

# Modeling extreme values with a GEV mixture probability distributions

Application to a wind speed data

Pascal Alain Dkengne Sielenou

2023-10-04

```
# 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"))

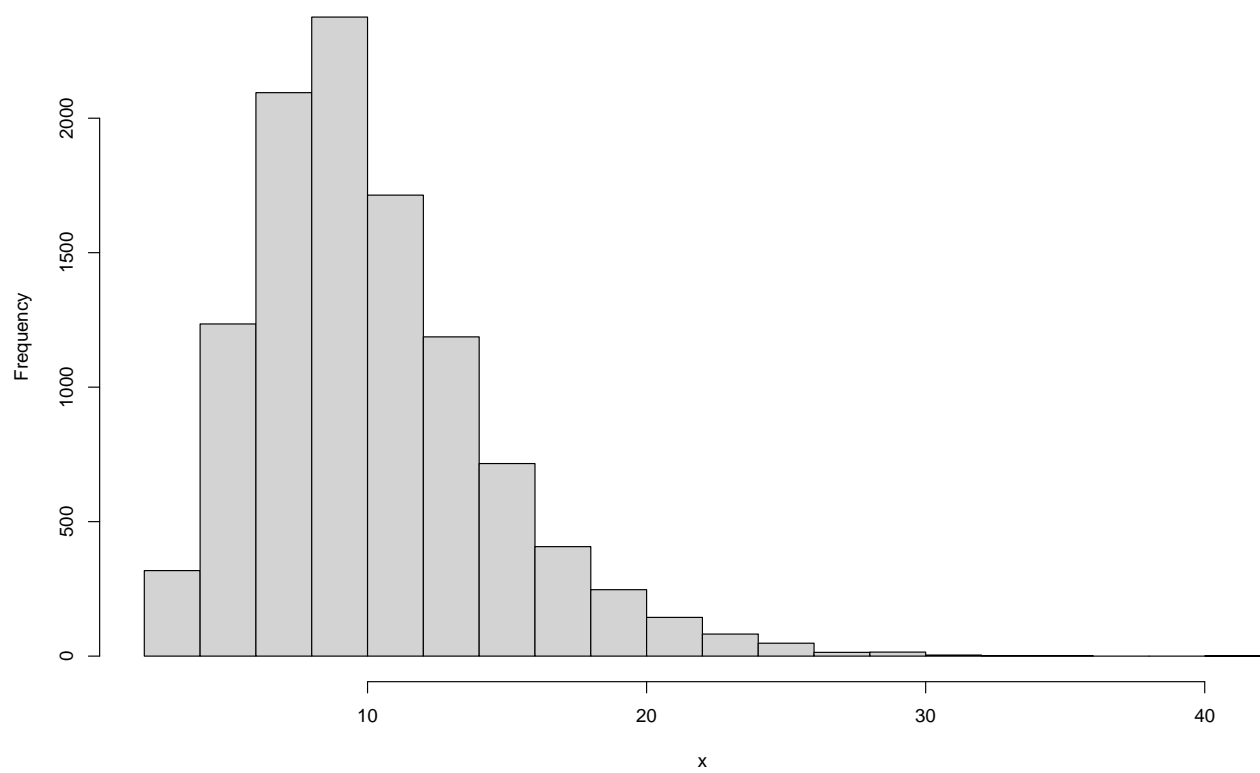
library(readr)

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

## Rows: 10627 Columns: 2
## -- Column specification -----
## Delimiter: ","
## dbl   (1): Vent
## date  (1): Date
##
## 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 <- vent$Vent
x <- x[!is.na(x)]

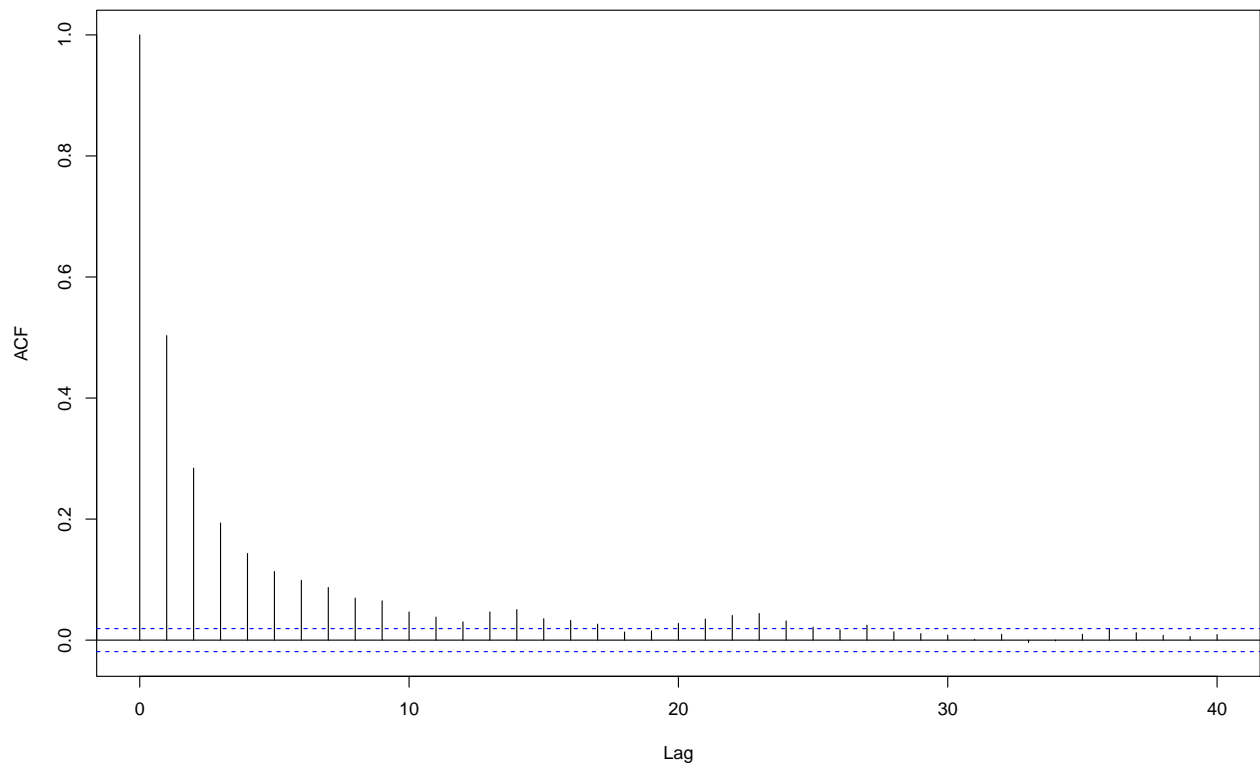
hist(x)
```

Histogram of x



`acf(x)`

Series x



```

n <- length(x)
n

## [1] 10607
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_object"
## [7] "block_maxima_indexes_object"
## [8] "gev_models_object"
## [9] "extremal_indexes"
## [10] "normalized_gev_parameters_object"
## [11] "full_normalized_gev_parameters_object"
## [12] "weighted_normalized_gev_parameters_object"
## [13] "identic_weights_mw"
## [14] "pessimistic_weights_mw"
## [15] "pessimistic_weights_pw_shape"
## [16] "pessimistic_weights_pw_scale"
## [17] "pessimistic_weights_pw_loc"
## [18] "automatic_weights_mw"
## [19] "automatic_weights_mw_statistics"
## [20] "automatic_weights_pw_shape"
## [21] "automatic_weights_pw_scale"
## [22] "automatic_weights_pw_loc"
## [23] "automatic_weights_pw_statistics"

gev_mixture_model$block_sizes

## [1] 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
## 5  17.6205160918168  2.50470685169596  0.048443189131169531
## 6  17.7609013780948  2.44020690168626  0.047839904072699273
## 7  17.3707071904242  2.67269753081899  0.024805736367325623
## 8  17.3726788708364  2.71302465197915  0.016893819861680943
## 9  16.1270910615283  3.44862156692913 -0.047127263464354092
## 10 17.0616684652882  2.70820231692934  0.023384212145713644
## 11 18.2991747477729  2.20576161475593  0.064044636014361653
## 12 16.4330744957099  3.00051523914643  0.000389020292740058
## 13 16.8127645422959  2.82262922050949  0.010157763065955368
## 14 17.8223118845530  2.27214299061722  0.059532096709861981
## 15 19.3792261213589  1.54721898409943  0.165658425510553209
## 16 15.9078762231531  3.32181734413804 -0.028518268095111637
## 17 17.7361555702709  2.55427288328847  0.026911861085332413
## 18 18.5234231200768  1.73473112583630  0.132984588680751692
## 19 18.1212732839698  2.26376993952106  0.064389250961314440
## 20 18.1781994279764  2.28574523049468  0.057578093832479700
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##          loc_star      scale_star      shape_star
## identic_weights      17.5329401546954  2.53100402452787  0.04171044163577961
## pessimistic_weights  18.3471110118577  2.81582360359462  0.04440495482380970
## automatic_weights    18.2018461385347  2.64601605817168 -0.00583879721999937
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 0.00489068692497834
##
## $gradient_value
## [1] 9.98270871810369e-06
##
## $function_reduction
## [1] 0.137738189105129
##
## $number_iterations
## [1] 1707
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.000596486499161229
##
## $gradient_value
## [1] 6.25192775425276e-05
##
## $function_reduction
## [1] 0.282472637502684
##
```

```
## $number_iterations
## [1] 3328
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##           5           6           7
## 6.18771193422427e-01 0.00000000000000e+00 0.00000000000000e+00
##           8           9          10
## 0.00000000000000e+00 1.04083408558608e-17 0.00000000000000e+00
##          11          12          13
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##          14          15          16
## 0.00000000000000e+00 3.81228806577574e-01 0.00000000000000e+00
##          17          18          19
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##          20
## 0.00000000000000e+00
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##           5           6           7           8
## 0.0628380075618947 0.0628001097635657 0.0613700943376514 0.0608864550550386
##           9          10          11          12
## 0.0571105951513385 0.0612829172389267 0.0638260588728235 0.0598897838719796
##          13          14          15          16
## 0.0604776986843522 0.0635386901424947 0.0706526306033375 0.0581833161255777
##          17          18          19          20
## 0.0614994836174342 0.0683814443168942 0.0638480580771257 0.0634146565795652
```

```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##           5           6           7           8
## 0.0503776071039353 0.0472465590717997 0.0595610502346524 0.0621907025611740
##           9          10          11          12
## 0.2017025698184525 0.0617394794165078 0.0362129822629125 0.0834283988407236
##          13          14          15          16
## 0.0695624414774665 0.0387632627025272 0.0187899506501598 0.1166099603275278
##          17          18          19          20
## 0.0524764702277870 0.0222042473322592 0.0390913157383527 0.0400430022337620
```

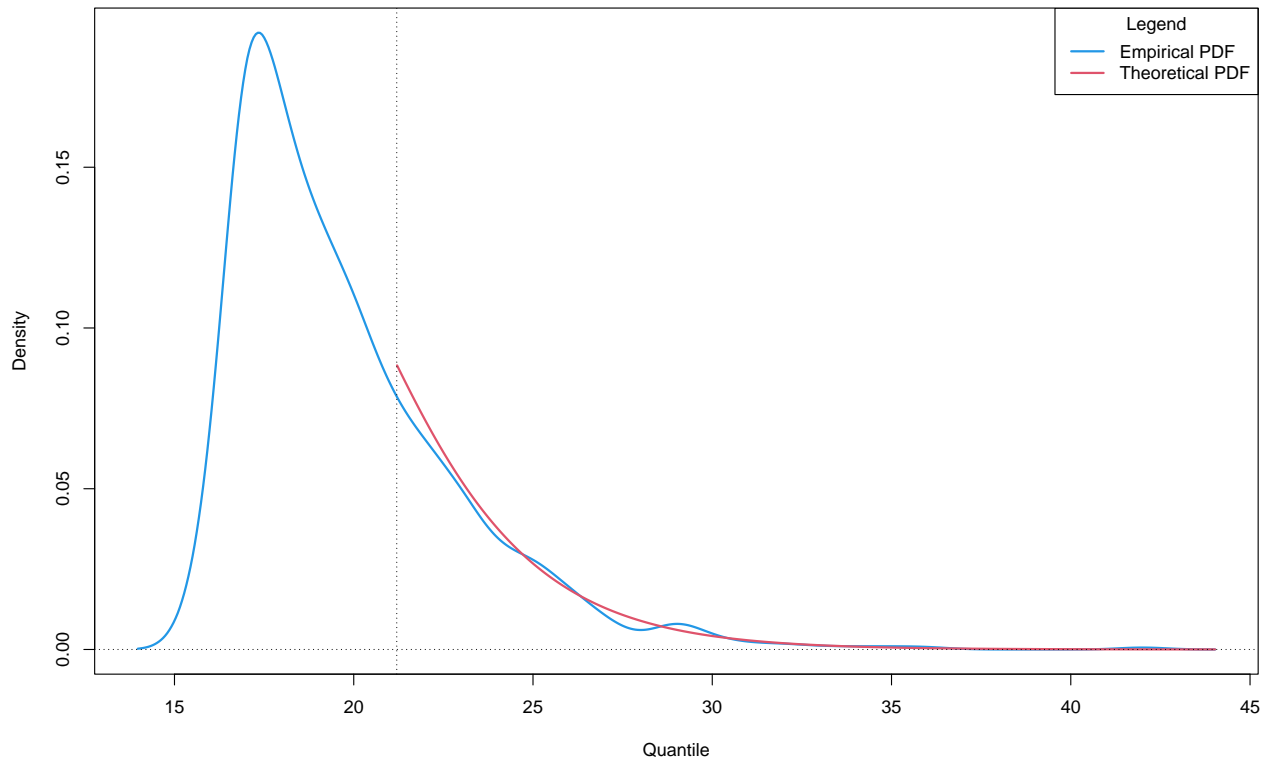
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##           5           6           7
## 5.63351676906206e-02 6.51293719467066e-02 3.61225311848163e-02
##           8           9          10
## 3.59813266349726e-02 1.50182856471364e-06 2.63862515950979e-02
##          11          12          13
## 7.13285000207967e-02 1.36086355367206e-02 2.02359988311359e-02
##          14          15          16
## 4.34336212709005e-02 3.12629926828425e-01 7.62148028020209e-03
##          17          18          19
```

```
## 3.95136635809661e-02 1.07664220410431e-01 7.98102427578386e-02
## 20
## 8.41975596018038e-02
```

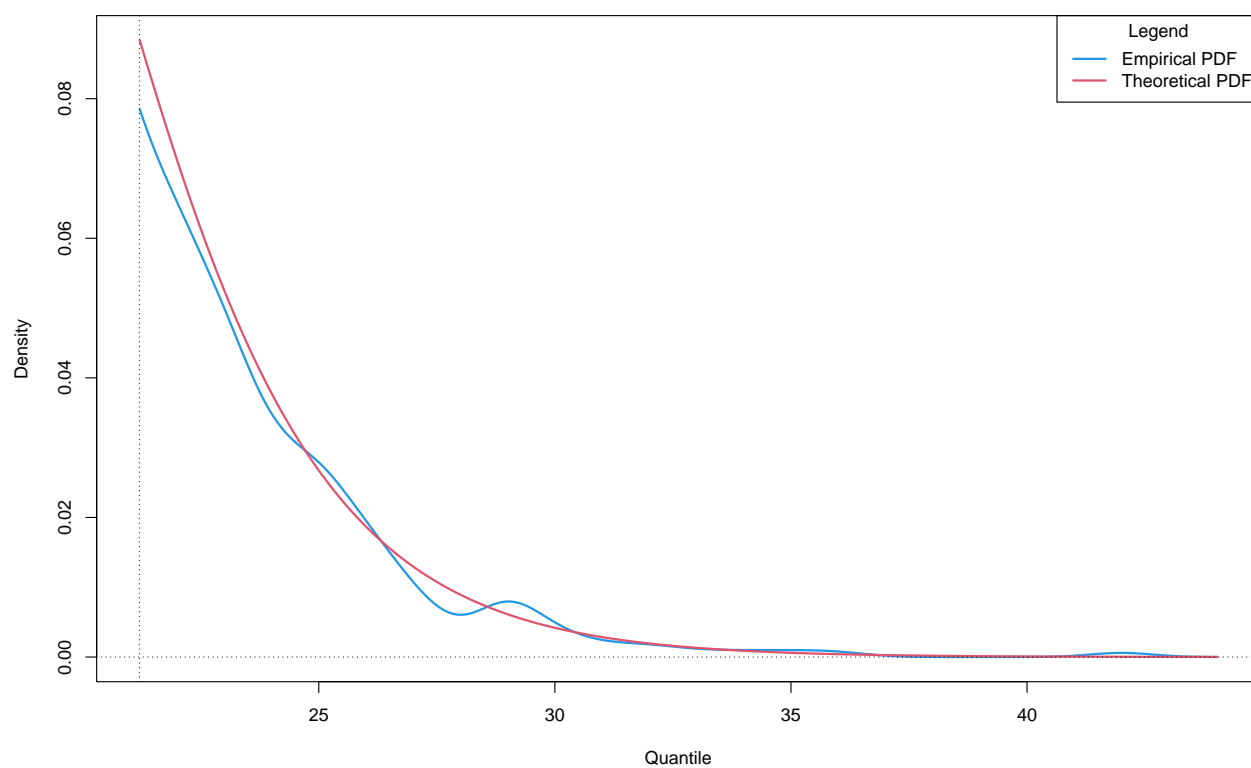
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                             type = "automatic_weights",
                             model_wise = FALSE,
                             zoom = FALSE,
                             xlab = "Quantile",
                             ylab = "Density",
      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

Probability Density Function (PDF) Plot : automatic\_weights – model\_wise = FALSE : zoom = FALSE

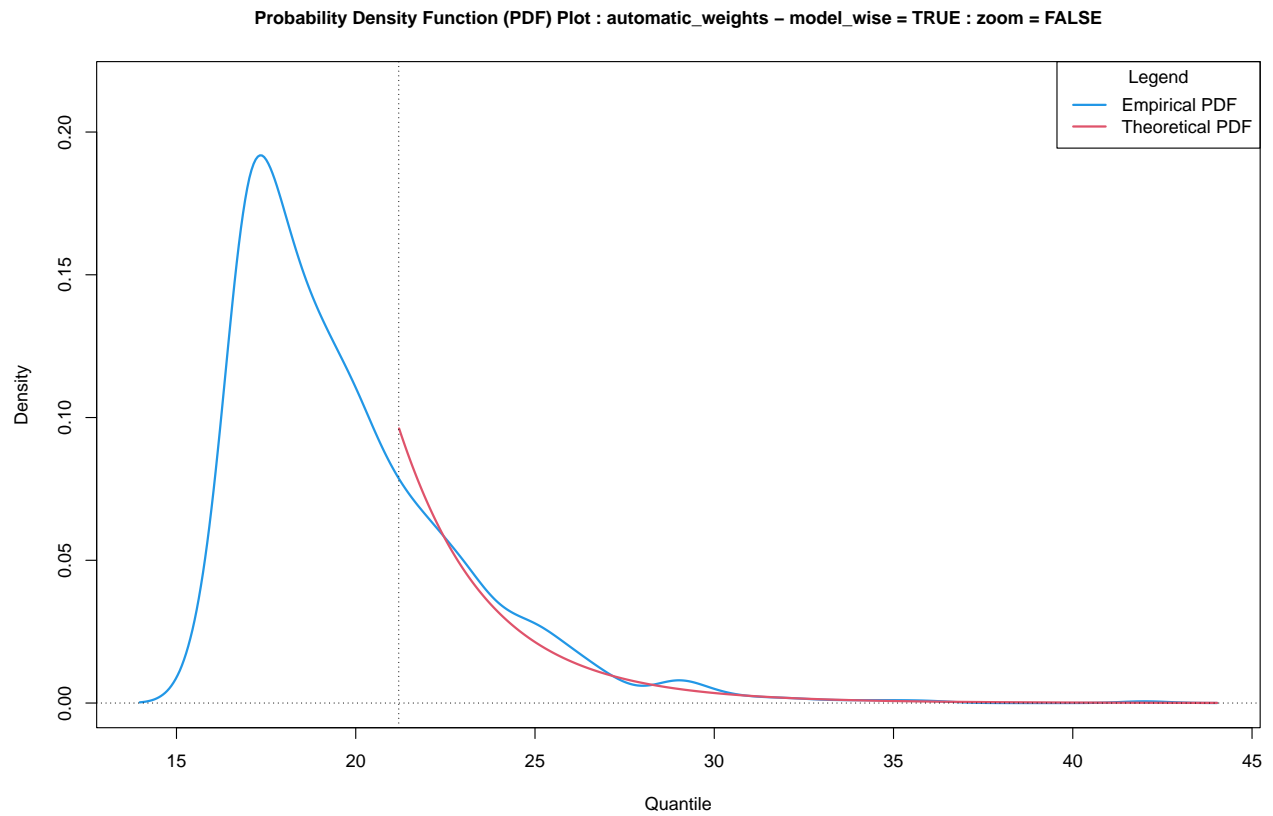


```
plot_gev_mixture_model_pdf(gev_mixture_model,
                             type = "automatic_weights",
                             model_wise = FALSE,
                             zoom = TRUE,
                             xlab = "Quantile",
                             ylab = "Density",
       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

Probability Density Function (PDF) Plot : automatic\_weights – model\_wise = FALSE : zoom = TRUE

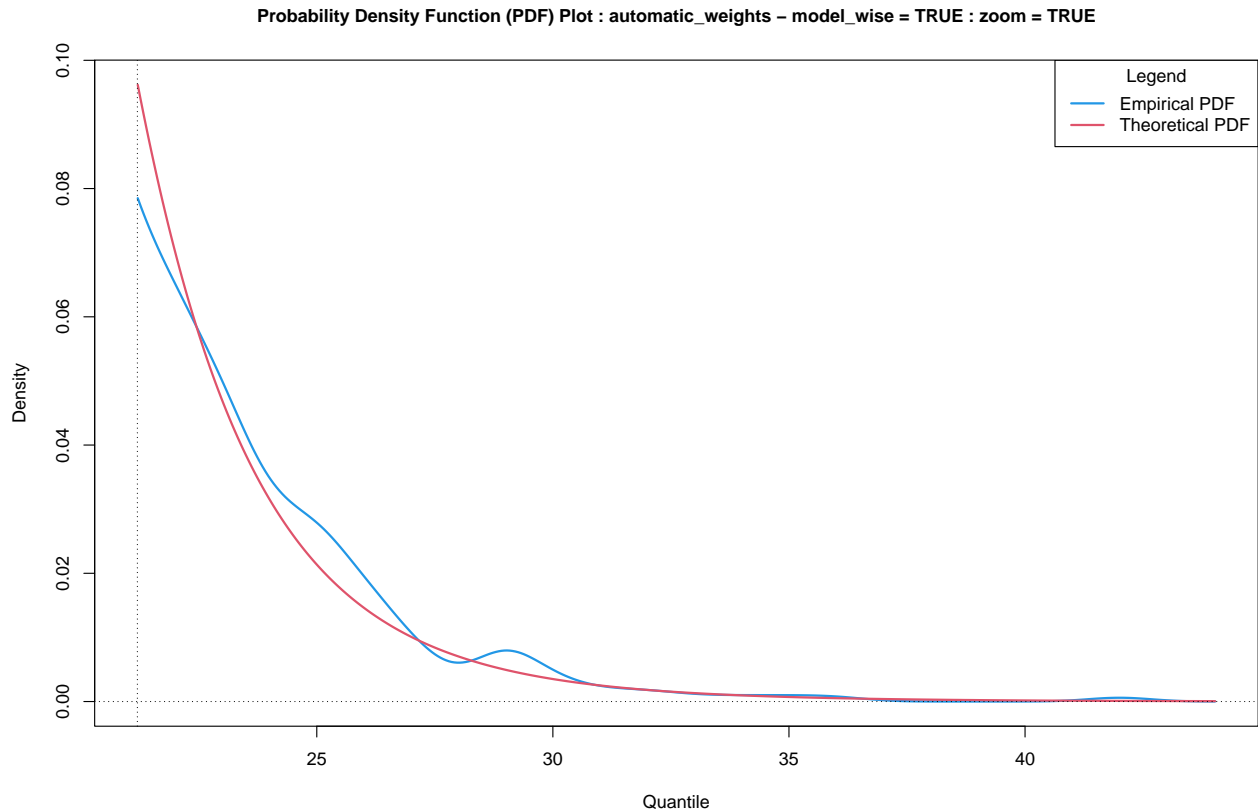


```
plot_gev_mixture_model_pdf(gev_mixture_model,  
                             type = "automatic_weights",  
                             model_wise = TRUE,  
                             zoom = FALSE,  
                             xlab = "Quantile",  
                             ylab = "Density",  
                             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(-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 64.1022969886653 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 47.5180664421353      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 41.936364000001      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
## 5  37.3607975255106  55.969842569361  74.5788876132114
## 6  36.2300132135648  54.9819985287164  73.733983843868
## 7  35.6271373126986  52.7754700408495  69.9238027690004
## 8  35.7410814419031  51.6577239923709  67.5743665428387
## 9  37.5355830862437  46.6514091423969   55.76723519855
## 10 34.7526697107768  52.6323512441578  70.5120327775388
## 11 33.2879128616886  55.5810750902905  77.8742373188925
## 12 34.9682650759231  50.8774534920081  66.7866419080931
## 13 34.0635036652888  51.0987339645002  68.1339642637116
## 14 32.9024180410562  55.1334636756482  77.3645093102403
## 15 14.9374069636966  72.3232465621667 129.709086160637
## 16 35.0624659206243  48.3663724445649  61.6702789685056
## 17 31.7296943126891  52.0029639301011  72.2762335475131
## 18 20.7186681866565  65.3127415515182 109.90681491638
## 19 27.8634590277109  56.4682591159479  85.073059204185
## 20 29.0428103741655  55.2488294222344  81.4548484703033
```

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

```
## [1] 14.9374069636966 129.7090861606369
```

```
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
## 5  37.3607975255106  55.969842569361  74.5788876132114
```

```
## 9 37.5355830862437 46.6514091423969 55.76723519855
## 15 14.9374069636966 72.3232465621667 129.709086160637
```

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

```
## [1] 14.9374069636966 129.7090861606369
```

```
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)
```

