Modeling extreme values with a GEV mixture probability distributions
Application to a wind speed data

Pascal Alain Dkengne Sielenou

2023-09-27

```r
# library(xfun)
```

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

```r
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"))
```

```r
library(readr)
```

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

```
## Rows: 10627 Columns: 2
## -- Column specification ----------------------------------------------------
## Delimiter: ","
## dbl  (1): Vent
## date (1): Date
##
## 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.
```

```r
x <- vent$Vent
x <- x[!is.na(x)]
```

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

```
## [1] 10607
```

```r
nlargest <- 2000
```

```r
gev_mixture_model <- estimate_gev_mixture_model_parameters(x,
                                                           nsloc = NULL,
                                                           std.err = FALSE,
                                                           block_sizes = NULL,
                                                           minimum_nblocks = 50,
                                                           threshold = NULL,
                                                           nlargest = nlargest,
                                                           confidence_level = 0.95,
```

```
                                                    log_mv = TRUE,
                                                    log_pw = TRUE,
                                                    trace = FALSE)
```

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

```r
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"
```

```r
gev_mixture_model$block_sizes
```

```
##  [1] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##            loc_star        scale_star            shape_star
## 19 12.20744497069555 3.91660441408066 -0.050064740184636587
## 20 13.33940412121185 3.38625979464296 -0.022202200772843829
## 21 12.12026007538837 3.90639604826376 -0.053305975219225470
## 22 12.52835542364349 3.65888345900501 -0.034942208227873930
## 23 13.58438624480374 3.11809198938246  0.000114768015159329
## 24 13.55429571915738 3.23998466633287 -0.018253845006561398
## 25 12.79536306994679 3.66948701032461 -0.037925539633644659
## 26 14.97859273509689 2.62816267679412  0.030405257958260011
## 27 10.22869453309226 4.59309685630972 -0.076962228519940543
## 28 12.41471387940332 3.51205682808384 -0.014430006323105390
## 29 12.49209556320523 3.83727364424011 -0.047476308610663057
## 30 13.01790714224697 3.46566019456546 -0.027167007870192311
## 31 14.58207969348401 2.52601749587086  0.052459998583495683
## 32 12.44009274859522 3.65501061733934 -0.032962149677229213
## 33 13.90066077030271 2.89708489564325  0.017560574802178455
## 34 10.57176442250722 4.40582389381168 -0.059758582510654711
## 35  7.95733706990223 5.75840658465214 -0.117504265224657581
## 36 14.59162520840709 2.68175239204769  0.030691321237269913
```

```
## 37  7.93235038981421 5.30169263952855 -0.092457513175610614
## 38 14.33883985064790 2.81337304751811  0.017404577634357241
## 39 10.89833480421348 4.09929293015722 -0.044270650024992315
## 40 12.04668518060133 3.72985767240219 -0.031592461133284934
```

`gev_mixture_model$weighted_normalized_gev_parameters_object`

```
##                              loc_star       scale_star          shape_star
## identic_weights     12.3873310734712 3.67273953413621 -0.0278472356311089
## pessimistic_weights 14.0734700711822 4.51598033717027 -0.0262077843911444
## automatic_weights   13.8830283506281 3.78878878906777 -0.1058485380576595
```

`gev_mixture_model$automatic_weights_mw_statistics`

```
## $function_value
## [1] 0.0533035045991523
##
## $gradient_value
## [1] 1.66533453693773e-15
##
## $function_reduction
## [1] 0.373734672367452
##
## $number_iterations
## [1] 493
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

`gev_mixture_model$automatic_weights_pw_statistics`

```
## $function_value
## [1] 0.00221772872200737
##
## $gradient_value
## [1] 6.8342247370512e-05
##
## $function_reduction
## [1] 0.437257153215912
##
## $number_iterations
## [1] 3836
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

`gev_mixture_model$automatic_weights_mw`

```
##                   19                  20                  21
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   22                  23                  24
```

```
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   25                   26                   27
## 0.00000000000000e+00 1.00000000000000e+00 0.00000000000000e+00
##                   28                   29                   30
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   31                   32                   33
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   34                   35                   36
## 0.00000000000000e+00 0.00000000000000e+00 1.11889664200504e-16
##                   37                   38                   39
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   40
## 0.00000000000000e+00
```

gev_mixture_model$pessimistic_weights_pw_shape

```
##                   19                   20                   21                   22
## 0.0444193578265033 0.0456743970325268 0.0442756173219218 0.0450961958381174
##                   23                   24                   25                   26
## 0.0467051702111629 0.0458550922912592 0.0449618594254480 0.0481415370045079
##                   27                   28                   29                   30
## 0.0432405137115701 0.0460307704353776 0.0445344832276305 0.0454481944527041
##                   31                   32                   33                   34
## 0.0492150809682790 0.0451855774074559 0.0475271286030306 0.0439908439043447
##                   35                   36                   37                   38
## 0.0415225160889668 0.0481553105003918 0.0425756540407733 0.0475197150838329
##                   39                   40
## 0.0446774746445280 0.0452475099796673
```

gev_mixture_model$pessimistic_weights_pw_scale

```
##                    19                    20                    21                    22
## 0.03917336543057505 0.02304969283040290 0.03877550360064979 0.03027360187365580
##                    23                    24                    25                    26
## 0.01762793109023462 0.01991309020683965 0.03059631750692833 0.01080009926482262
##                    27                    28                    29                    30
## 0.07705269240709239 0.02613954956904284 0.03618578248748711 0.02495446697001174
##                    31                    32                    33                    34
## 0.00975139297808406 0.03015658374954477 0.01413250614137868 0.06389343019785955
##                    35                    36                    37                    38
## 0.24710151229089405 0.01139466246038515 0.15650458207868331 0.01299761262023116
##                    39                    40
## 0.04702529171819848 0.03250033252699785
```

gev_mixture_model$pessimistic_weights_pw_loc

```
##                    19                    20                    21
## 0.013334051387203023 0.041358591234235402 0.012220759728616498
##                    22                    23                    24
## 0.018379418356759495 0.052839673052984981 0.051273382942204487
##                    25                    26                    27
## 0.024004447298542869 0.213037622685339267 0.001843324215430216
##                    28                    29                    30
## 0.016405061688340748 0.017724920913485785 0.029987576810727938
##                    31                    32                    33
## 0.143302207695417128 0.016826731724925045 0.072496393633930303
```

4

```
##                        34                        35                        36
## 0.002597736450423388 0.000190179506314946 0.144676650503425014
##                        37                        38                        39
## 0.000185486428220063 0.112360887170228560 0.003600997360061663
##                        40
## 0.011353899213183197
```
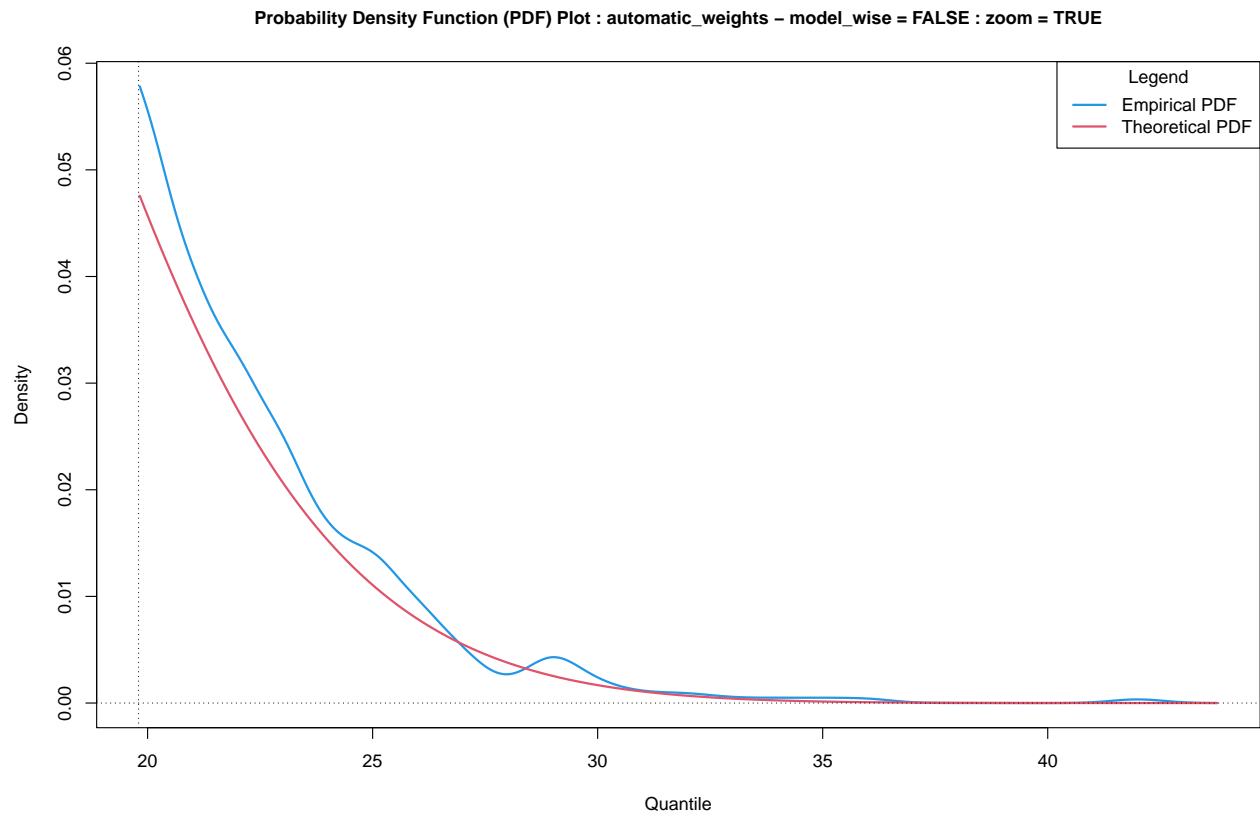
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                           main = "Probability Density Function (PDF) Plot")
```

Probability Density Function (PDF) Plot : automatic_weights – model_wise = FALSE : zoom = FALSE
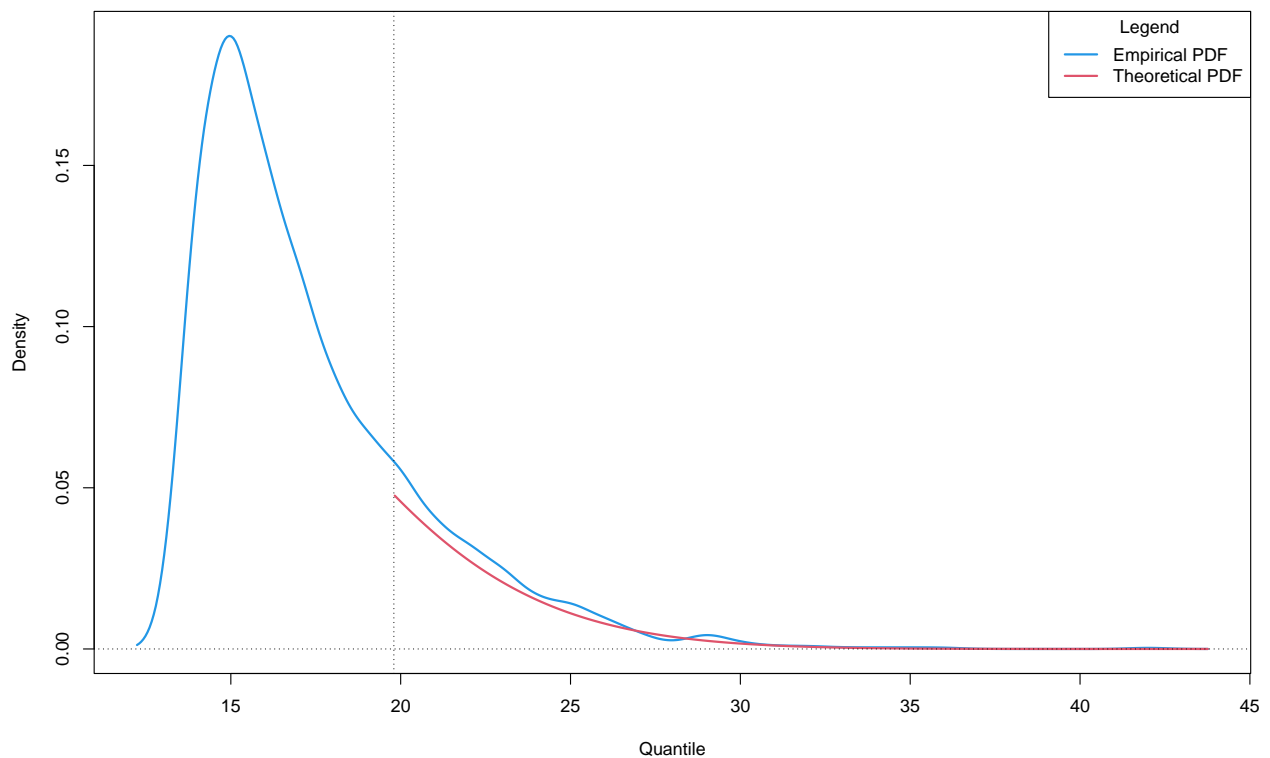

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                           main = "Probability Density Function (PDF) Plot")
```
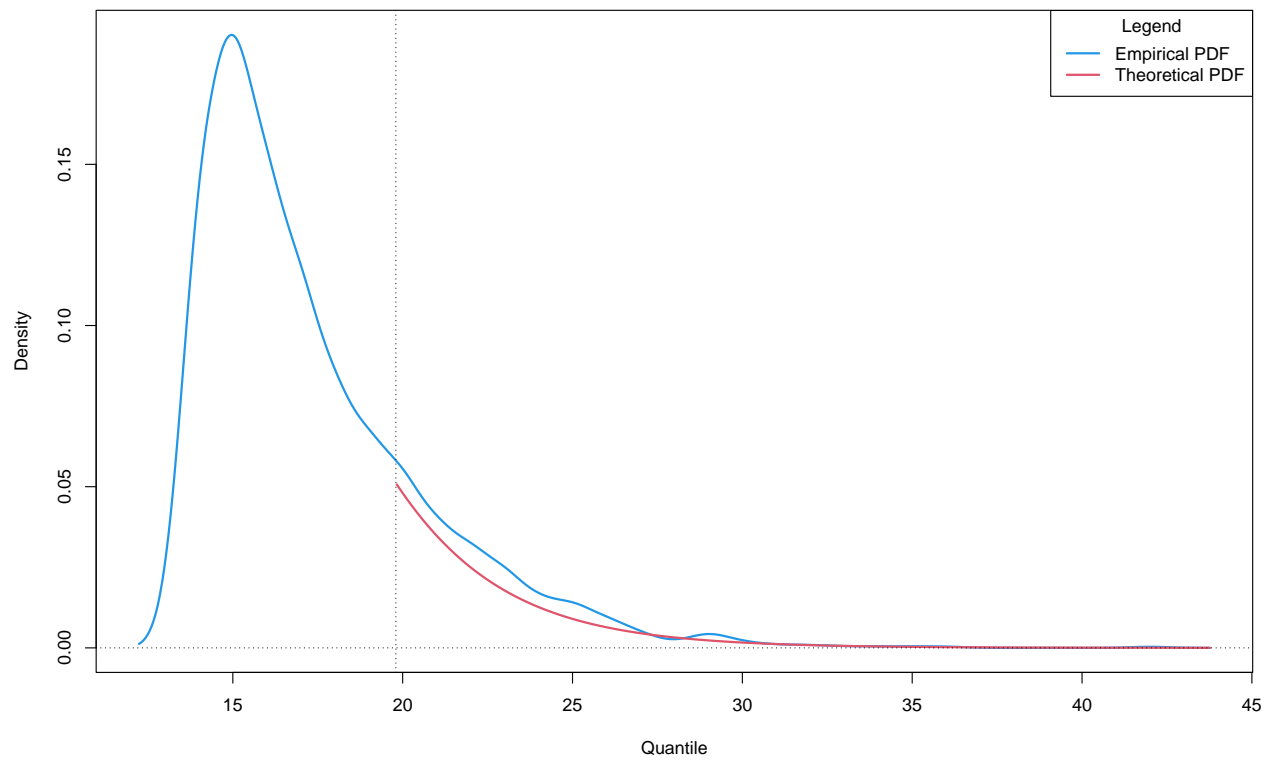
5

Probability Density Function (PDF) Plot : automatic_weights – model_wise = FALSE : zoom = TRUE


```r
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")
```
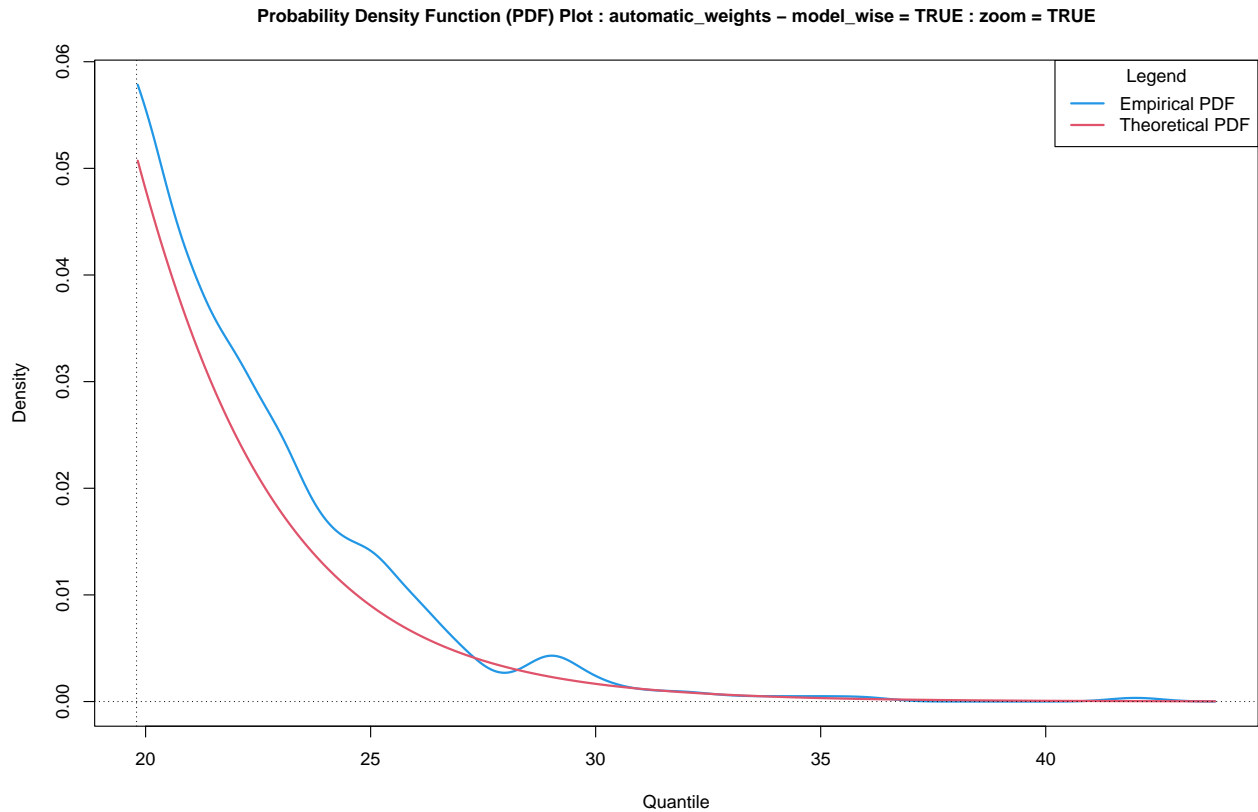
Probability Density Function (PDF) Plot : automatic_weights – model_wise = TRUE : zoom = FALSE


```
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 = TRUE


```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^(-6)
```

```r
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 53.596828657496    NA
```

```r
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 39.7824076741767    NA
```

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

```
##       99.9999%
## 41.936364000001
```

```
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
## 19 36.7389474241984 47.8523616921686 58.9657759601387
## 20 35.2855370482326 49.3926492296612 63.4997614110898
## 21 36.5882220051241 47.0508936708328 57.5135653365416
## 22 35.3387807097112 48.7450901237228 62.1513995377344
## 23 33.6672616830829 51.4862486335974 69.3052355841119
## 24 35.6033958368424  48.852536539198 62.1016772415536
## 25 35.7671007720629 48.5122806886458 61.2574606052286
## 26 32.9328879810057 53.5963917358147 74.2598954906237
## 27 36.6150187667301 46.4759033509195 56.3367879351089
## 28 33.1036771620926 51.5461707271353  69.988664292178
## 29 35.9546517536884 47.9137539478768 59.8728561420651
## 30 34.7711811468292  48.874277806638 62.9773744664467
## 31 28.7623100952125 57.4960802559006 86.2298504165888
## 32 34.3976752527474 49.0257110310343 63.6537468093212
## 33 31.2168062908376 53.1265675345949 75.0363287783522
## 34 34.5897890561079 48.6226182462672 62.6554474364266
## 35 37.2040262874502 45.2037648631293 53.2035034388085
## 36 31.0494032755755 54.0696482413894 77.0898932072033
## 37 36.3095360790161 46.6315076869183 56.9534792948205
## 38  31.480437363601 52.3937845329068 73.3071317022127
## 39 34.0870260811822 49.4129276229699 64.7388291647576
## 40 33.1432446259424 49.6748467085401 66.2064487911378
```

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

```
## [1] 28.7623100952125 86.2298504165888
```

```
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
## 26 32.9328879810057 53.5963917358147 74.2598954906237
```

```
## 36 31.0494032755755 54.0696482413894 77.0898932072033

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

```
## [1] 31.0494032755755 77.0898932072033
```

```
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)
```


Estimates of a quantile