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 <- 100000
```

```
loc <- 1
scale <- 0.5
shape <- 0.1
set.seed(1122)
x <- 10^(4)*generate_gev_sample(n = n, loc = loc, scale = scale, shape = shape)
```

```
nlargest <- 1000
```

```
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.
```

```
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]  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
## 8  42307.6041461125  7575.33845637881  0.084886007732018876
## 9  38695.7903610545 10105.79377438252 -0.034721885785258644
## 10 40018.9787293499  8750.84020793509  0.033433604293382345
## 11 40317.2775396159  9130.22700070248  0.007467319567499974
## 12 42019.4486380160  8113.64305712375  0.046847051264966084
## 13 41289.5821417095  8221.52278920339  0.046949124253755199
## 14 39471.9341320387  9374.56394848612  0.003909095611221790
## 15 39315.5379086070  9433.76276964432  0.000242462438563962
## 16 40533.0866004263  8799.47029281573  0.022956566917236354
## 17 40517.2610558888  9255.56585790857 -0.000552669808677966
## 18 38220.4950006056 11011.71983222787 -0.059579087959913082
## 19 38594.3584743171  9625.46392790441 -0.008800326544034551
## 20 37528.9885889457 11042.03946136243 -0.054543063738776119
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                           loc_star       scale_star          shape_star
## identic_weights     39910.026408976 9264.61164431350 0.00680724601861417
## pessimistic_weights             NaN              NaN 0.00838703203416191
## automatic_weights   40578.322821552 7773.90497565418 0.07561985441901581
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 0.00274513898928342
##
## $gradient_value
## [1] 3.88578058618805e-16
##
## $function_reduction
## [1] 0.0299235894420111
##
## $number_iterations
```

```
## [1] 1547
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.000761397128384956
##
## $gradient_value
## [1] 1.35751418063323e-05
##
## $function_reduction
## [1] 0.0343080764403879
##
## $number_iterations
## [1] 3648
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                      8                      9                     10
##  0.00000000000000e+00  0.00000000000000e+00 -2.22044604925031e-16
##                     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  1.00000000000000e+00
##                     20
##  0.00000000000000e+00
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                  8                  9                 10                 11
## 0.0831041728601264 0.0737356958628789 0.0789364037216100 0.0769131010960602
##                 12                 13                 14                 15
## 0.0800023459850406 0.0800105124803876 0.0766399133769666 0.0763594174802623
##                 16                 17                 18                 19
## 0.0781137013368381 0.0762987257772059 0.0719254251063759 0.0756720280241508
##                 20
## 0.0722885568920966
```

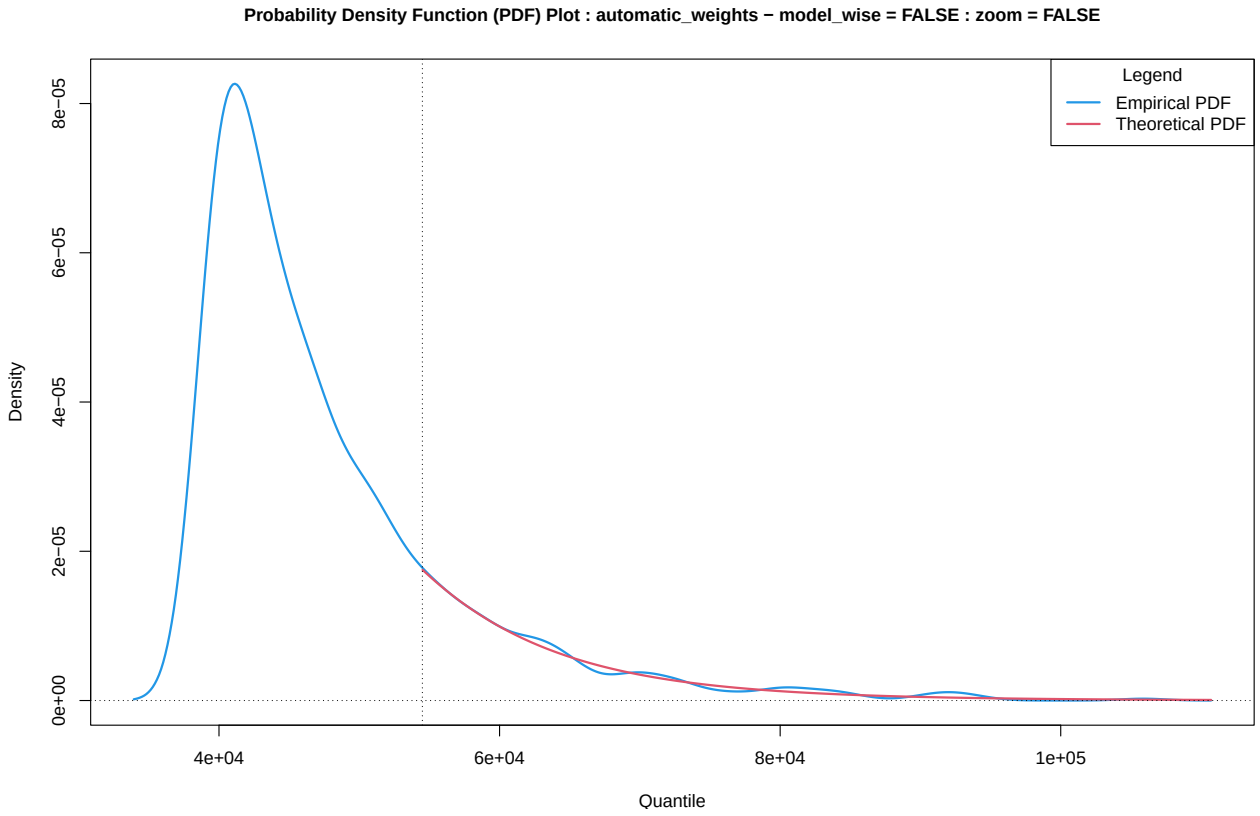
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##   8   9  10  11  12  13  14  15  16  17  18  19  20
## NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
```

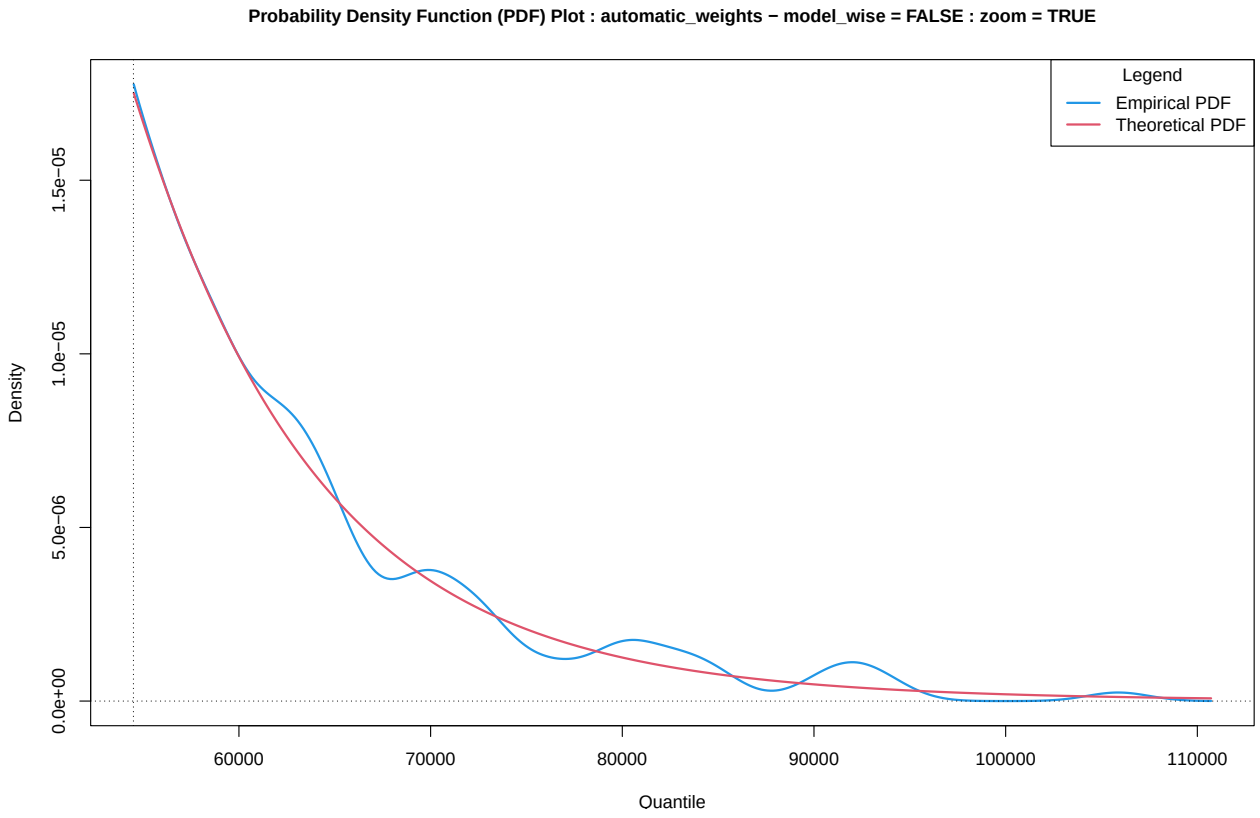
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##   8   9  10  11  12  13  14  15  16  17  18  19  20
## NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
```

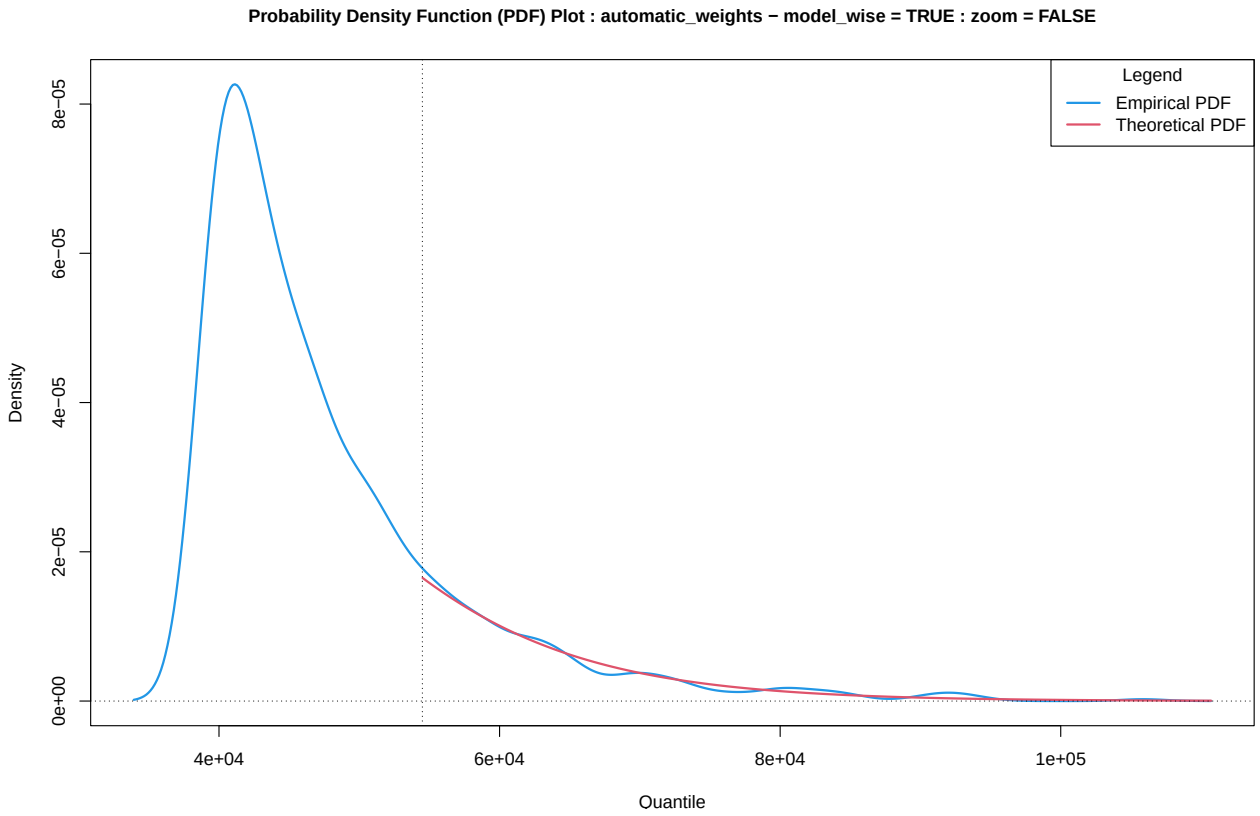
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



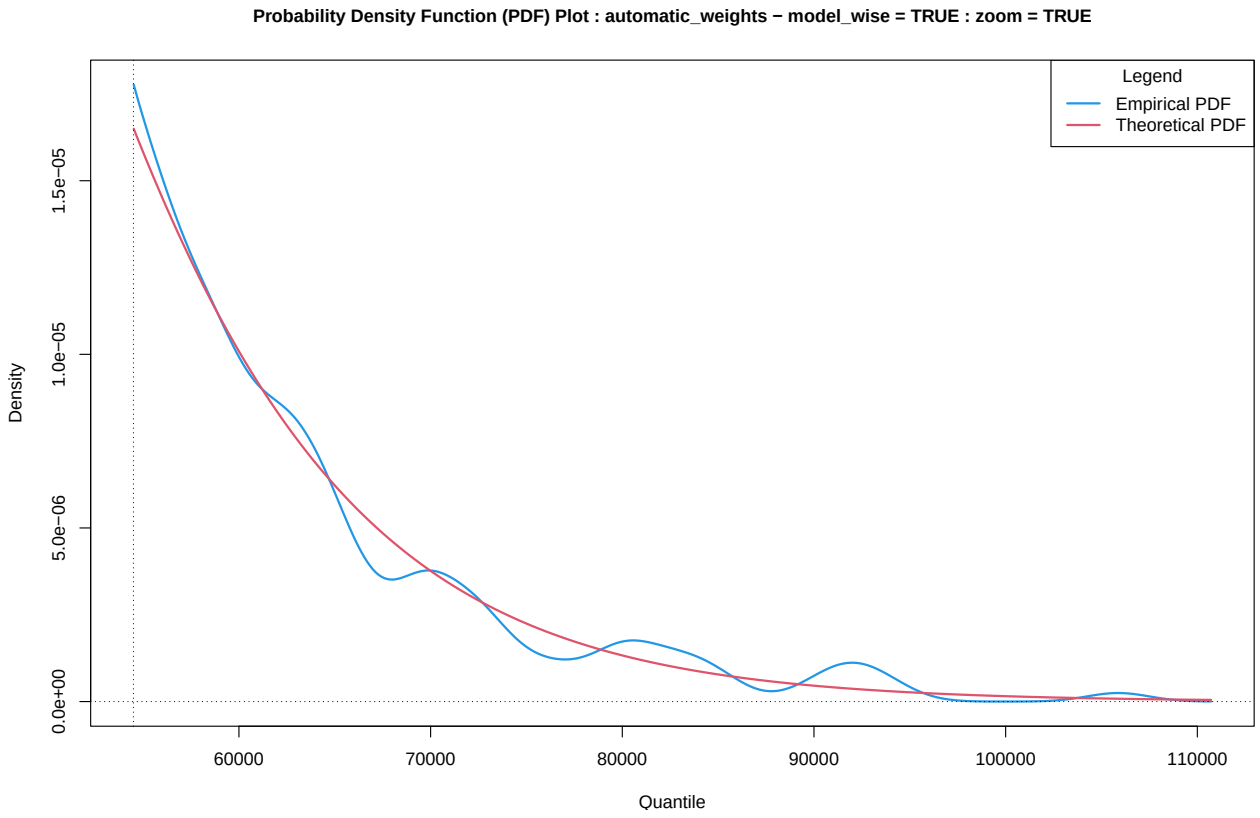

```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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

```
results_mw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                  alpha = alpha,
                                                  confidence_level = 0.95,
                                                  do.ci = TRUE,
                                                  estimator_type = estimator_types[1])

results_mw
```

```
##   lower         estimate upper
## 1    NA 274689.135334543    NA
```

```
results_pw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                  alpha = alpha,
                                                  confidence_level = 0.95,
                                                  do.ci = TRUE,
                                                  estimator_type = estimator_types[4])
```

```
results_pw
```

```
##   lower         estimate upper
## 1    NA 768472.370653925    NA
```

```
quantile(x = x, probs = 1 - alpha)
```

```
##             100%
## 105868.368546653
```

```
true_rl <- calculate_gev_inverse_cdf(p = 1 - alpha, loc = loc, scale = scale, shape = shape)
true_rl
```

```
## [1] 121.604364466665
```

```
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
## 8  -2273751.03599685 1184812.46454244 4643375.96508173
## 9    23174.2447834694  224238.59436895 425302.943954431
## 10 -633505.612906617  457772.01299453 1549049.63889568
## 11 -267716.887207007 301443.589860912  870604.06692883
## 12 -1042049.65606288 524596.982698748 2091243.62146037
## 13 -1122867.42303492 513131.365123911 2149130.15328274
## 14 -256476.496443485 251473.433723506 759423.363890498
## 15 -255854.040993292  251234.06324963 758322.167492552
## 16 -495396.053218785 287486.756922882 1070369.56706455
## 17 -213126.327626711 218902.271764158 650930.871155028
## 18  33218.7263423406  146691.61636566 260164.506388979
## 19 -161701.619297798 201496.663344882 564694.945987562
## 20  17901.2435689669 149130.046538253 280358.849507539
```

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

```
## [1] -2273751.03599685  4643375.96508173
```

```
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
## 19 -161701.619297798 201496.663344882 564694.945987562
```

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

```
## [1] -161701.619297798  564694.945987562
```

```r
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 = true_rl, col = 4, lwd = 2)
abline(h = results_mw[2], col = 7, lwd = 2)
abline(h = results_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)
```