Modeling extreme values with a GEV mixture probability distributions

Pascal Alain Dkengne Sielenou

September 28th, 2023

```
# library(xfun)
```

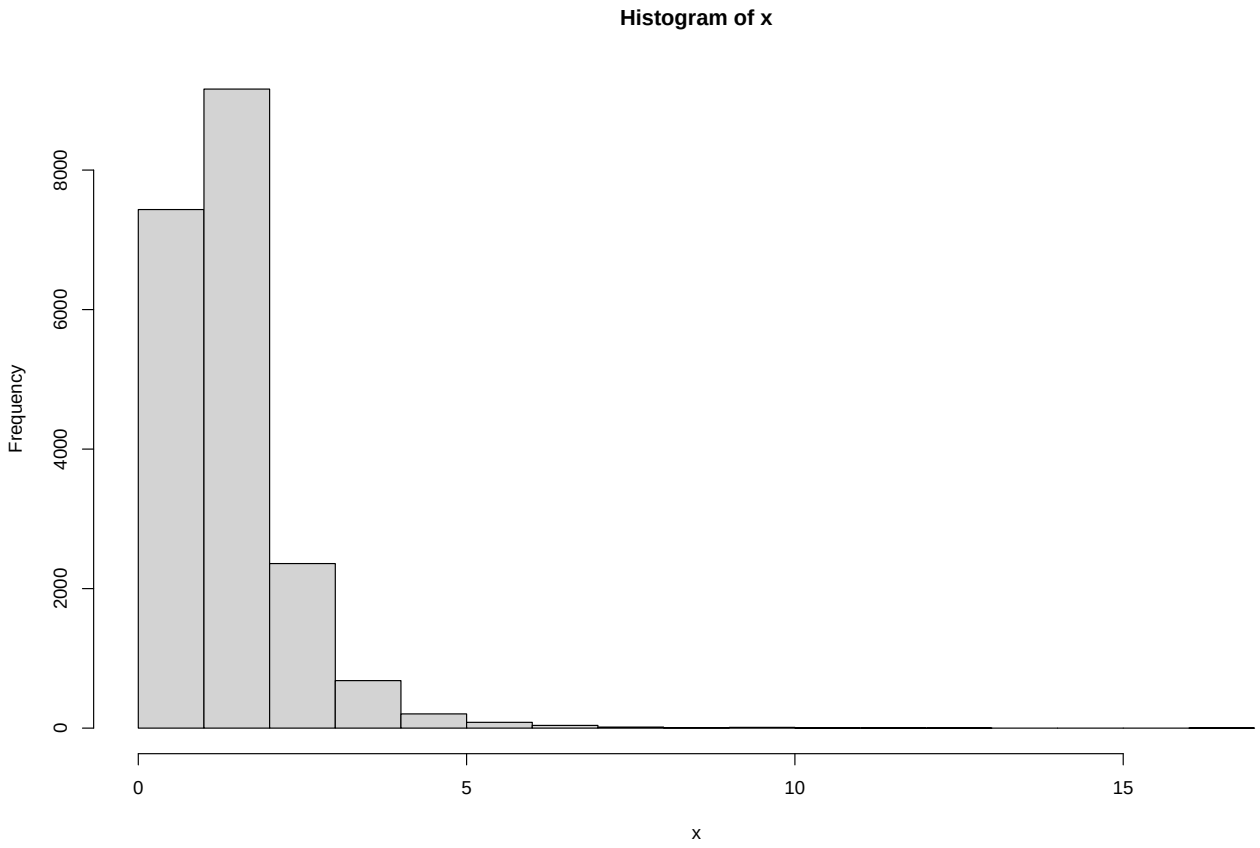
```
path <- ".."
```

```
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/plot_several_standardized_block_maxima_mean.R"))
xfun::in_dir(dir = path, expr = source("./src/estimate_gev_mixture_model_quantile.R"))
```

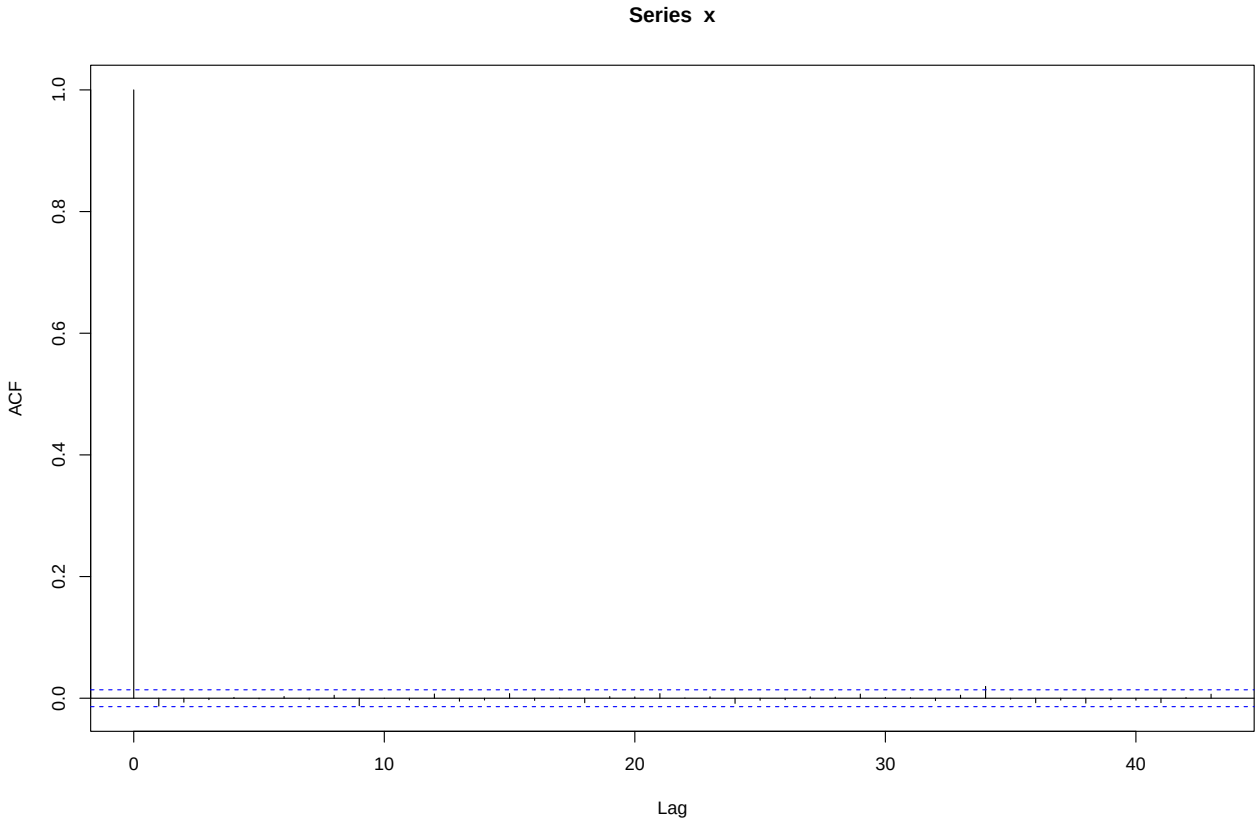
```
n <- 20000
```

```
loc <- 1
scale <- 0.5
shape <- +0.2
set.seed(1122)
x <- generate_gev_sample(n = n, loc = loc, scale = scale, shape = shape)
```

```
hist(x)
```


Histogram of x
Frequency
0
2000
4000
6000
8000
0
5
10
15
x

```r
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] 13 14 15 16 17 18 19 20
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##            loc_star        scale_star         shape_star
## 13 2.69054403952503 1.269396453338834 0.0631980865046416
## 14 3.15756693633157 1.058442164368152 0.1014298547687266
## 15 2.92843477417179 1.120671414751082 0.0892364007649310
## 16 2.88348675521493 1.202048311864730 0.0740391888027813
## 17 2.53264930529822 1.334088288932666 0.0472414448223002
```

```
## 18 3.39945390070561 0.899662955913661 0.1485764968893955
## 19 1.98719280221051 1.601264551990957 0.0118212931082462
## 20 3.44843386195449 0.896750177156583 0.1477941025087067
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                            loc_star       scale_star         shape_star
## identic_weights     2.87847029692652 1.17279053978958 0.0854171085212161
## pessimistic_weights 3.05456592749605 1.22338344229340 0.0873744898284566
## automatic_weights   2.85947661852415 1.14469517725063 0.0785542034032756
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 0.00237017485148604
##
## $gradient_value
## [1] 9.86839071503987e-06
##
## $function_reduction
## [1] 0.0110712356964391
##
## $number_iterations
## [1] 1697
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.0018071308020465
##
## $gradient_value
## [1] 3.44267217272931e-05
##
## $function_reduction
## [1] 0.011951276748745 9
##
## $number_iterations
## [1] 480
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                13                14                15                16
## 0.000000000000000 0.000000000000000 0.749015537152418 0.000000000000000
##                17                18                19                20
## 0.000000000000000 0.000000000000000 0.250984462847581 0.000000000000000
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                13                14                15                16
## 0.122133682967109 0.126893477557559 0.125355602832204 0.123464949880029
##                17                18                19                20
## 0.120200305676956 0.133019351446370 0.116017311083817 0.132915318555957
```

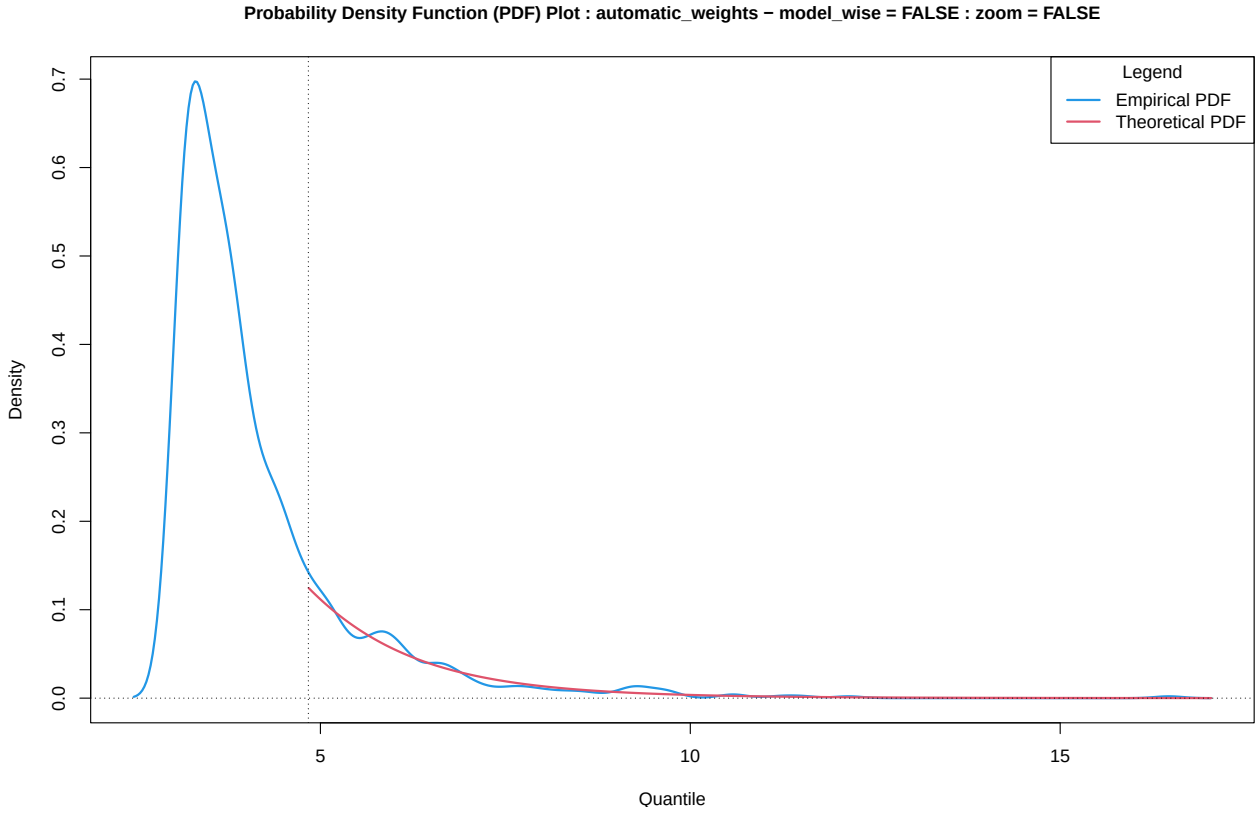
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                 13                 14                 15                 16
## 0.1342874302146783 0.1087474493402086 0.1157297192247521 0.1255412466131191
##                 17                 18                 19                 20
## 0.1432618879971623 0.0927816614794751 0.1871388028928032 0.0925118022378013
```

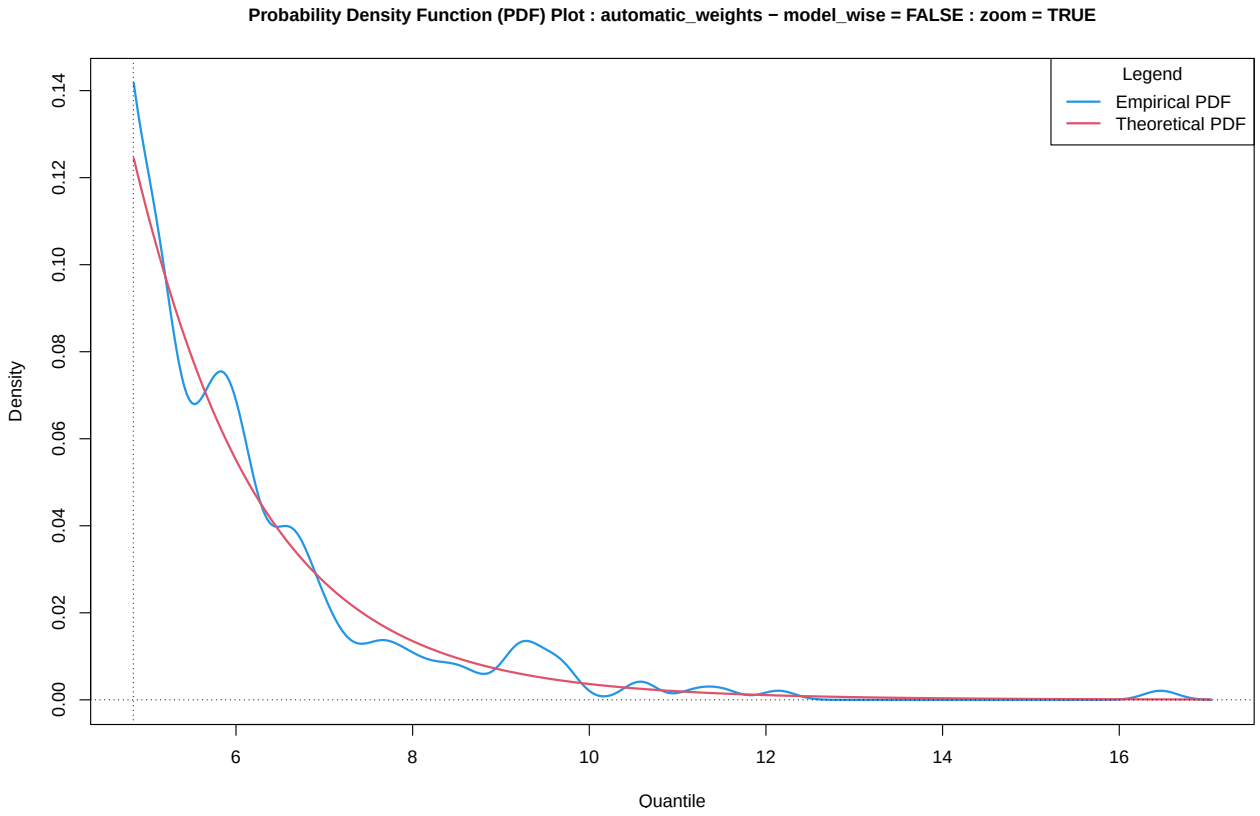
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                 13                 14                 15                 16
## 0.0944195285450295 0.1506216131594574 0.1197778355647250 0.1145132615433783
##                 17                 18                 19                 20
## 0.0806285808449421 0.1918392502597717 0.0467304826257412 0.2014694474569548
```

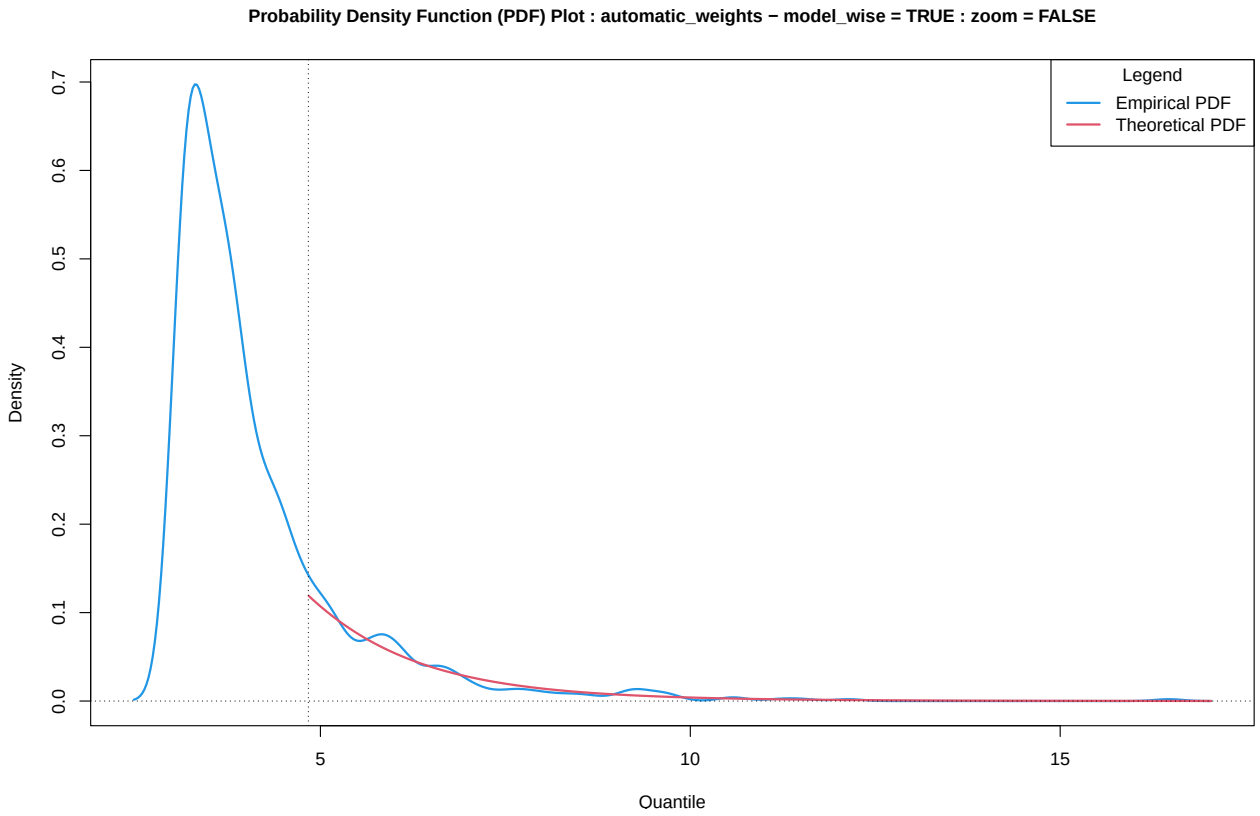
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



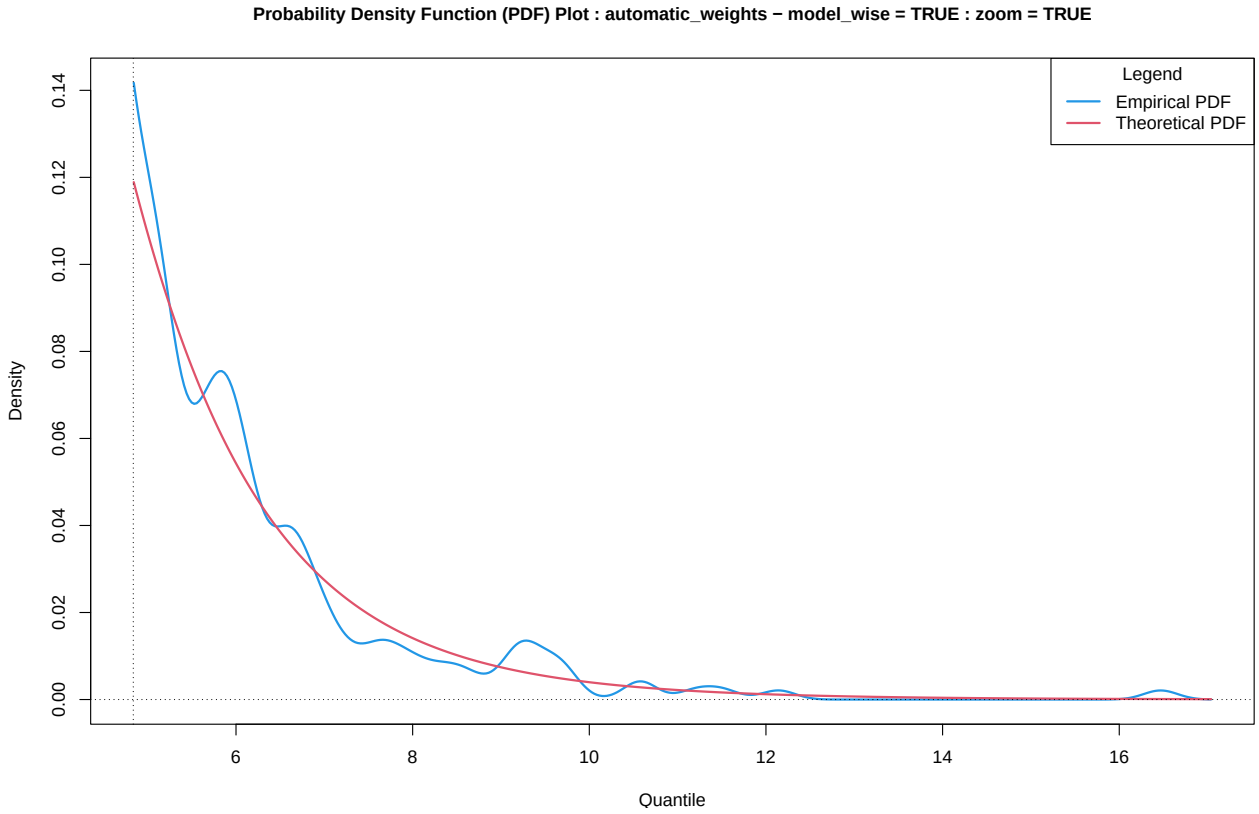
```r
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                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




```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")
```




```
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 156.68819525798    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 133.193387409948    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 16.4691752920578    NA
```

```
true_rl <- calculate_gev_inverse_cdf(p = 1 - alpha, loc = loc, scale = scale, shape = shape)
true_rl
```

```
## [1] 1576.14563730748
```

```
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
## 13 -176.122885626966 110.086972186096 396.296829999158
## 14 -414.404207150067 195.288131291723 804.980469733513
## 15 -330.378479257648 161.031135269381  652.44074979641
## 16 -254.787915920599 128.121775051664 511.031466023927
## 17 -125.646677385188 86.6903961244531 299.027469634095
## 18 -1628.57691372336 464.633990295639 2557.84489431464
## 19 -68.4291111685574 57.9188932873447 184.266897743247
## 20  -1669.4440979172 455.699723461769 2580.84354484074
```

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

```
## [1] -1669.44409791720  2580.84354484074
```

```
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
## 15 -330.378479257648 161.031135269381  652.44074979641
## 19 -68.4291111685574 57.9188932873447 184.266897743247
```

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

```
## [1] -330.378479257648  652.440749796410
```

```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)
```