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"))
```

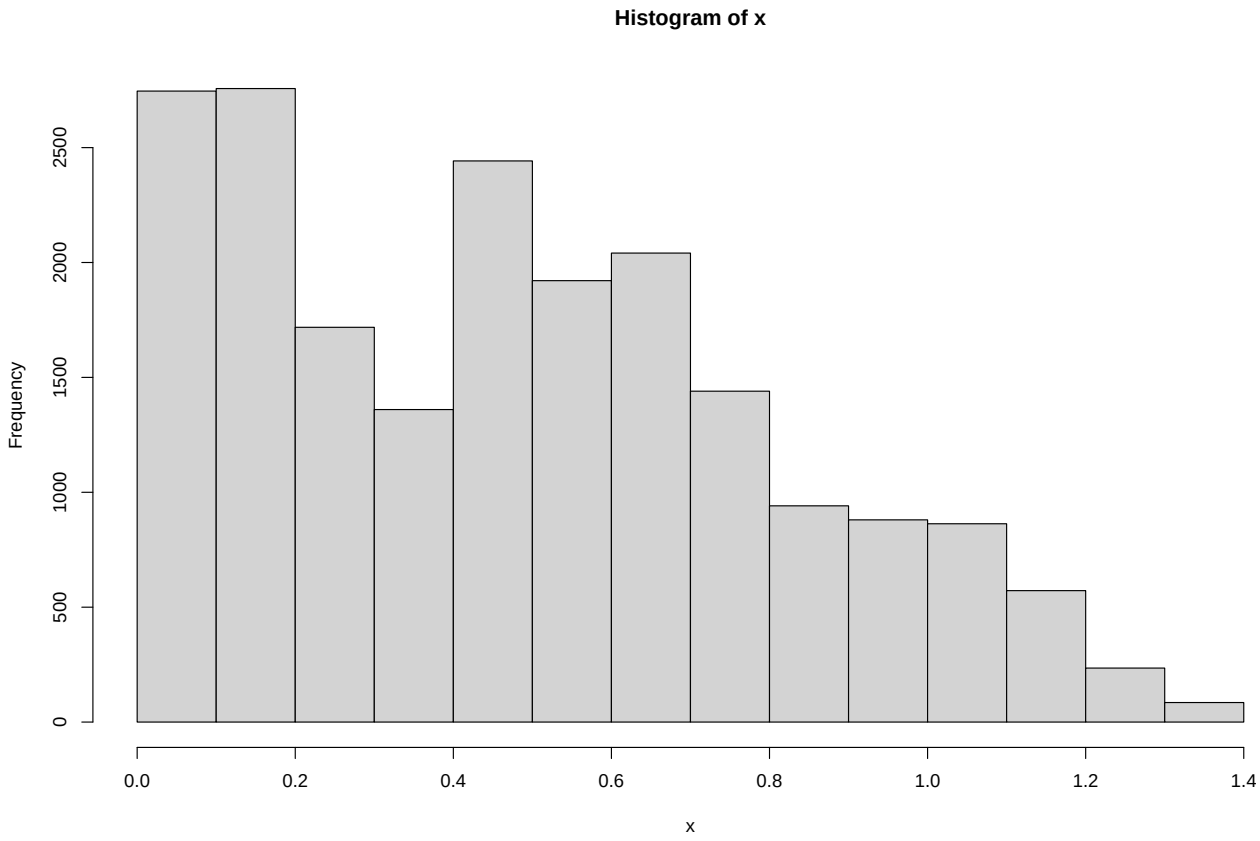
```
## Rows: 20001 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify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

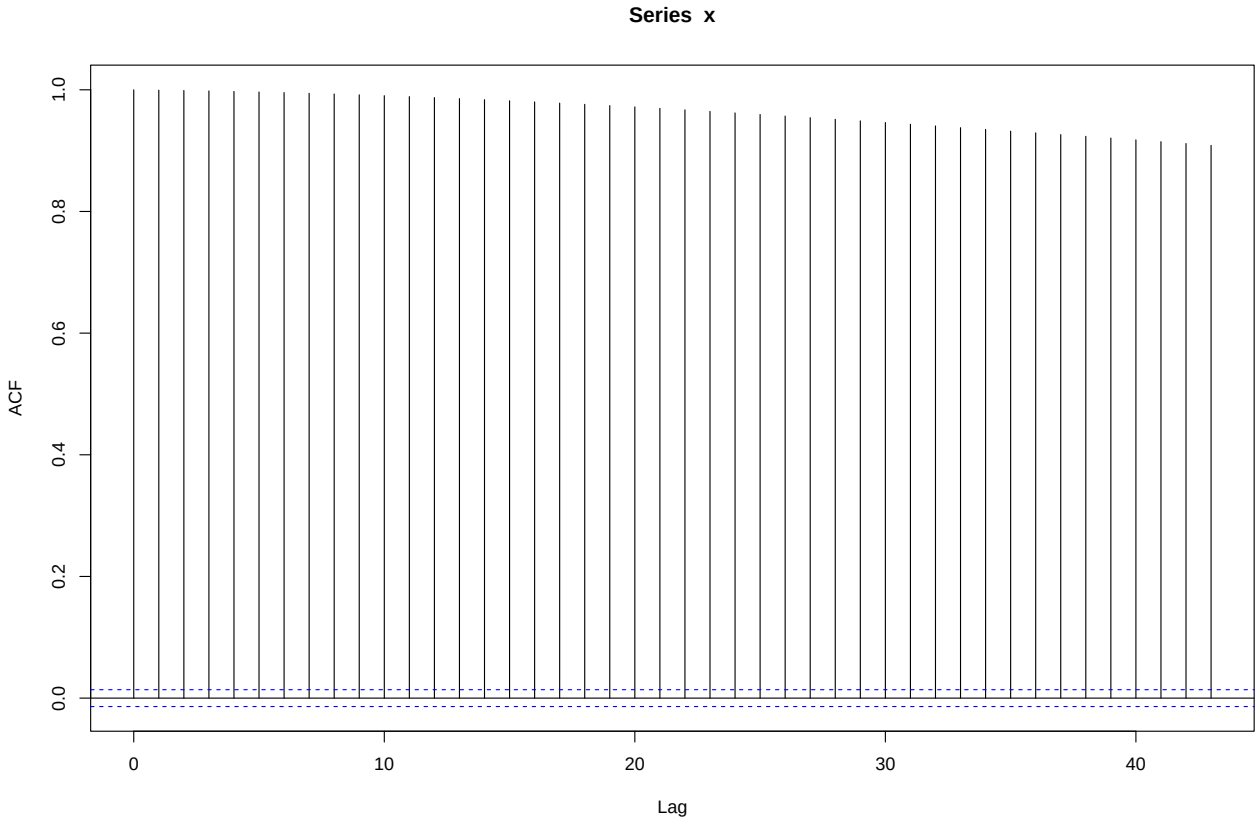
```
longitude_Gnss_standard_errors <- Gnss_imar$longitude[timestamp_position] - Gnss_standard$longitude
```

```
coefficient <- 10^(4)
x <- coefficient*abs(longitude_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  1.11097575884424 0.0437683135246788 0.1584944198604022
## 3  1.09599318821926 0.0423716103008420 0.1418345677806985
## 4  1.08564649967897 0.0408005169236056 0.1454188217254399
## 5  1.07801790184958 0.0415700035796163 0.1191402487429947
## 6  1.07277466256394 0.0388988975348902 0.1419456442135157
## 7  1.06499623163615 0.0405359077955685 0.1186983656507314
## 8  1.06360174122789 0.0383838704193530 0.1323597994256946
## 9  1.05528723695830 0.0401552093149766 0.1134280042117844
## 10 1.05747720132588 0.0362872761696244 0.1430424921910143
## 11 1.05283689293564 0.0363283526241252 0.1401732986769047
## 12 1.04963791392624 0.0366477486700707 0.1344092947694165
## 13 1.05081668518771 0.0349280369250175 0.1409125495654031
## 14 1.04123005522348 0.0390930708216718 0.1082298118187518
## 15 1.04814073770671 0.0347858846126763 0.1365648976792762
## 16 1.04051866112567 0.0375208280172684 0.1122834787717200
## 17 1.03415443703117 0.0398475627733267 0.0935629716116705
## 18 1.04265169508298 0.0349909704999583 0.1264161478523425
## 19 1.06646659520100 0.0219576157887703 0.2484245448487418
## 20 1.03503758227236 0.0361393082528887 0.1194348777660872
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                            loc_star         scale_star        shape_star
## identic_weights     1.06032956199985 0.0376321570815226 0.135514433534873
## pessimistic_weights 1.06074660058789 0.0376523022639200 0.136497630551664
## automatic_weights   1.11097575883414 0.0437683135245868 0.248424544848753
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 138.016540594381
```

```
##
## $gradient_value
## [1] 2.8421709430404e-13
##
## $function_reduction
## [1] 377.733727143781
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 133.634362382552
##
## $gradient_value
## [1] 3.63797880709171e-11
##
## $function_reduction
## [1] 411.843570964308
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                     2                     3                     4
##  9.99999999999204e-01  0.00000000000000e+00  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
## -1.42108547152020e-14
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

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

```
## 0.0538289133153459 0.0529395603958899 0.0531296496847090 0.0517516633629966
##                  6                  7                  8                  9
## 0.0529454410600092 0.0517288002297481 0.0524403390622078 0.0514568879232039
##                 10                 11                 12                 13
## 0.0530035460202823 0.0528516865510971 0.0525479255042358 0.0528907716524066
##                 14                 15                 16                 17
## 0.0511900991303152 0.0526613201383530 0.0513980278957340 0.0504447812163421
##                 18                 19                 20
## 0.0521295764057682 0.0588940972999338 0.0517669131514215
```

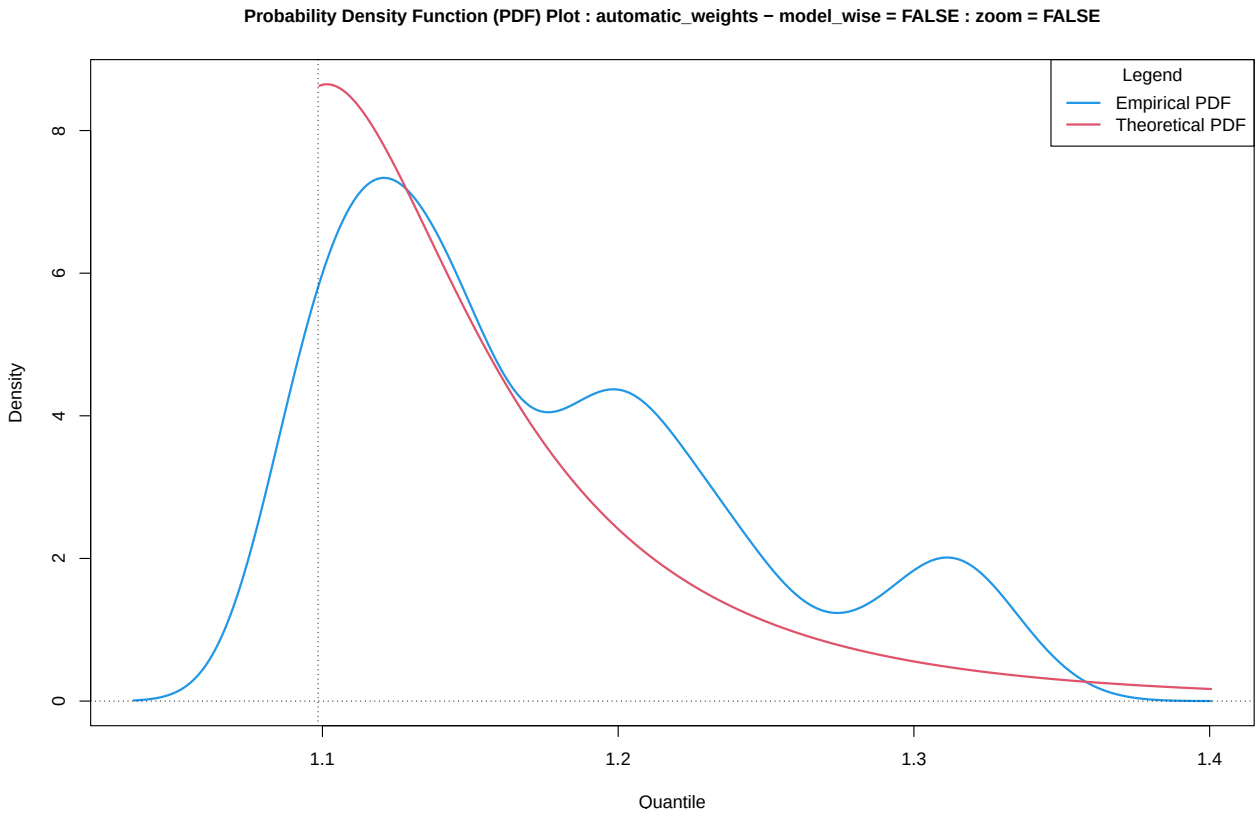
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                  2                  3                  4                  5
## 0.0529549932428189 0.0528810824607675 0.0527980665722645 0.0528387096150892
##                  6                  7                  8                  9
## 0.0526977601477319 0.0527840975701511 0.0526706263602456 0.0527640065689561
##                 10                 11                 12                 13
## 0.0525603131093616 0.0525624721450140 0.0525792630721174 0.0524889196004959
##                 14                 15                 16                 17
## 0.0527079936385289 0.0524814587095064 0.0526251889863628 0.0527477764015171
##                 18                 19                 20
## 0.0524922230197980 0.0518125123345825 0.0525525364446905
```

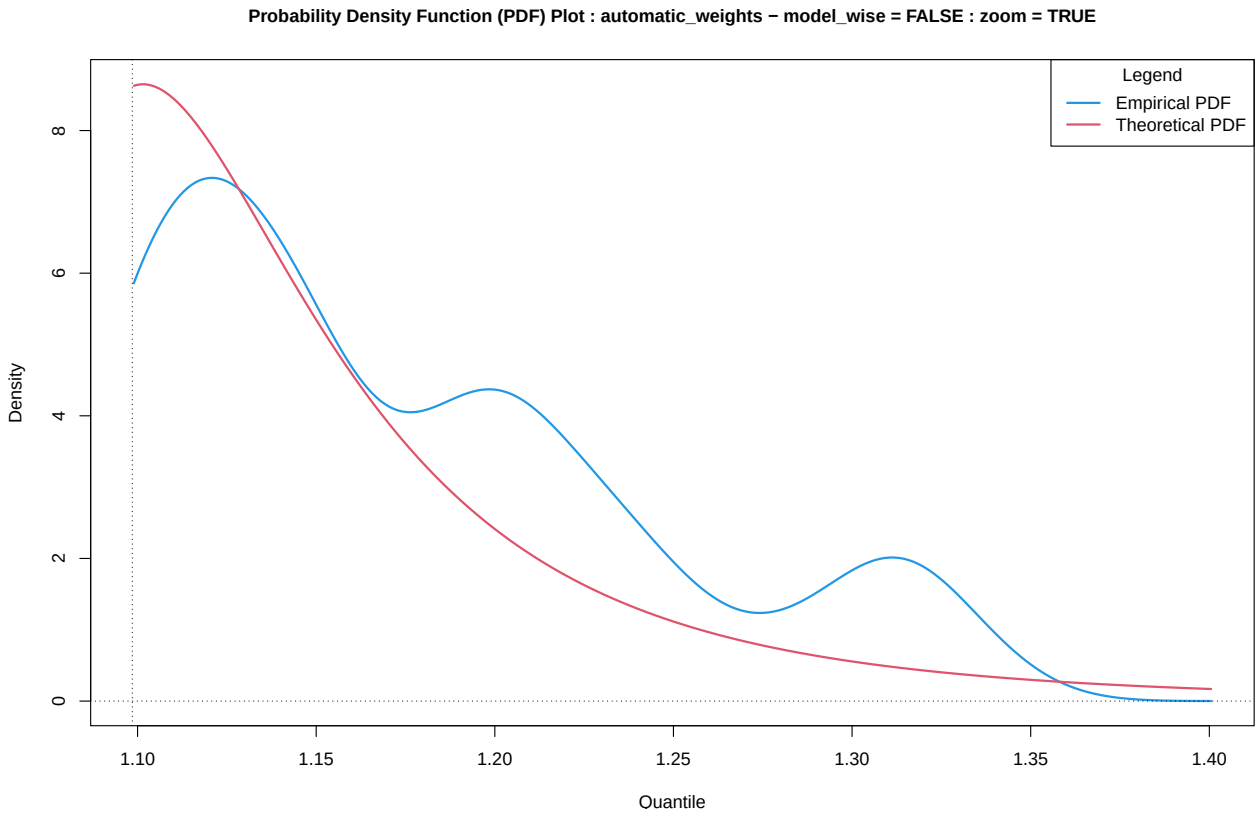
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                  2                  3                  4                  5
## 0.0553543138638852 0.0545311459313582 0.0539698379953671 0.0535596902216524
##                  6                  7                  8                  9
## 0.0532795988843856 0.0528667748476867 0.0527931040158186 0.0523559752996212
##                 10                 11                 12                 13
## 0.0524707586598173 0.0522278421976966 0.0520610333780090 0.0521224376116605
##                 14                 15                 16                 17
## 0.0516251465706002 0.0519831471561195 0.0515884338062013 0.0512611559921026
##                 18                 19                 20
## 0.0516985911302043 0.0529445654033982 0.0513064470344155
```

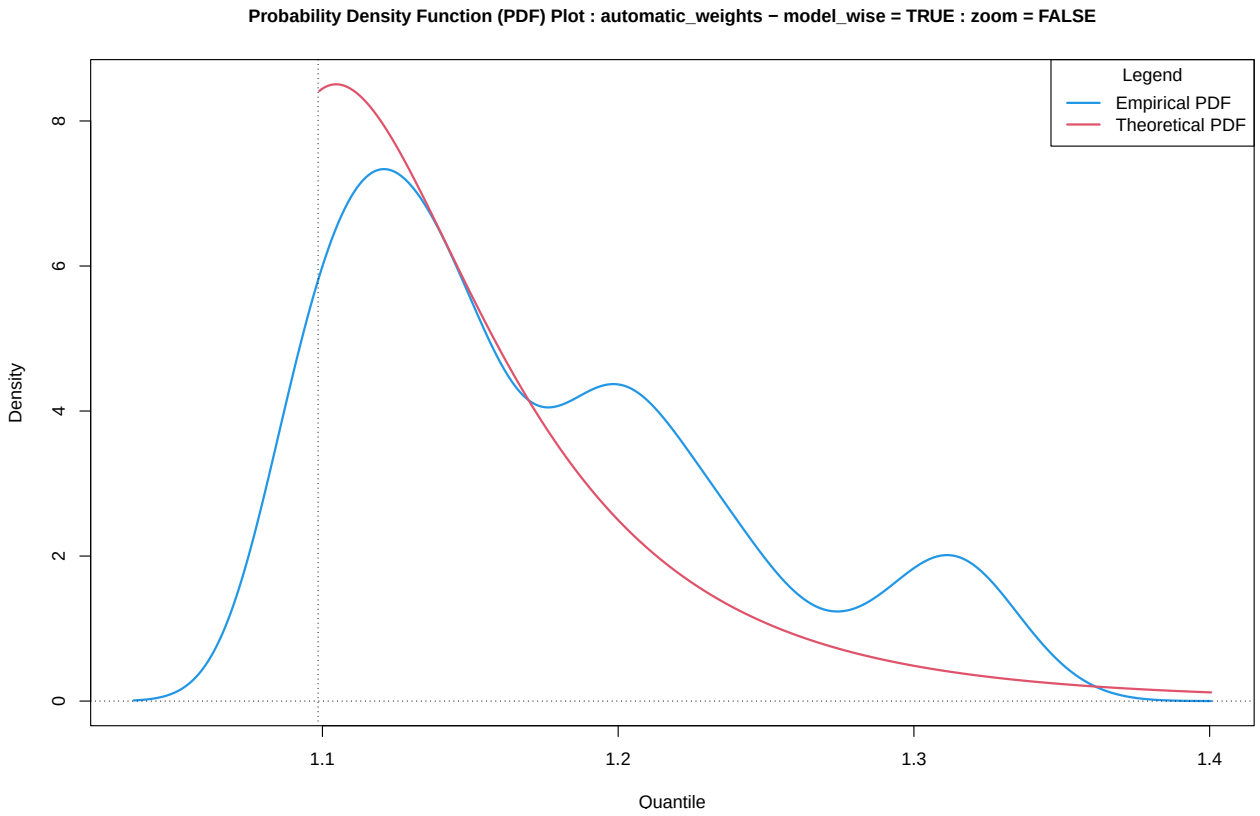
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



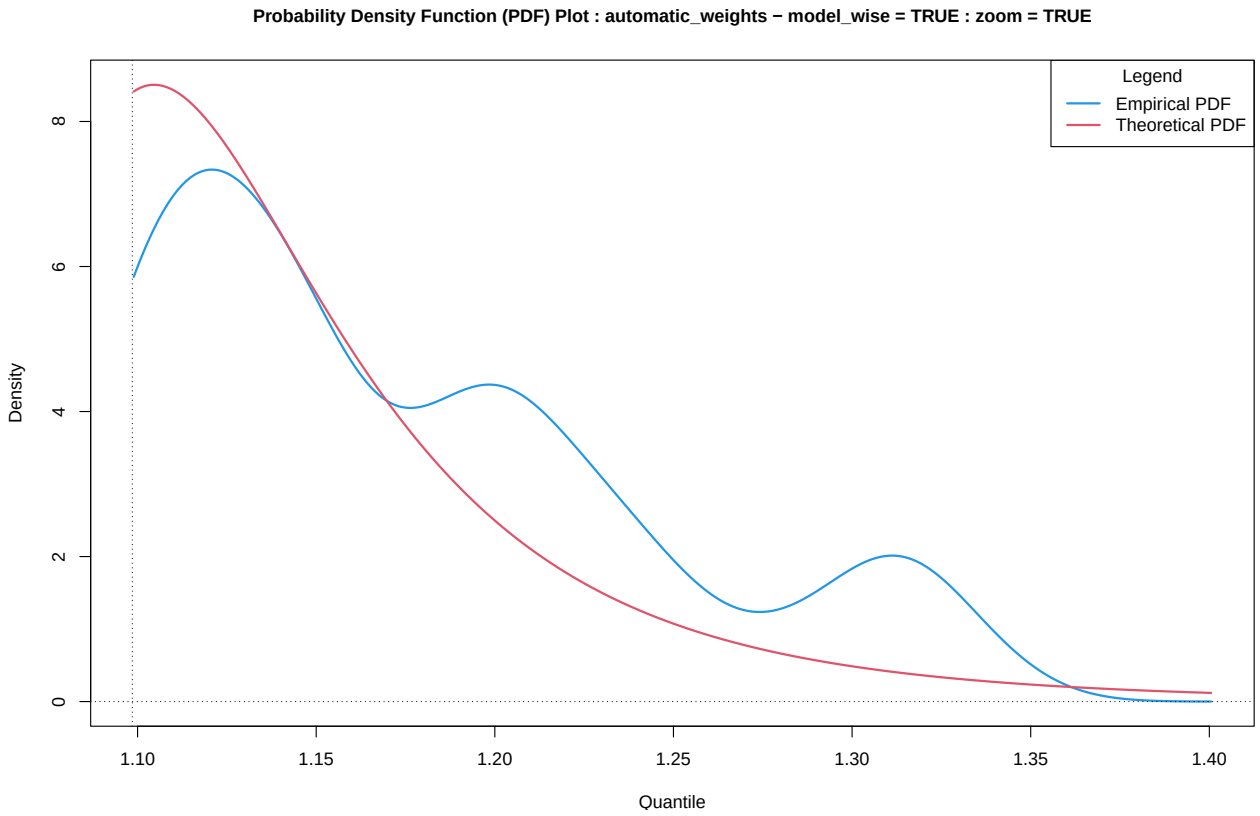

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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 29.2689798198916    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 252.509867236709    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 1.35415352500097    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  -36.1319073825935 29.4623239395381 95.0565552616698
## 3  -29.2071084472518 19.7595086481908 68.7261257436335
## 4  -38.6540827899554  20.562679478267 79.7794417464895
## 5  -21.6289136113142 12.1202170884523 45.8693477882188
## 6  -42.8613882008291 18.2329269896379 79.3272421801049
## 7  -25.9203060091211 11.6968841384384 49.3140742859979
## 8  -40.9325569738831 14.6950543180462 70.3226656099755
## 9  -26.3088500839829 10.4713156339783 47.2514813519395
## 10 -57.5403544872437 17.4931926853178 92.5267398578793
## 11 -58.6499606102254  16.482751842339 91.6154642949035
## 12  -51.230615538127 14.6947716152772 80.6201587686814
## 13 -55.9079292174909  16.105488295057 88.1189058076049
## 14 -29.3230268320319 9.24599177761938 47.8150103872707
## 15 -57.3591288249517 14.6468815757504 86.6528919764525
## 16 -34.8477370254002 9.64045260748223 54.1286422403647
## 17 -21.6982953859776 7.18988127462724 36.0780579352321
## 18 -52.0360401285937 11.9593156961942  75.954671520982
## 19 -853.065296012808 131.494913828091 1116.05512366899
## 20  -44.119980048772 10.6920331446367 65.5040463380454
```

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

```
## [1] -853.065296012808 1116.055123668989
```

```
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 -36.1319073825935 29.4623239395381 95.0565552616698
```

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

```
## [1] -36.1319073825935  95.0565552616698
```

```
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)
```