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)))
```

```
longitude_Gnss_standard_errors <- Gnss_imar$longitude[timestamp_position] - Gnss_standard$longitude
```

```
coefficient <- 10^(3)
x <- coefficient*longitude_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 = 0.05,
                                                           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  0.1035812743434626 0.01167399322549353 -2.67446366448166e-01
## 3  0.0988679045527408 0.01315341019546154 -2.77027773608026e-01
## 4  0.0955208636584127 0.01389491041163969 -2.63030005846708e-01
## 5  0.0919793883212749 0.01509511827307151 -2.70057196374030e-01
## 6  0.0893194749613544 0.01584021988784539 -2.70439559901730e-01
## 7  0.0870994318494083 0.01625790550580849 -2.62488819989725e-01
## 8  0.0923819325351427 0.00885743294338813 -1.63755302225507e-05
## 9  0.0815969632174938 0.01854685779970587 -2.82105973492069e-01
## 10 0.0811561492098496 0.01825312229832896 -2.74599714036745e-01
## 11 0.0791871241066561 0.01886775549696216 -2.74614115293918e-01
## 12 0.0774052084427616 0.01946936393561964 -2.77100658548703e-01
## 13 0.0766487189986225 0.01936497101016049 -2.69200815550277e-01
## 14 0.0753745526374527 0.01927475881692605 -2.56923445315080e-01
## 15 0.0739761279772026 0.01986960286795838 -2.62891217167125e-01
## 16 0.0723501436279786 0.02046967234186035 -2.65905873011640e-01
## 17 0.0729707067485712 0.01968676256694617 -2.55002037872495e-01
## 18 0.0704553241083254 0.02098553709239033 -2.66467903241546e-01
## 19 0.0739213417603818 0.01816244541858470 -2.28155218761755e-01
## 20 0.0712841080557923 0.01957178448424240 -2.42783834609083e-01
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                              loc_star         scale_star         shape_star
## identic_weights     0.0823724599533097 0.0172260855038102 -0.250855626557844
## pessimistic_weights 0.0824713096868426 0.0172365574009869 -0.246767262205383
## automatic_weights   0.1035812743434658 0.0197654719639519 -0.282105973492069
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 176.786220525751
##
## $gradient_value
## [1] 4.54747350886464e-13
##
## $function_reduction
## [1] 453.843753673286
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 146.698048224414
##
```

```
## $gradient_value
## [1] 7.07550976871557e-06
##
## $function_reduction
## [1] 538.895264294955
##
## $number_iterations
## [1] 1576
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                    2                    3                    4
## 1.00000000000000e+00 1.13686837721616e-13 0.00000000000000e+00
##                    5                    6                    7
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                    8                    9                   10
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   11                   12                   13
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   14                   15                   16
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   17                   18                   19
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   20
## 0.00000000000000e+00
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                  2                  3                  4                  5
## 0.0516638014297418 0.0511711534123276 0.0518924719810340 0.0515290919591635
##                  6                  7                  8                  9
## 0.0515093928801275 0.0519205630535140 0.0675040341604618 0.0509119547553179
##                 10                 11                 12                 13
## 0.0512955509823840 0.0512948122672817 0.0511674239417596 0.0515732393861896
##                 14                 15                 16                 17
## 0.0522103260120293 0.0518996745685078 0.0517434505104419 0.0523107397579757
##                 18                 19                 20
## 0.0517143772978329 0.0537341380925761 0.0529538035513330
```

```
gev_mixture_model$pessimistic_weights_pw_scale
```

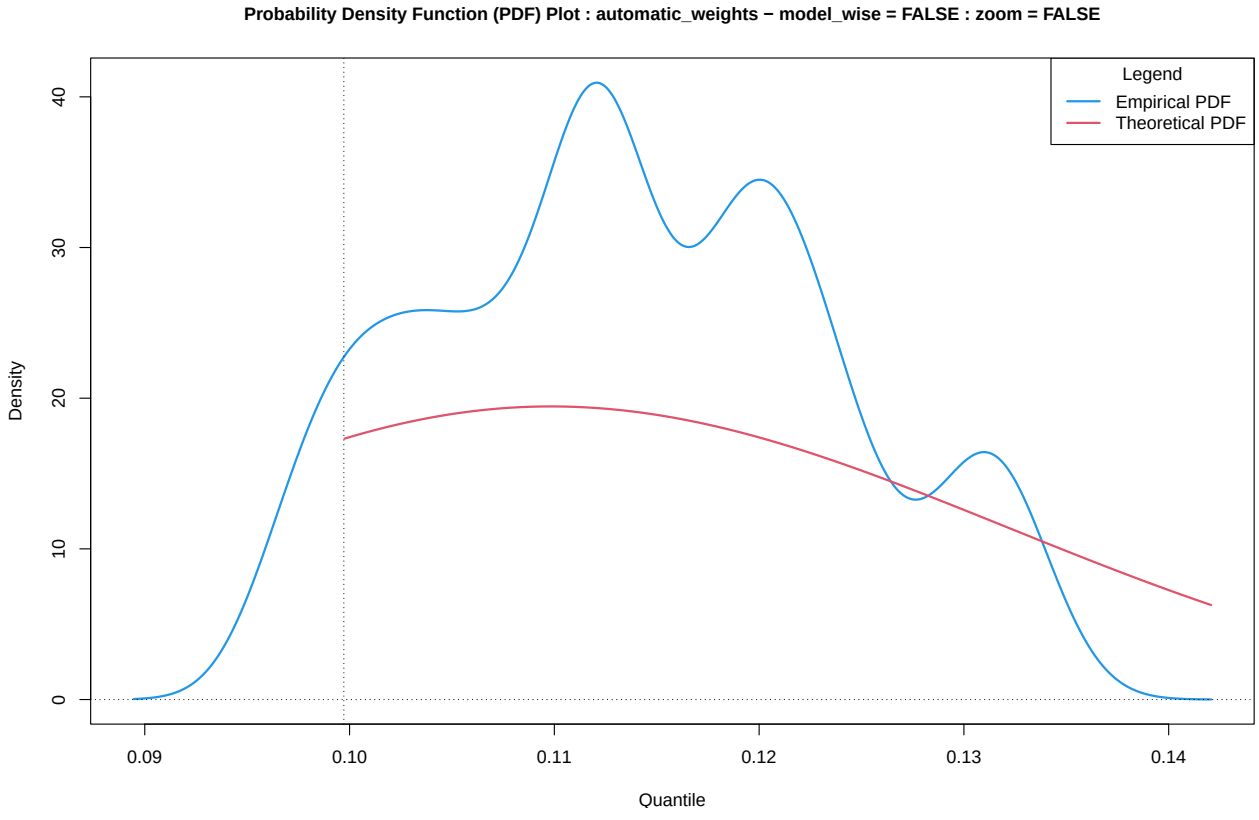
```
##                  2                  3                  4                  5
## 0.0523398990573769 0.0524173888980081 0.0524562708169016 0.0525192670422377
##                  6                  7                  8                  9
## 0.0525584138152684 0.0525803712941643 0.0521926879882666 0.0527008631030455
##                 10                 11                 12                 13
## 0.0526853852619059 0.0527177774023934 0.0527495024042035 0.0527439960167499
##                 14                 15                 16                 17
## 0.0527392380798045 0.0527706190342940 0.0528022945747073 0.0527609713204475
##                 18                 19                 20
```

```
## 0.0528295404442131 0.0526806081321531 0.0527549053138590
```

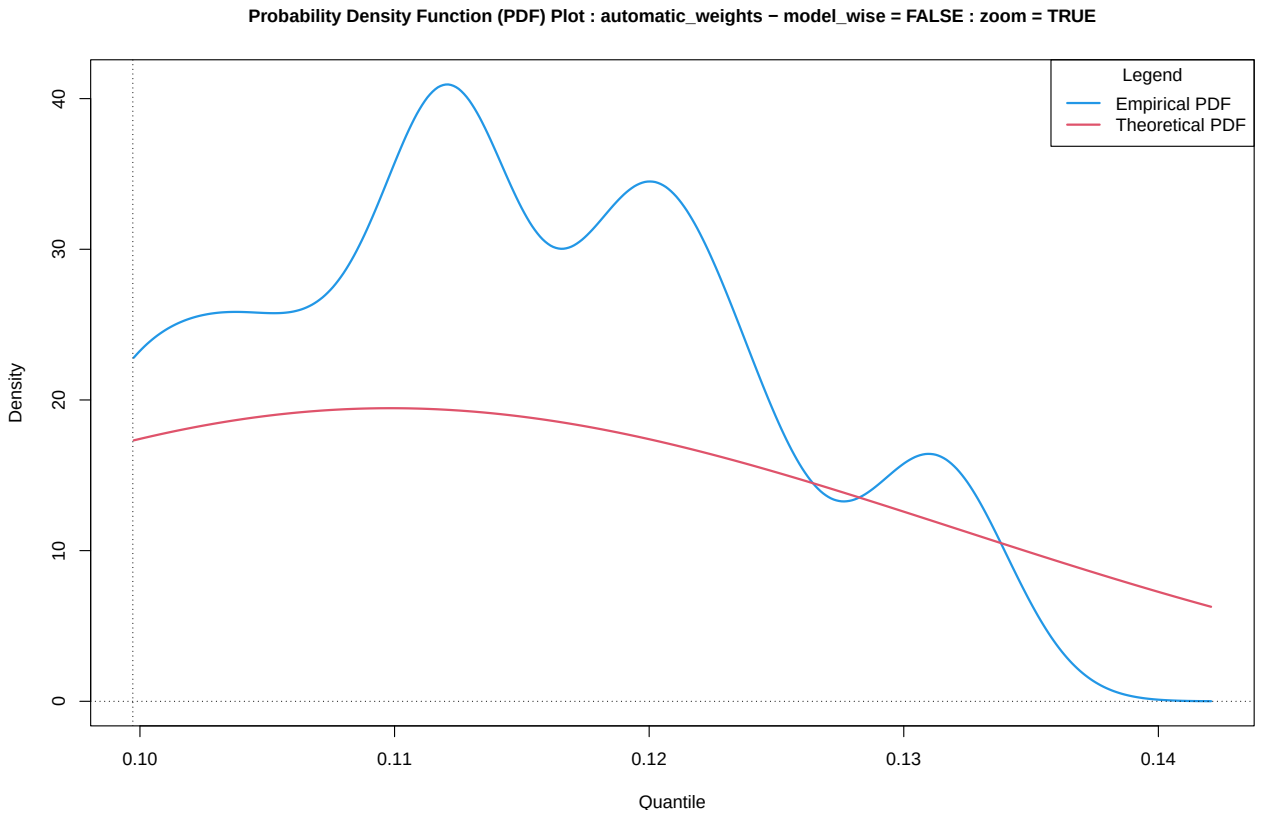
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                  2                  3                  4                  5
## 0.0537570994451162 0.0535043185494519 0.0533255367691116 0.0531370197070995
##                  6                  7                  8                  9
## 0.0529958676478209 0.0528783450377866 0.0531584140127129 0.0525881826397627
##                 10                 11                 12                 13
## 0.0525650061408565 0.0524616061562057 0.0523682072376544 0.0523286062224342
##                 14                 15                 16                 17
## 0.0522619733323671 0.0521889399777897 0.0521041505300215 0.0521364944789632
##                 18                 19                 20
## 0.0520055160453267 0.0521860808215304 0.0520486352479881
```

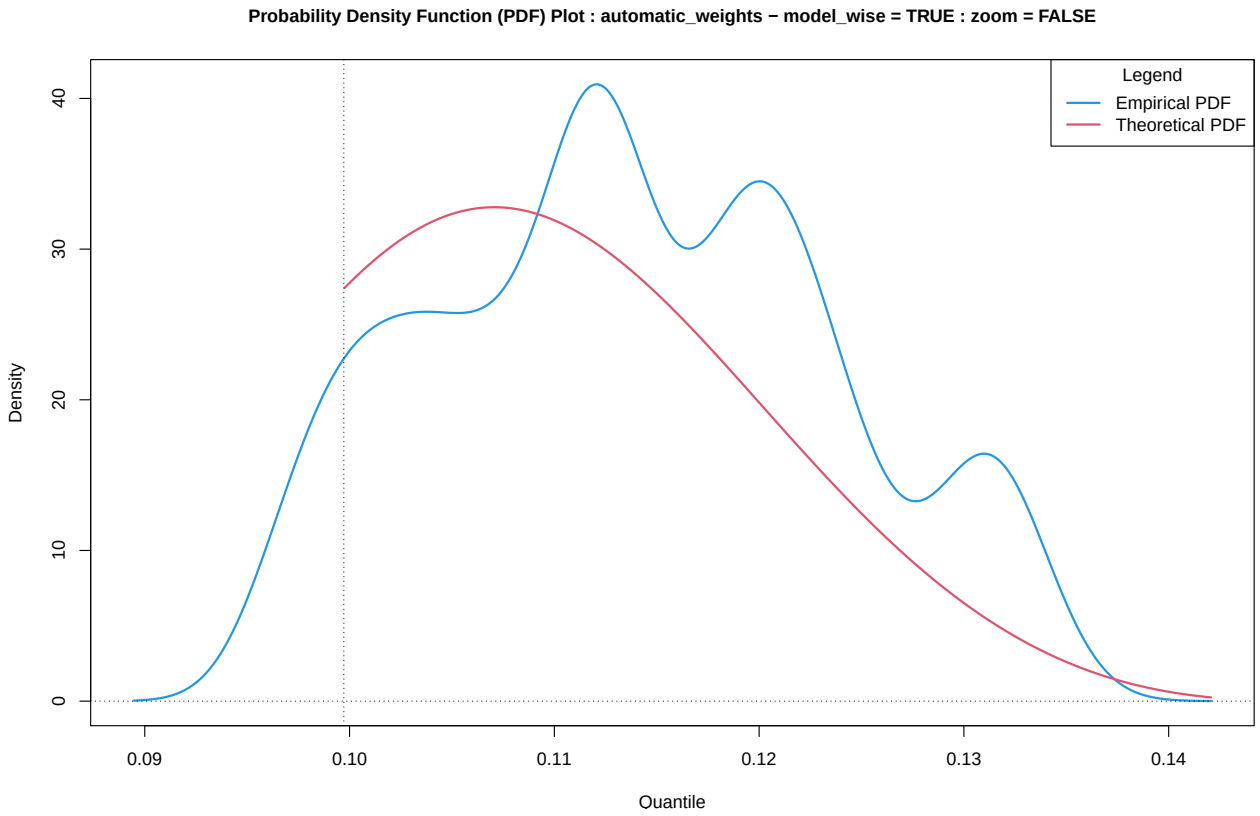
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



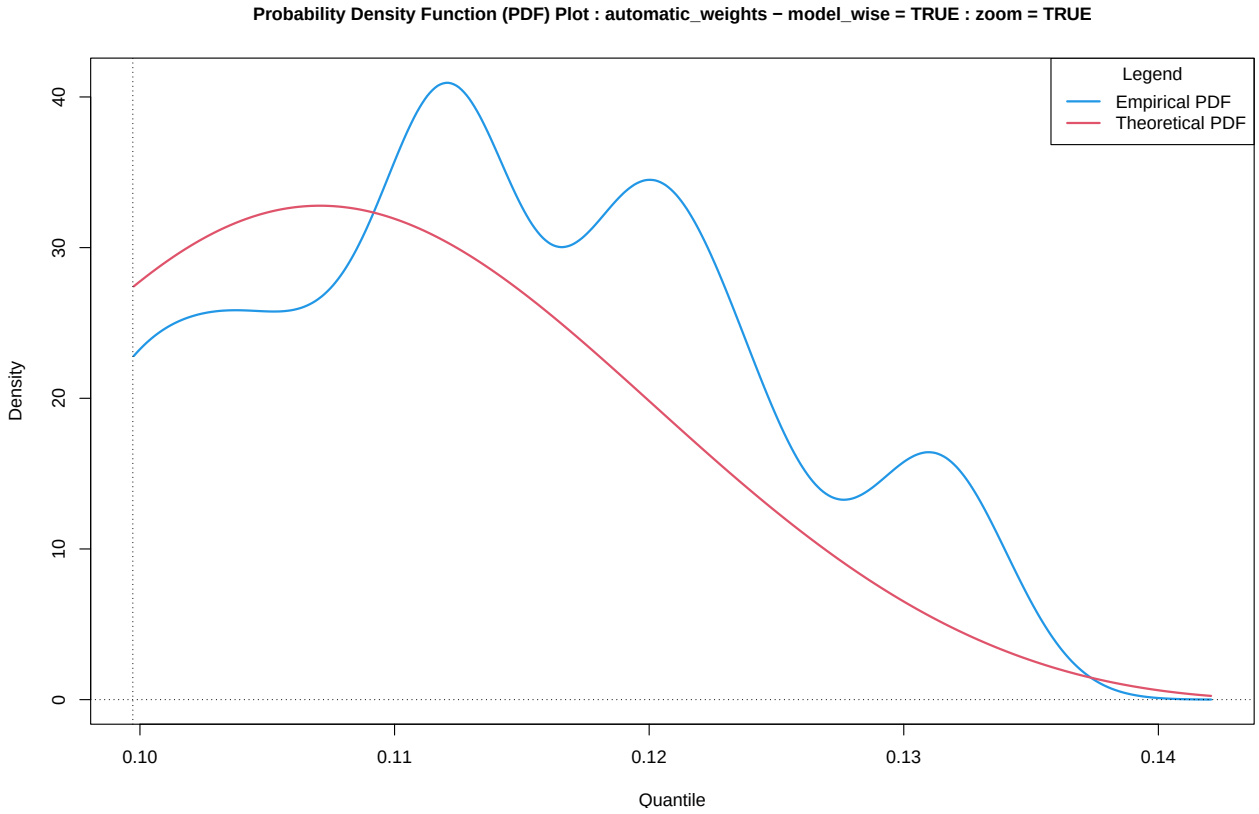

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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 0.147213593253778    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 0.173626944507789    NA
```

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

```
##              100%
## 0.135415352500097
```

```
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  0.138129281437151 0.146827998360211  0.15552671528327
## 3  0.136338757844029 0.146932397783181 0.157526037722333
## 4  0.134959782518006  0.14798964702532 0.161019511532635
## 5  0.133347606074153 0.148821003034056  0.16429439999396
## 6   0.13239991497598  0.14777951425293  0.16315911352988
## 7  0.130661402602816 0.148997599322063 0.167333796041309
## 8  0.129597700996349 0.149199163131364 0.168800625266379
## 9  0.128957357440838 0.148447079763703 0.167936802086567
## 10 0.127750145287134  0.14882971320025 0.169909281113366
## 11  0.12624185372688 0.149205445206551 0.172169036686222
## 12 0.125703573358383 0.148844582733274 0.171985592108164
## 13 0.123402885910373 0.149721746228553 0.176040606546733
## 14 0.122036088705979 0.151181203446786 0.180326318187593
## 15 0.121517521234689  0.15005496908564 0.178592416936591
## 16 0.121650553382753 0.149578055319203 0.177505557255653
## 17 0.118113368049048 0.150529248433548 0.182945128818047
## 18 0.120139000555453 0.149282692059401  0.17842638356335
## 19 0.108342074171995 0.155358074253686 0.202374074335377
## 20 0.111720001137797 0.153541429121345 0.195362857104893
```

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

```
## [1] 0.108342074171995 0.202374074335377
```

```
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 0.138129281437151 0.146827998360211  0.15552671528327
## 3 0.136338757844029 0.146932397783181 0.157526037722333
```

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

```
## [1] 0.136338757844029 0.157526037722333
```

```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)
```