Modeling extreme values with a GEV mixture probability distributions

Standard Gumbel distribution

Pascal Alain Dkengne Sielenou

2023-10-12

```
# Load useful functions
```

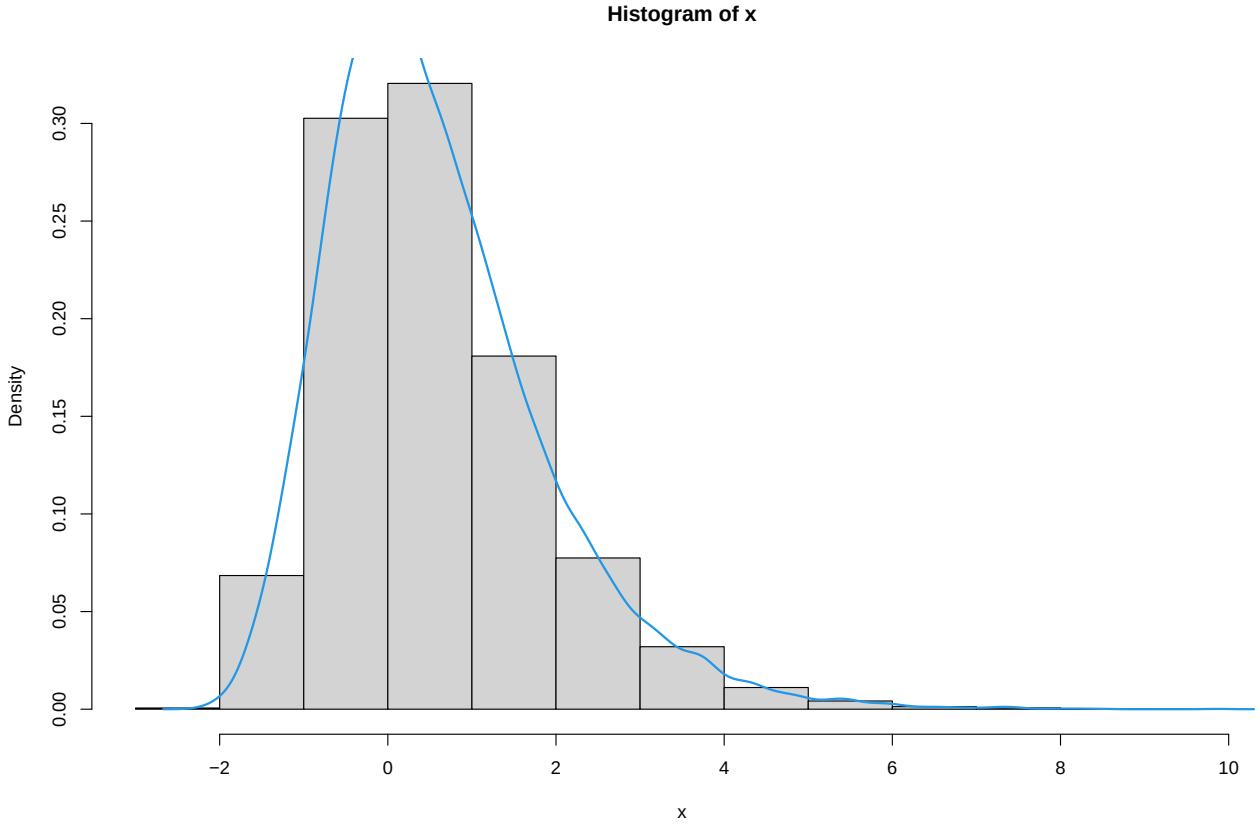
```
path <- ".."
```

```
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
```

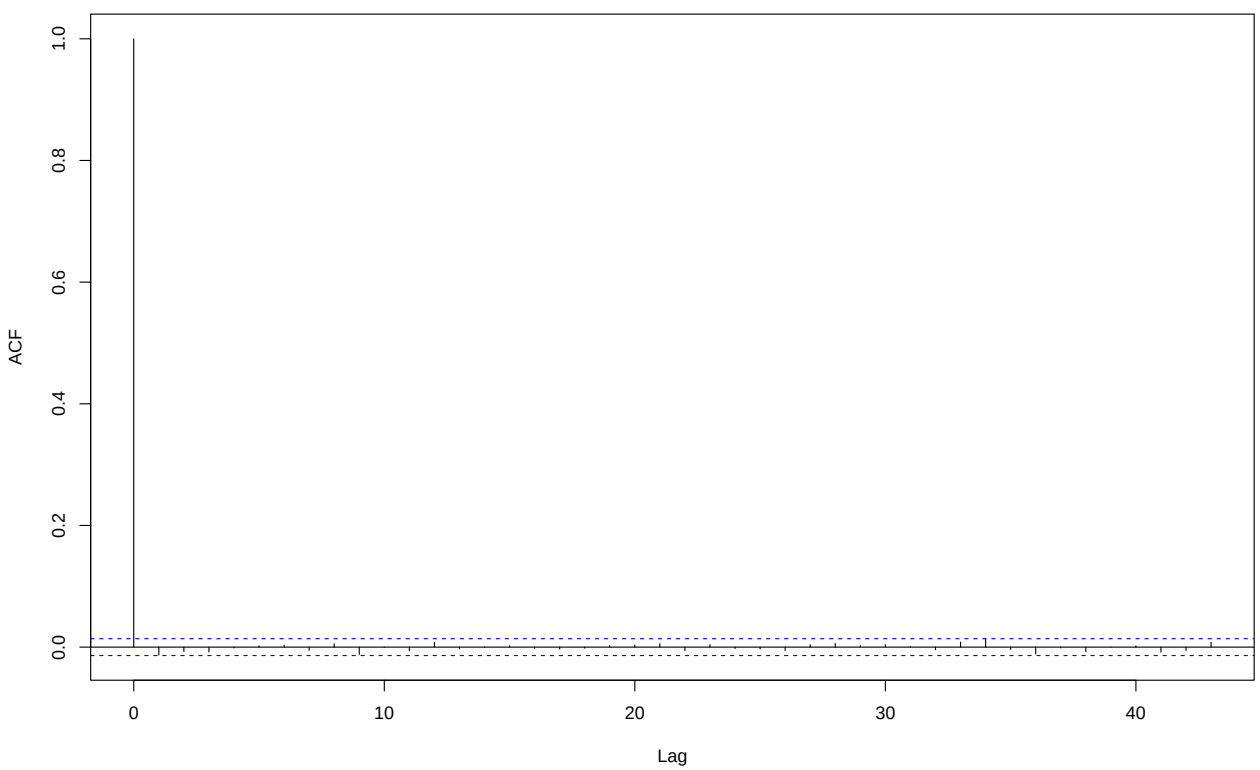
```
loc <- 0
scale <- 1
shape <- 0

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)
```



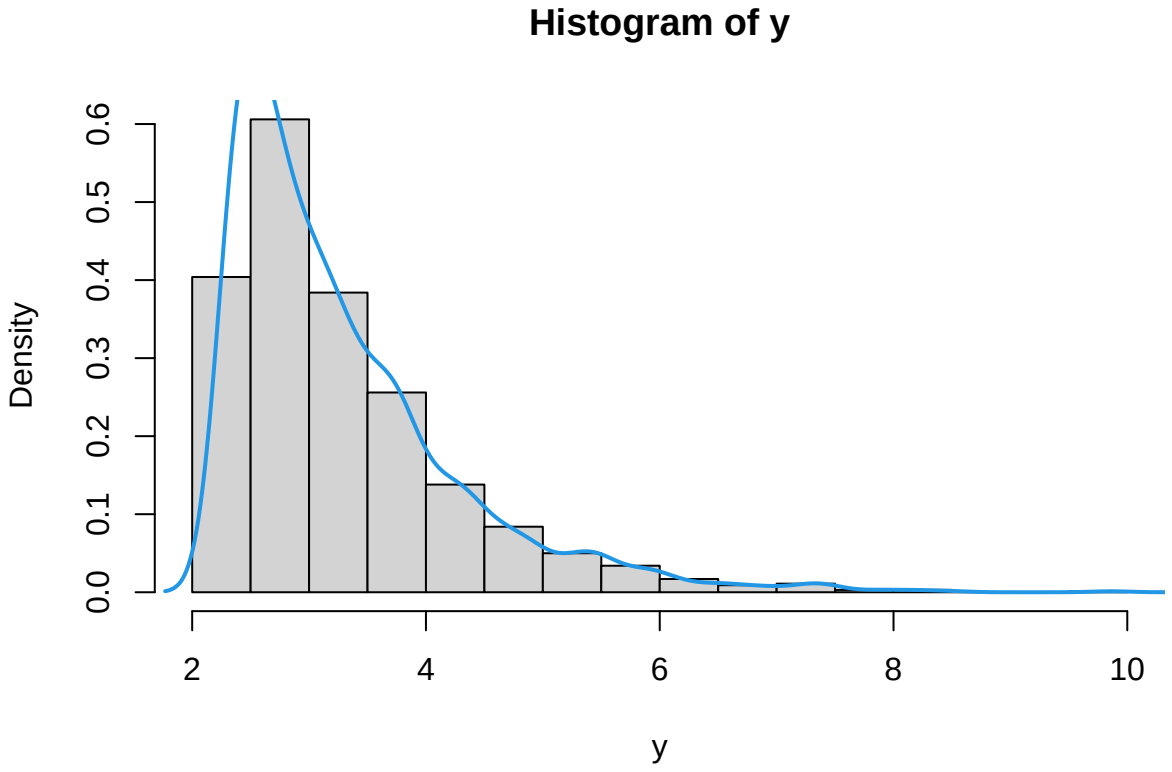

```
# Autocorrelation function of all data
```

```
acf(x)
```



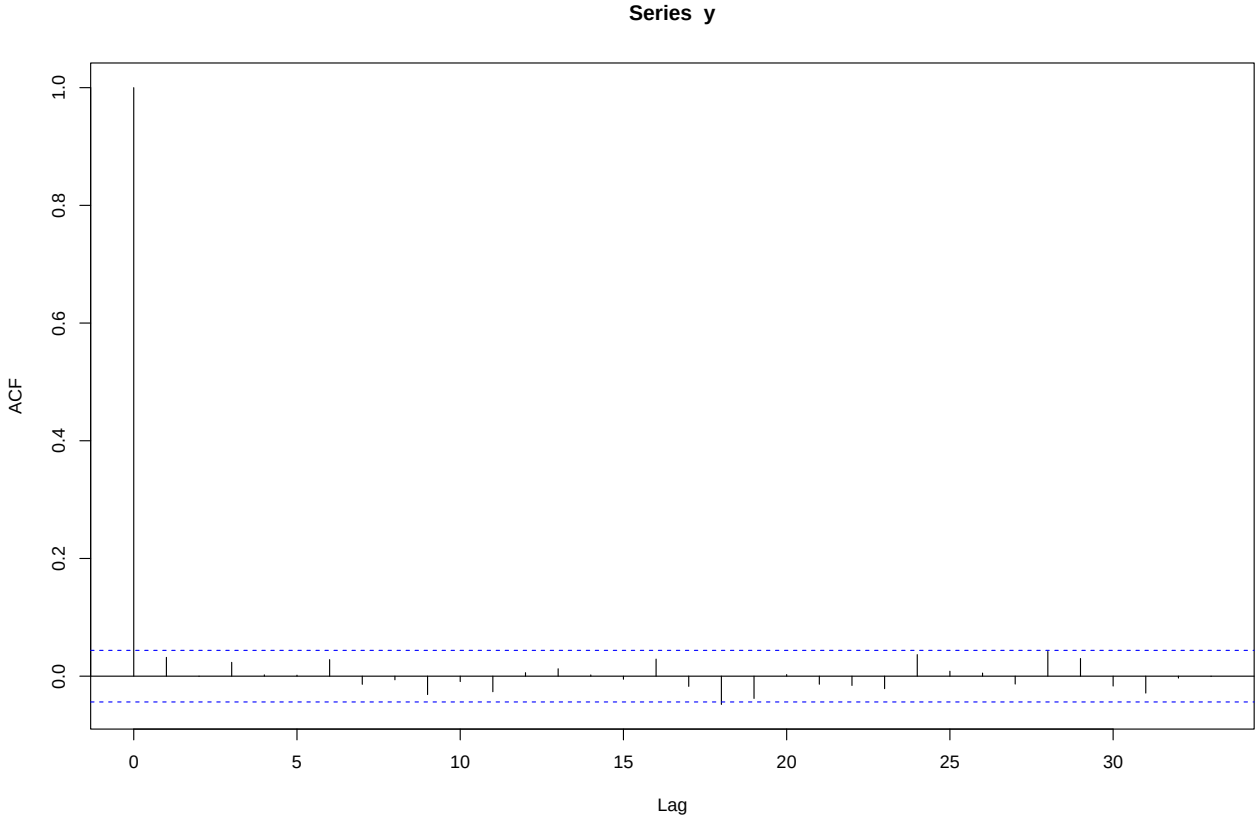

```
# Histogram of the largest data
```

```
nlargest <- 2000
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.
##   Unsuccessful convergence.
```

```r
gev_mixture_model$extremal_indexes
```

```
##           10           11           12           13           14           15
## 0.9633798283 1.0000000000 1.0000000000 1.0000000000 1.0000000000 1.0000000000
##           16           17           18           19           20           21
## 1.0000000000 0.9263642548 0.9813979955 0.9367100621 1.0000000000 0.9813979955
##           22           23           24           25           26           27
## 0.8368217356 0.9813979955 0.9813979955 0.8463210638 0.9606390346 0.9498712026
##           28           29           30           31           32           33
## 0.9367100621 0.9367100621 0.8980942875 0.9259440014 0.8723100811 0.9606390346
##           34           35           36           37           38           39
## 0.8723100811 0.8980942875 0.8858757883 0.9440029012 0.8723100811 0.8980942875
##           40
## 0.8980942875
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##        loc_star   scale_star     shape_star
## 10 2.4743152810 0.9557938021 -0.01012988691
## 11 2.3646577741 1.0074297647 -0.02480194476
## 12 2.2833859490 1.0875011227 -0.04752730423
## 13 2.1199629589 1.2225641874 -0.08099912265
## 14 2.1283482993 1.1646111259 -0.06372424212
## 15 2.2397372906 1.0802667310 -0.04164278549
## 16 2.2328499350 1.1433964465 -0.06405775333
## 17 2.0067504104 1.2361965584 -0.07706432056
## 18 1.7937013685 1.3491422537 -0.09739968931
## 19 1.5202854355 1.5160492015 -0.12402966252
## 20 1.6135559544 1.5278667081 -0.13132823606
## 21 1.7695345008 1.3286358431 -0.09141759474
## 22 1.6679822654 1.4572556589 -0.11805927067
## 23 1.4569247451 1.5307125907 -0.12250247532
## 24 1.2463547104 1.6868682296 -0.14773888631
## 25 1.5297152336 1.4807951307 -0.11627500051
## 26 1.4084136804 1.5550662827 -0.12551611990
## 27 1.6505940510 1.4328397029 -0.11120643090
## 28 1.1889284125 1.7044107485 -0.14947018060
```

```
## 29 1.2397040485 1.7089066562 -0.15219413305
## 30 1.7292479530 1.3767188645 -0.10113552612
## 31 1.3167110841 1.5911446540 -0.13156528165
## 32 1.1430339059 1.7159331181 -0.14630505136
## 33 1.3657358842 1.5743592736 -0.12735561353
## 34 0.9802154249 1.7592216822 -0.14787153152
## 35 1.8059951446 1.3164070326 -0.09167879195
## 36 2.2076734828 1.1566965467 -0.07029811509
## 37 0.8571865339 1.7828089630 -0.14753680445
## 38 0.9197965649 1.7509783908 -0.14246529421
## 39 1.9415549028 1.3416923231 -0.10518514342
## 40 1.4159397001 1.5927219515 -0.13591824724
```

```
gev_mixture_model$full_normalized_gev_parameters_object
```

```
##        loc_star   scale_star     shape_star
## 10 2.4386502429 0.9561550849 -0.01012988691
## 11 2.3646577741 1.0074297647 -0.02480194476
## 12 2.2833859490 1.0875011227 -0.04752730423
## 13 2.1199629589 1.2225641874 -0.08099912265
## 14 2.1283482993 1.1646111259 -0.06372424212
## 15 2.2397372906 1.0802667310 -0.04164278549
## 16 2.2328499350 1.1433964465 -0.06405775333
## 17 1.9119172860 1.2435048087 -0.07706432056
## 18 1.7683450775 1.3516119485 -0.09739968931
## 19 1.4207609111 1.5283931947 -0.12402966252
## 20 1.6135559544 1.5278667081 -0.13132823606
## 21 1.7445650180 1.3309184931 -0.09141759474
## 22 1.4056314501 1.4882286048 -0.11805927067
## 23 1.4281491690 1.5342376700 -0.12250247532
## 24 1.2146360766 1.6915543053 -0.14773888631
## 25 1.2802225580 1.5098048916 -0.11627500051
## 26 1.3458099376 1.5629240616 -0.12551611990
## 27 1.5766935849 1.4410579100 -0.11120643090
## 28 1.0769452296 1.7211488951 -0.14947018060
## 29 1.1274154604 1.7259963205 -0.15219413305
## 30 1.5804707639 1.3917655238 -0.10113552612
## 31 1.1936642557 1.6073333446 -0.13156528165
## 32 0.9062614549 1.7505741237 -0.14630505136
## 33 1.3023531028 1.5824314267 -0.12735561353
## 34 0.7374437542 1.7951207009 -0.14787153152
## 35 1.6638080476 1.3294425739 -0.09167879195
## 36 2.0669079804 1.1665920962 -0.07029811509
## 37 0.7540123441 1.7980309533 -0.14753680445
## 38 0.6782519716 1.7853901124 -0.14246529421
## 39 1.7965312935 1.3569466523 -0.10518514342
## 40 1.2434970969 1.6161600479 -0.13591824724
```

```
gev_mixture_model$automatic_weights_pw_shape
```

```
##              10              11              12              13              14
## 2.879635301e-01 2.261153759e-01 1.303177903e-01 9.726153732e-03 6.340479264e-02
##              15              16              17              18              19
## 1.551238945e-01 6.236695042e-02 2.195259104e-02 0.000000000e+00 1.973064011e-06
##              20              21              22              23              24
```

```
## 2.774467335e-06 0.000000000e+00 1.317494326e-06 1.805374381e-06 4.576395874e-06
##              25              26              27              28              29
## 1.121577758e-06 2.136281397e-06 5.650293755e-07 4.766497640e-06 5.065592199e-06
##              30              31              32              33              34
## 0.000000000e+00 2.800496988e-06 4.418956747e-06 2.338262559e-06 4.590961526e-06
##              35              36              37              38              39
## 0.000000000e+00 4.297684081e-02 4.554210311e-06 3.997343149e-06 0.000000000e+00
##              40
## 3.278463611e-06
```

```
gev_mixture_model$automatic_weights_pw_scale
```

```
##              10              11              12              13              14
## 3.313952613e-01 2.569171025e-01 1.413392470e-01 2.909562647e-05 3.020161784e-02
##              15              16              17              18              19
## 1.517652488e-01 6.077970623e-02 2.620958713e-05 1.131005082e-05 0.000000000e+00
##              20              21              22              23              24
## 0.000000000e+00 1.416208873e-05 0.000000000e+00 3.575751437e-08 1.287216002e-05
##              25              26              27              28              29
## 0.000000000e+00 2.376613149e-06 0.000000000e+00 1.528672575e-05 1.568220853e-05
##              30              31              32              33              34
## 5.775920801e-06 6.000340139e-06 1.768739335e-05 3.968404902e-06 2.132159637e-05
##              35              36              37              38              39
## 1.436550467e-05 2.734628421e-02 2.155901386e-05 2.052776532e-05 1.057479916e-05
##              40
## 6.720565009e-06
```

```
gev_mixture_model$automatic_weights_pw_loc
```

```
##              10              11              12              13              14
## 3.329259586e-01 2.498632622e-01 1.592126202e-01 1.747402446e-02 2.397578012e-02
##              15              16              17              18              19
## 1.109469469e-01 1.050479022e-01 2.453898857e-05 1.382407001e-05 8.220385605e-06
##              20              21              22              23              24
## 2.271544275e-06 1.204930081e-05 9.060179247e-06 7.810279624e-06 2.039886407e-05
##              25              26              27              28              29
## 1.602123413e-05 1.238069720e-05 0.000000000e+00 3.351669445e-05 2.870856571e-05
##              30              31              32              33              34
## 0.000000000e+00 2.239693893e-05 4.977578387e-05 1.479284519e-05 6.585487960e-05
##              35              36              37              38              39
## 6.022111451e-06 3.610560032e-05 6.427686667e-05 7.149210905e-05 1.592766200e-05
##              40
## 1.805972985e-05
```

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

```
##                      loc_star  scale_star     shape_star
## automatic_weights 2.349989965 1.029914959 -0.03472214012
```

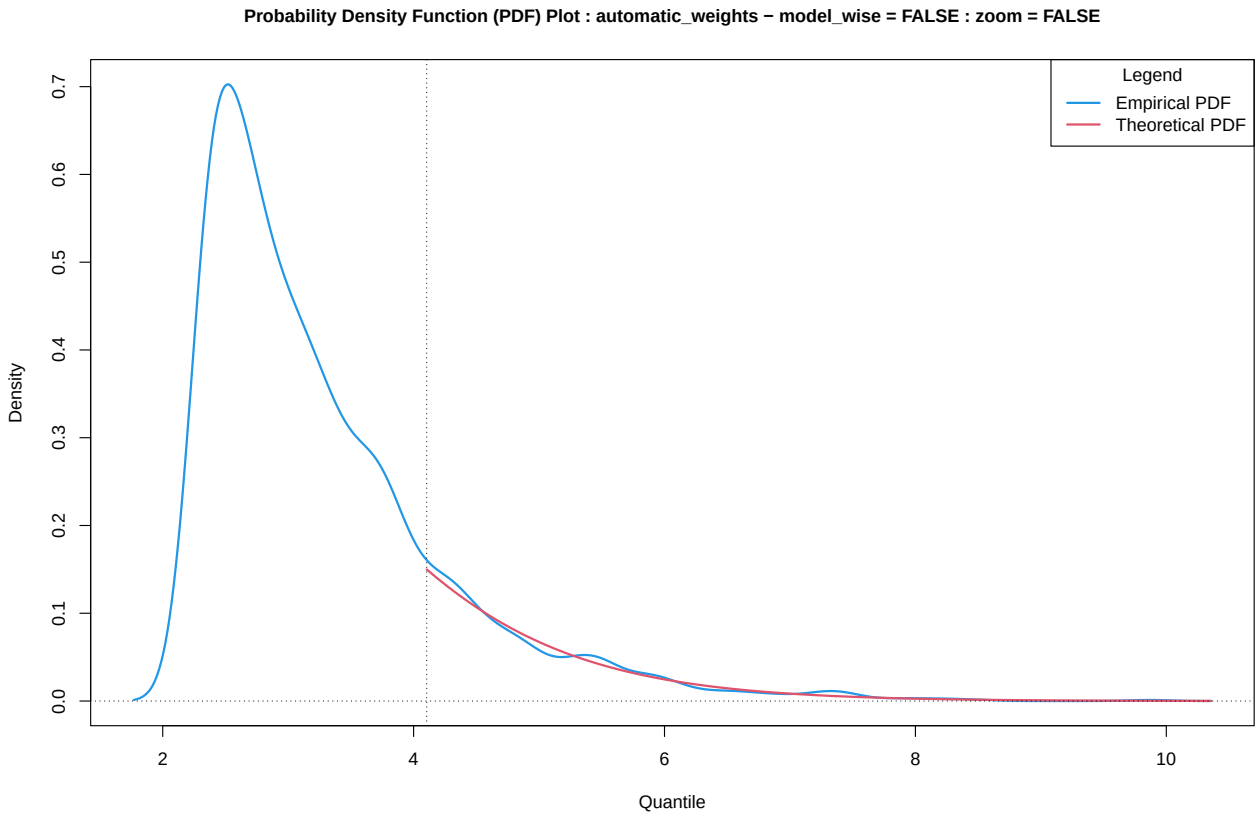
```
gev_mixture_model$automatic_weights_mw
```

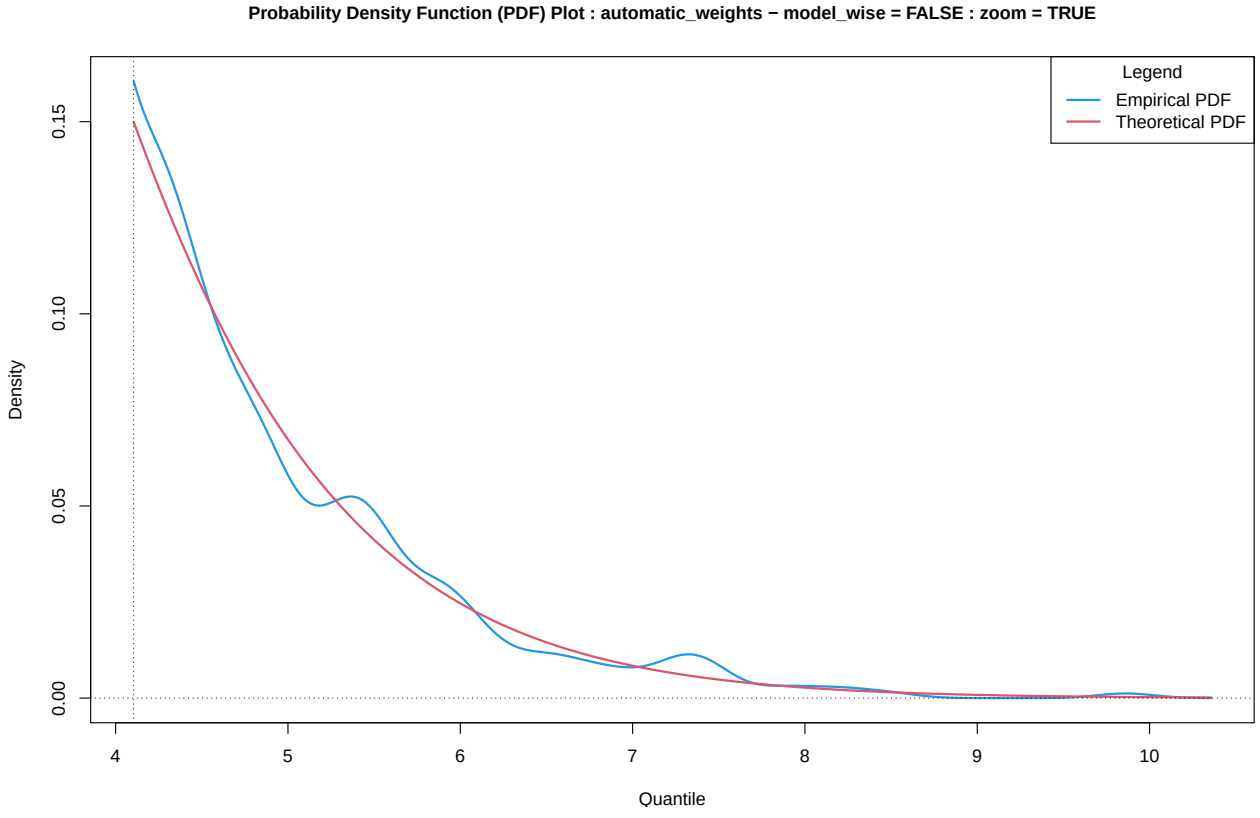
```
##            10            11            12            13            14
## 0.96184120774 0.00000000000 0.00000000000 0.00000000000 0.00000000000
##            15            16            17            18            19
## 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000
##            20            21            22            23            24
## 0.02308228199 0.00000000000 0.00000000000 0.00000000000 0.01507651027
```

```
##            25            26            27            28            29
## 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000
##            30            31            32            33            34
## 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000
##            35            36            37            38            39
## 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000
##            40
## 0.00000000000
```

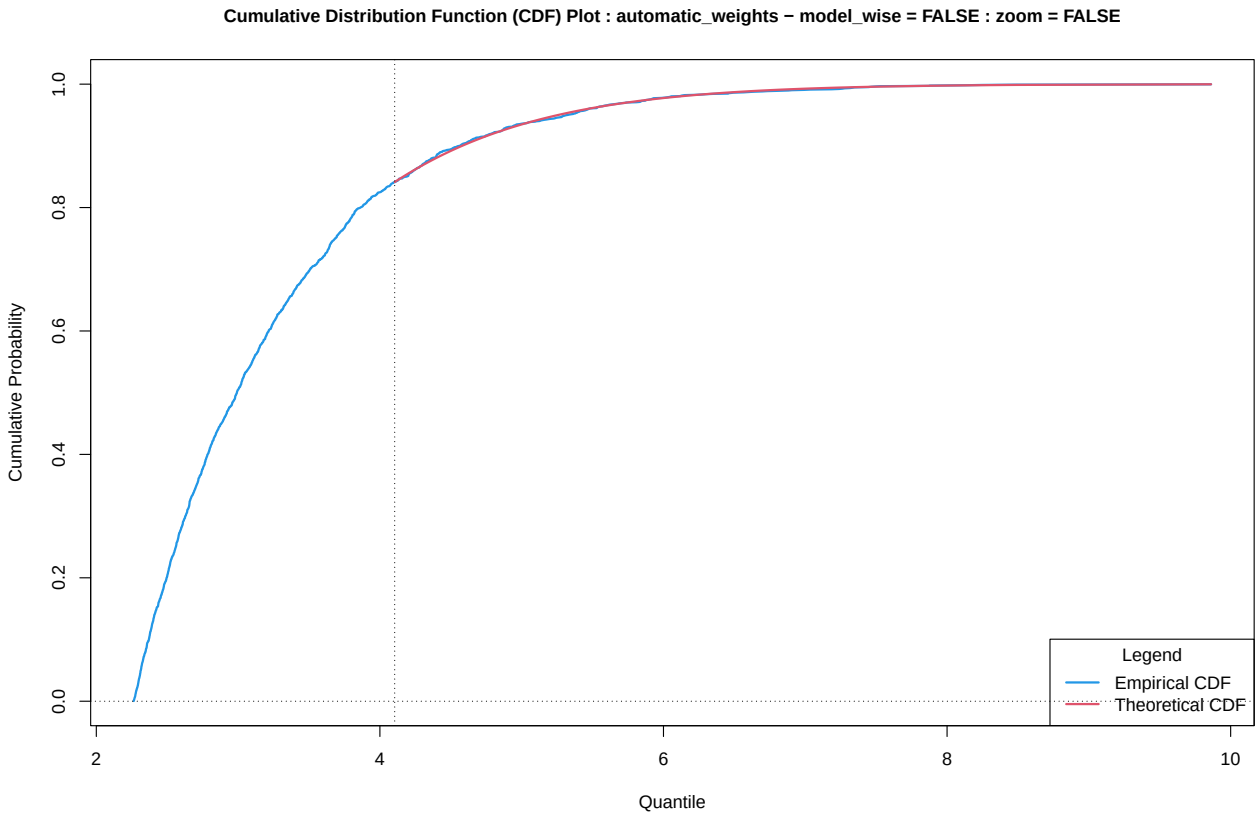
```r
# Model diagnostics

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

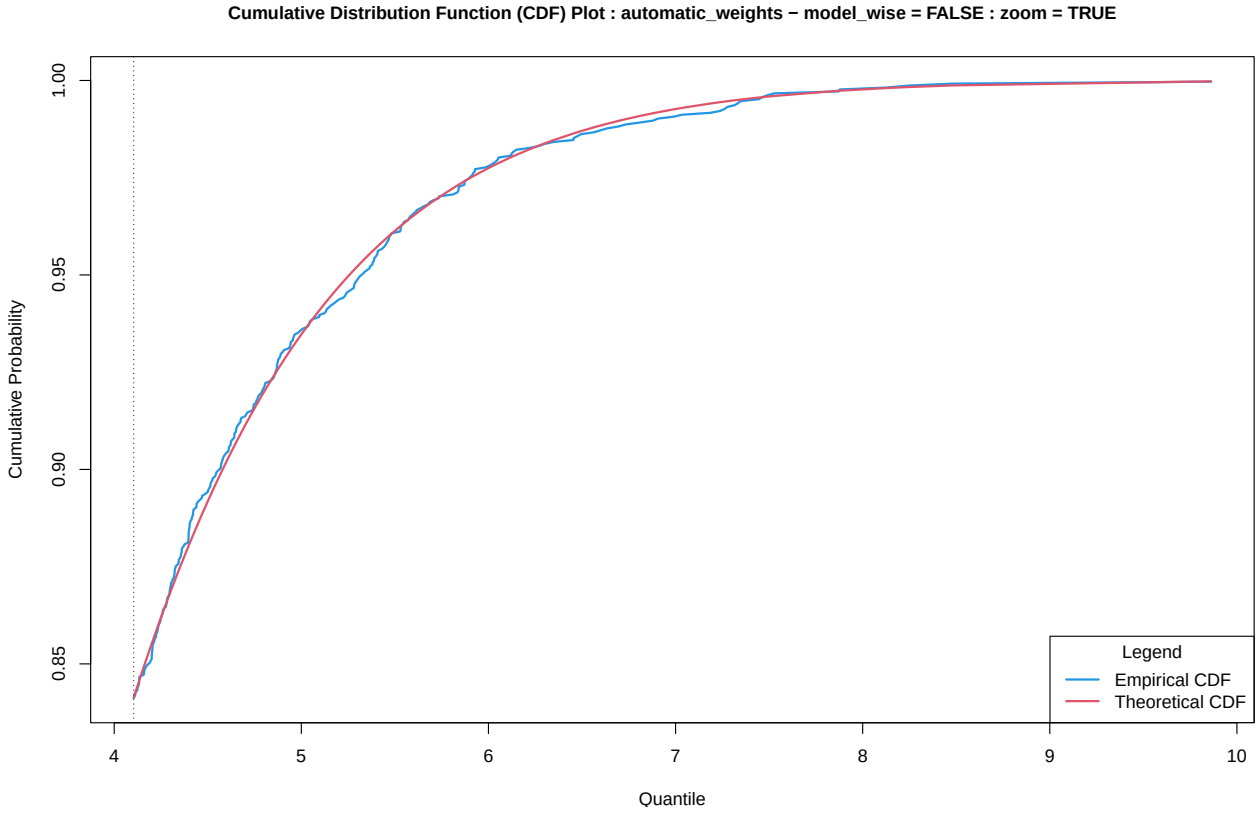
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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
2
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
4
5
6
7
8
9
10
Quantile

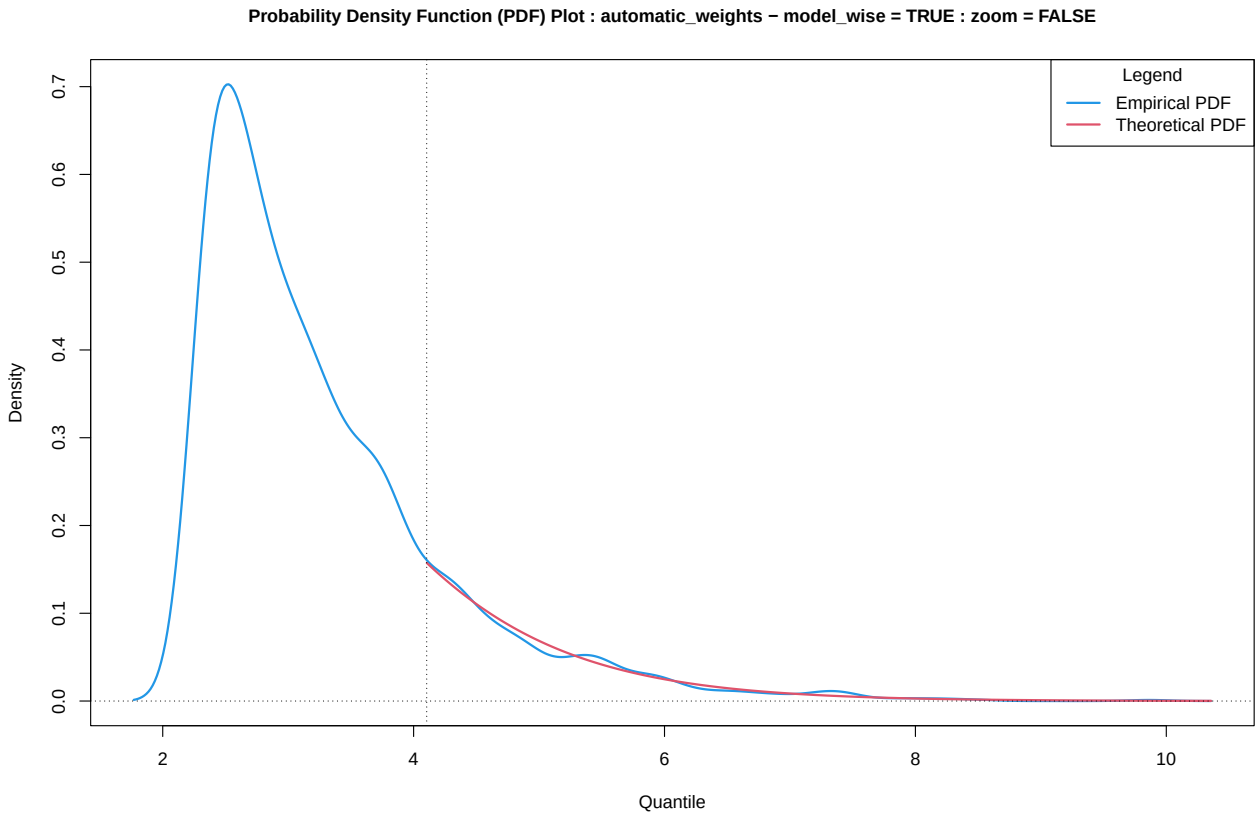
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Cumulative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

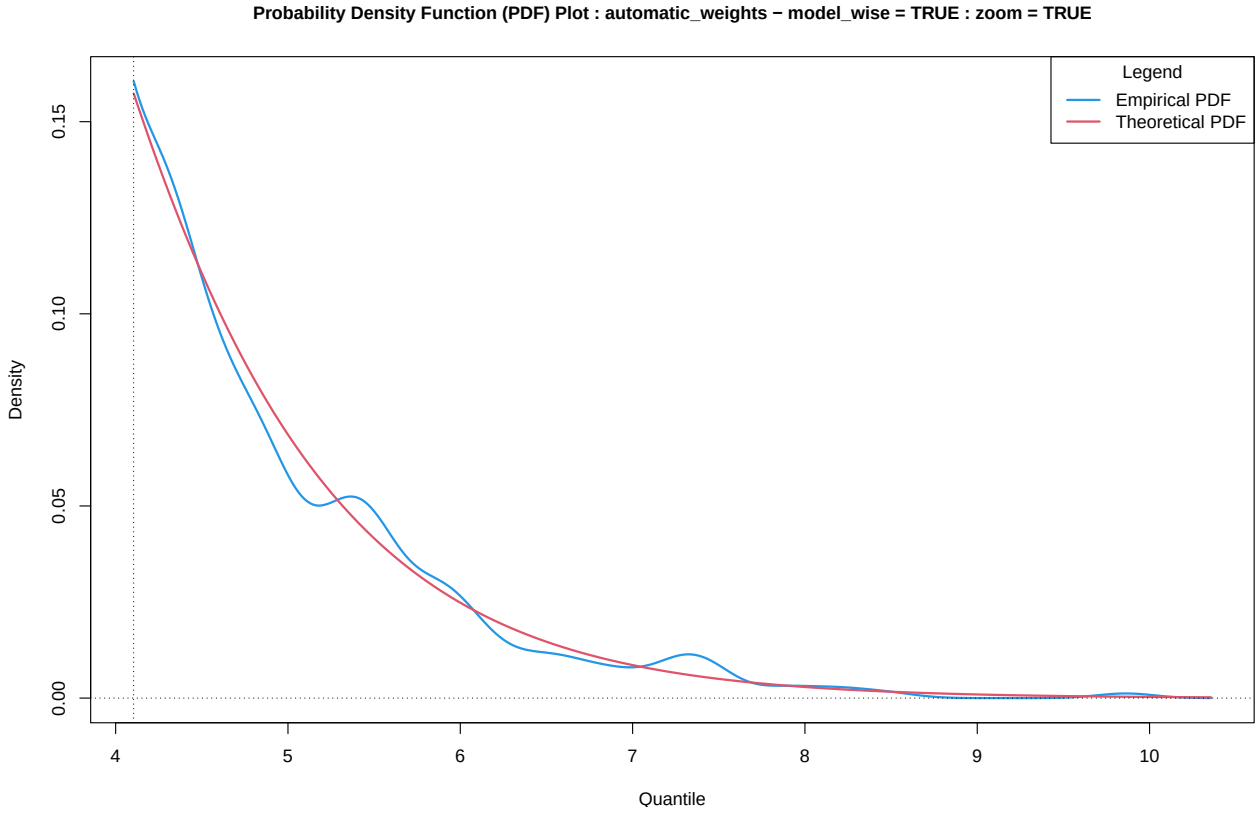

Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = FALSE : zoom = FALSE
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
Quantile
Legend
Empirical CDF
Theoretical CDF


Cumulative Distribution Function (CDF) Plot : automatic_weights − model_wise = FALSE : zoom = TRUE
Cumulative Probability
0.85
0.90
0.95
1.00
4
5
6
7
8
9
10
Quantile
Legend
Empirical CDF
Theoretical CDF

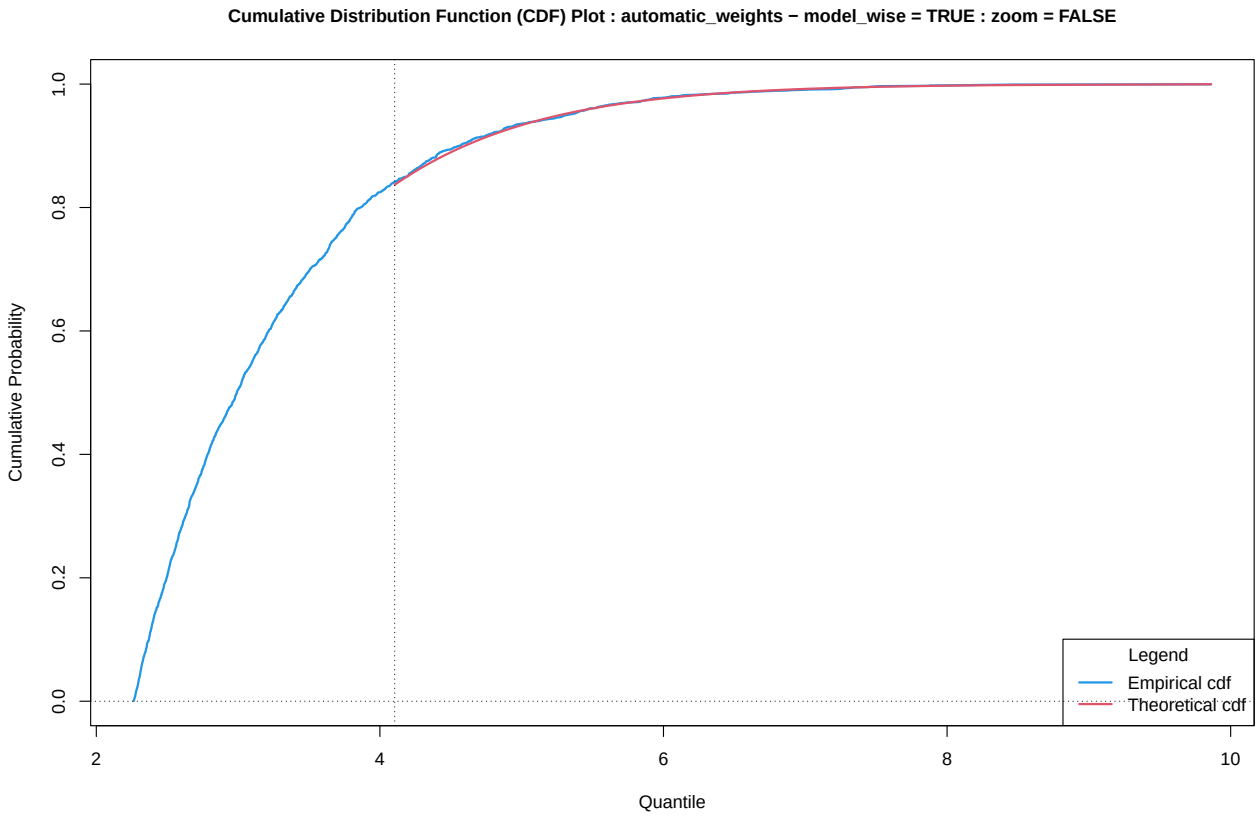
```r
## GEV mixture model with respect to distribution functions
```

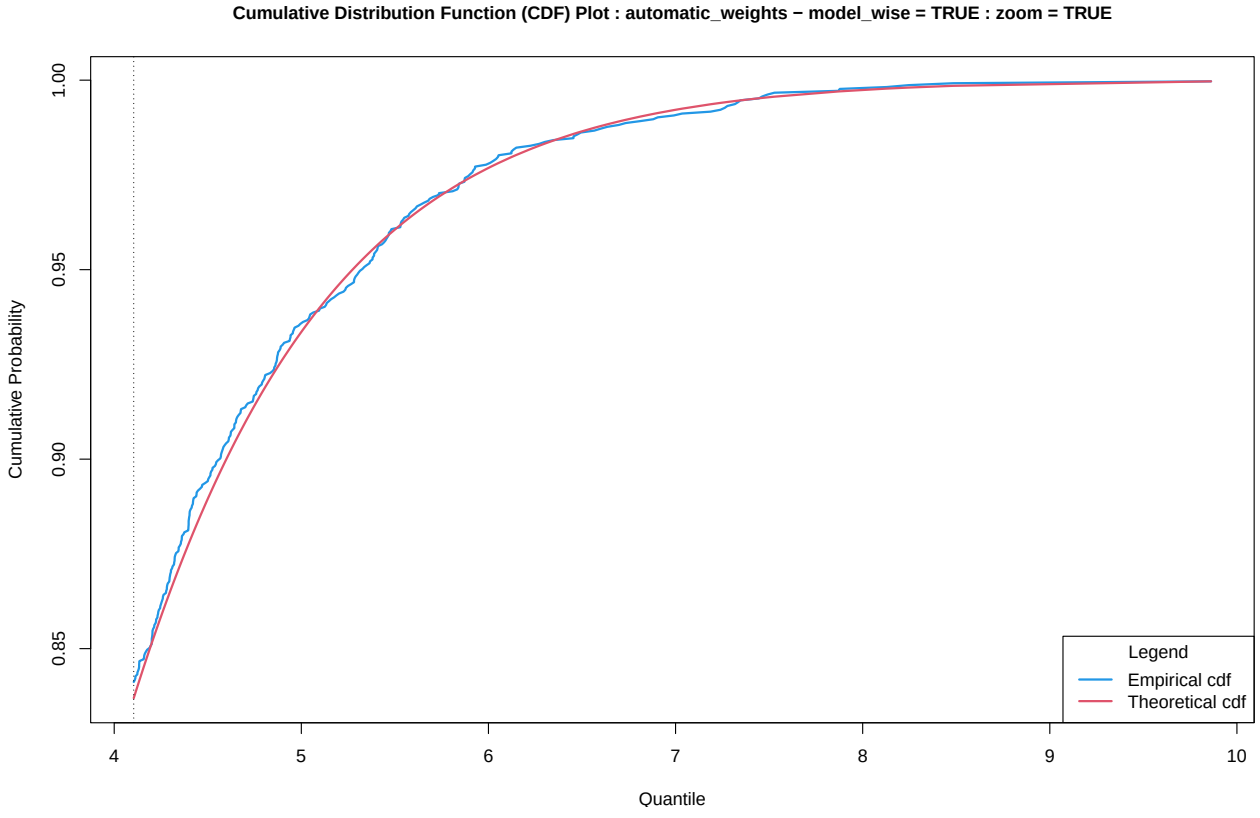
```r
par(mfrow = c(2, 1))
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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
2
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
4
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
2
4
6
8
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
4
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 21.52073507
```

```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 27.12585499
```

```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
```

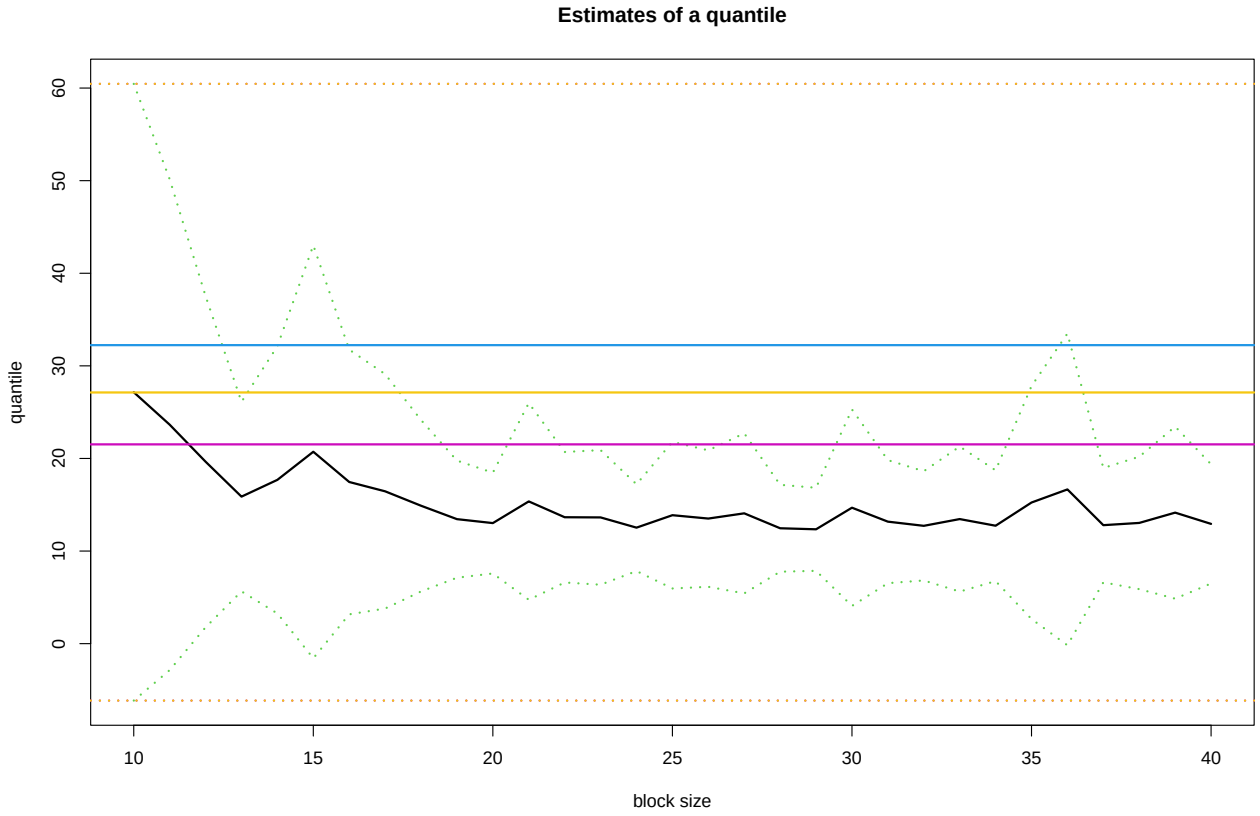
```
##            lower    estimate       upper
## 10  -6.148672532 27.15412375 60.45692002
## 11  -2.858141606 23.65043581 50.15901322
## 12    1.76278456 19.64889176 37.53499897
## 13   5.645494235 15.87752617  26.1095581
## 14   3.259943963 17.69107276 32.12220157
## 15   -1.57715302 20.72278535 43.02272373
## 16   3.165006185 17.45884715 31.75268812
## 17   3.793205164 16.45051108   29.107817
## 18   5.637632199 14.89488887 24.15214555
## 19   7.121331742 13.44516173 19.76899172
## 20   7.577175804 13.01923499 18.46129417
## 21   4.741279309 15.36145884 25.98163836
## 22   6.603424581  13.6511054 20.69878621
## 23   6.365834664  13.6329714 20.90010813
## 24    7.83857729 12.52717555 17.21577382
## 25   5.958715538 13.87286583 21.78701611
## 26   6.139963413   13.508494  20.8770246
## 27   5.429346315 14.07337321 22.71740011
## 28   7.770798052 12.46195456 17.15311106
## 29   7.864320352 12.35019319 16.83606603
## 30   4.074042657 14.68241414 25.29078562
## 31   6.531665725 13.17504278 19.81841984
## 32   6.825719759 12.72450934 18.62329892
## 33   5.621349214 13.45446191  21.2875746
## 34   6.753938707 12.73490803 18.71587736
## 35   2.661084506 15.24169774 27.82231097
## 36 -0.1770961519 16.65551446 33.48812507
## 37   6.621468506  12.7950542 18.96863989
## 38   5.885896563 13.03756613  20.1892357
## 39   4.858584674 14.14970225 23.44081982
## 40   6.496851997 12.93376189 19.37067178
```

```
## 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
```