

# Modeling extreme values with a GEV mixture probability distributions

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)

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

## Rows: 20002 Columns: 25
## -- Column specification -----
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## 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.
Gnss_map_matching <- xfun::in_dir(dir = path, expr = read_csv("./applications/Gnss_map_matching.csv"))

## Rows: 20001 Columns: 25
## -- Column specification -----
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## 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.
timestamp_position <- sapply(Gnss_map_matching$timestamp,
                             function(ts)
                               which.min(abs(ts - Gnss_imar$timestamp)))

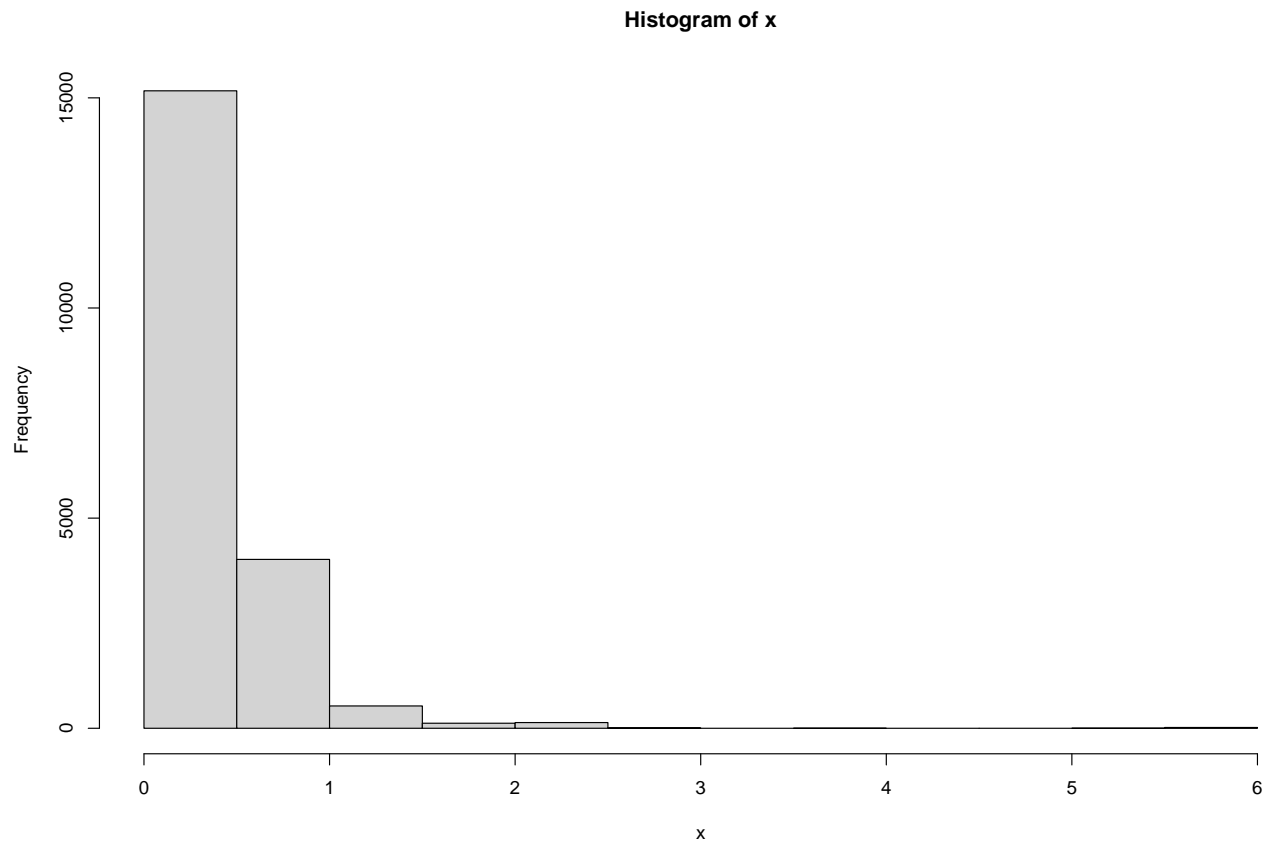
latitude_Gnss_map_matching_errors <- Gnss_imar$latitude[timestamp_position] - Gnss_map_matching$latitude

# timestamp_diff <- Gnss_imar$timestamp[timestamp_position] - Gnss_map_matching$timestamp
#
# head(timestamp_diff)
#
```

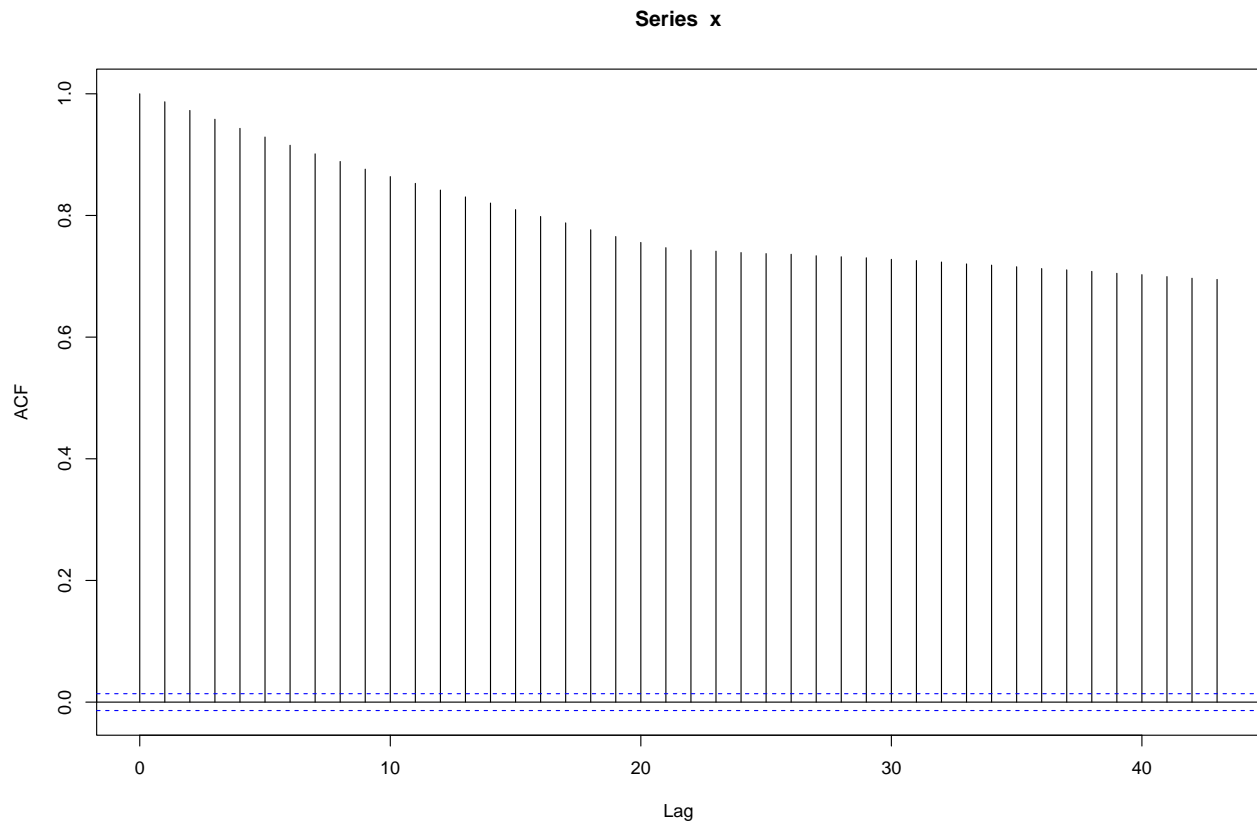
```
# tail(timestamp_diff)
#
# range(timestamp_diff)

coefficient <- 10^(5)
x <- coefficient*abs(latitude_Gnss_map_matching_errors)

hist(x)
```



```
acf(x)
```



```
n <- length(x)
n

## [1] 20001
nlargest <- 2000

#
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_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"

gev_mixture_model$block_sizes

## [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## [26] 27 28 29 30 31 32 33 34 35 36 37 38 39 40

gev_mixture_model$normalized_gev_parameters_object

##          loc_star      scale_star      shape_star
## 2  0.805710426836804 0.0925627976926149 0.698222823927651
## 3  0.779856695044170 0.0743406403952020 0.674803594819466
## 4  0.763739358981927 0.0662365180298286 0.648693798435795
## 5  0.753682630161842 0.0569595953846753 0.662663165005161
## 6  0.743895854617002 0.0550795362384526 0.635097291949382
## 7  0.737841546000396 0.0504019877453781 0.643250851393584
## 8  0.734564741142007 0.0456260891899596 0.645747454676042
## 9  0.724827957921471 0.0468923841950375 0.616776982885949
## 10 0.719466818262841 0.0463329044039577 0.607438354881596
## 11 0.719644885710446 0.0403323097066101 0.648074254397211
## 12 0.703043977524664 0.0473295834961854 0.583339671502519
## 13 0.715820752741766 0.0376673830403789 0.650152017300982
## 14 0.705269616082112 0.0418686021123029 0.598987261325323
## 15 0.702757528856571 0.0391711927072970 0.611037124637749
## 16 0.698376408476461 0.0381135255887596 0.616757968769042
## 17 0.700300449479832 0.0372172911460309 0.610649463494003
## 18 0.693560969490841 0.0381994968318737 0.603538782712026
## 19 0.693283330623280 0.0381063226336259 0.596643675292459
## 20 0.696284841746089 0.0349938278991347 0.615029996051900
## 21 0.673667032503377 0.0414490795738468 0.568920316897178
## 22 0.671443939690212 0.0403358213824187 0.580349010859472
## 23 0.661535904625353 0.0444064517245657 0.544168707258970
## 24 0.681432444198506 0.0363455344257737 0.583168277009720
## 25 0.659503249595194 0.0415530647055034 0.573896343937682
## 26 0.678772807428923 0.0359318966660696 0.597708301137729
## 27 0.650040932867760 0.0407223023790315 0.556692435491310
## 28 0.668882650811449 0.0346895815215517 0.586609058584254
## 29 0.661730030643846 0.0350119919715672 0.601807278625393

```

```
## 30 0.651039314283511 0.0380930736613254 0.574243965360771
## 31 0.659783522532237 0.0371985544018593 0.568923233505324
## 32 0.680123960057229 0.0285258401672996 0.639848699316876
## 33 0.635503986643970 0.0386847808222243 0.592623121836322
## 34 0.652619248722293 0.0346673658670043 0.599333653294762
## 35 0.620262954312987 0.0456308905151669 0.528763108202815
## 36 0.643962137328050 0.0370018093612843 0.555194473862787
## 37 0.610832735007749 0.0442049799724191 0.516111366264953
## 38 0.628766851391194 0.0390228600884451 0.559028851604275
## 39 0.664529951539311 0.0288351523502549 0.646039600063804
## 40 0.666742929222648 0.0271588055347410 0.625875942866908
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##               loc_star      scale_star      shape_star
## identic_weights    0.690079624951444 0.0429974827058887 0.604261802036901
## pessimistic_weights 0.691974866913750 0.0431471702613059 0.605842349960313
## automatic_weights   0.805710426828560 0.0925627976926096 0.698222823924852
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 140.165074348661
##
## $gradient_value
## [1] 1.13686837721616e-12
##
## $function_reduction
## [1] 388.408555603683
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 140.165074369517
##
## $gradient_value
## [1] 2.38742359215394e-11
##
## $function_reduction
## [1] 406.754502433245
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
```

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

```
gev_mixture_model$automatic_weights_mw
```

```
##          2          3          4
## 1.00000000000000e+00 1.13686837721616e-12 0.00000000000000e+00
##          5          6          7
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##          8          9         10
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         11         12         13
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         14         15         16
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         17         18         19
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         20         21         22
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         23         24         25
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         26         27         28
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         29         30         31
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         32         33         34
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         35         36         37
## 0.00000000000000e+00 0.00000000000000e+00 0.00000000000000e+00
##         38         39         40
## 0.00000000000000e+00 0.00000000000000e+00 1.42108547152020e-14
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##          2          3          4          5
## 0.0281448554037349 0.0274933828545581 0.0267848266086899 0.0271616193204813
##          6          7          8          9
## 0.0264231111456470 0.0266394342563065 0.0267060254466171 0.0259434388713065
##         10         11         12         13
## 0.0257022904965592 0.0267682373684530 0.0250903028018009 0.0268239132396566
##         14         15         16         17
## 0.0254859932962452 0.0257949537591276 0.0259429455844166 0.0257849559958479
##         18         19         20         21
## 0.0256022577291063 0.0254263346132563 0.0258981555913322 0.0247311126967734
##         22         23         24         25
## 0.0250153783133928 0.0241264913923648 0.0250860028305842 0.0248544820724081
##         26         27         28         29
## 0.0254534185608132 0.0244305449819592 0.0251724669536794 0.0255579656722261
##         30         31         32         33
## 0.0248631235247262 0.0247311848278433 0.0265489568459541 0.0253243119070539
##         34         35         36         37
## 0.0254948229688318 0.0237576566937603 0.0243939763590473 0.0234589743621902
##         38         39         40
## 0.0244876916340868 0.0267138286285531 0.0261805743906087
```

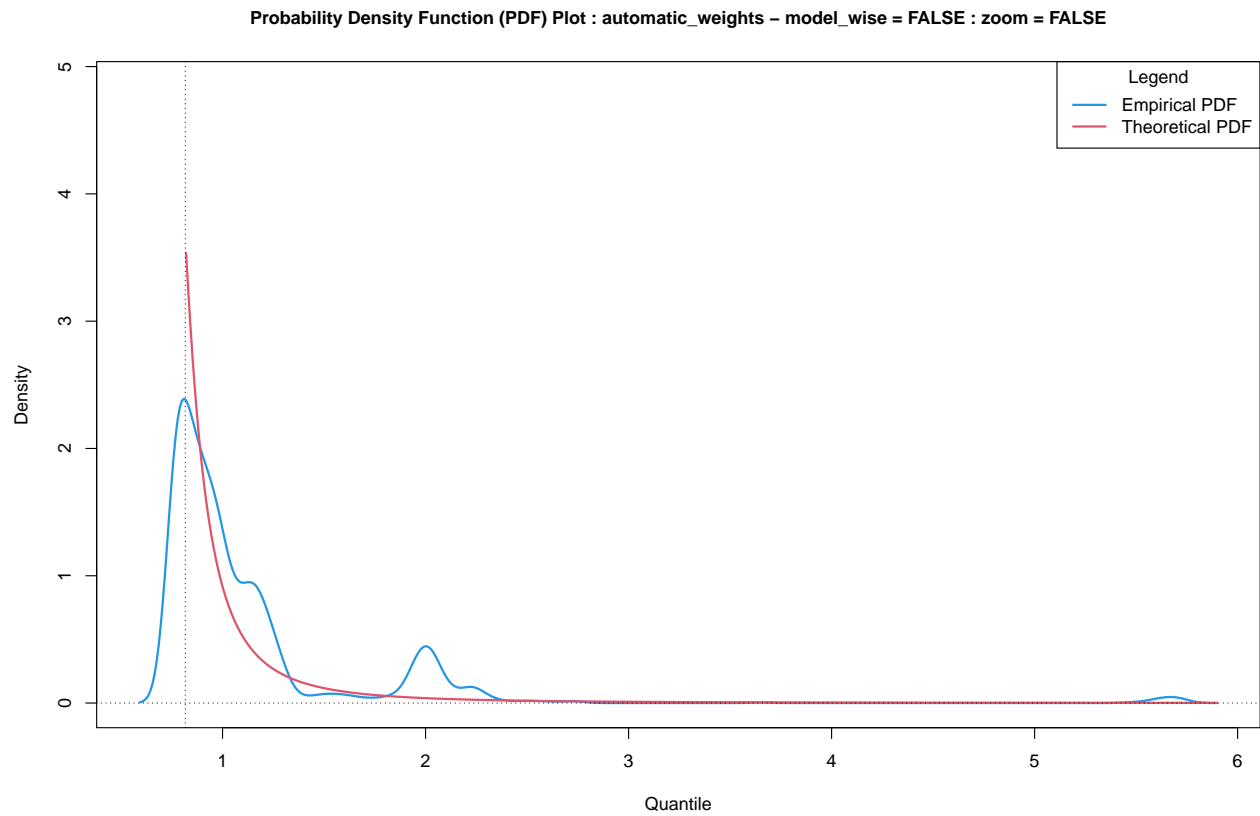
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##           2           3           4           5
## 0.0269419470764177 0.0264554526295714 0.0262419208168720 0.0259996022704505
##           6           7           8           9
## 0.0259507674010096 0.0258296648803202 0.0257065991298206 0.0257391718869035
##          10          11          12          13
## 0.0257247753680499 0.0255708736301585 0.0257504274951573 0.0255028198465895
##          14          15          16          17
## 0.0256101881606983 0.0255412000846621 0.0255142002780854 0.0254913438169213
##          18          19          20          21
## 0.0255163938598915 0.0255140165011074 0.0254347277160649 0.0255994463629215
##          22          23          24          25
## 0.0255709634269345 0.0256752655101264 0.0254691312500244 0.0256021084631289
##          26          27          28          29
## 0.0254585984341679 0.0255808480283464 0.0254269904693668 0.0254351897184973
##          30          31          32          33
## 0.0255136784688488 0.0254908661966081 0.0252707470932384 0.0255287795623716
##          34          35          36          37
## 0.0254264255984049 0.0257067225558593 0.0254858514884296 0.0256700931904529
##          38          39          40
## 0.0255374117725370 0.0252785648521884 0.0252362247087951
```

```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##           2           3           4           5
## 0.0287569802205355 0.0280230334791713 0.0275749971062693 0.0272990726125897
##           6           7           8           9
## 0.0270332058277006 0.0268700329047604 0.0267821291507722 0.0265226227938217
##          10          11          12          13
## 0.0263808117828870 0.0263855097649733 0.0259511020973267 0.0262848007522442
##          14          15          16          17
## 0.0260089241924366 0.0259436695032745 0.0258302557850622 0.0258800020979466
##          18          19          20          21
## 0.0257061707660435 0.0256990347245698 0.0257762865412373 0.0251998271085153
##          22          23          24          25
## 0.0251438677784162 0.0248959715649661 0.0253962759063745 0.0248454180394787
##          26          27          28          29
## 0.0253288207799935 0.0246114315975128 0.0250795494735475 0.0249008049899341
##          30          31          32          33
## 0.0246360154634302 0.0248523825136959 0.0253630670134965 0.0242562444629331
##          34          35          36          37
## 0.0246749695170176 0.0238893572289068 0.0244622775347096 0.0236651342487736
##          38          39          40
## 0.0240933761123505 0.0249706229709394 0.0250259435913857
```

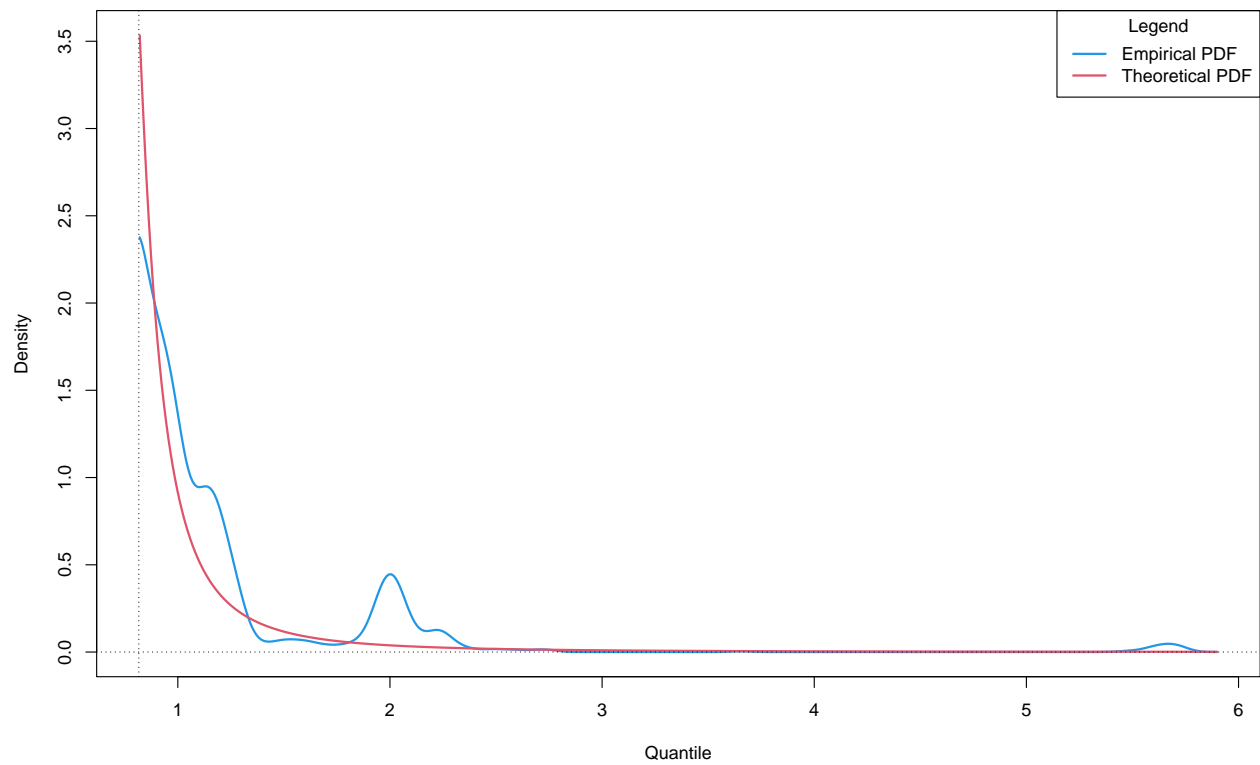
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                             type = "automatic_weights",
                             model_wise = FALSE,
                             zoom = FALSE,
                             xlab = "Quantile",
                             ylab = "Density",
      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



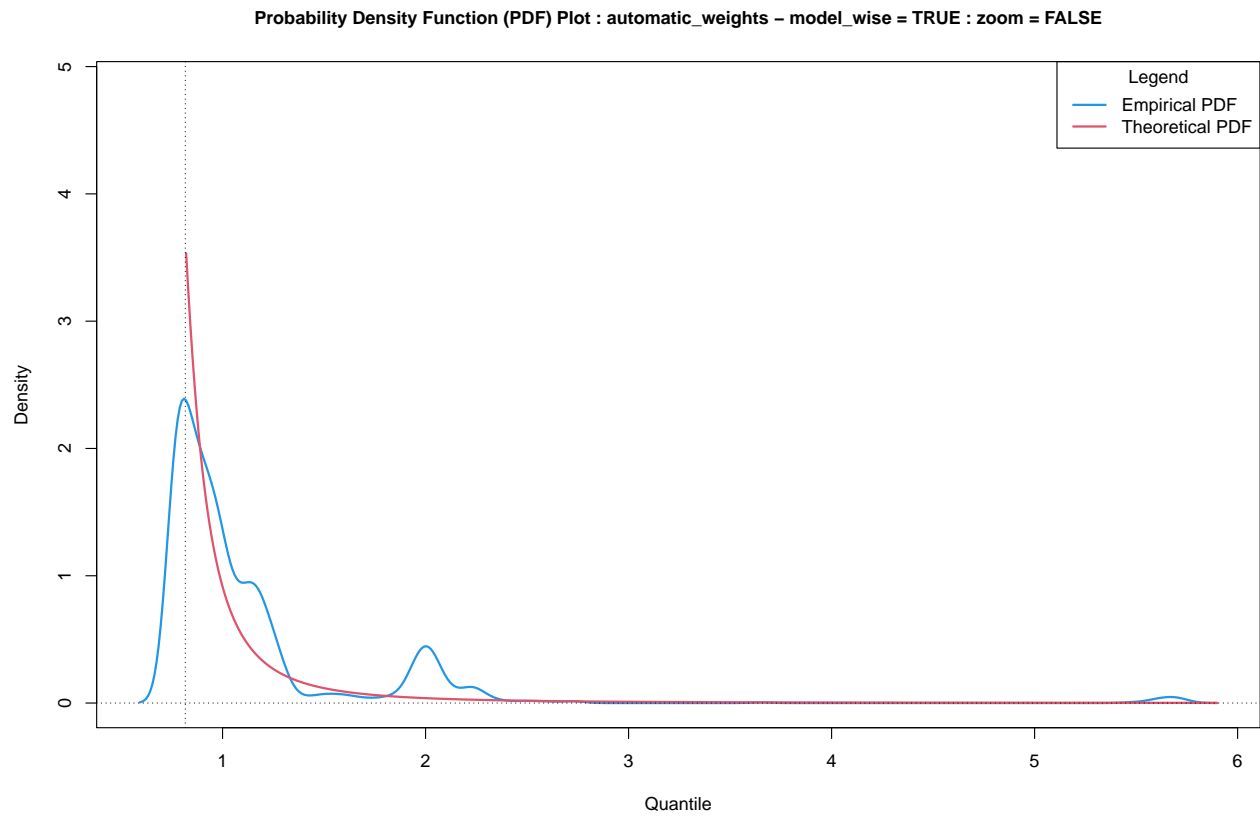
```
plot_gev_mixture_model_pdf(gev_mixture_model,  
    type = "automatic_weights",  
    model_wise = FALSE,  
    zoom = TRUE,  
    xlab = "Quantile",  
    ylab = "Density",  
    main = "Probability Density Function (PDF) Plot")
```



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

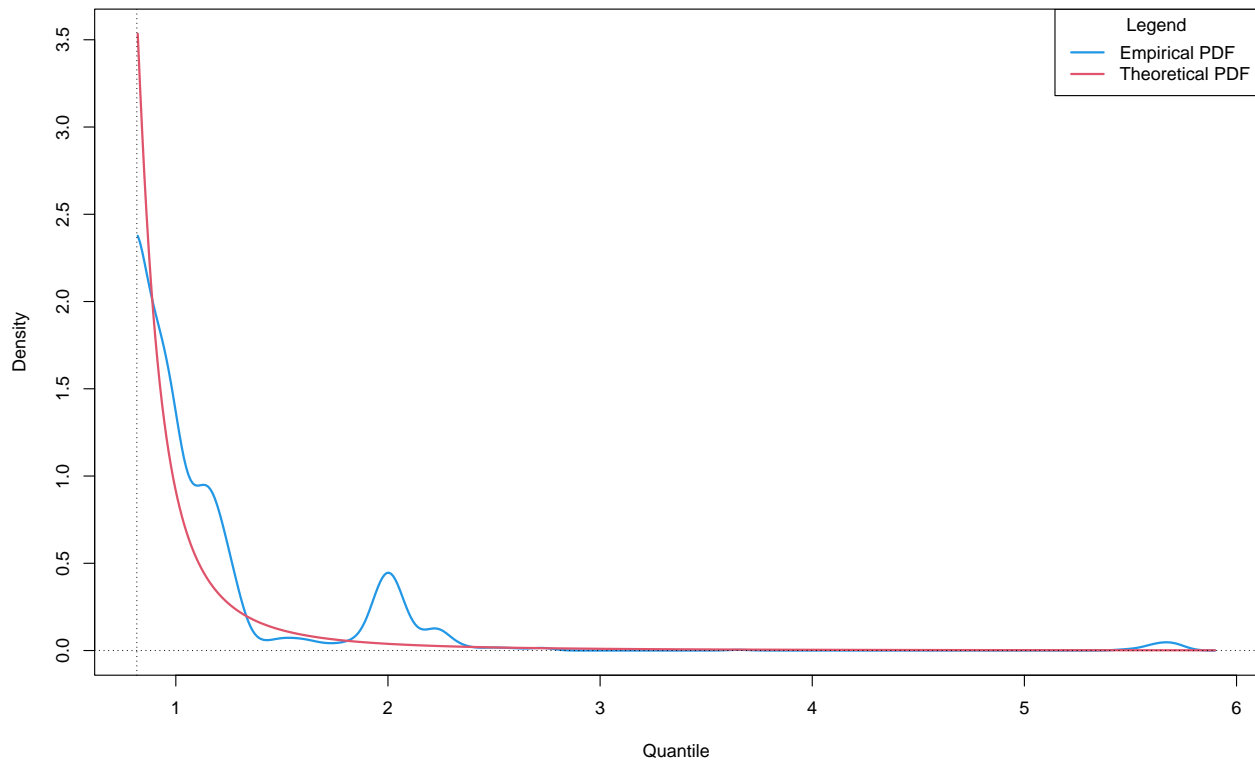


```
plot_gev_mixture_model_pdf(gev_mixture_model,  
                             type = "automatic_weights",  
                             model_wise = TRUE,  
                             zoom = FALSE,  
                             xlab = "Quantile",  
                             ylab = "Density",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

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



```
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 158152710.644808    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 158213985.098109      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 5.7475778000048      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
## 2 -199189602.081222 158983582.015779 517156766.11278
## 3 -105201487.907062 65086936.4996284 235375360.906319
## 4 -52104458.1343202 27659967.5261659 107424393.186652
## 5 -81902605.5629126 35307377.5267747 152517360.616462
## 6 -37665472.5760388 15623089.6204425 68911651.8169237
## 7 -47770122.607734 18017433.0324003 83804988.6725346
## 8 -51595370.7775297 17439904.5444856 86475179.866501
## 9 -24016541.5537509 7903043.5191041 39822628.5919591
## 10 -19206940.6852464 5997119.64195696 31201179.9691603
## 11 -58231531.5587385 16538676.5743643 91308884.7074672
## 12 -11096285.4866883 3094896.36862844 17286078.2239451
## 13 -61735369.0216596 16392844.8145478 94521058.6507553
## 14 -16592935.4438193 4274322.88755697 25141581.2189332
## 15 -22274983.5301244 5626065.65880416 33527114.8477327
## 16 -28147004.9797781 6433388.87592238 41013782.7316229
## 17 -22902312.9119946 5284573.59247322 33471460.096941
## 18 -19106844.2977533 4436156.25383335 27979156.80542
## 19 -15875940.1469478 3641886.30996825 23159712.7668843
## 20 -26598077.8940404 5625191.75963099 37848461.4133023
## 21 -8336284.94376717 1804879.048744 11946043.0412552
## 22 -11664672.3669833 2432163.91719127 16529000.2013659
## 23 -4827030.71275312 967250.003726135 6761530.72020539
## 24 -13179718.3168353 2372493.42841056 17924705.1736564
## 25 -10734397.4658647 2092431.85738651 14919261.1806377
## 26 -19632234.001966 3544347.54665353 26720929.0952731
## 27 -6601837.34772114 1262366.23994471 9126569.82761055
## 28 -15084284.2904934 2494965.27100685 20074214.8325071
## 29 -23727889.7240123 3868233.66616603 31464357.0563444
## 30 -11514401.8263575 1934363.32540782 15383128.4771731
## 31 -9356772.47054333 1626127.640265 12609027.7510733

```

```
## 32 -58378132.19828 9266715.20355324 76911562.6053865
## 33 -22118018.0807432 3301395.03431397 28720808.1493711
## 34 -23222800.8761358 3573438.21550858 30369677.307153
## 35 -3841356.07306431 645280.033069709 5131916.13920373
## 36 -8150101.34464101 1099138.35099692 10348378.0466348
## 37 -3070018.07963593 438609.771301075 3947237.62223808
## 38 -8410893.07187495 1291493.80414459 10993880.6801641
## 39 -84069221.6779872 11020275.7071963 106109773.09238
## 40 -49016110.9624712 5937584.92821086 60891280.8188929
```

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

```
## [1] -199189602.081222 517156766.112780
```

```
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
## 2 -199189602.081222 158983582.015779 517156766.11278
## 3 -105201487.907062 65086936.4996284 235375360.906319
## 40 -49016110.9624712 5937584.92821086 60891280.8188929
```

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

```
## [1] -199189602.081222 517156766.112780
```

```
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)
```

Estimates of a quantile

