Modeling extreme values with a GEV mixture probability distributions

Application to a wind speed data

Pascal Alain Dkengne Sielenou

2023-09-27

```
# library(xfun)
```

```
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_several_standardized_block_maxima_mean.R"))
xfun::in_dir(dir = path, expr = source("./src/estimate_gev_mixture_model_quantile.R"))
```

```
library(readr)
```

```
Gnss_imar <- xfun::in_dir(dir = path, expr = read_csv("./applications/Gnss_imar.csv"))
```

```
## Rows: 20002 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Gnss_map_matching <- xfun::in_dir(dir = path, expr = read_csv("./applications/Gnss_map_matching.csv"))
```

```
## Rows: 20001 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Gnss_standard <- xfun::in_dir(dir = path, expr = read_csv("./applications/Gnss_standard.csv"))
```

```
## Rows: 20001 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
timestamp_position <- sapply(Gnss_standard$timestamp,
                             function(ts)
                               which.min(abs(ts - Gnss_imar$timestamp)))
```

```
latitude_Gnss_standard_errors <- Gnss_imar$latitude[timestamp_position] - Gnss_standard$latitude
```

```
coefficient <- 10^(4)
x <- coefficient*latitude_Gnss_standard_errors
```

```
n <- length(x)
n
```

```
## [1] 20001
```

```
nlargest <- 1000
```

```
gev_mixture_model <- estimate_gev_mixture_model_parameters(x,
                                                           nsloc = NULL,
                                                           std.err = FALSE,
                                                           block_sizes = NULL,
                                                           minimum_nblocks = 50,
                                                           threshold = 0.5,
                                                           nlargest = nlargest,
                                                           confidence_level = 0.95,
                                                           log_mv = TRUE,
                                                           log_pw = TRUE,
                                                           trace = FALSE)
```

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

```
names(gev_mixture_model)
```

```
##  [1] "data"
##  [2] "data_largest"
##  [3] "block_sizes"
##  [4] "equivalent_block_sizes"
##  [5] "rejected_block_sizes"
##  [6] "block_maxima_indexes_object"
##  [7] "gev_models_object"
##  [8] "extremal_indexes"
##  [9] "normalized_gev_parameters_object"
## [10] "weighted_normalized_gev_parameters_object"
## [11] "identic_weights_mw"
## [12] "pessimistic_weights_mw"
## [13] "pessimistic_weights_pw_shape"
## [14] "pessimistic_weights_pw_scale"
## [15] "pessimistic_weights_pw_loc"
## [16] "automatic_weights_mw"
## [17] "automatic_weights_mw_statistics"
## [18] "automatic_weights_pw_shape"
## [19] "automatic_weights_pw_scale"
## [20] "automatic_weights_pw_loc"
## [21] "automatic_weights_pw_statistics"
```

```
gev_mixture_model$block_sizes
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
gev_mixture_model$normalized_gev_parameters_object
```

```
##             loc_star         scale_star        shape_star
## 2  0.557482439042738 0.0416443473867565 0.201823171461190
## 3  0.541947295716754 0.0387152900016861 0.198939083599126
## 4  0.533985653749949 0.0360125945405177 0.202912086889310
## 5  0.525621222663993 0.0356131226056433 0.191244488969175
## 6  0.520437644631545 0.0340356630762777 0.194463876941935
## 7  0.517506552452747 0.0328160198597940 0.192475600620196
## 8  0.515642321316783 0.0304155330997995 0.215288517415016
## 9  0.506227880699435 0.0332998729217927 0.179214405675663
## 10 0.506653371680722 0.0321530651984181 0.182809618416344
## 11 0.502423131150597 0.0317279393149894 0.183924608459844
## 12 0.502693310361283 0.0302289188598985 0.193769045902675
## 13 0.497065691194356 0.0320515493623219 0.170445690251289
## 14 0.499948797708687 0.0303000646723038 0.179933348484317
## 15 0.499721525303604 0.0288276146658321 0.190418482334089
## 16 0.490678546938413 0.0321720083296132 0.156867459743239
## 17 0.499798285812077 0.0268232113025282 0.206075687384584
## 18 0.482204662456251 0.0344930573898525 0.139507638629199
## 19 0.492642626919278 0.0292463557714775 0.174917571935242
## 20 0.519516368609514 0.0182009254187604 0.290125927706145
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                             loc_star         scale_star        shape_star
## identic_weights     0.511168280442565 0.0320409028304349 0.191850332148346
## pessimistic_weights 0.511508617897970 0.0320631504020369 0.192705143511690
## automatic_weights   0.557482439041978 0.0416443473867420 0.290125927706167
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 122.997119229964
##
## $gradient_value
## [1] 3.97903932025656e-13
##
## $function_reduction
## [1] 257.043000236854
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 117.985881957549
##
```

```
## $gradient_value
## [1] 7.73070496506989e-12
##
## $function_reduction
## [1] 279.86749265168
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                2                3                4                5
## 0.999999999999602 0.000000000000000 0.000000000000000 0.000000000000000
##                6                7                8                9
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##               10               11               12               13
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##               14               15               16               17
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##               18               19               20
## 0.000000000000000 0.000000000000000 0.000000000000000
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                 2                 3                 4                 5
## 0.0531365606125187 0.0529835308846892 0.0531944533475926 0.0525774085695252
##                 6                 7                 8                 9
## 0.0527469484071375 0.0526421770902554 0.0538569017210245 0.0519486873320165
##                10                11                12                13
## 0.0521357900500784 0.0521939533566223 0.0527103109200636 0.0514951554282483
##                14                15                16                17
## 0.0519860488959975 0.0525339972126938 0.0508006679633300 0.0533630058223172
##                18                19                20
## 0.0499263880823998 0.0517259513319916 0.0580420629714982
```

```
gev_mixture_model$pessimistic_weights_pw_scale
```

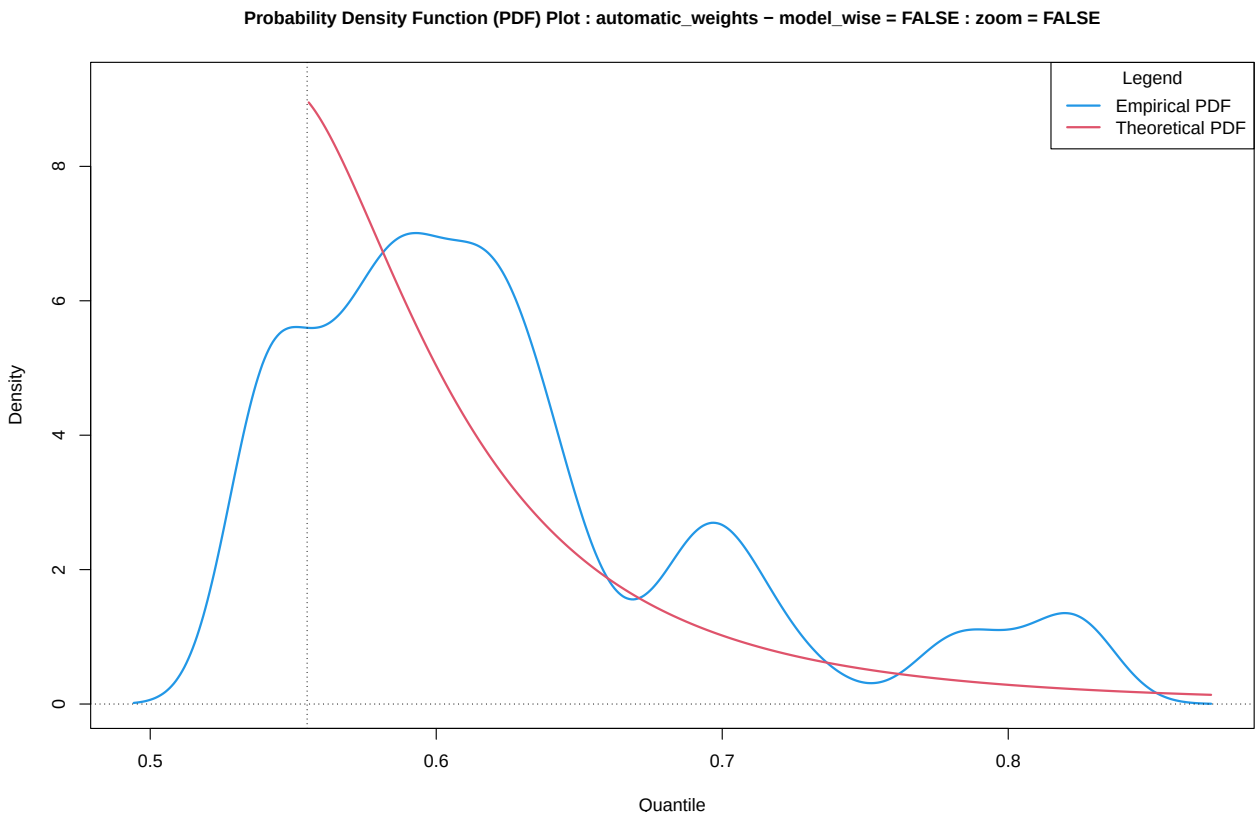
```
##                 2                 3                 4                 5
## 0.0531388667225690 0.0529834476593550 0.0528404428720927 0.0528193388136591
##                 6                 7                 8                 9
## 0.0527360841270262 0.0526718041270579 0.0525455177936036 0.0526972957073581
##                10                11                12                13
## 0.0526368966812943 0.0526145241300040 0.0525357129666290 0.0526315534739328
##                14                15                16                17
## 0.0525394507955722 0.0524621460085923 0.0526378937983780 0.0523570960231297
##                18                19                20
## 0.0527602108293137 0.0524841186657228 0.0519075988047097
```

```
gev_mixture_model$pessimistic_weights_pw_loc
```

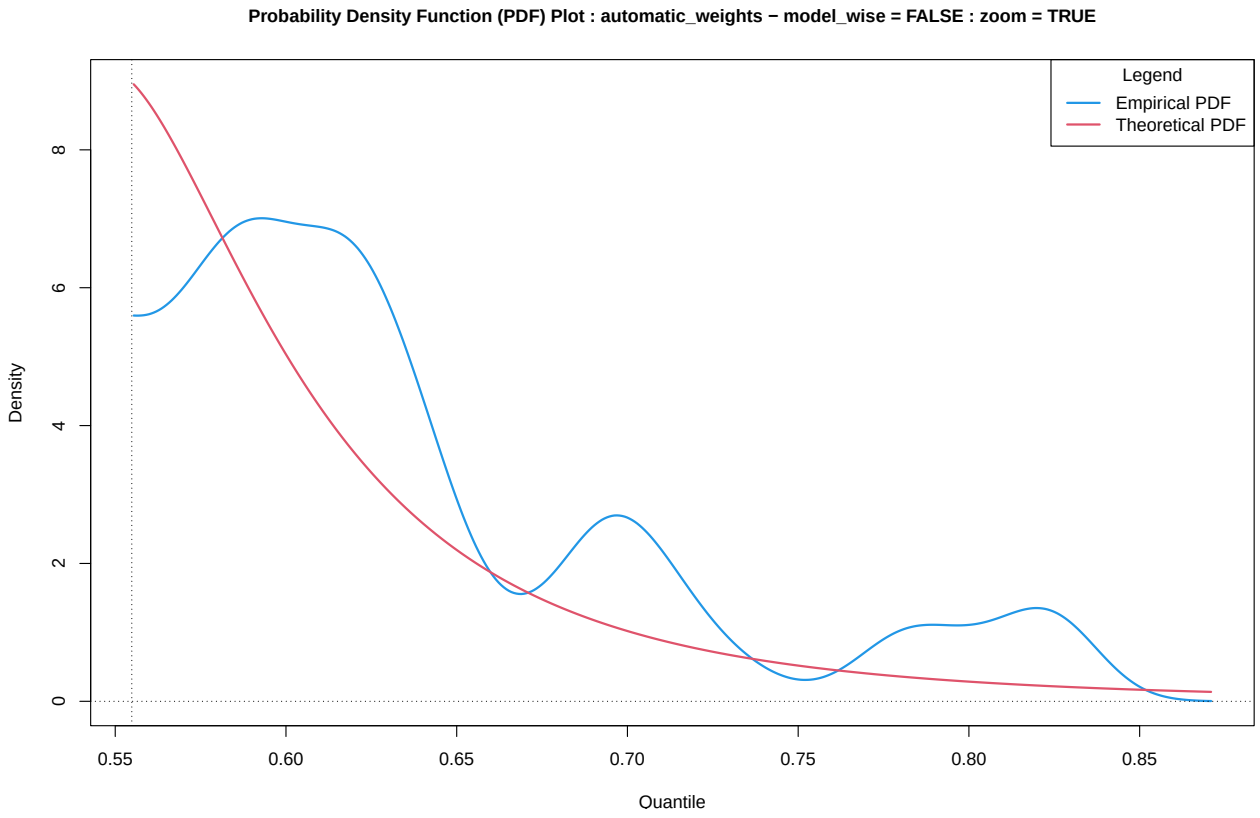
```
##                 2                 3                 4                 5
```

```
## 0.0551171388008947 0.0542675028456489 0.0538371598093519 0.0533887206798959
##                  6                  7                  8                  9
## 0.0531126921060867 0.0529572418404818 0.0528586092668906 0.0523633101690650
##                 10                 11                 12                 13
## 0.0523855950259589 0.0521644594168607 0.0521785550734228 0.0518857387389901
##                 14                 15                 16                 17
## 0.0520355467027367 0.0520237218026754 0.0515553931463687 0.0520277153232840
##                 18                 19                 20
## 0.0511203644948455 0.0516567515673924 0.0530637831891493
```

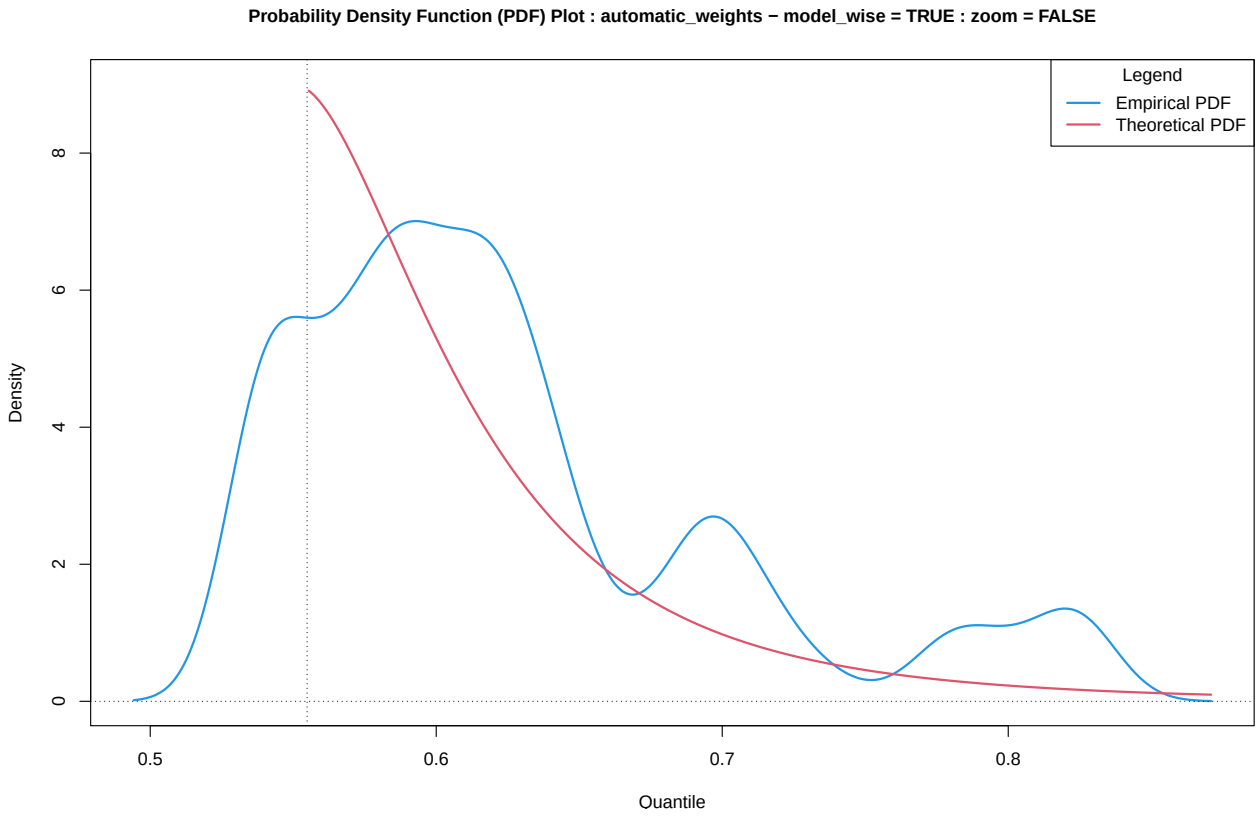
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



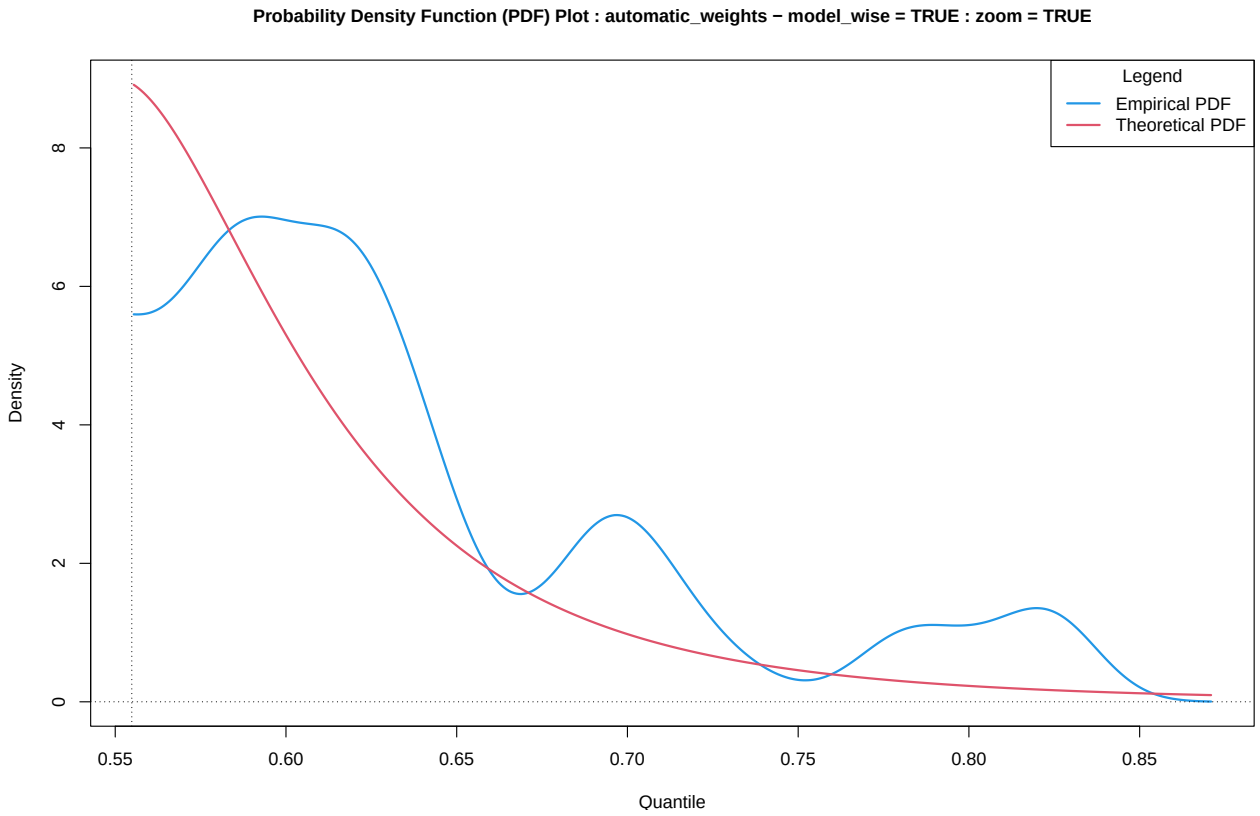

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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




```
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")
```




```
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")
```

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

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

results_mw
```

```
##   lower         estimate upper
## 1    NA 75.7742079559771    NA
```

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

```
results_pw
```

```
##   lower         estimate upper
## 1    NA 694.208357520279    NA
```

```
quantile(x = x, probs = 1 - alpha)
```

```
##              100%
## 0.831998570021065
```

```
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
## 2  -83.6967982858339 75.9302218669204 235.557242019675
## 3  -100.579793408004 65.6887986196679  231.95739064734
## 4  -127.716722320508 68.9399156621235 265.596553644755
## 5  -104.974522719039 50.3984871732308 205.771497065501
## 6  -127.672577293595 53.0949046291485 233.862386551892
## 7  -121.607409854917 47.9378491230093 217.483108100936
## 8  -232.129086918294 77.2069048989667 386.542896716227
## 9  -104.173695627453 35.4427172702819 175.059130168017
## 10 -110.795645967518 37.2552375217694 185.306121011057
## 11 -122.947768556307 37.7805338773116  198.50883631093
## 12 -165.872971031718 45.5787088767365 257.030388785191
## 13 -97.3609248190241 27.7948280996157 152.950581018255
## 14  -116.03465187045 32.7153419792998 181.465335829049
## 15 -153.712047761546 40.0273580175832 233.766763796712
## 16 -74.0003680900996 20.3636558121958 114.727679714491
## 17 -240.315299252002 54.3128337443456 348.940966740693
## 18  -56.179617117764 14.9305512851581 86.0407196880802
## 19 -114.944704487537 28.7640369824917  172.47277845252
## 20 -1674.57733173974 303.806502822202 2282.19033738414
```

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

```
## [1] -1674.57733173974  2282.19033738414
```

```
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
```

```
##                lower         estimate            upper
## 2 -83.6967982858339 75.9302218669204 235.557242019675
```

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

```
## [1] -83.6967982858339 235.5572420196746
```

```r
matplot(x = rownames(est_rl_pw),
        y = est_rl_pw,
        xlab = "block size",
        ylab = "quantile",
        main = "Estimates of a quantile",
        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 = results_mw[2], col = 7, lwd = 2)
abline(h = results_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)
```