Modeling extreme values with a GEV mixture probability distributions

Pascal Alain Dkengne Sielenou

September 28th, 2023

```
# library(xfun)
```

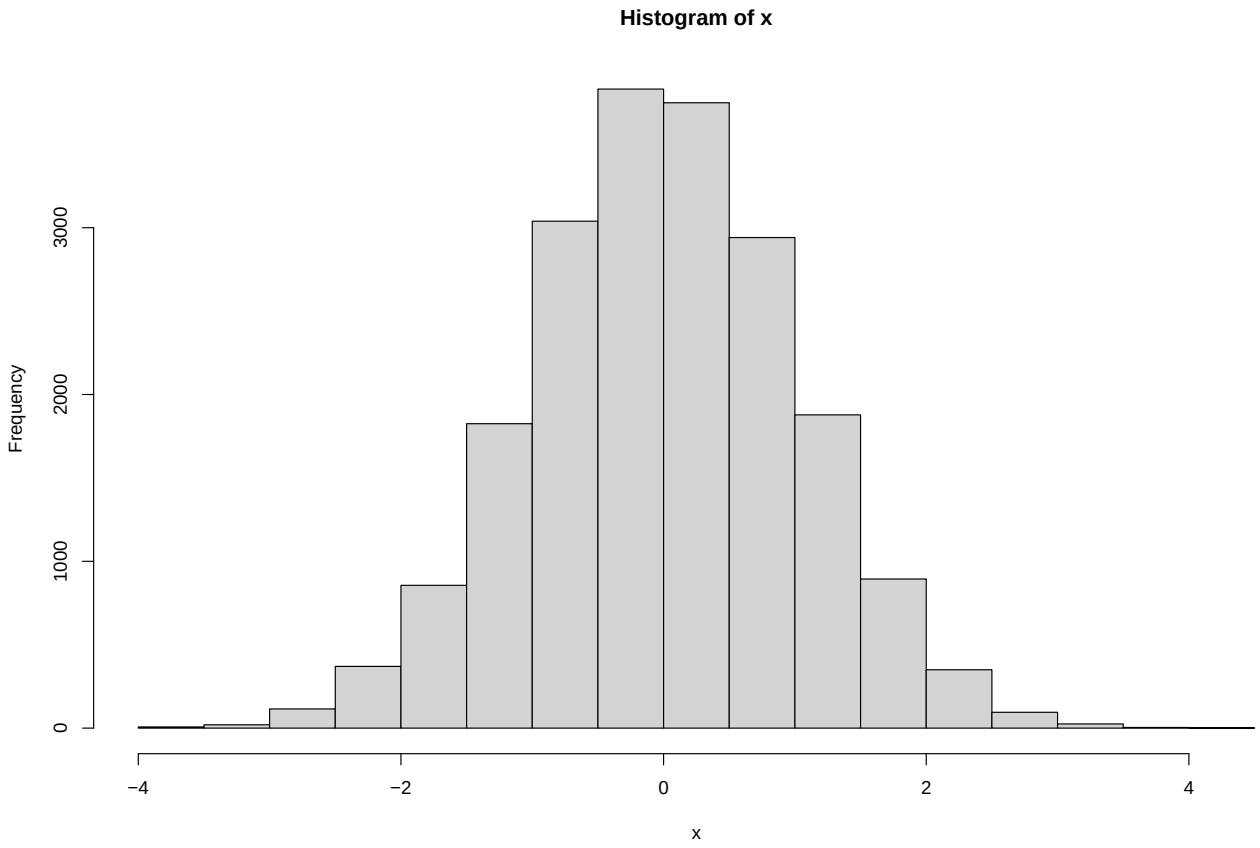
```
path <- ".."
```

```
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"))
```

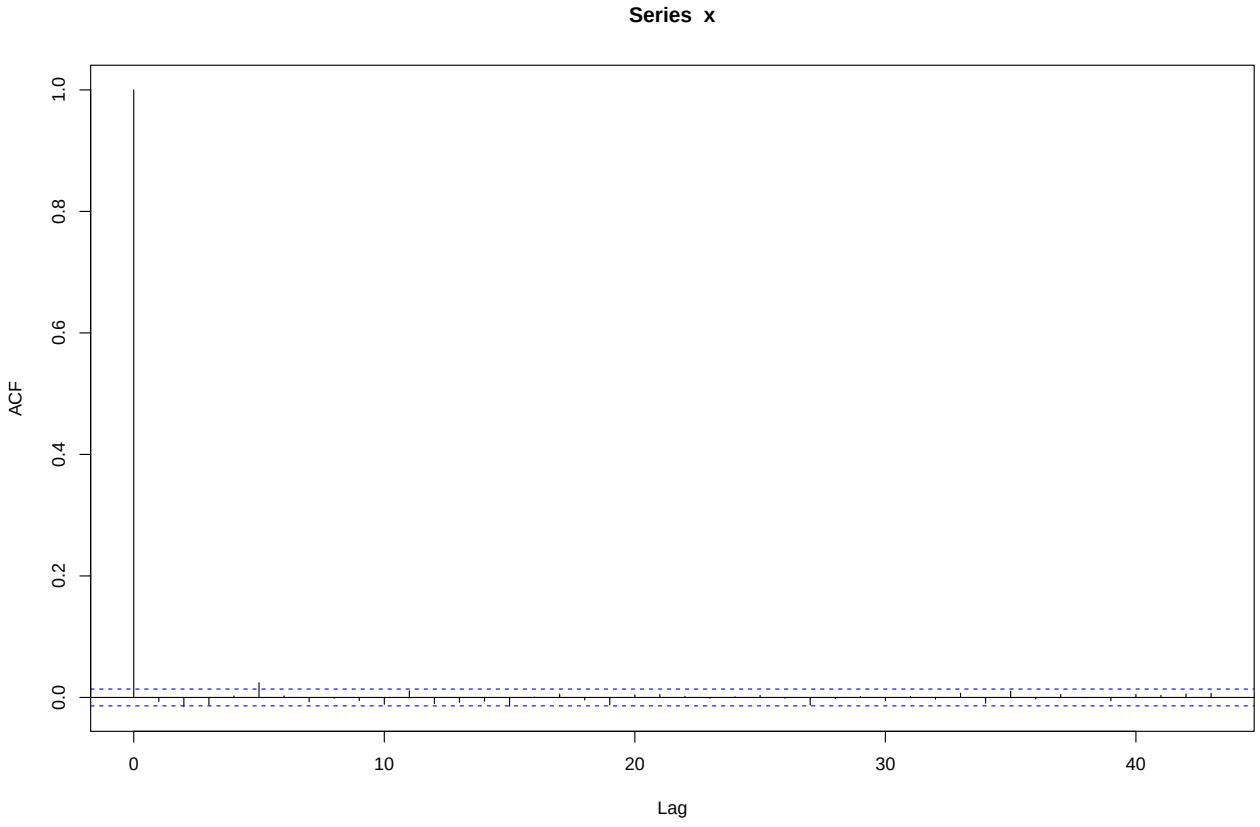
```
n <- 20000
```

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

```
hist(x)
```


Histogram of x
Frequency
0
1000
2000
3000
−4
−2
0
2
4
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]  9 10 11 12 13 14 15 16 17 18 19 20
```

```r
gev_mixture_model$normalized_gev_parameters_object
```

```
##            loc_star        scale_star           shape_star
## 9  1.84562202024446 0.320697665498849 -0.00108309532027692
## 10 1.83756777761176 0.331458485445656 -0.00627101236012475
## 11 1.80875035231887 0.346351183960770 -0.02088954665989408
## 12 1.72211697636964 0.417675469856155 -0.07445443503712171
## 13 1.95902989459704 0.271204757056876  0.03348220912887317
```

```
## 14 1.74623071631591 0.410521693896988 -0.07346327545129022
## 15 1.90790642921463 0.315001385835597 -0.01204999168988620
## 16 1.80296667255663 0.380320612667436 -0.05698063065925631
## 17 1.93664593401627 0.296584284504634  0.00561669223118209
## 18 1.90927122937761 0.317187130592407 -0.01476009599078024
## 19 1.88571229091319 0.301656011085101  0.00465175527901195
## 20 1.87587367386084 0.333090708297259 -0.02554801536941992
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                             loc_star        scale_star          shape_star
## identic_weights     1.85314116394974 0.336812449058144 -0.0201457868249153
## pessimistic_weights 1.85801544005142 0.338693505354884 -0.0191541237366313
## automatic_weights   1.73699632937740 0.393920263852699 -0.0504962812627854
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 0.00186622918270171
##
## $gradient_value
## [1] 9.99010397134681e-06
##
## $function_reduction
## [1] 0.0227711426797029
##
## $number_iterations
## [1] 1879
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.000833577257819185
##
## $gradient_value
## [1] 8.55174365929634e-06
##
## $function_reduction
## [1] 0.023795114276474 8
##
## $number_iterations
## [1] 3143
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```r
gev_mixture_model$automatic_weights_mw
```

```
##                  9                 10                 11                 12
## 0.000000000000000 0.000000000000000 0.697882917819512 0.302117082180488
##                 13                 14                 15                 16
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                 17                 18                 19                 20
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
```

```r
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                  9                 10                 11                 12
## 0.0848949251032901 0.0844556377531802 0.0832300004529702 0.0788890927478254
##                 13                 14                 15                 16
## 0.0878806479831745 0.0789673231913425 0.0839689779212754 0.0802796995344303
##                 17                 18                 19                 20
## 0.0854656126737091 0.0837417213167136 0.0853831835217636 0.0828431778003249
```

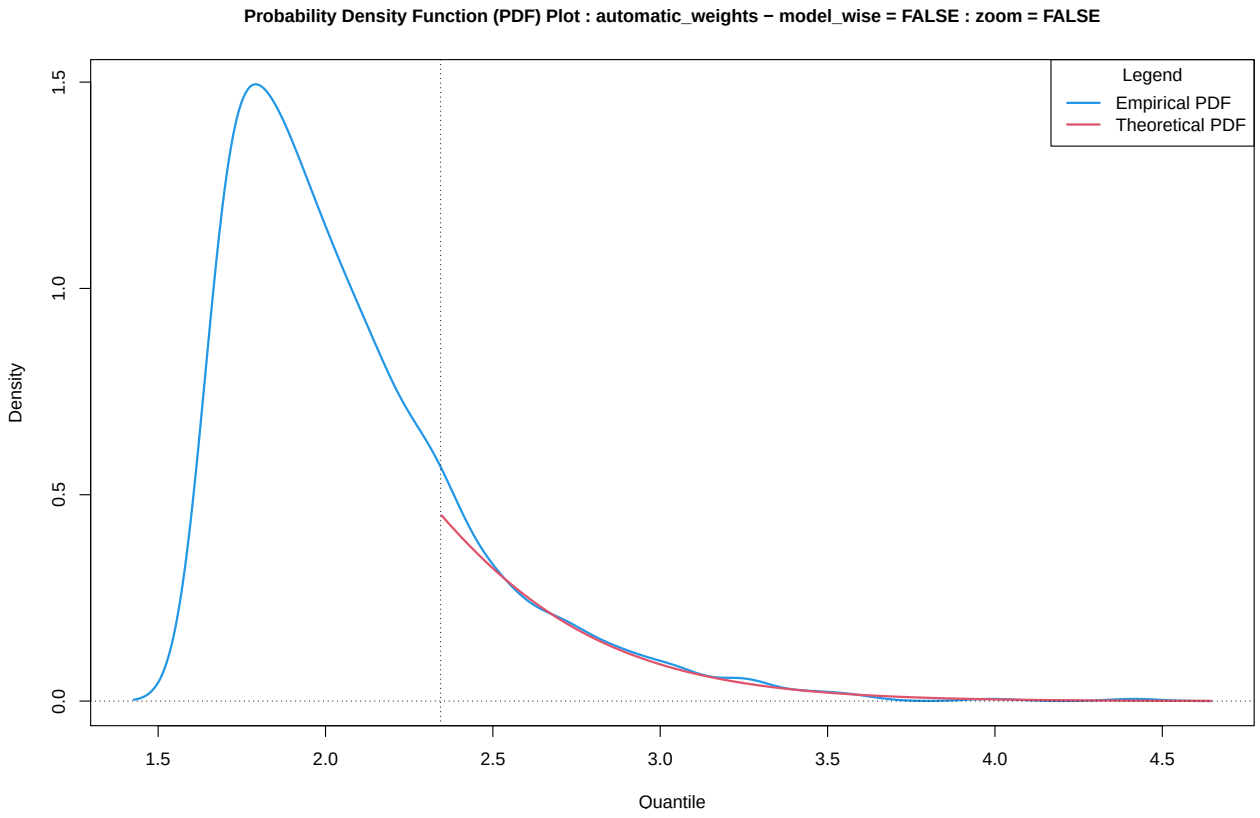
```r
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                  9                 10                 11                 12
## 0.0819244420604935 0.0828107765207590 0.0840532816113730 0.0902672927746558
##                 13                 14                 15                 16
## 0.0779684670023709 0.0896238450701800 0.0814591041330052 0.0869575728540069
##                 17                 18                 19                 20
## 0.0799725941772586 0.0816373476692328 0.0803792235938428 0.0829460525328213
```

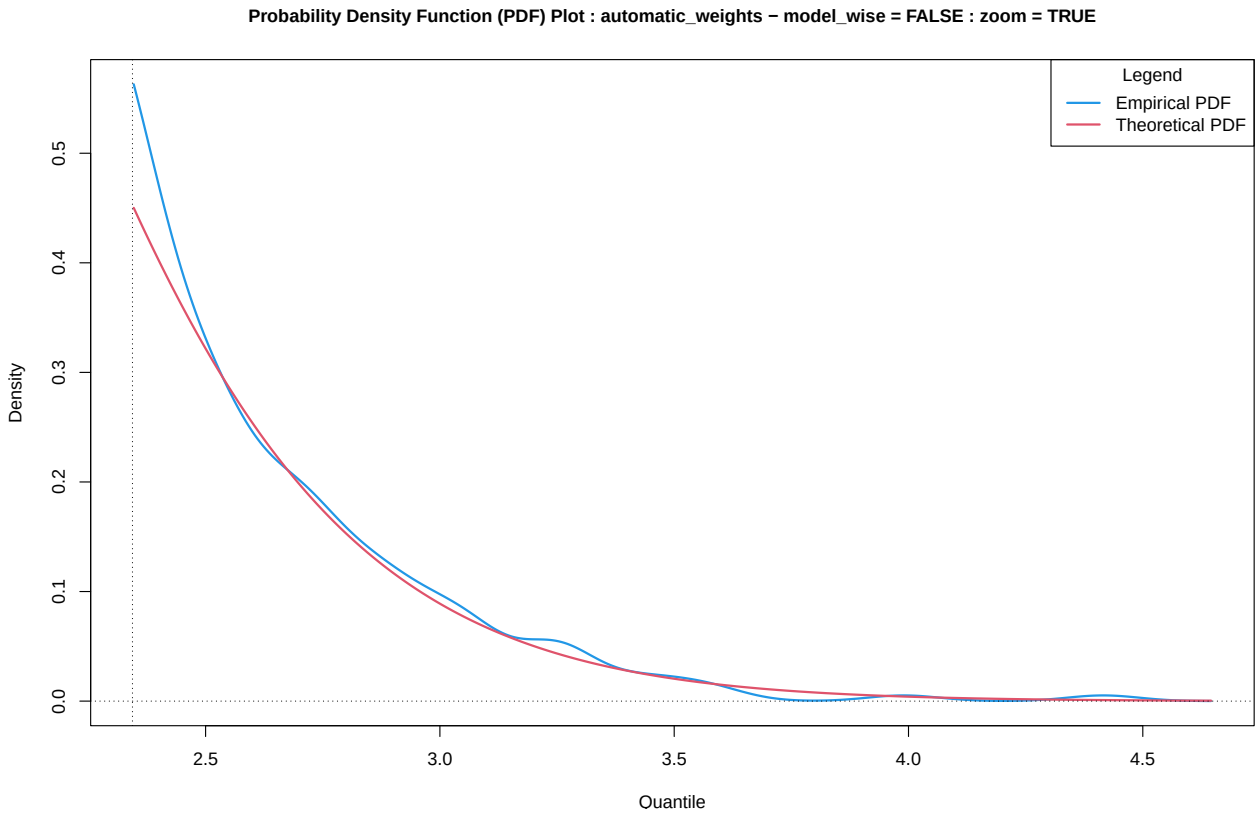
```r
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                  9                 10                 11                 12
## 0.0825068141696351 0.0818449532416649 0.0795200521374202 0.0729209401082058
##                 13                 14                 15                 16
## 0.0924149498319106 0.0747007088957586 0.0878091136085026 0.0790614610714728
##                 17                 18                 19                 20
## 0.0903693173280154 0.0879290373184012 0.0858817333836440 0.0850409189053688
```

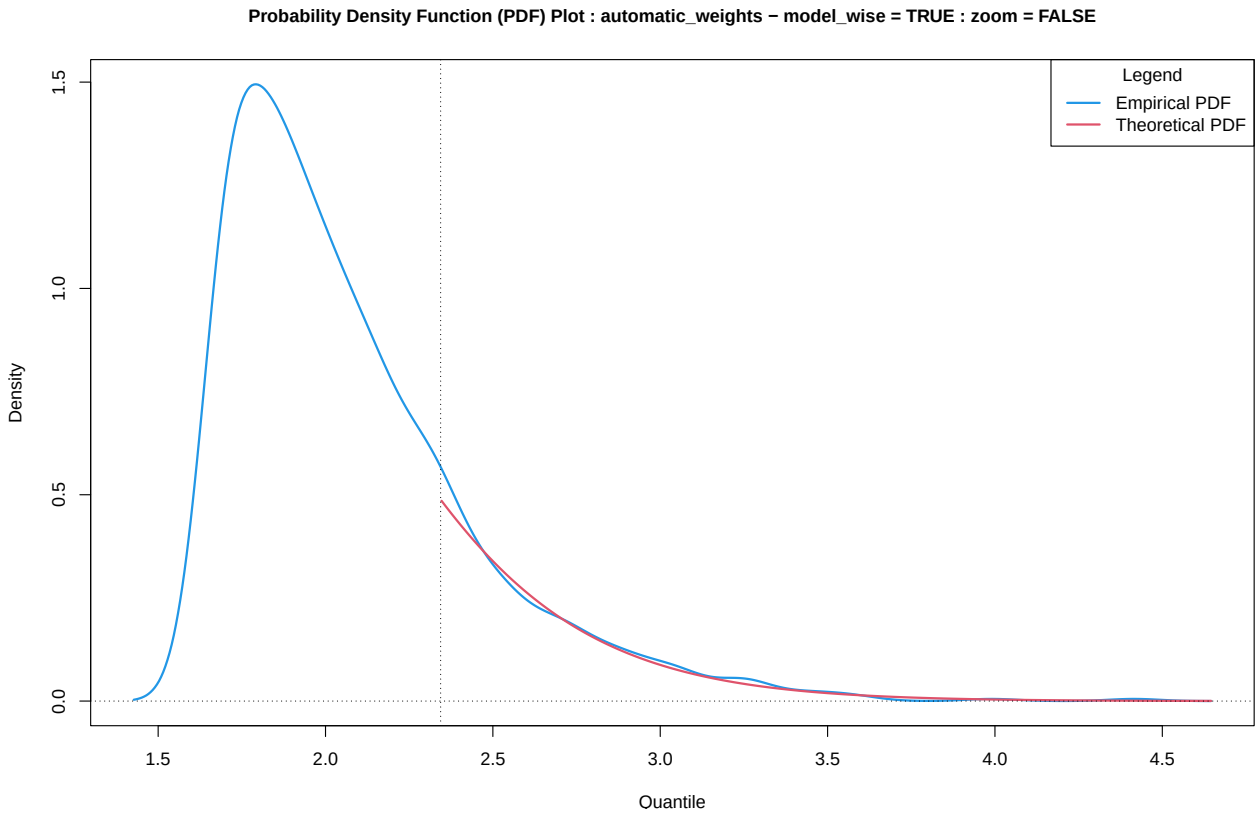
```r
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
               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



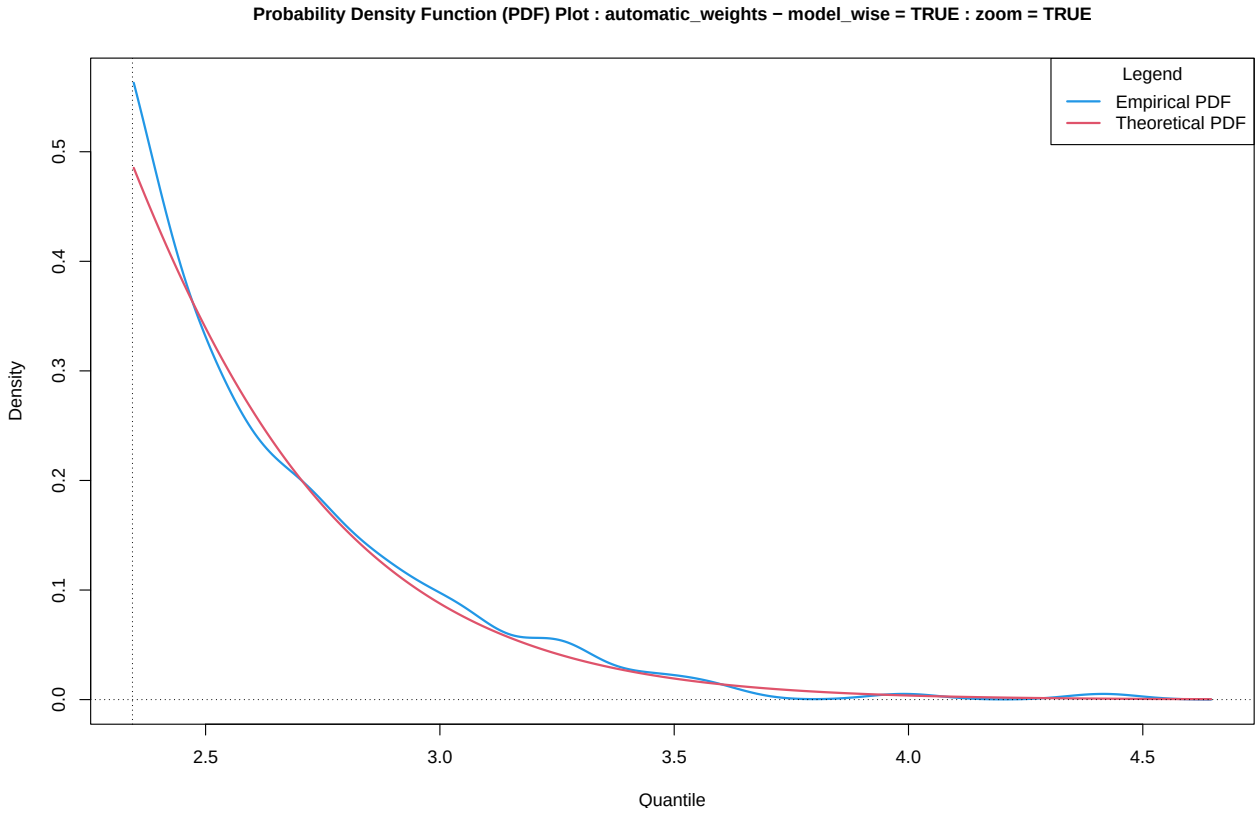

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                           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 9.31952393550542    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 7.75603314109019    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 4.41549649951448    NA
```

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

```
## [1] 7.65073090515564
```

```
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
## 9  -4.80373058808842 11.0761369507686 26.9560044896256
## 10   -5.638565949889 10.6925323455795 27.0236306410479
## 11 -2.73828358859176 9.38717209640206 21.5126277813959
## 12  2.21417786063728 6.69723385856345 11.1802898564896
## 13 -19.1948210254486 15.4205818274004 50.0359846802494
## 14   2.0585661431743 6.68280963766255 11.3070531321508
## 15 -4.28943365652196 9.67078540808062 23.6310044726832
## 16  1.06179849287471 7.21640517439165 13.3710118559086
## 17 -9.97168411646543 11.3617438026365 32.6951717217384
## 18 -4.13937212473434 9.44193051892511 23.0232331625846
## 19 -9.96106648302088 11.3343684101284 32.6298033032777
## 20 -2.93336123575422 8.73670284148468 20.4067669187236
```

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

```
## [1] -19.1948210254486  50.0359846802494
```

```
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
```

```
## 11 -2.73828358859176 9.38717209640206 21.5126277813959
## 12  2.21417786063728 6.69723385856345 11.1802898564896
```

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

```
## [1] -2.73828358859176 21.51262778139589
```

```
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)
```