Modeling extreme values with a GEV mixture probability distributions

Pascal Alain Dkengne Sielenou

2023-09-28

```
# 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"))
```

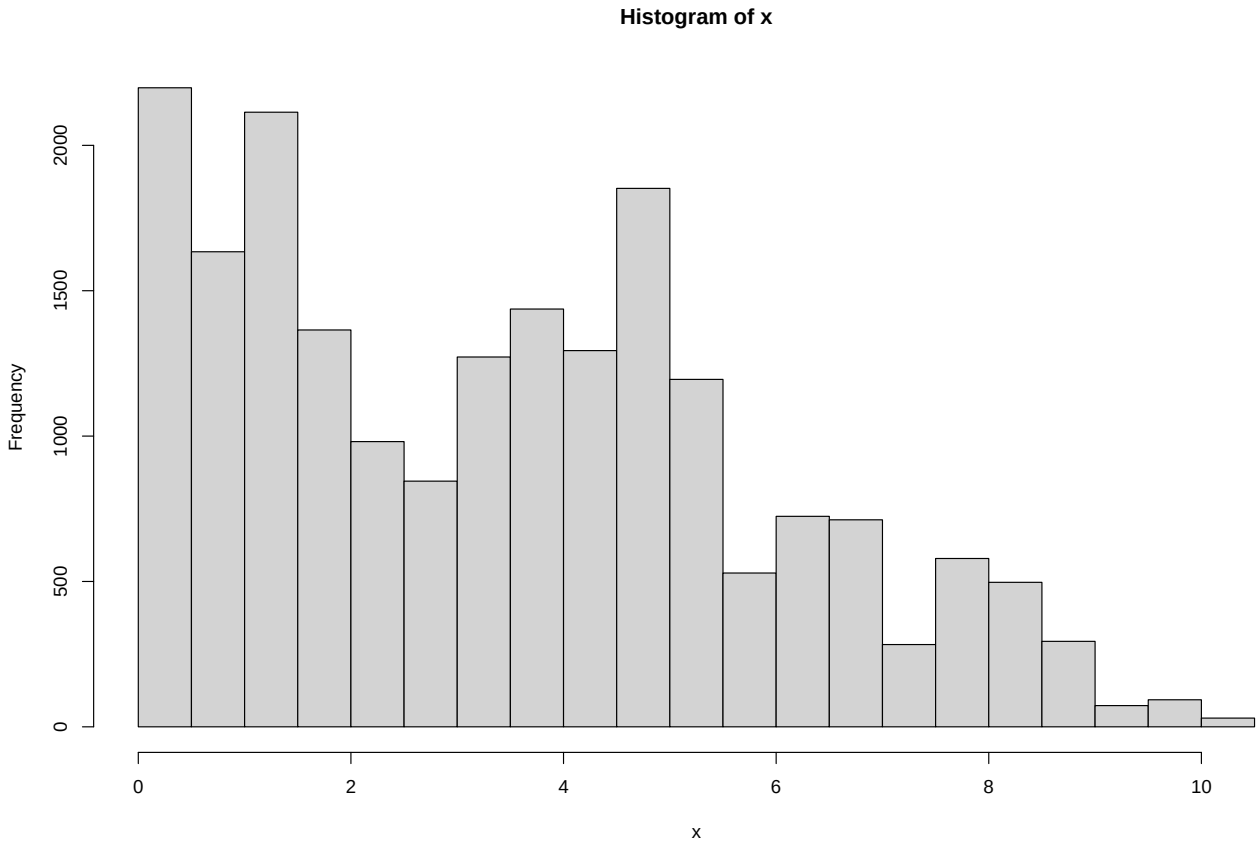
```
## Rows: 20001 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify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

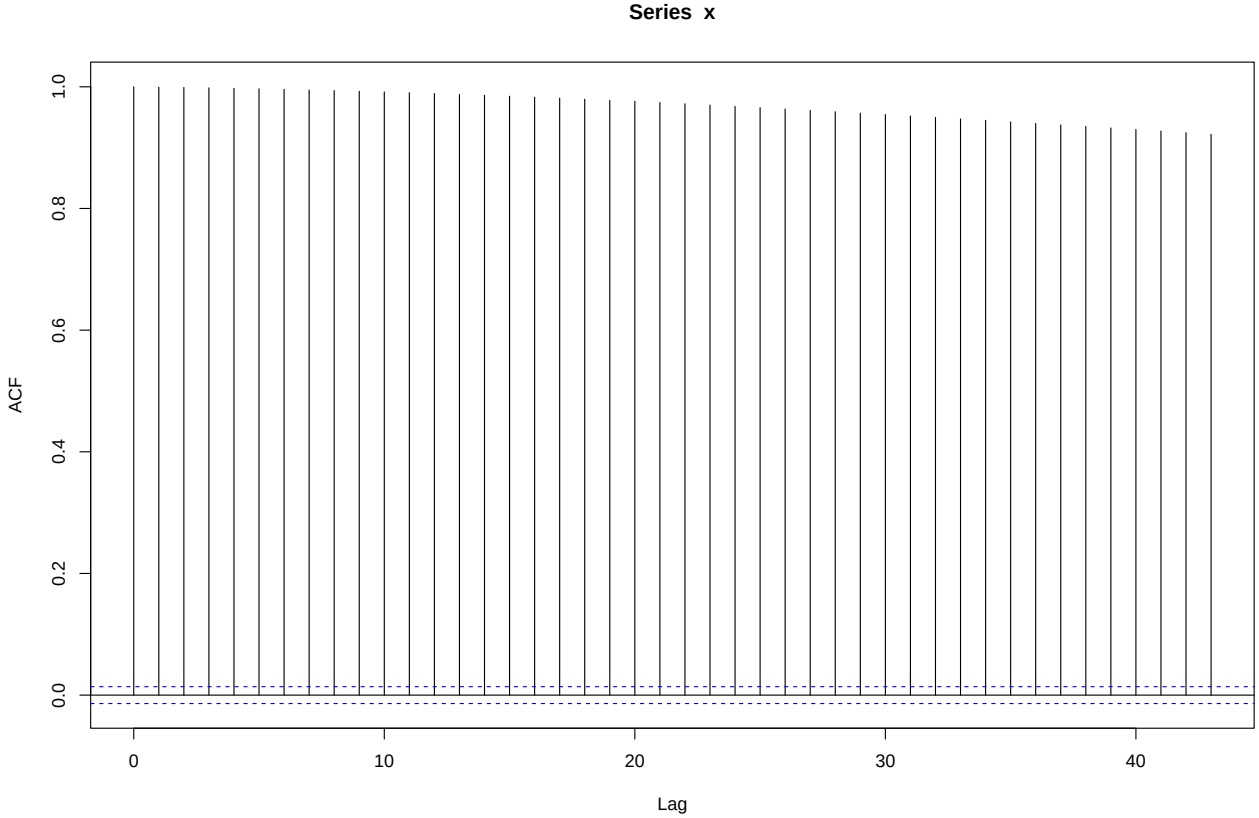
```
latitude_Gnss_standard_errors <- Gnss_imar$latitude[timestamp_position] - Gnss_standard$latitude
```

```
coefficient <- 10^(5)
x <- coefficient*abs(latitude_Gnss_standard_errors)
```

```
hist(x)
```




```
acf(x)
```




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

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

```
nlargest <- 1000

#
y <- extract_nlargest_sample(x, n = nlargest)
```

```
gev_mixture_model <- estimate_gev_mixture_model_parameters(x,
                                                           nsloc = NULL,
                                                           std.err = FALSE,
                                                           block_sizes = NULL,
                                                           minimum_nblocks = 50,
                                                           threshold = min(y),
                                                           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  8.19568741323931 0.238667499398149 0.305645586958560
## 3  8.10976688328106 0.213554633628678 0.303266791270471
## 4  8.07033411507302 0.189451007045208 0.318741441612394
## 5  8.03452014047604 0.179739649225530 0.311541753777175
## 6  8.00369853740988 0.170005044676169 0.313268425095372
## 7  7.98231479655153 0.166213009157696 0.303249152001071
## 8  7.99028308134887 0.142972562044951 0.342339279651148
## 9  7.96716698537096 0.142564155227287 0.328822005939105
## 10 7.97769046420849 0.130308627658875 0.347583112573954
## 11 7.93628705432173 0.144026987256888 0.306276267142409
## 12 7.95509046010435 0.121332410544804 0.353683618127263
## 13 7.93783685942683 0.131770246289268 0.315433678268880
## 14 7.89172659908504 0.140769361751004 0.296982311000943
## 15 7.93733494314746 0.115713169973633 0.346557866932923
## 16 7.93593459664920 0.112909321892428 0.345697460100454
## 17 7.88157539247376 0.137070693986474 0.288186940521290
## 18 7.91049847667249 0.118114499485689 0.320882649055202
## 19 7.88458597607089 0.125950462480757 0.306112096452279
## 20 7.90458608622845 0.122335783519788 0.307829155323856
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                           loc_star        scale_star        shape_star
## identic_weights     7.97404836111260 0.149656269749646 0.319057873252882
## pessimistic_weights 7.98066913830468 0.150829392892462 0.319417305290492
## automatic_weights   8.19568741350484 0.238667499398416 0.353683618127278
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 189.771413364467
```

```
##
## $gradient_value
## [1] 5.11590769747272e-13
##
## $function_reduction
## [1] 406.733575557649
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 187.505839269259
##
## $gradient_value
## [1] 7.27595761418343e-12
##
## $function_reduction
## [1] 430.989192163356
##
## $number_iterations
## [1] 1
##
## $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
```

```
## 0.0519210637432592 0.0517977009263466 0.0526054862145258 0.0522281032851557
##                  6                  7                  8                  9
## 0.0523183619542049 0.0517967872608039 0.0538616247446057 0.0531384610324154
##                 10                 11                 12                 13
## 0.0541448079394886 0.0519538196574537 0.0544761282240858 0.0524317671846699
##                 14                 15                 16                 17
## 0.0514732000246298 0.0540893246580205 0.0540428058489415 0.0510224592822664
##                 18                 19                 20
## 0.0527182461512673 0.0519452910631173 0.0520345608047421
```

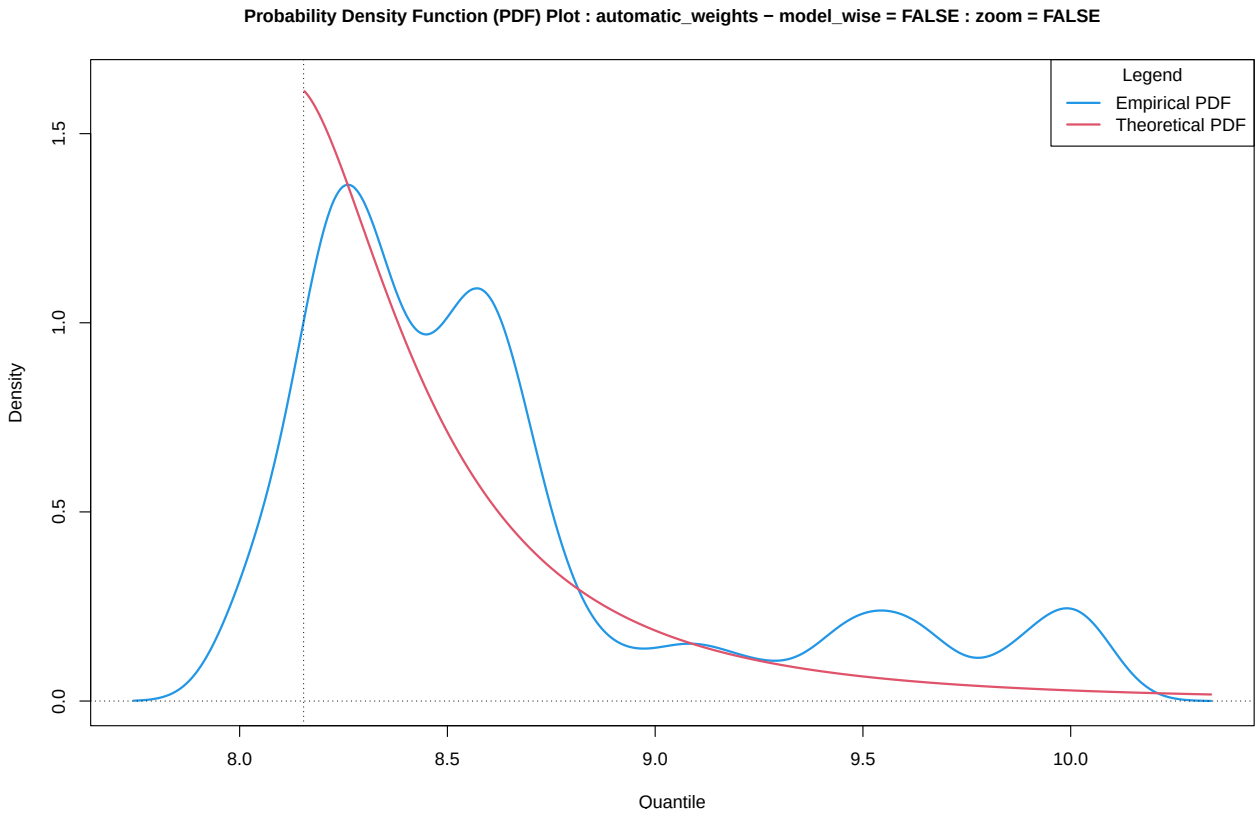
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                  2                  3                  4                  5
## 0.0574976937755434 0.0560717417206290 0.0547363677264048 0.0542073760448638
##                  6                  7                  8                  9
## 0.0536822487754206 0.0534790692567971 0.0522505230538248 0.0522291879409824
##                 10                 11                 12                 13
## 0.0515929980723808 0.0523056463792744 0.0511319604028019 0.0516684624962304
##                 14                 15                 16                 17
## 0.0521355314074033 0.0508454433743962 0.0507030801512445 0.0519430555695133
##                 18                 19                 20
## 0.0509676867526989 0.0513686365222141 0.0511832905773765
```

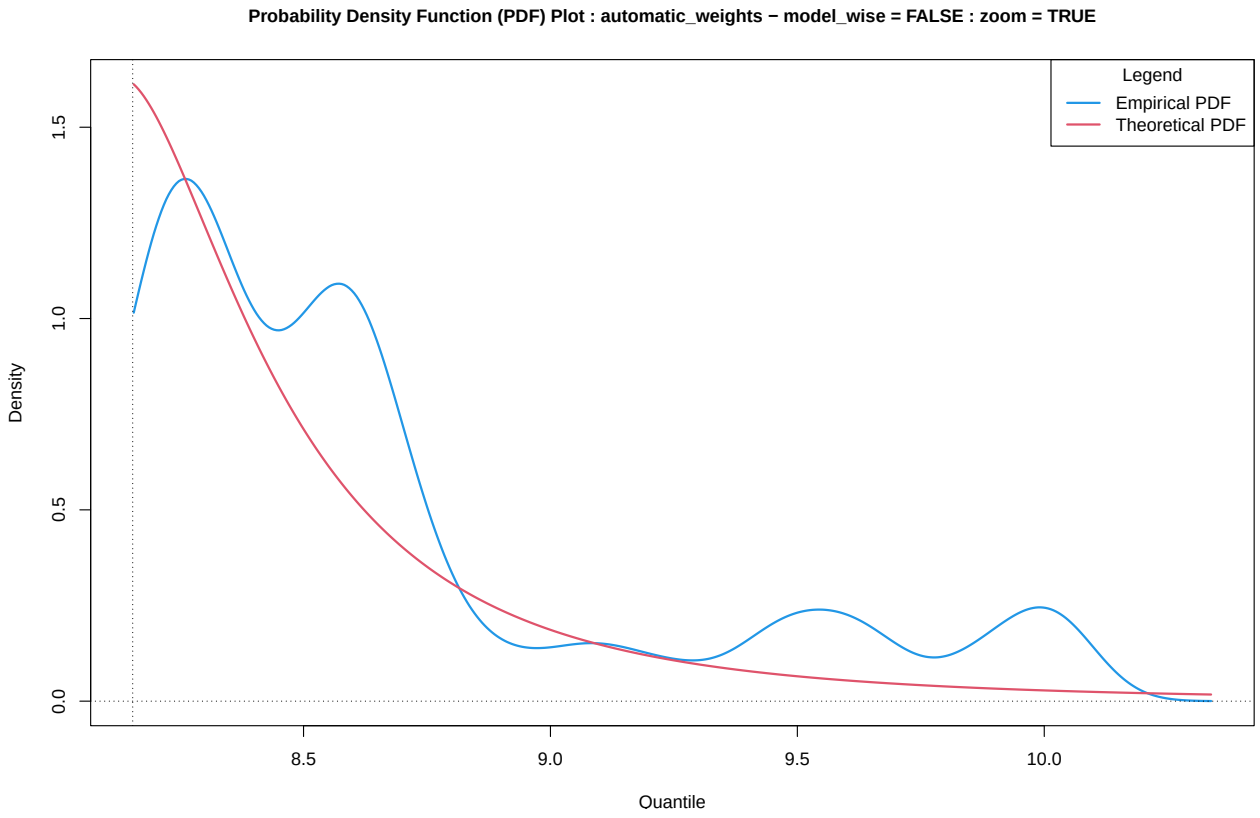
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                  2                  3                  4                  5
## 0.0654769244072274 0.0600860229631318 0.0577627718933664 0.0557306635865375
##                  6                  7                  8                  9
## 0.0540391565784467 0.0528958647512683 0.0533190378088670 0.0521006462948862
##                 10                 11                 12                 13
## 0.0526518213967687 0.0505163690076686 0.0514752355189538 0.0505947202331298
##                 14                 15                 16                 17
## 0.0483147533930156 0.0505693322912381 0.0504985672631884 0.0478267812920897
##                 18                 19                 20
## 0.0492302782047372 0.0479709847747535 0.0489400683407252
```

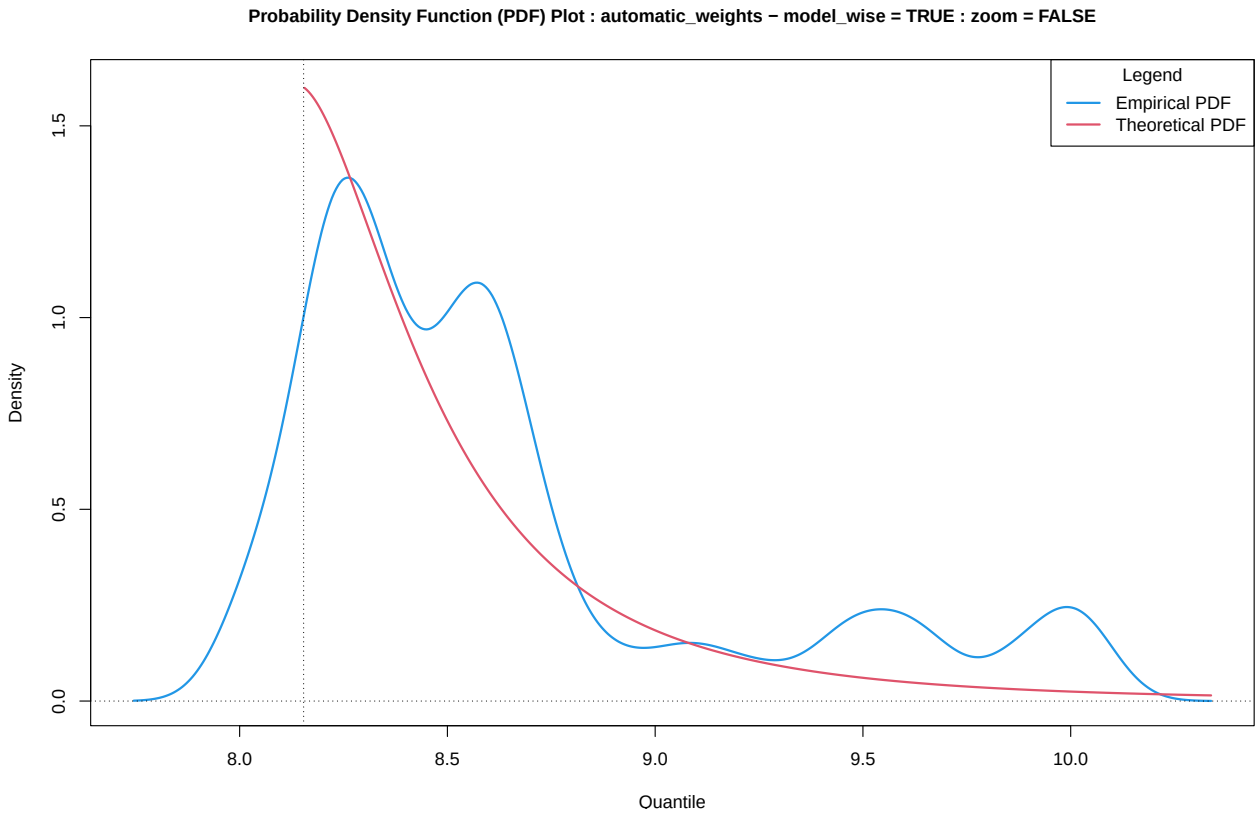
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



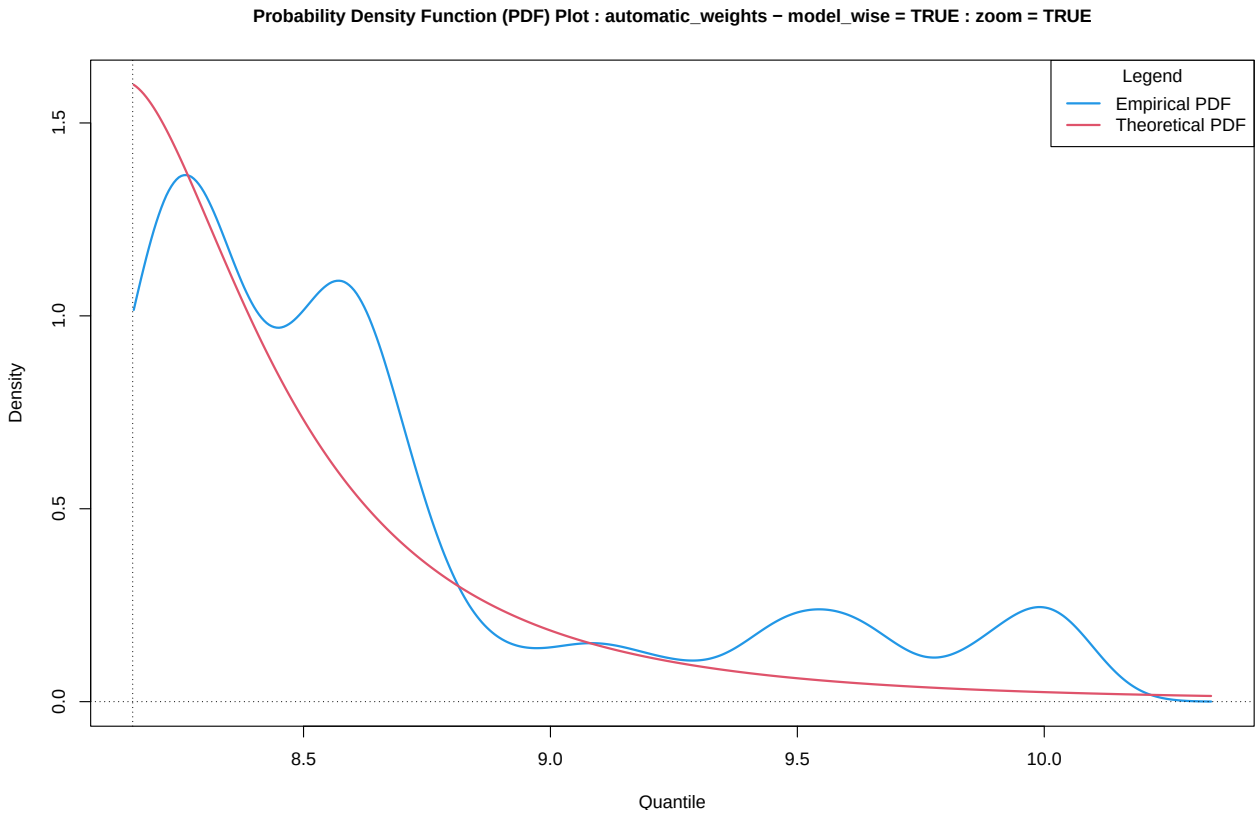

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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

```
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
```

```
##   lower         estimate upper
## 1    NA 5948.71095283378    NA
```

```
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
```

```
##   lower         estimate upper
## 1    NA 20926.8735339421    NA
```

```
rl_empirical <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                    alpha = alpha,
                                                    confidence_level = 0.95,
                                                    do.ci = TRUE,
                                                    estimator_type = estimator_types[7])

rl_empirical
```

```
##   lower        estimate upper
## 1    NA 10.097498900307    NA
```

```
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  -6275.17243174258  5946.4143856484 18168.0012030394
## 3  -7469.99397515395 5005.55487574333 17481.1037266406
## 4  -12984.9846538234 6637.90313927138 26260.7909323661
## 5  -11536.5026417734 5226.11602260648 21988.7346869863
## 6   -12737.915949967 5132.70978028836 23003.3355105438
## 7  -10546.7661505393 3895.83465355086  18338.435457641
## 8  -31258.6026750104 9299.37820779407 49857.3590905985
## 9  -23025.4400319397 6504.87510249116  36035.190236922
## 10 -37681.4301645868 9730.81736471867 57143.0648940241
## 11 -13436.3649108304  3652.3004021727 20740.9657151758
## 12 -47214.8483475772 10642.6759892364   68500.20032605
## 13 -18403.3691500322 4239.45836292525 26882.2858758827
## 14 -12218.6287750451 2807.21232253921 17833.0534201235
## 15 -41094.5170112152 8412.21185367664 57918.9407185685
## 16 -39732.9125865019   8024.293489516 55781.4995655339
## 17 -10786.7912463279 2177.32575222081 15141.4427507695
## 18  -22304.632794126 4368.72094755086 31042.0746892277
## 19 -16859.0061589495 3178.52424279078  23216.054644531
## 20 -19048.7520374799 3231.53281966909 25511.8176768181
```

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

```
## [1] -47214.8483475772  68500.2003260500
```

```
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 -6275.17243174258  5946.4143856484 18168.0012030394
## 3 -7469.99397515395 5005.55487574333 17481.1037266406
```

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

```
## [1] -7469.99397515395 18168.00120303939
```

```
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 = 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)
```