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"))
```

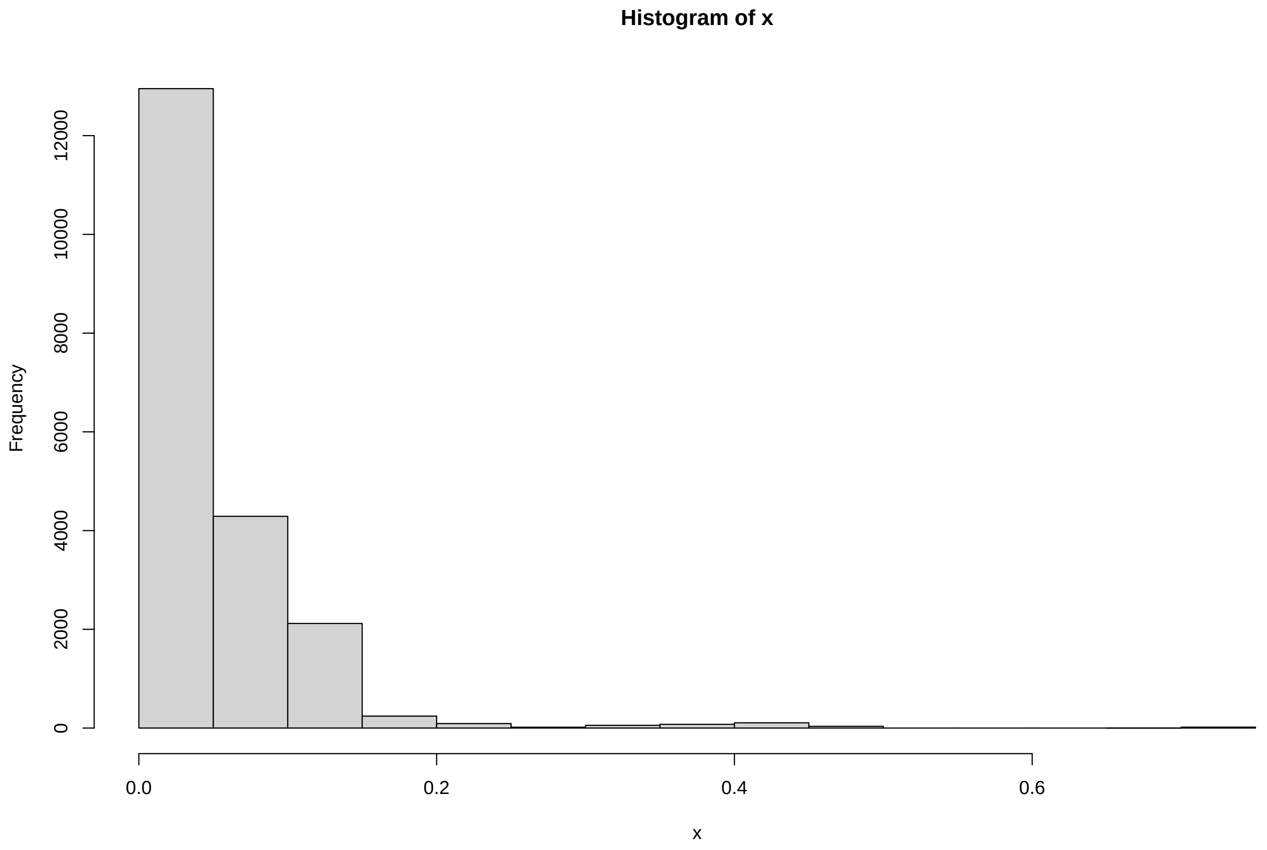
```
## Rows: 20001 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify 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)))
```

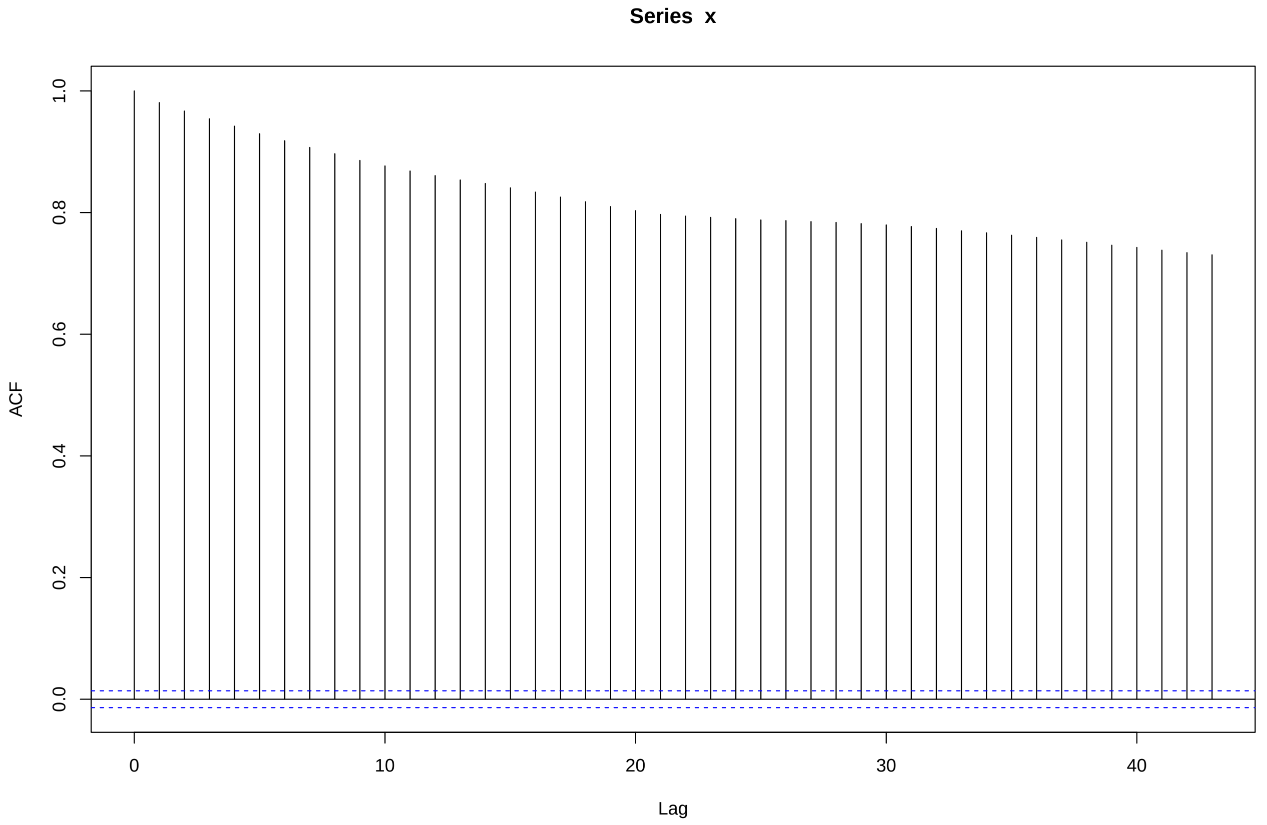
```
longitude_Gnss_map_matching_errors <- Gnss_imar$longitude[timestamp_position] - Gnss_map_matching$longi
```

```
coefficient <- 10^(4)
x <- coefficient*abs(longitude_Gnss_map_matching_errors)
```

```
hist(x)
```


Histogram of x
Frequency
0
2000
4000
6000
8000
10000
12000
0.0
0.2
0.4
0.6
x

```
acf(x)
```


Series x
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
Lag

```
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  0.119795382166518 0.01043519353869735 0.934433282582886
## 3  0.117225729245122 0.00776915471962147 0.919473648356441
## 4  0.115687144632795 0.00654195489999514 0.905150628209515
## 5  0.114750048330102 0.00560688814154006 0.890800643595627
## 6  0.113594010075366 0.00489887669594115 0.892623215839623
## 7  0.113071947338929 0.00447791662445303 0.886453847836877
## 8  0.112739777837855 0.00404667247181760 0.898267190609773
## 9  0.112270365422346 0.00371314994029802 0.892661447737409
## 10 0.111920959751720 0.00361228255822290 0.885995002578321
## 11 0.111601057054473 0.00331342430853533 0.892445212755723
## 12 0.110803598353364 0.00334042532308328 0.883480534984953
## 13 0.110766589141358 0.00310290660249475 0.887590880259133
## 14 0.109955952874506 0.00308985844908163 0.877885667053274
## 15 0.110641039960343 0.00280365072205724 0.886811916691246
## 16 0.110606597282878 0.00257107597746282 0.901836476931882
## 17 0.109235632760966 0.00289809727035658 0.859737724241165
## 18 0.109672742383695 0.00245203208701325 0.892478825163727
## 19 0.108591319774807 0.00261480112008392 0.876277871516123
## 20 0.108432040271048 0.00281352973708659 0.842479929817399
## 21 0.109634240070361 0.00217317355366519 0.911605872855531
## 22 0.108733757810629 0.00241879156972262 0.870975545750284
## 23 0.109030464956411 0.00219135434388233 0.892326480037792
## 24 0.109400000770947 0.00191220258047273 0.938681457382233
## 25 0.108159641157958 0.00224635568636447 0.869054181361073
## 26 0.108913339426601 0.00194896931523222 0.906992861648218
## 27 0.109401607913934 0.00177218883833911 0.929747207731878
## 28 0.107276328181549 0.00202095164172255 0.909862321900553
## 29 0.106893046565196 0.00215108094753769 0.897017799529763
## 30 0.106820081334573 0.00223576022332469 0.876072059737570
## 31 0.108898848461315 0.00165023801385915 0.930154513676650
## 32 0.109840445905791 0.00153656809888160 0.928690552087475
## 33 0.107704306809585 0.00182044390659071 0.883954719593402
## 34 0.103991037282217 0.00267799175833500 0.801936955650600
## 35 0.106797571180625 0.00171658872597110 0.929186507257484
## 36 0.104866203241029 0.00215407521469256 0.862579515030586
## 37 0.106963863171517 0.00163245614816882 0.927488261482935
## 38 0.109672818171278 0.00131216116686627 0.976028612735869
## 39 0.109993163669245 0.00147339360085294 0.924350742310328
## 40 0.107837332772911 0.00153185700668312 0.929060900705827
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                              loc_star          scale_star        shape_star
## identic_weights     0.110056154705432 0.00304303829561556 0.897503872185311
## pessimistic_weights 0.110065906026538 0.00304641303624093 0.898410858128731
## automatic_weights   0.119795382168251 0.01043519353866773 0.976028612733484
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 172.495711066447
##
## $gradient_value
## [1] 5.11590769747272e-13
##
```

```
## $function_reduction
## [1] 535.875820396137
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 169.703856335085
##
## $gradient_value
## [1] 6.59383658785373e-12
##
## $function_reduction
## [1] 550.534222177703
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                 2                 3                 4                 5
## 0.999999999998408 0.000000000000000 0.000000000000000 0.000000000000000
##                 6                 7                 8                 9
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                10                11                12                13
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                14                15                16                17
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                18                19                20                21
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                22                23                24                25
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                26                27                28                29
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                30                31                32                33
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                34                35                36                37
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                38                39                40
## 0.000000000000000 0.000000000000000 0.000000000000000
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                  2                  3                  4                  5
## 0.0265935519235331 0.0261986830250891 0.0258261132915322 0.0254581553725991
##                  6                  7                  8                  9
## 0.0255045970087273 0.0253477341339498 0.0256489512935445 0.0255055721165132
##                 10                 11                 12                 13
## 0.0253361061142622 0.0255000575158400 0.0252724793237707 0.0253765717210197
##                 14                 15                 16                 17
## 0.0251314779494901 0.0253568119932244 0.0257406633204485 0.0246795069018141
##                 18                 19                 20                 21
## 0.0255009146485824 0.0250911041365163 0.0242572471612552 0.0259933664202278
##                 22                 23                 24                 25
## 0.0249584150198964 0.0254970300044396 0.0267067662875791 0.0249105068493308
##                 26                 27                 28                 29
## 0.0258737348725958 0.0264692240809660 0.0259480851479567 0.0256169257329775
##                 30                 31                 32                 33
## 0.0250859406231222 0.0264800073491832 0.0264412699974589 0.0252844659862042
##                 34                 35                 36                 37
## 0.0232934557164113 0.0264543869344532 0.0247497406447511 0.0264094990097759
##                 38                 39                 40
## 0.0277230475197885 0.0263267685523587 0.0264510642988110
```

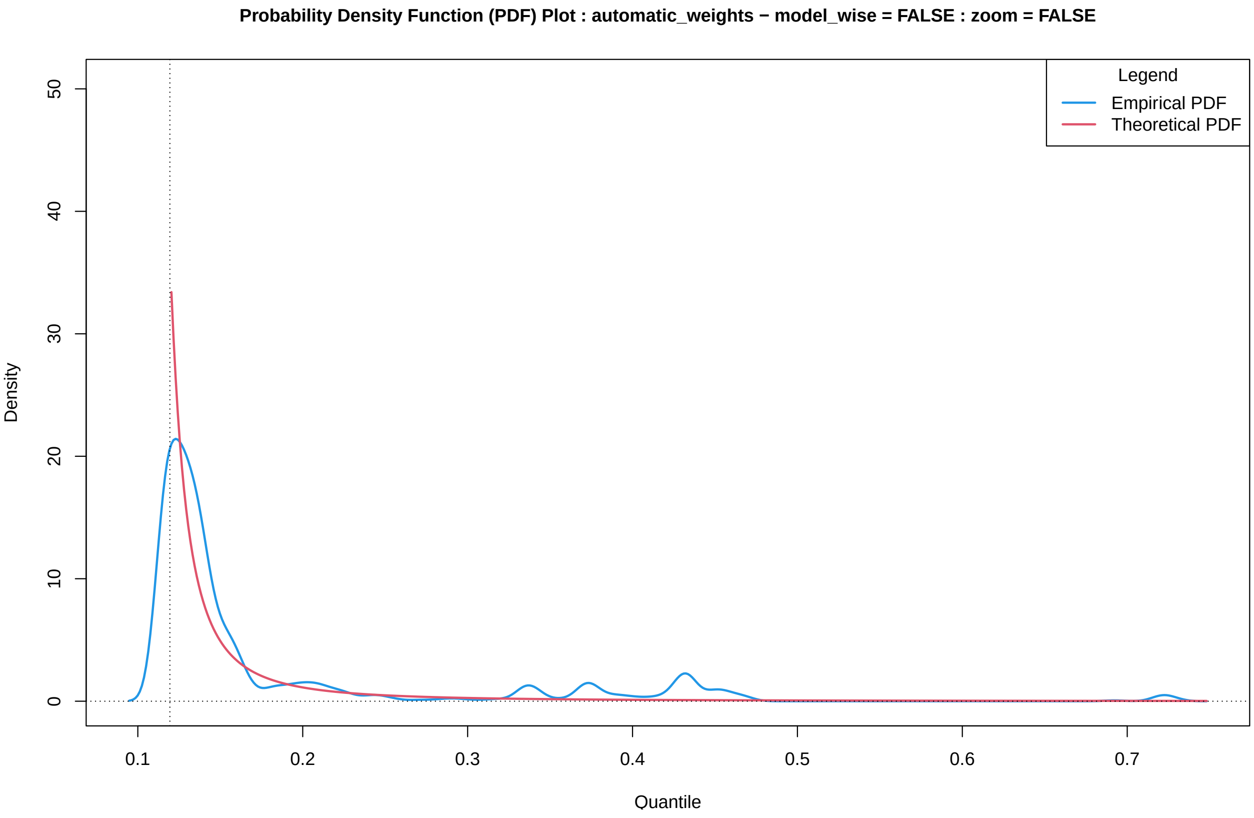
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                  2                  3                  4                  5
## 0.0258312268193750 0.0257624514855217 0.0257308552011481 0.0257068063791622
##                  6                  7                  8                  9
## 0.0256886121076522 0.0256778005034495 0.0256667294894564 0.0256581704842511
##                 10                 11                 12                 13
## 0.0256555825422872 0.0256479163054100 0.0256486088345208 0.0256425175331932
##                 14                 15                 16                 17
## 0.0256421829478734 0.0256348450071137 0.0256288836828371 0.0256372662440778
##                 18                 19                 20                 21
## 0.0256258329024076 0.0256300043339315 0.0256350982553837 0.0256186879164950
##                 22                 23                 24                 25
## 0.0256249811006231 0.0256191536887197 0.0256120030548929 0.0256205628153174
##                 26                 27                 28                 29
## 0.0256129447419272 0.0256084172735368 0.0256147884876357 0.0256181219391654
##                 30                 31                 32                 33
## 0.0256202913550292 0.0256052944963532 0.0256023841101196 0.0256096530392759
##                 34                 35                 36                 37
## 0.0256316239614358 0.0256069934822407 0.0256181986467813 0.0256048391904936
##                 38                 39                 40
## 0.0255966394022495 0.0256007667434439 0.0256022634952117
```

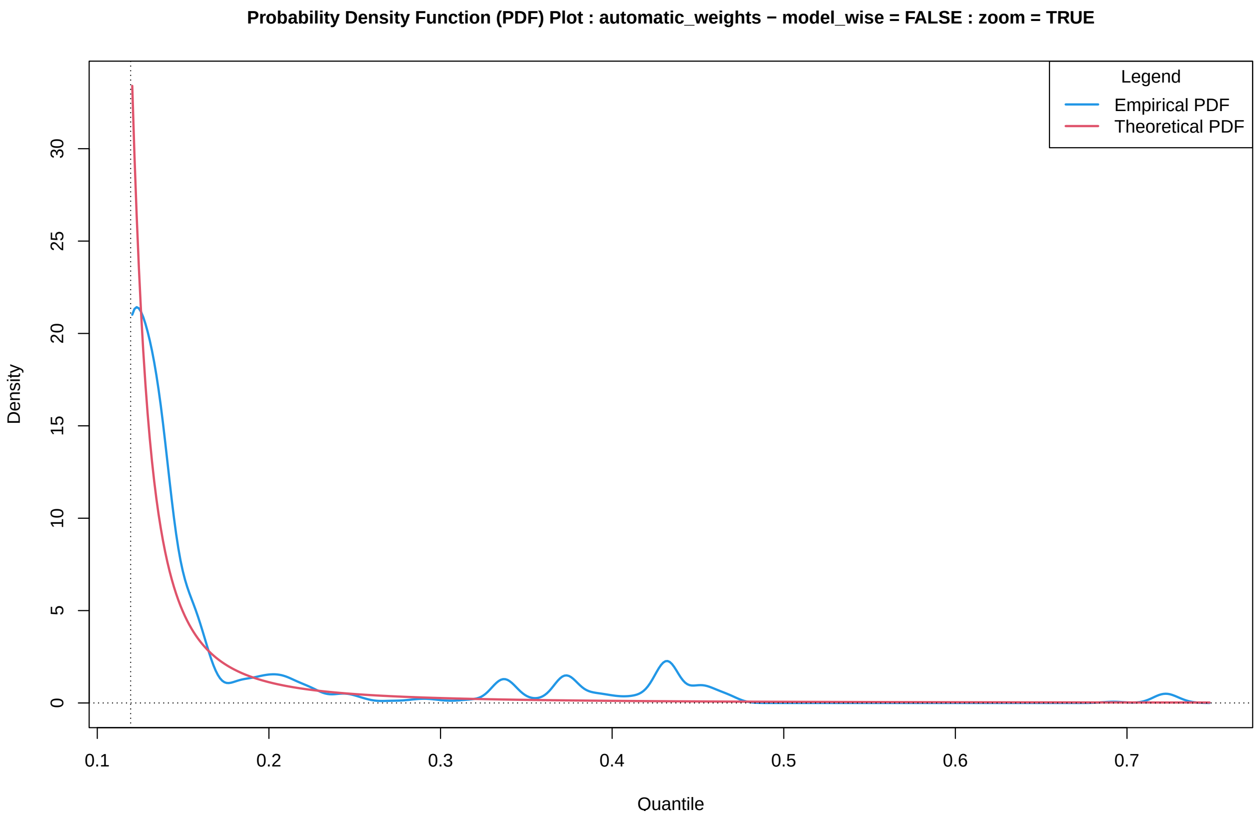
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                  2                  3                  4                  5
## 0.0258918432602556 0.0258253956198317 0.0257856916153414 0.0257615392573764
##                  6                  7                  8                  9
## 0.0257317751400379 0.0257183450450760 0.0257098036139139 0.0256977379450062
##                 10                 11                 12                 13
## 0.0256887605781087 0.0256805439886355 0.0256600729788599 0.0256591233373517
##                 14                 15                 16                 17
## 0.0256383315498259 0.0256559020576499 0.0256550184149079 0.0256198703937023
##                 18                 19                 20                 21
## 0.0256310715334691 0.0256033684952685 0.0255992907282010 0.0256300846969197
##                 22                 23                 24                 25
## 0.0256070156485279 0.0256146145603230 0.0256240818269178 0.0255923184537747
##                 26                 27                 28                 29
## 0.0256116146107105 0.0256241230085143 0.0255697224079475 0.0255599238813308
##                 30                 31                 32                 33
## 0.0255580589636280 0.0256112434763813 0.0256353703148906 0.0255806680447452
##                 34                 35                 36                 37
## 0.0254858562693847 0.0255574836542613 0.0255081703860763 0.0255617340124909
##                 38                 39                 40
## 0.0256310734759861 0.0256392855902689 0.0255840711641010
```

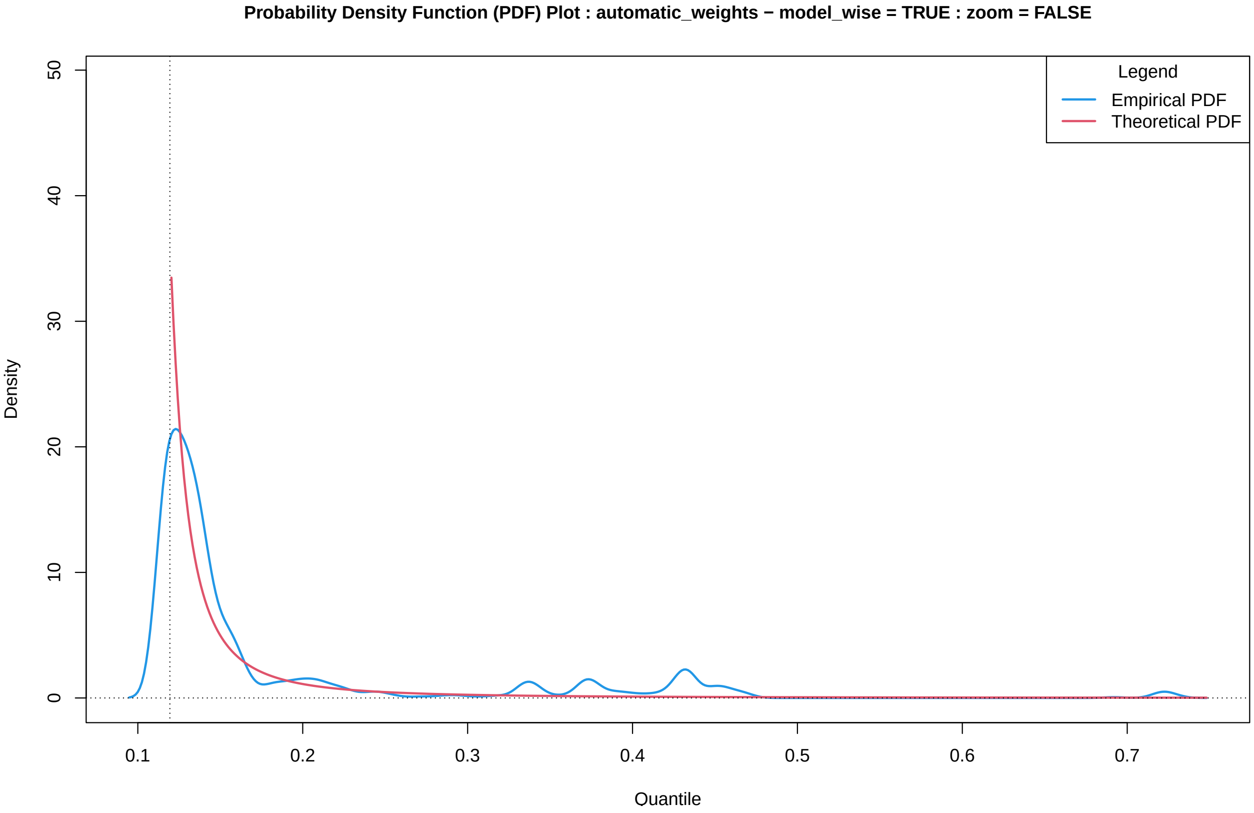
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



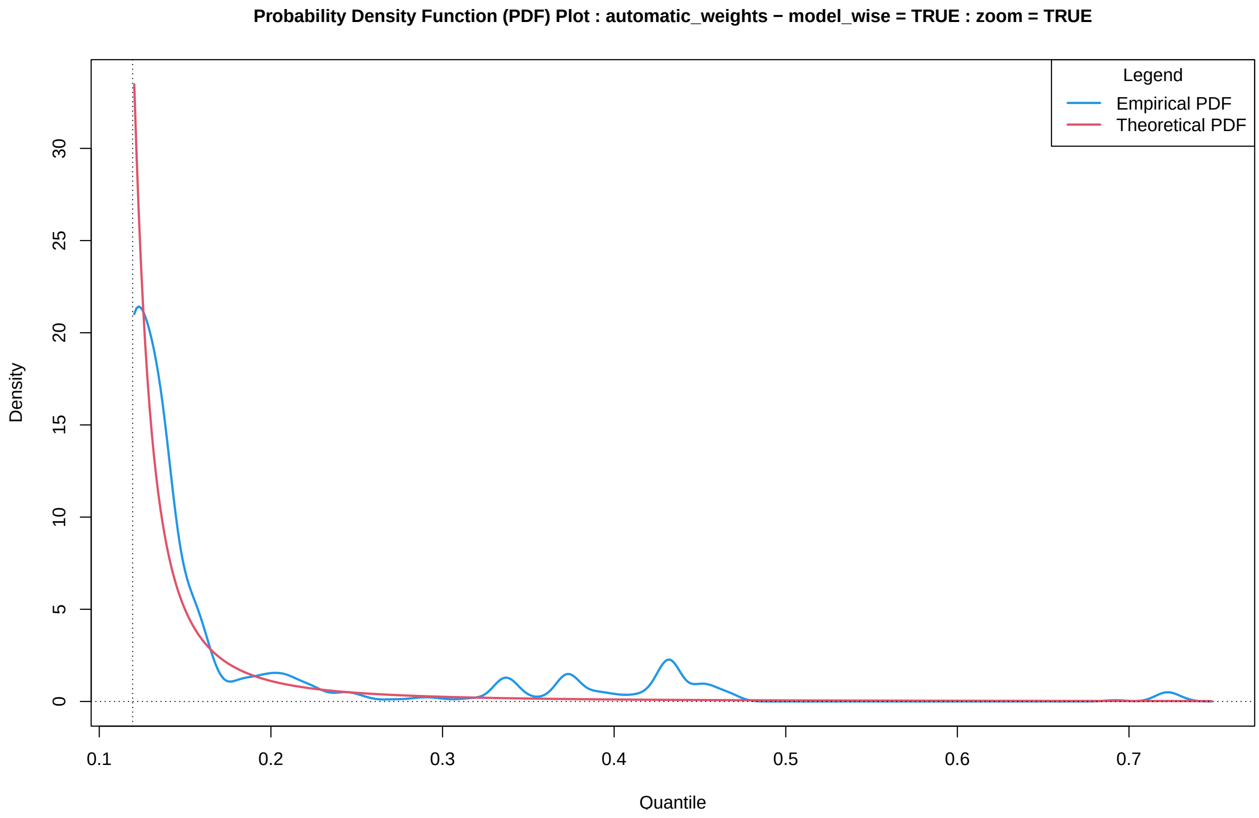
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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



```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")
```



```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")
```



```r
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")
```

```r
alpha <- 10^(-14)
```

```r
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 15684067949.4583    NA
```

```r
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 52152861272.9493    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 0.730388515002199    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  -22394119152.6843 16018848645.4187 54431816443.5217
## 3  -9819022396.06219 5593166112.37945 21005354620.8211
## 4  -5688739932.17032 2736566516.66393 11161872965.4982
## 5  -4118111955.30355  1771716669.5336 7661545294.37075
## 6  -4688740896.46386 1757840980.22361 8204422856.91108
## 7   -3687913805.2733 1296721393.39905  6281356592.0714
## 8  -5465567399.37965 1728737420.34991 8923042240.07947
## 9  -4455377333.41015 1326467672.61868 7108312678.64752
## 10 -3704136200.57024 1044113310.74081 5792362822.05186
## 11 -4528536041.45221  1186284060.5032 6901104162.45862
## 12 -3757297794.61984 955224472.092455 5667746738.80475
## 13 -3981735240.62832 967247245.786242 5916229732.20081
## 14 -3413619317.64924 794118954.650671 5001857226.95059
## 15 -4361124394.04268 946795101.492749 6254714597.02818
## 16 -6477653451.22694 1353902566.83257 9185458584.89208
## 17 -2062817085.99914 429565039.029866 2921947164.05888
## 18 -4768885014.97815 953582742.971247 6676050500.92064
## 19 -3639631140.03466 713441932.156436 5066515004.34753
## 20 -1450024621.66404 273338340.233632  1996701302.1313
## 21 -7979537918.45668 1455651423.15401 10890840764.7647
## 22  -3356105460.4504 577420139.403482 4510945739.25736
## 23 -5154361915.74938 864709086.739381 6883780089.22814
## 24  -20584795026.089 3257117608.02349  27099030242.136
## 25 -2796168891.89829 460659266.947198 3717487425.79269
## 26 -7502106222.91864 1196813777.43556 9895733777.78976
## 27 -15224630328.5002 2294151684.91302 19812933698.3262
## 28 -9630465718.83013 1406723884.68745  12443913488.205
## 29 -7071092218.14829 1028892667.96551  9128877554.0793
## 30 -3981888204.54443 581073284.202332  5144034772.9491
## 31 -14523187604.0564 2038056850.15837 18599301304.3732
## 32 -13093562172.5842 1742784365.39564 16579130903.3755
## 33 -4424075984.20879 608700209.280949 5641476402.77069
## 34 -634376675.059219 85194583.0647062 804765841.188631
## 35 -16275065476.4684 2043121957.91347 20361309392.2954
## 36 -2968551485.18791 393487648.796844  3755526782.7816
## 37 -15838199034.3311 1903835109.80814 19645869253.9474
## 38 -55867008549.1698 6137742453.56624 68142493456.3022
## 39 -12433440069.3417 1526144576.33745 15485729222.0166
## 40 -16908165669.2011 1829525455.44511 20567216580.0913
```

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

```
## [1] -55867008549.1698  68142493456.3022
```

```
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
```

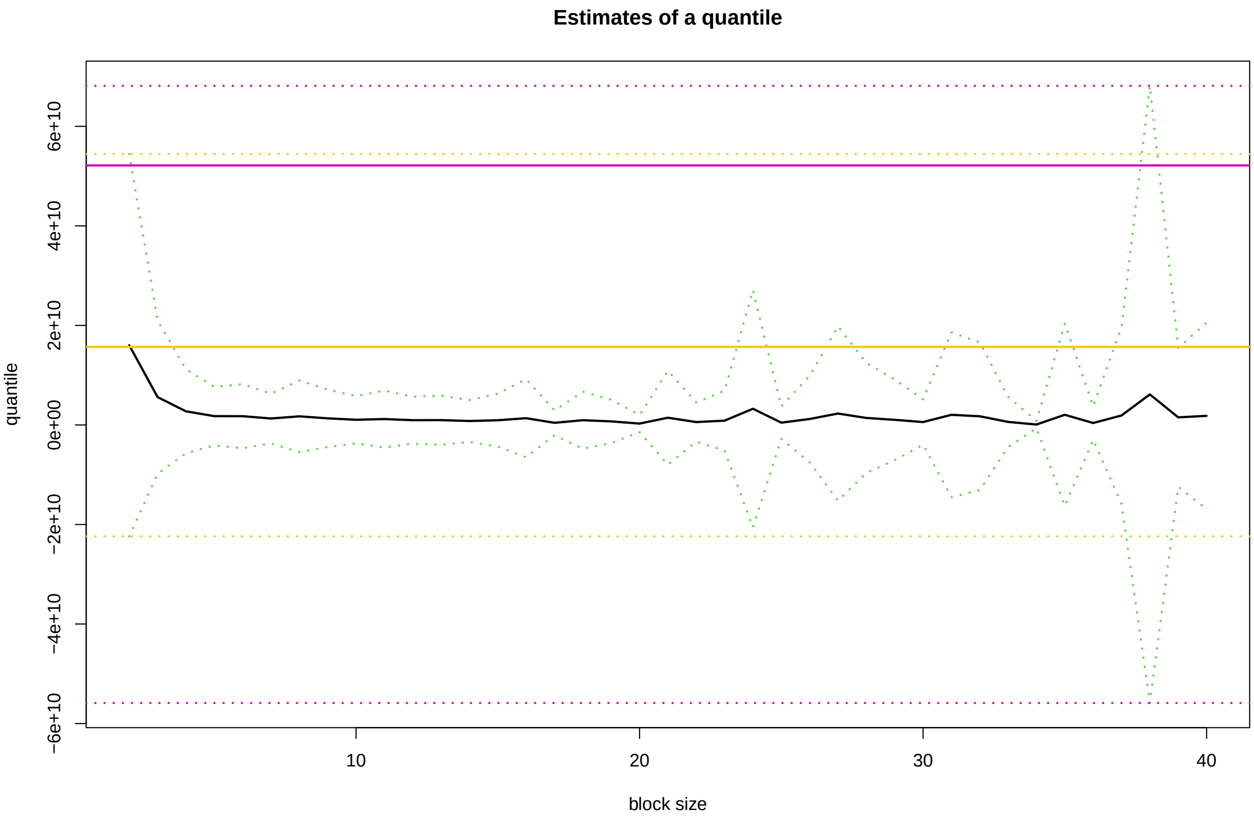
```
##               lower         estimate            upper
## 2 -22394119152.6843 16018848645.4187 54431816443.5217
```

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

```
## [1] -22394119152.6843  54431816443.5217
```

```
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
quantile
block size
−6e+10
−4e+10
−2e+10
0e+00
2e+10
4e+10
6e+10
10
20
30
40