Modeling extreme values with a GEV mixture probability distributions

Pascal Alain Dkengne Sielenou

September 28th, 2023

```
# library(xfun)
```

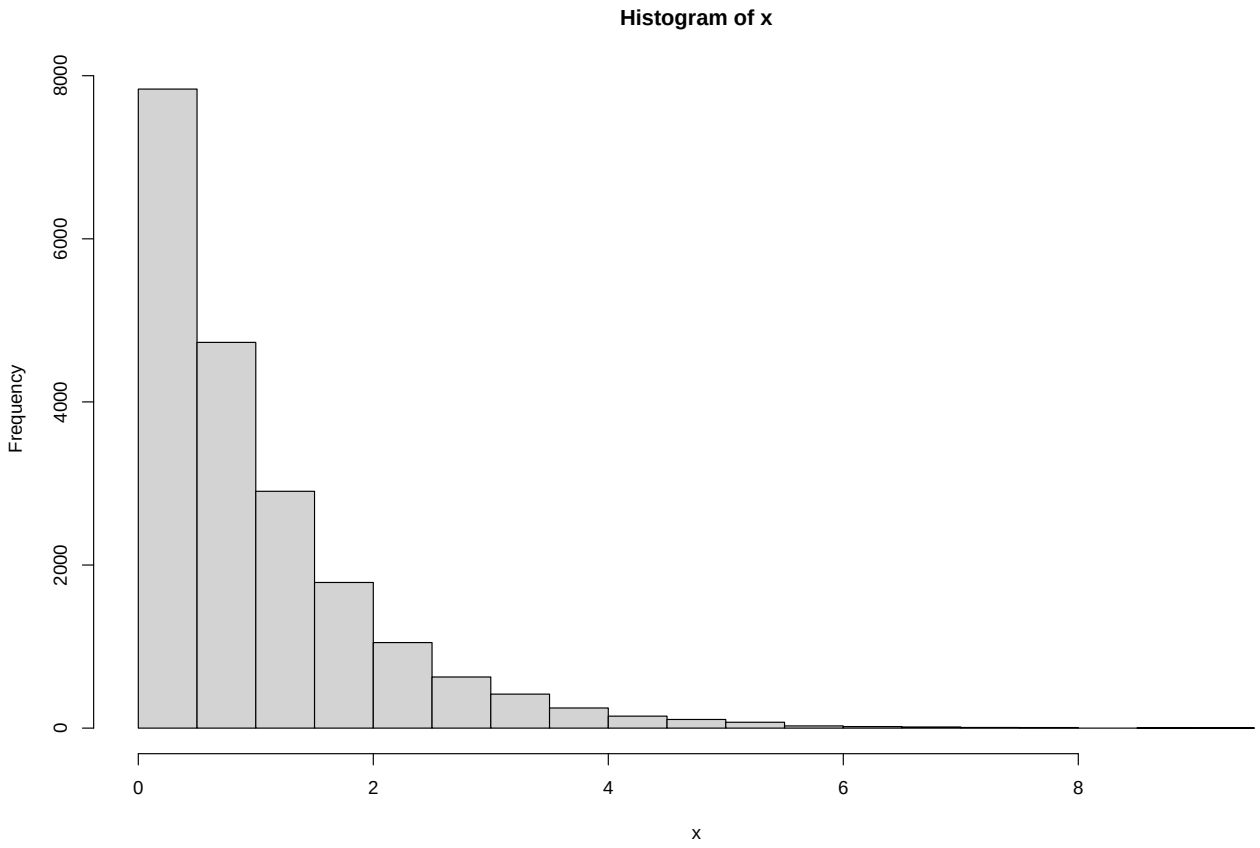
```
path <- ".."
```

```
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/estimate_gev_mixture_model_quantile.R"))
```

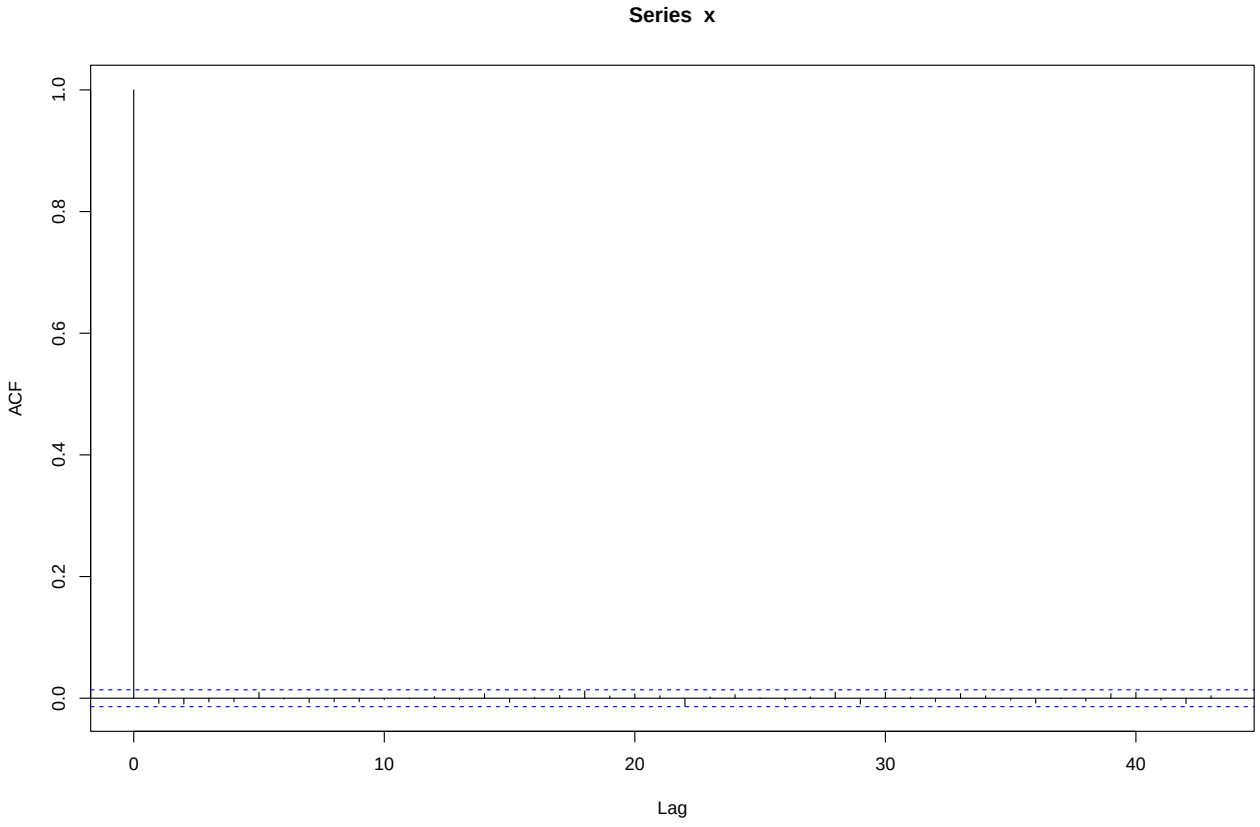
```
n <- 20000
```

```
set.seed(1122)
x <- rexp(n = n)
```

```
hist(x)
```


Histogram of x
Frequency
0
2000
4000
6000
8000
0
2
4
6
8
x

```
acf(x)
```


Series x
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
Lag

```r
nlargest <- 1000

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

```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] 11 12 13 14 15 16 17 18 19 20
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##            loc_star        scale_star           shape_star
## 11 3.29574918040805 0.976445970509164 -0.03667709095381891
## 12 3.52277758042938 0.804214867250803  0.02327082826820088
## 13 3.34734924573158 0.975487676545606 -0.03914751087346882
## 14 3.37760466022731 0.934922331849203 -0.02311039263952277
## 15 3.12226861242757 1.169635359507689 -0.09470560157345080
```

```
## 16 2.93110818059460 1.177531657286269 -0.08031243590085080
## 17 2.65709377754162 1.369077387579498 -0.11932062611815418
## 18 3.75244477098718 0.776090074273978  0.00850247899967913
## 19 3.87855287007820 0.689409149289278  0.04546195765427703
## 20 3.57806473766632 0.970271024691920 -0.06303282380142441
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                            loc_star        scale_star          shape_star
## identic_weights     3.34630136160918 0.984308549878341 -0.0379071216938534
## pessimistic_weights 3.45993502872538 1.024148504273390 -0.0353681535590280
## automatic_weights   3.60700208661585 0.738869635008201  0.0454619576542770
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 0.00255261684711061
##
## $gradient_value
## [1] 9.90838708142538e-06
##
## $function_reduction
## [1] 0.0157651151485194
##
## $number_iterations
## [1] 1789
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.00170124879625866
##
## $gradient_value
## [1] 6.95493901186608e-06
##
## $function_reduction
## [1] 0.0190034725999947
##
## $number_iterations
## [1] 3617
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                11                12                13                14
## 0.000000000000000 0.844460821843412 0.000000000000000 0.000000000000000
```

```
##                15                16                17                18
## 0.000000000000000 0.155539178156588 0.000000000000000 0.000000000000000
##                19                20
## 0.000000000000000 0.000000000000000
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                11                12                13                14
## 0.0999961068269341 0.1061739909730453 0.0997493793384735 0.1013619680046930
##                15                16                17                18
## 0.0943586306897419 0.0957265709739753 0.0920643434883038 0.1046174980960356
##                19                20
## 0.1085564486893274 0.0973950629194701
```

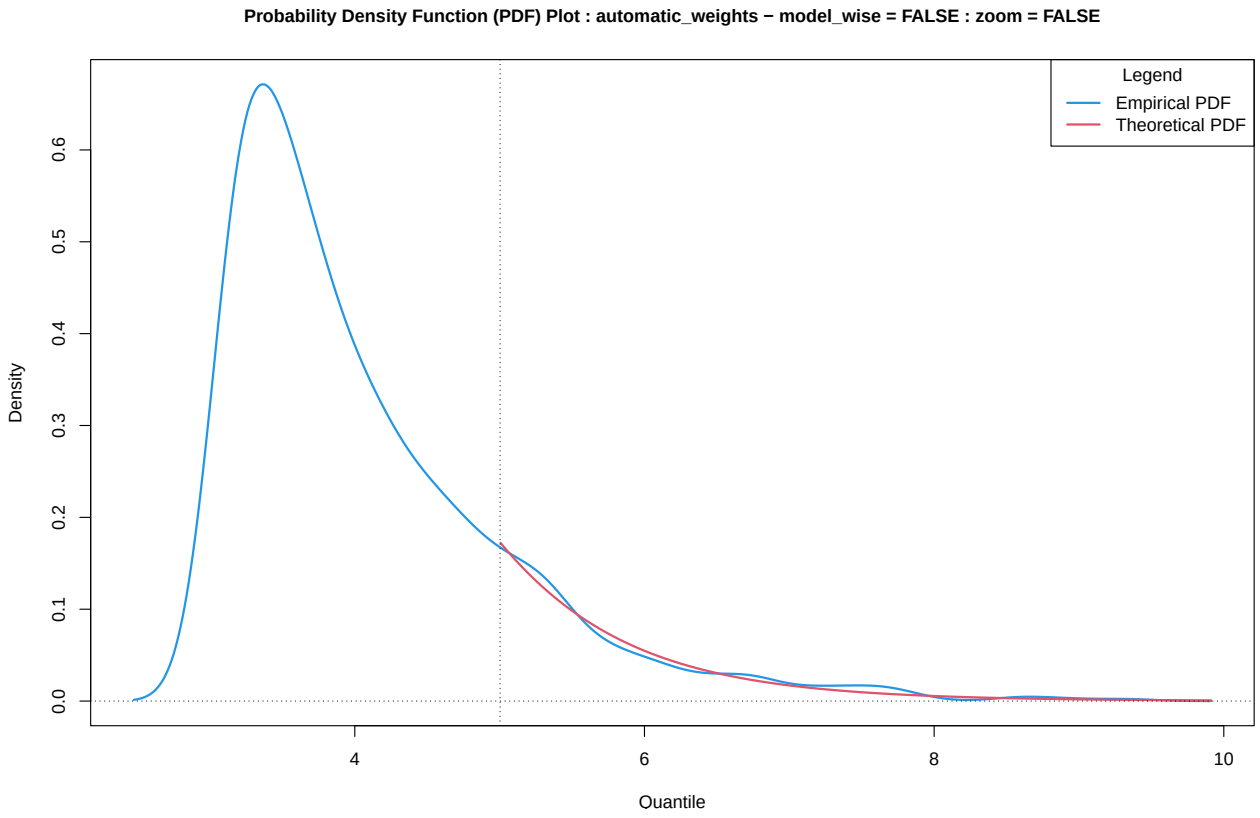
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                11                12                13                14
## 0.0972797133258867 0.0818885656908950 0.0971865354168743 0.0933230222641345
##                15                16                17                18
## 0.1180112383653980 0.1189467790375891 0.1440588488104659 0.0796175523205252
##                19                20
## 0.0730068775772010 0.0966808671910303
```

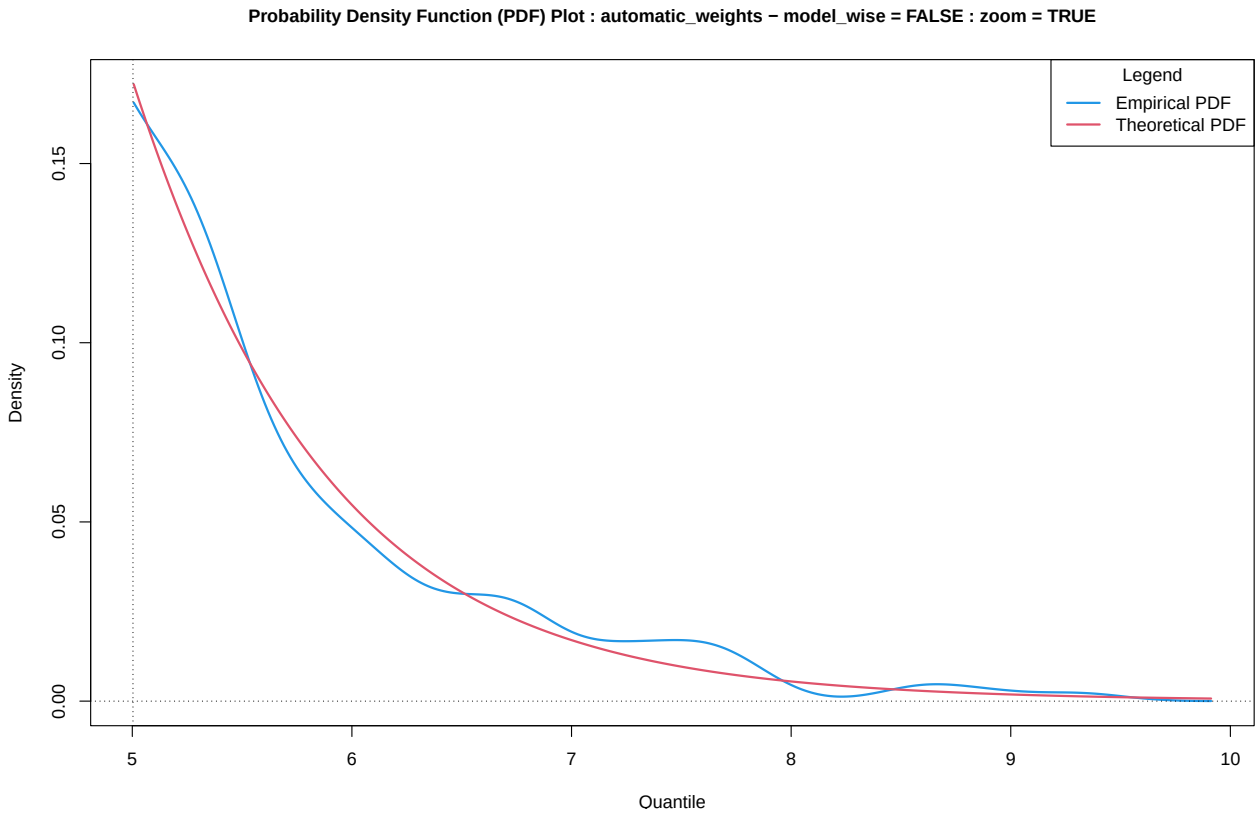
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                11                12                13                14
## 0.0896668094859343 0.1125197862248132 0.0944150746069881 0.0973152942651538
##                15                16                17                18
## 0.0753858895106338 0.0622687493205185 0.0473442454881823 0.1415702802926139
##                19                20
## 0.1605980040547069 0.1189158667504553
```

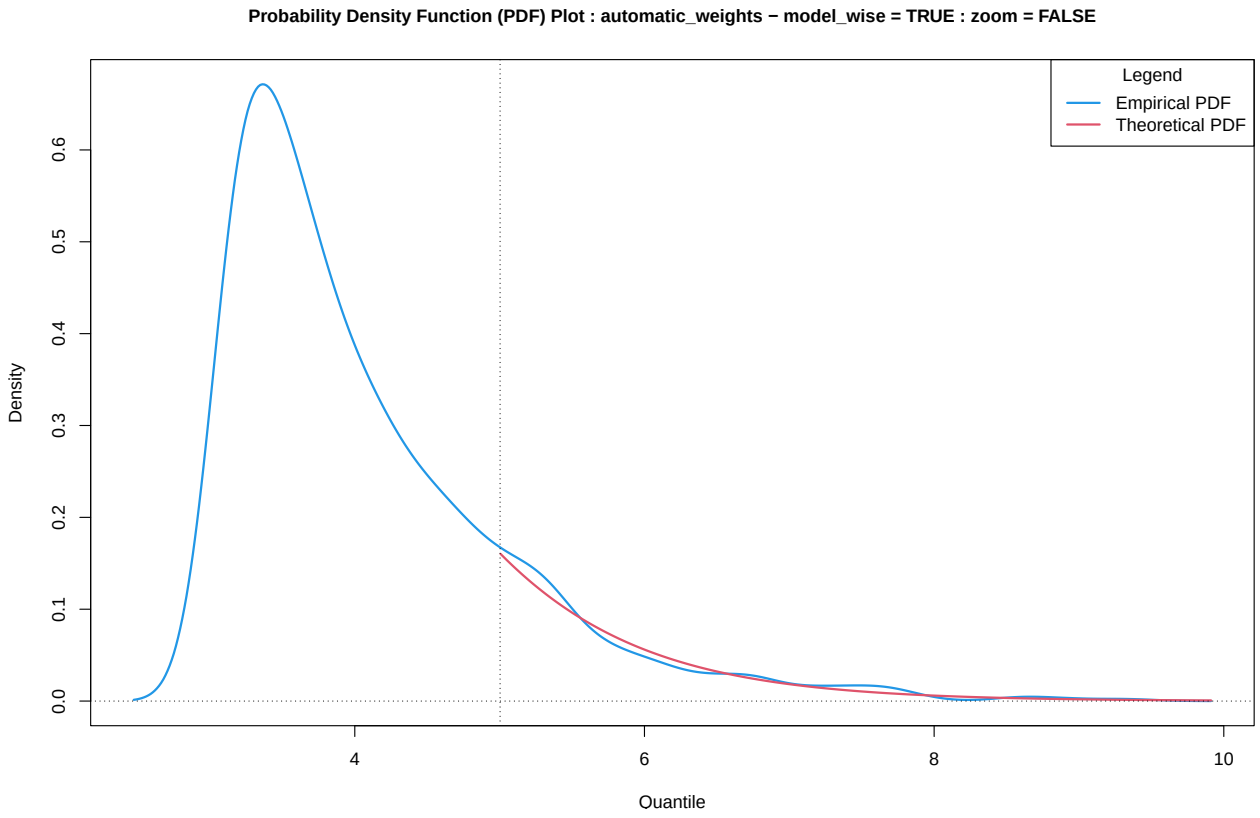
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



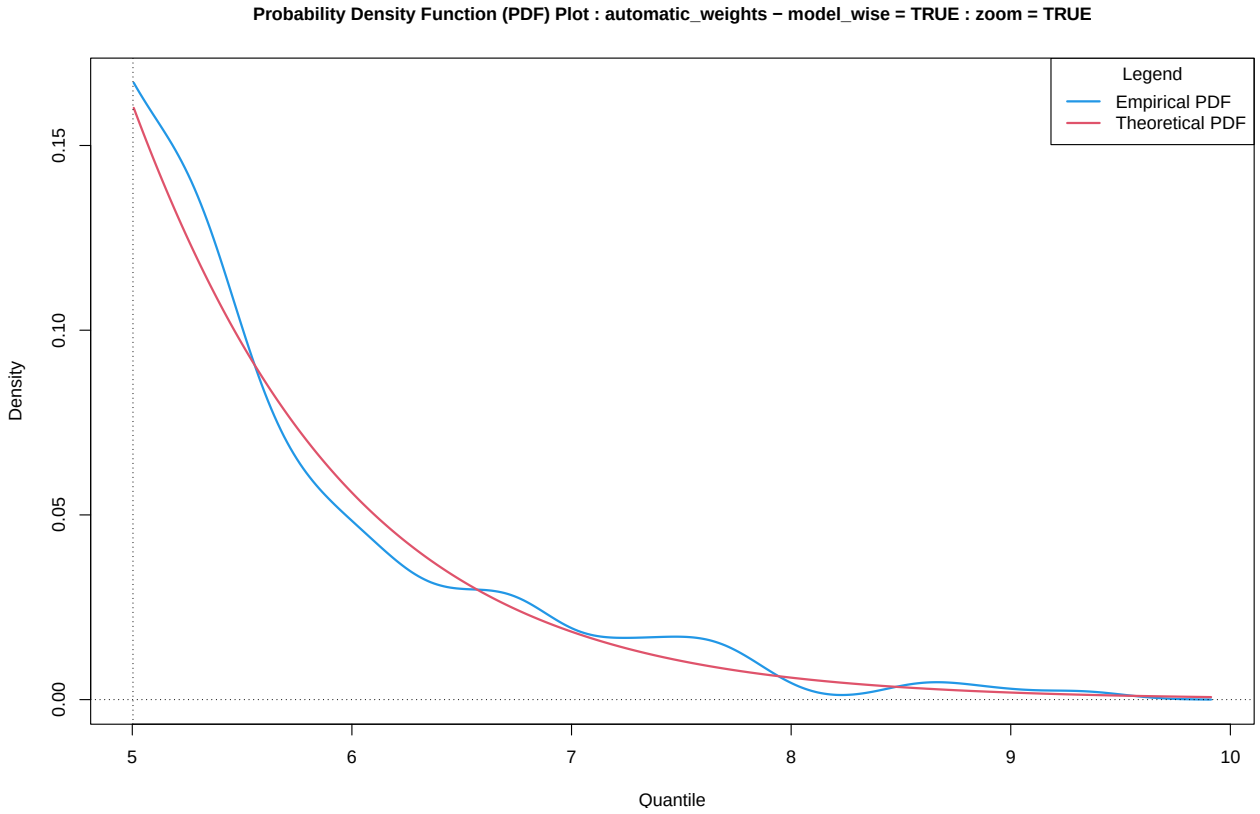

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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 36.9414820328362    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 48.7652905178263    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 9.32637366225866    NA
```

```
true_rl <- qexp(p = 1 - alpha)
true_rl
```

```
## [1] 32.2369908993468
```

```
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
## 11 -5.33725533795543 20.8088856145853 46.9550265671261
## 12 -54.6851376503337 37.2083453260977 129.101828302529
## 13 -7.20051572831085  20.333368467944 47.8672526641989
## 14 -20.7925982801311  23.249751888565 67.2921020572611
## 15  3.25947733267602 14.6973568099349 26.1352362871937
## 16 -1.04892346751356 16.1923500364021 33.4336235403178
## 17  3.47565308886628 13.7806549395963 24.0856567903263
## 18 -55.5777942109169 29.5201084977206 114.618011206358
## 19 -122.562697705713 46.0134907413773 214.589679188468
## 20  -5.0597127061962 16.5338799886942 38.1274726835847
```

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

```
## [1] -122.562697705713  214.589679188468
```

```
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
## 12 -54.6851376503337 37.2083453260977 129.101828302529
## 16 -1.04892346751356 16.1923500364021 33.4336235403178
```

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

```
## [1] -54.6851376503337 129.1018283025291
```

```r
matplot(x = rownames(est_rl_pw),
        y = est_rl_pw,
        xlab = "block size",
        ylab = "quantile",
        main = "Estimates of a quantile",
        ylim = range(c(est_rl_pw_range, true_rl)),
        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 = true_rl, col = 4, lwd = 2)
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)
```