Modeling extreme values with a GEV mixture probability distributions

Application to a rain data

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"))
```

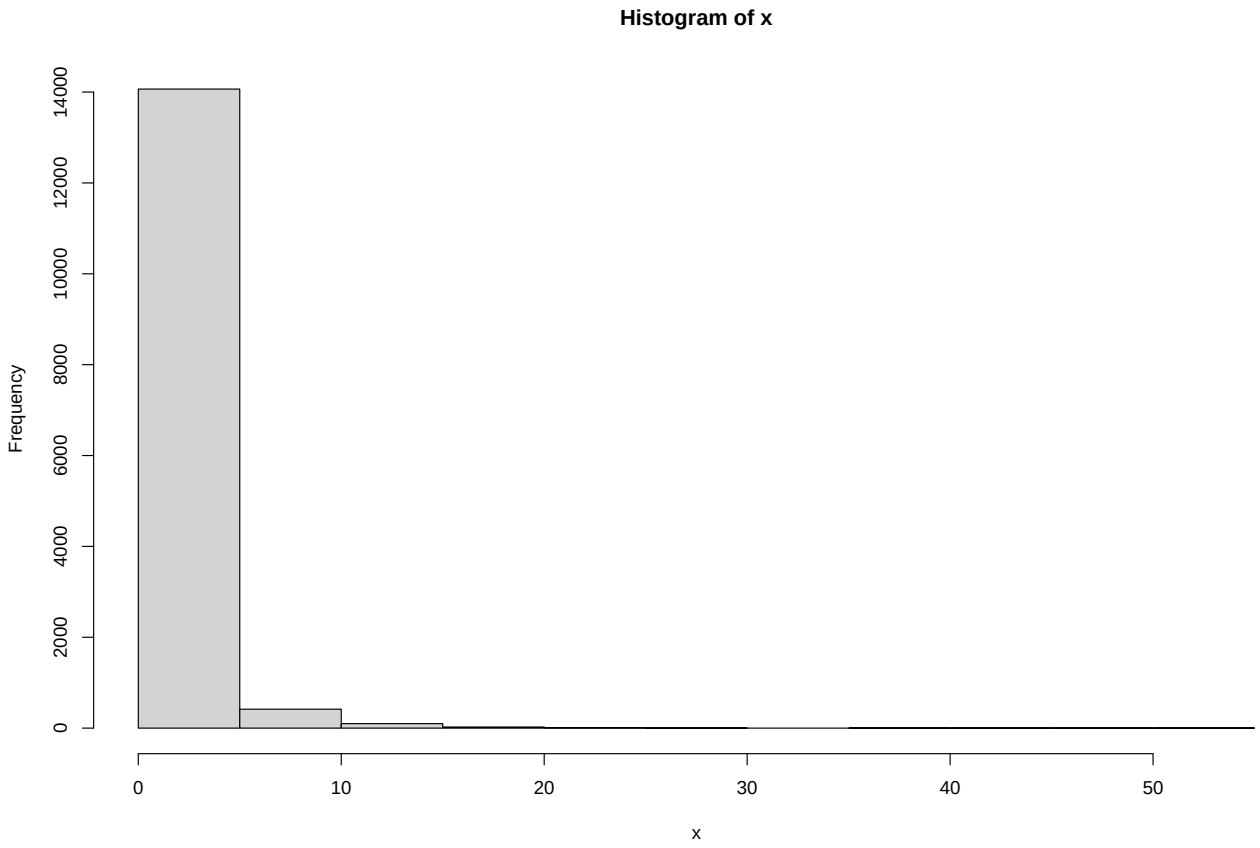
```
library(readr)
```

```
pluie <- xfun::in_dir(dir = path, expr = read_csv("./applications/pluie.csv"))
```

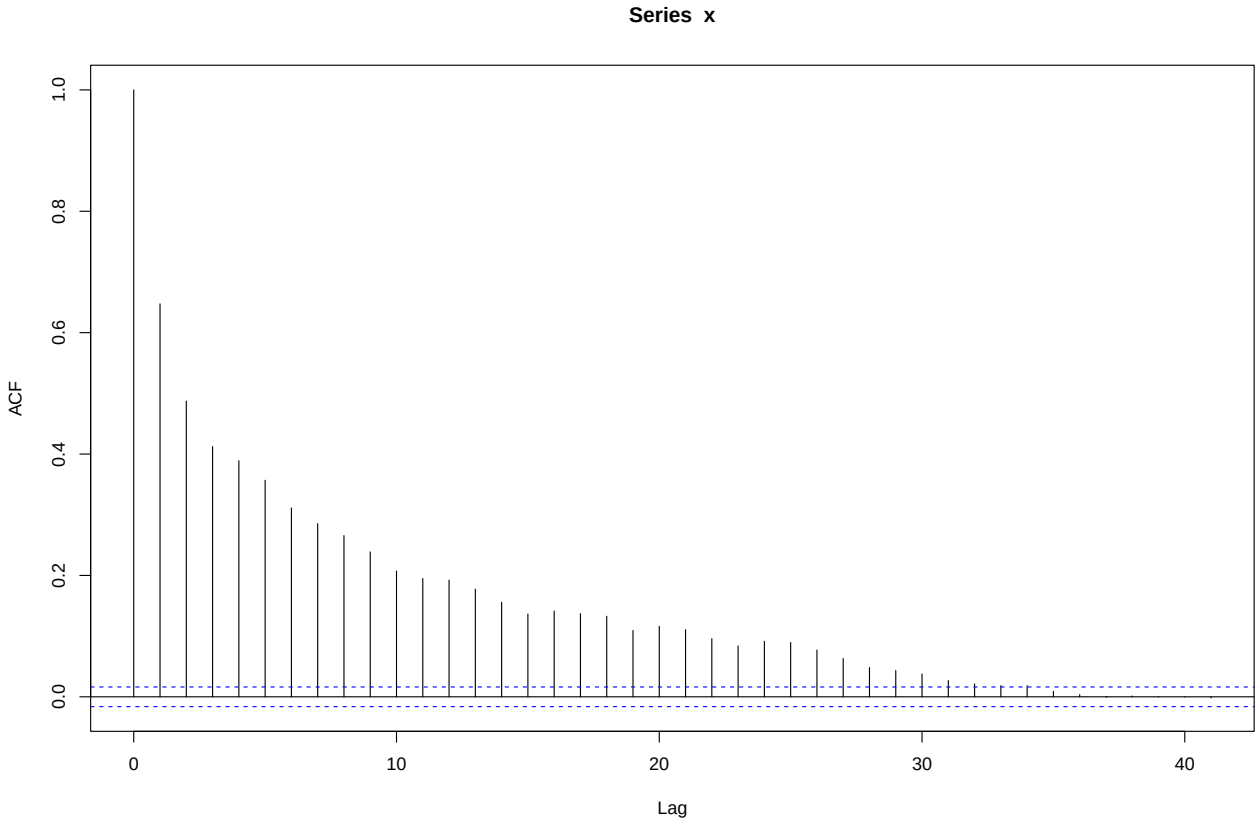
```
## Rows: 14623 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): x
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
x <- pluie$x
x <- x[!is.na(x)]
```

```
hist(x)
```


Histogram of x
Frequency
0
2000
4000
6000
8000
10000
12000
14000
0
10
20
30
40
50
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
n <- length(x)
n
```

```
## [1] 14623
```

```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 = min(y),
                                                           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]  4  5  6  7  8  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
## 4  4.18322505020593 1.46348036116415 0.406157236370832
```

```
## 5  3.74497497382970 1.66176975901827 0.369286610600768
## 6  3.67929321413234 1.55792809828217 0.364605955292967
## 7  3.53372938774498 1.64362681811017 0.345935009853569
## 8  3.80357909816534 1.33528049324323 0.421751991033149
## 9  2.69133502806732 1.96454915417608 0.286322445628405
## 10 3.11374436622655 1.70125970871134 0.339497311662390
## 11 3.58309043320250 1.50495431333029 0.369635533291584
## 12 2.90978609260982 1.79009924877999 0.297673225275297
## 13 2.64180322736059 1.95833243861661 0.279748317573393
## 14 2.29259794098438 1.89886642257077 0.297062713713395
## 15 1.99771987115931 2.07485328281770 0.268263878883505
## 16 2.80784688057857 1.72509589340007 0.317152665315506
## 17 2.29665791876580 2.03598596896766 0.275893993472597
## 18 1.79854684225651 2.05130920761442 0.270622058491534
## 19 2.30891140615748 2.03978441598490 0.265272474084615
## 20 2.70804531344522 1.78095748352167 0.302955571366602
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                            loc_star       scale_star        shape_star
## identic_weights     2.94675806146426 1.77577253342997 0.322225705406477
## pessimistic_weights 3.39174973004701 1.82315043049293 0.324597340240921
## automatic_weights   4.18322505020593 2.07485328281770 0.413301924352270
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 3.91757684019216
##
## $gradient_value
## [1] 6.03961325396085e-14
##
## $function_reduction
## [1] 6.63975814472612
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.18116529184934
##
## $gradient_value
## [1] 3.02043459687873e-05
##
## $function_reduction
## [1] 10.742595139697
##
## $number_iterations
```

```
## [1] 1589
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                    4                    5                    6
## 1.00000000000000e+00 6.75015598972095e-14 0.00000000000000e+00
##                    7                    8                    9
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   10                   11                   12
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   13                   14                   15
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   16                   17                   18
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##                   19                   20
## 0.00000000000000e+00 0.00000000000000e+00
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                  4                  5                  6                  7
## 0.0638983181742725 0.0615852513425797 0.0612976655789646 0.0601637983351490
##                  8                  9                 10                 11
## 0.0649026072371018 0.0566820878927754 0.0597777260013776 0.0616067435835400
##                 12                 13                 14                 15
## 0.0573291391030572 0.0563106747848884 0.0572941496826106 0.0556676775908460
##                 16                 17                 18                 19
## 0.0584568263305871 0.0560940529275302 0.0557991068789940 0.0555014018559665
##                 20
## 0.0576327726997593
```

```
gev_mixture_model$pessimistic_weights_pw_scale
```

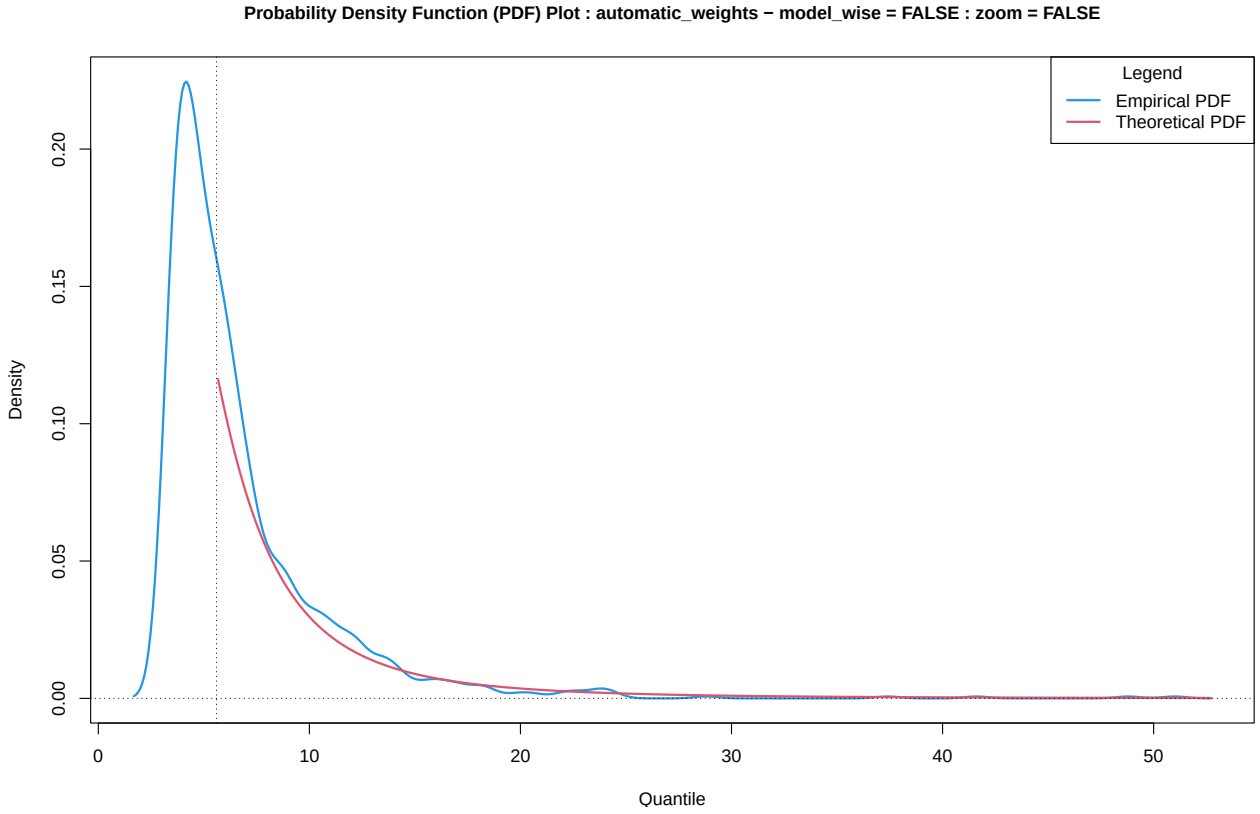
```
##                  4                  5                  6                  7
## 0.0420227067320397 0.0512389254825823 0.0461851281978549 0.0503176829795323
##                  8                  9                 10                 11
## 0.0369664313768048 0.0693578199945891 0.0533028316400945 0.0438022007185267
##                 12                 13                 14                 15
## 0.0582549458585966 0.0689279796378306 0.0649485988461184 0.0774461634899959
##                 16                 17                 18                 19
## 0.0545886311789990 0.0744937862951433 0.0756440628340556 0.0747772850821950
##                 20
## 0.0577248196550414
```

```
gev_mixture_model$pessimistic_weights_pw_loc
```

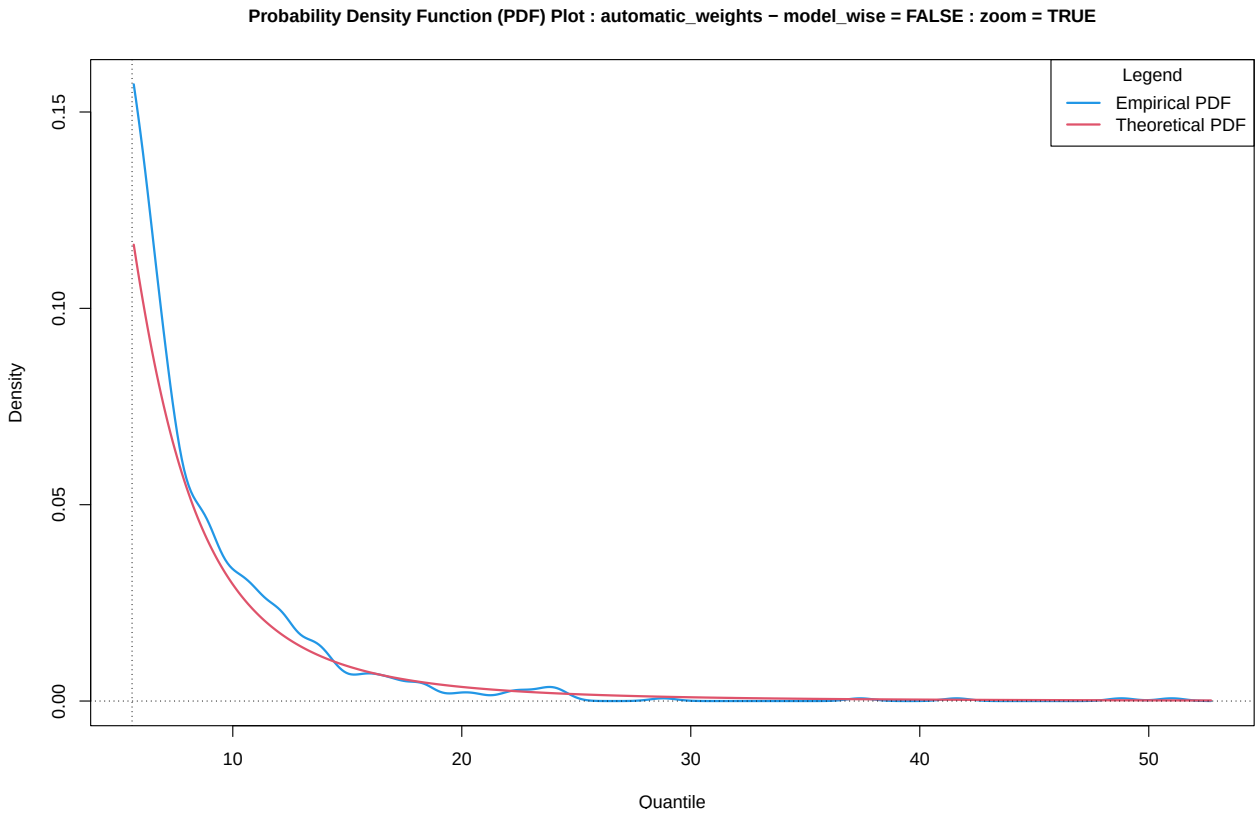
```
##                  4                  5                  6                  7
## 0.1610670866859514 0.1039147539334635 0.0973087710580927 0.0841268099906244
##                  8                  9                 10                 11
## 0.1101865693036741 0.0362315738160773 0.0552760809110835 0.0883835926297549
##                 12                 13                 14                 15
## 0.0450774448891938 0.0344806791607621 0.0243174416551340 0.0181073111316253
```

```
##                  16                  17                  18                  19
## 0.0407087393750328 0.0244163706167572 0.0148372774065199 0.0247173968503090
##                  20
## 0.0368421005859441
```

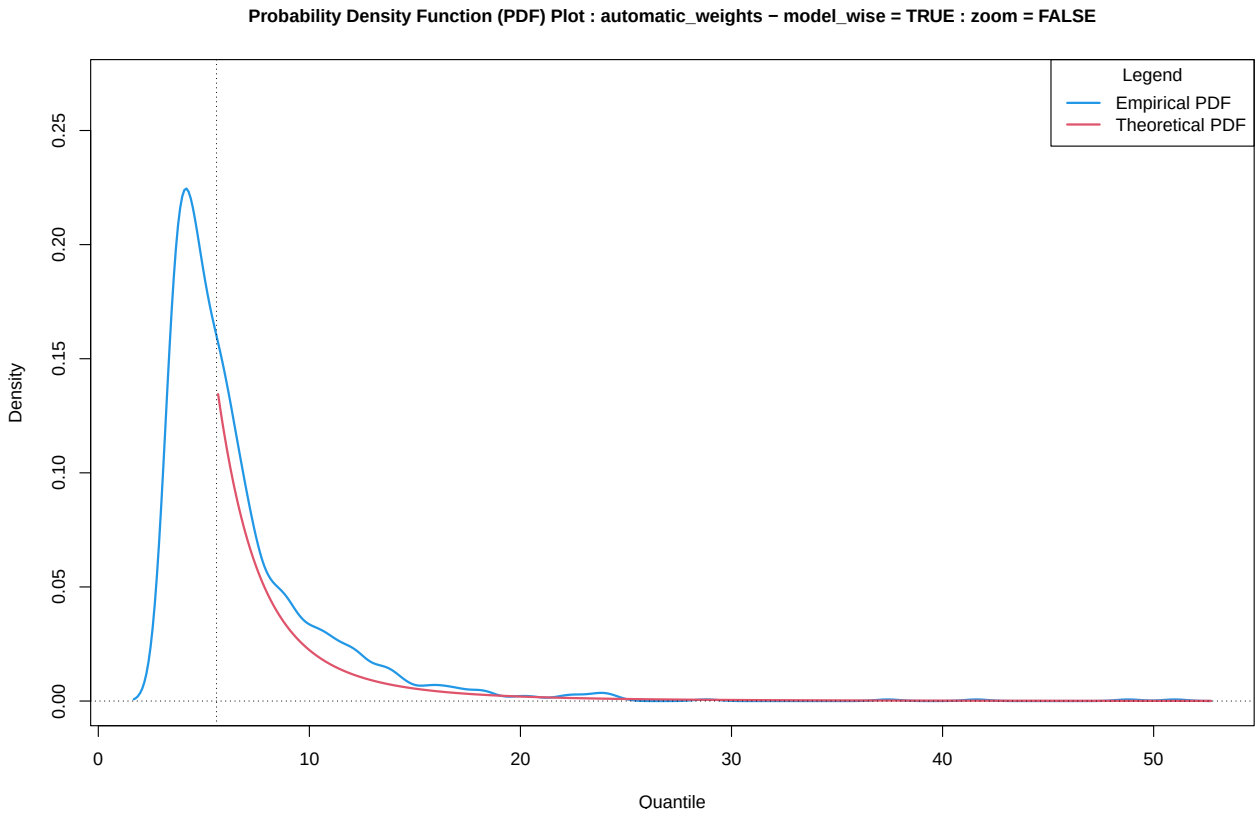
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



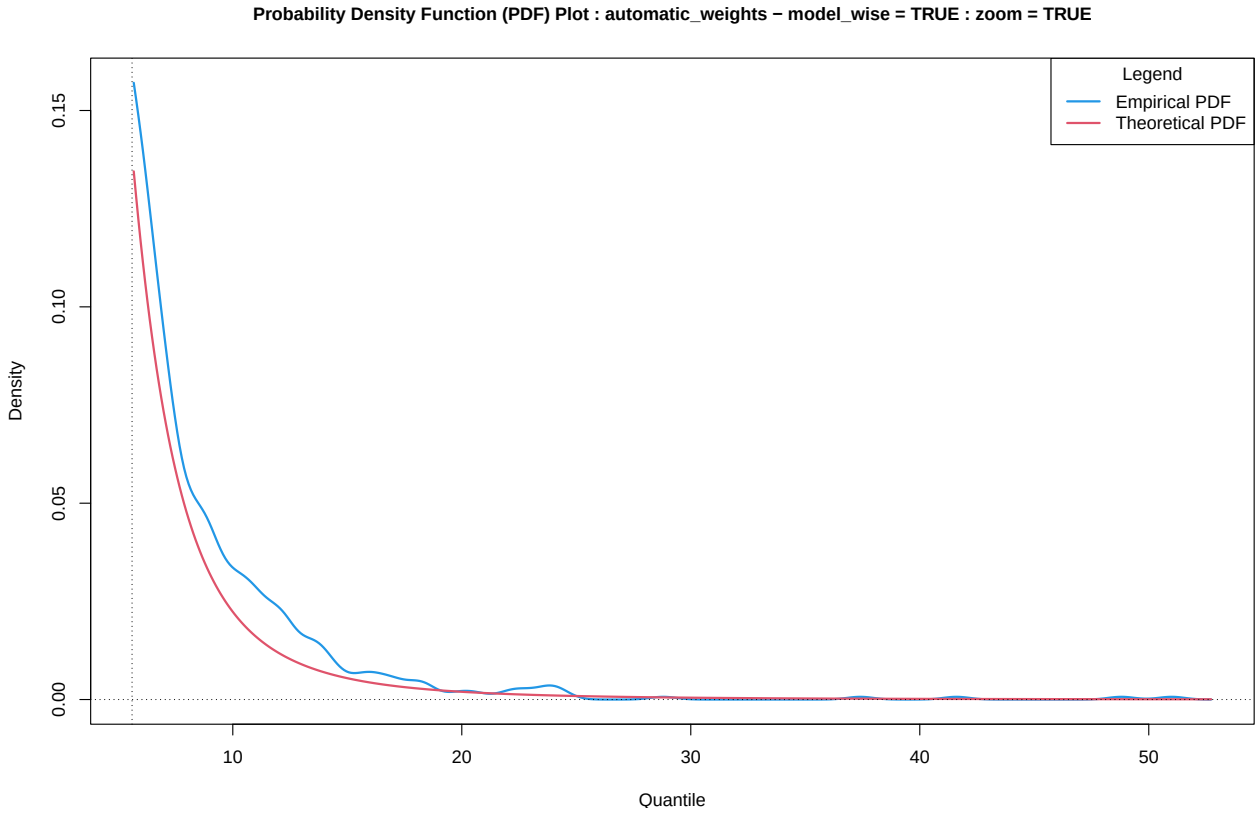

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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




```r
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^(-6)
```

```
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 332.05197871585    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 499.218734302632    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 50.9678315999987    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
## 4   18.4846725218455  332.04987106377 645.615069605695
## 5   7.49138998056429 273.821952257197 540.152514533831
## 6  -8.34166926641652 246.904713851617  502.15109696965
## 7  -6.22766951126607 222.336837812117 450.901345135501
## 8  -119.385616601887 347.099937304273 813.585491210433
## 9   7.31215173588544 162.077913094222 316.843674452559
## 10   -43.4502954787l  217.60163195305 478.653559384811
## 11  -93.964102640854 248.928996820948 591.822096282749
## 12 -23.0994674564878 162.170714872423 347.440897201334
## 13 -15.8264651632802  153.28486583488 322.396196833041
## 14 -42.9906573995647 170.451488586127 383.893634571819
## 15 -27.3336587204775 147.534253000828 322.402164722133
## 16 -64.3900355400181 183.133267487354 430.656570514727
## 17 -38.5963618927651 154.101042484214 346.798446861194
## 18 -41.5607166565588 148.571223799056 338.703164254672
## 19 -33.0346446119713 142.025924991238 317.086494594448
## 20 -64.0211745989149 168.239071333296 400.499317265507
```

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

```
## [1] -119.385616601887  813.585491210433
```

```
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
```

```
## 4 18.4846725218455  332.04987106377 645.615069605695
## 5 7.49138998056429 273.821952257197 540.152514533831
```

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

```
## [1]   7.49138998056429 645.61506960569523
```

```
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)
```