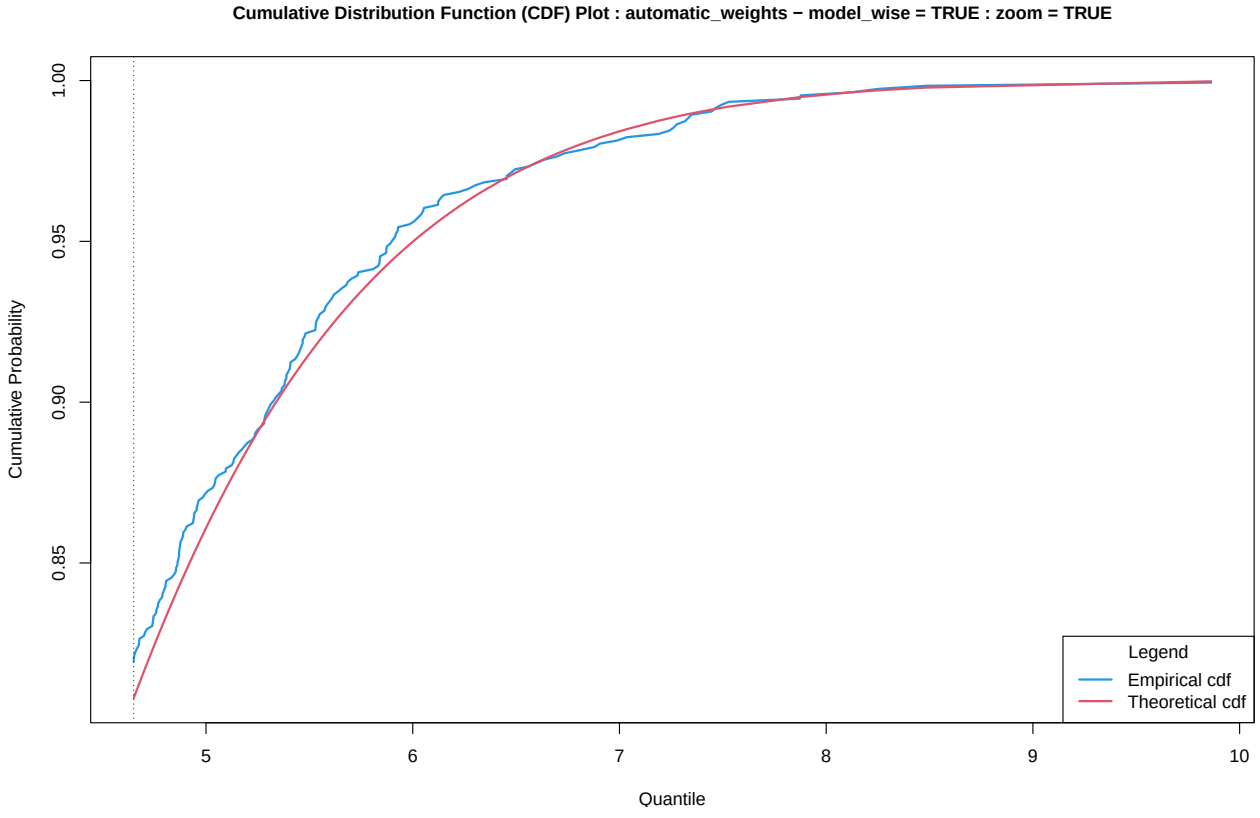
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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
0.1
0.2
0.3
0.4
0.5
0.6
0.7
4
6
8
10
Quantile


Probability Density Function (PDF) Plot : automatic_weights − model_wise = TRUE : zoom = TRUE
Legend
Empirical PDF
Theoretical PDF
Density
0.00
0.05
0.10
0.15
5
6
7
8
9
10
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
3
4
5
6
7
8
9
10
Quantile
Legend
Empirical cdf
Theoretical cdf


Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = TRUE : zoom = TRUE
Cumulative Probability
0.85
0.90
0.95
1.00
5
6
7
8
9
10
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 <- calculate_gev_inverse_cdf(p = 1 - alpha,
                                     loc = loc,
                                     scale = scale,
                                     shape = shape)
true_rl
```

```
## [1] 32.2369909
```

```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 14.94246142
```

```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 16.62950934
```

```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,
```

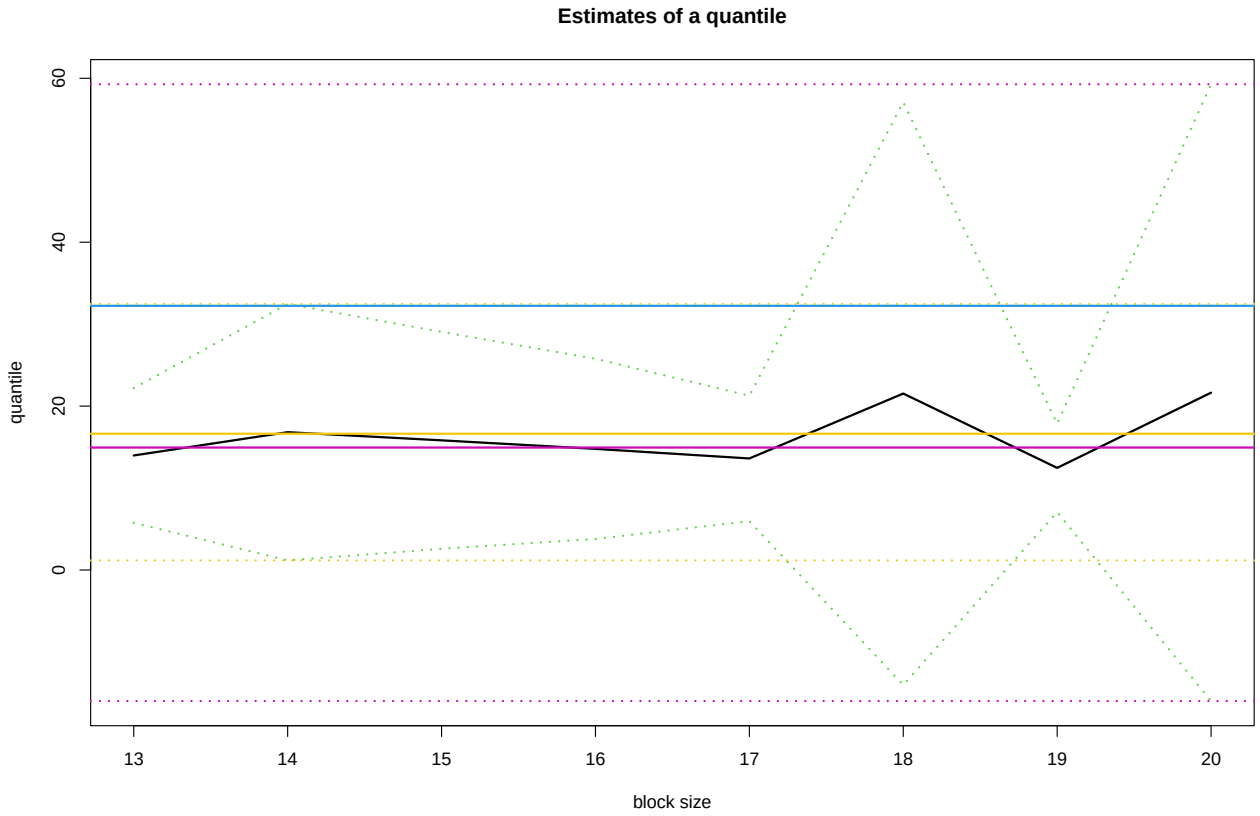
```
                                                estimator_type = estimator_types[9])

est_rl_pw
```

```
##           lower    estimate       upper
## 13  5.767240436 13.97538891 22.18353739
## 14  1.170296259 16.81563188  32.4609675
## 15  2.588521355 15.82128038 29.05403941
## 16  3.787271938 14.78148292  25.7756939
## 17  5.959090952 13.61031275 21.26153455
## 18 -14.02932298  21.5313576 57.09203818
## 19  7.071913005 12.45937114 17.84682927
## 20 -15.99101196 21.63956997 59.27015189
```

```
## 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
```