

# 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"))

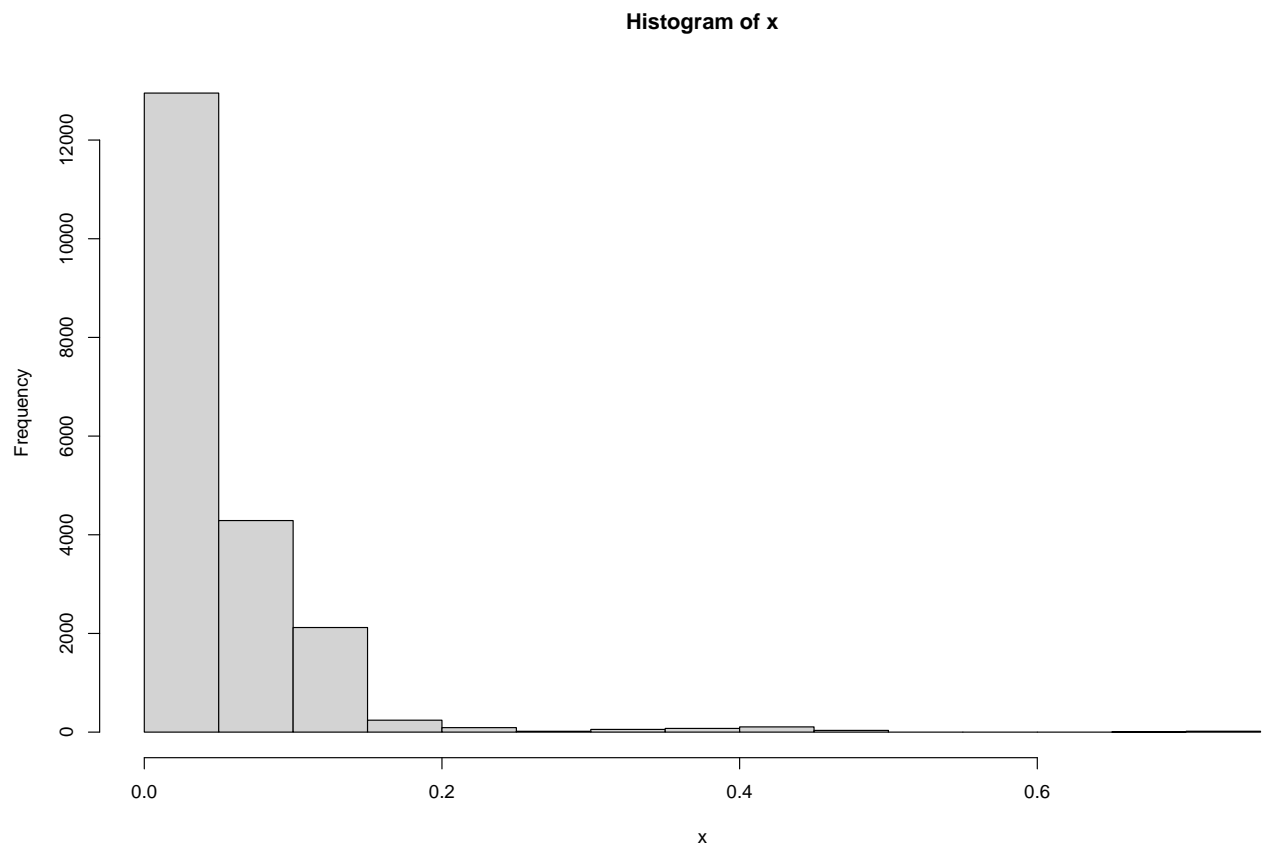
## Rows: 20002 Columns: 25
## -- Column specification -----
## Delimiter: ","
## dbl (25): version_major, version_minor, status, timestamp, latitude, longitu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` 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)))

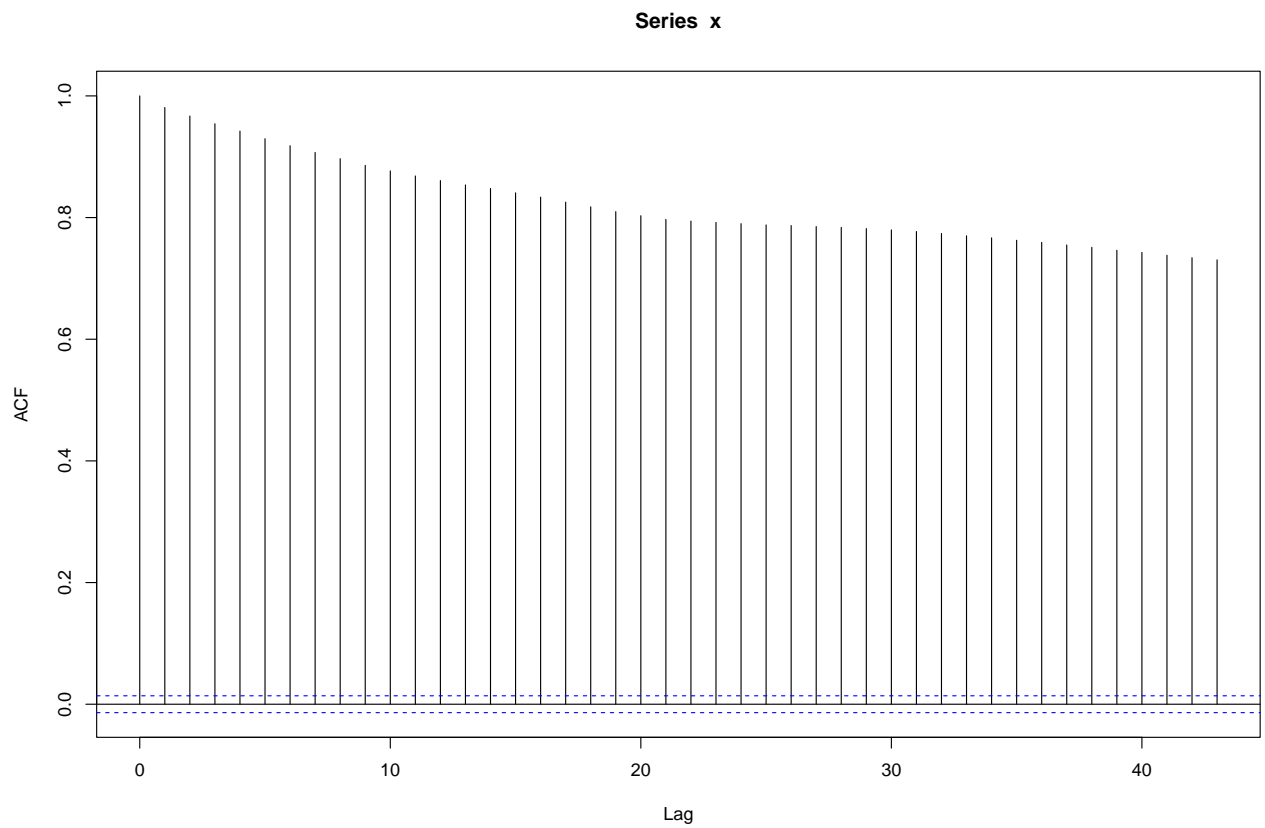
longitude_Gnss_map_matching_errors <- Gnss_imar$longitude[-1] - Gnss_map_matching$longitude

coefficient <- 10^(4)
x <- coefficient*abs(longitude_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 0.119631497151790 0.01014176570616628 0.956722983287248
## 3 0.117188570020768 0.00790568792520071 0.905385120336903
## 4 0.115740889965141 0.00646303138317857 0.901615632379177
## 5 0.114738919628887 0.00557386985577229 0.891592091545506
## 6 0.113571334814682 0.00494422553557732 0.891429545535832
## 7 0.113096446012622 0.00450785263935304 0.879744153845632
## 8 0.112763380859748 0.00399370448900549 0.904701169436551
## 9 0.112138908663039 0.00384519281442667 0.887590296378895
## 10 0.111856088991415 0.00363535896621732 0.886064956004230
## 11 0.111611664418508 0.00331434658269994 0.890882274507027
## 12 0.110800190487461 0.00337793722899094 0.875561907521457
## 13 0.110875813017731 0.00305370155946316 0.889233129667844
## 14 0.110003153040959 0.00306359775738704 0.882755061442448
## 15 0.110695360295408 0.00279132826161237 0.887694873080802
## 16 0.110989104240870 0.00240117173501272 0.920883205438043
## 17 0.109047655405595 0.00300408680024902 0.848178144639363
## 18 0.109673912625266 0.00246098039522860 0.889261570417668
## 19 0.108607868169546 0.00259367210354268 0.881176438990934
## 20 0.108439749258483 0.00282799954942772 0.840651053382885
## 21 0.109650257950319 0.00217632177420957 0.910999751503950
## 22 0.108749826650790 0.00239738632878670 0.874441256703715
## 23 0.108968050138820 0.00219668592785877 0.891545270418111
## 24 0.109446944865619 0.00189951556478164 0.940581568508024
## 25 0.108190966005075 0.00223510599327559 0.870894737869011
## 26 0.108919421885922 0.00194893404896263 0.906848962354428
## 27 0.109404149116376 0.00175799690029414 0.934921981681594
## 28 0.107414460923344 0.00197027456220046 0.919184315701300
## 29 0.106886470536705 0.00215220343302379 0.896746495381974
## 30 0.106732544465036 0.00226338145018104 0.872205468372028
## 31 0.108888227868922 0.00165497747540811 0.929019008725688
## 32 0.109902419697542 0.00152633606277656 0.930727734970948
## 33 0.107742061580202 0.00181932782972015 0.883966788783612
## 34 0.103991037282217 0.00267799175833500 0.801936955650600
## 35 0.106762295713534 0.00172747905426040 0.926541146535516
## 36 0.104857622797681 0.00215711742652299 0.861953672422285
## 37 0.106963863171517 0.00163245614816882 0.927488261482935
## 38 0.109662351743917 0.00131995282183322 0.972626873677816
## 39 0.109977299396664 0.00147190162973292 0.924487841332315
## 40 0.107837332772911 0.00153185700668312 0.929060900705827
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##               loc_star          scale_star          shape_star
## identic_weights    0.110062002862334 0.00303632601244942 0.897879553862055
## pessimistic_weights 0.110071679496250 0.00303962418969507 0.898885078195306
## automatic_weights  0.119631497149179 0.01014176570616286 0.972626873675936
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 182.666313769475
##
## $gradient_value
## [1] 1.02318153949454e-12
##
```

```
## $function_reduction
## [1] 530.813176321496
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 181.536880570035
##
## $gradient_value
## [1] 2.27373675443232e-12
##
## $function_reduction
## [1] 543.732049359678
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##           2           3           4           5
## 0.999999999998181 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$peessimistic_weights_pw_shape
```

```
##           2           3           4           5
## 0.0271814164982303 0.0258211948341299 0.0257240453680532 0.0254674873070577
##           6           7           8           9
## 0.0254633480050421 0.0251675305473985 0.0258035404427316 0.0253657752905124
##          10          11          12          13
## 0.0253271133431378 0.0254494164648944 0.0250624935337121 0.0254074812791729
##          14          15          16          17
## 0.0252434218498332 0.0253684280983421 0.0262244909959024 0.0243854997760901
##          18          19          20          21
## 0.0254082038972674 0.0252036034547234 0.0242026369698280 0.0259665790769412
##          22          23          24          25
## 0.0250344229614315 0.0254662949185047 0.0267461919994235 0.0249457951618582
##          26          27          28          29
## 0.0258590206628350 0.0265952471491778 0.0261799763007841 0.0255990959119072
##          30          31          32          33
## 0.0249785138144826 0.0264387185710032 0.0264839337224731 0.0252740285340062
##          34          35          36          37
## 0.0232835591397930 0.0263732881671389 0.0247237473260672 0.0263982785350820
##          38          39          40
## 0.0276171626115553 0.0263191913150001 0.0264398261644759
```

```
gev_mixture_model$peessimistic_weights_pw_scale
```

```
##           2           3           4           5
## 0.0258238226531376 0.0257661430891125 0.0257289981943274 0.0257061311267592
##           6           7           8           9
## 0.0256899505018549 0.0256787425493517 0.0256655432648514 0.0256617319150641
##          10          11          12          13
## 0.0256563477800113 0.0256481130964465 0.0256497441283931 0.0256414289145497
##          14          15          16          17
## 0.0256416826684609 0.0256347021707813 0.0256247025752526 0.0256401567527875
##          18          19          20          21
## 0.0256262352002137 0.0256296358147517 0.0256356422455595 0.0256189415095949
##          22          23          24          25
## 0.0256246055755275 0.0256194632229682 0.0256118510088986 0.0256204475433299
##          26          27          28          29
## 0.0256131167390275 0.0256082267104078 0.0256136633419167 0.0256183236306739
##          30          31          32          33
## 0.0256211719834322 0.0256055887015052 0.0256022949742627 0.0256097973349163
##          34          35          36          37
## 0.0256317969879095 0.0256074452144127 0.0256184495192590 0.0256050120361565
##          38          39          40
## 0.0255970116348655 0.0256009013657817 0.0256024363234874
```

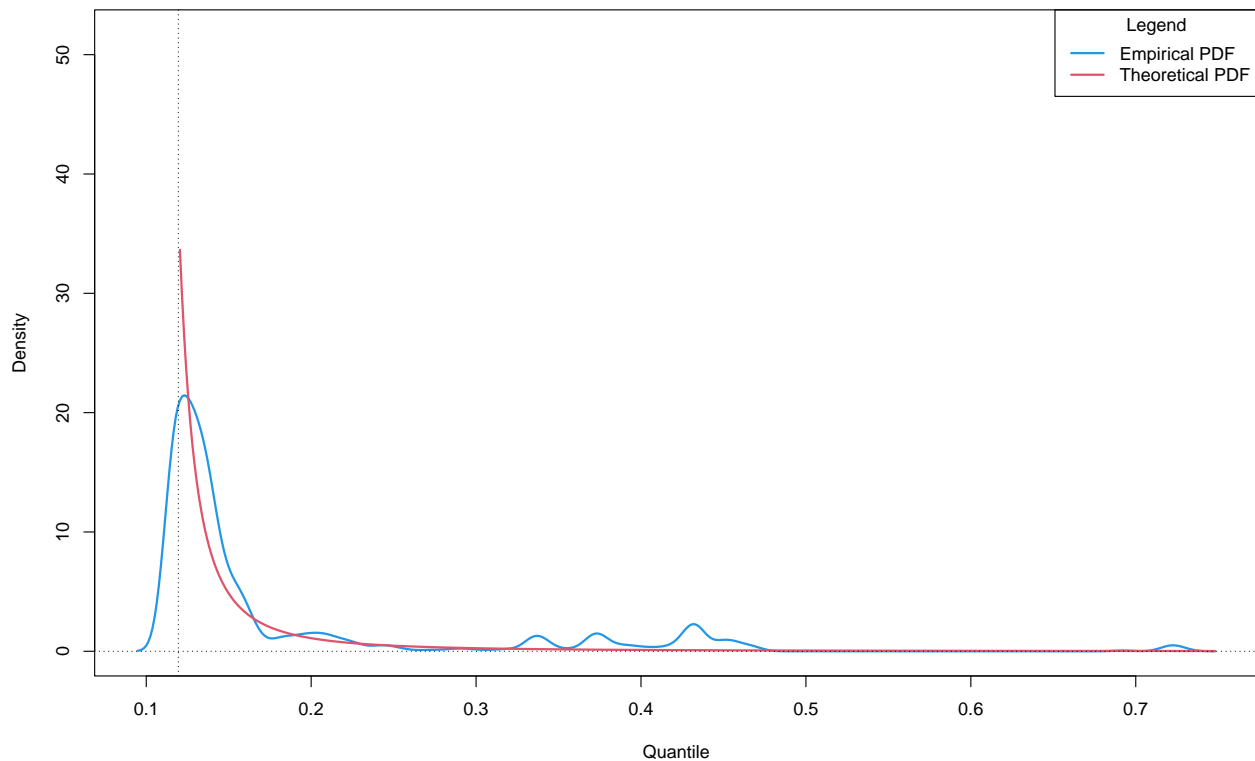
```
gev_mixture_model$peessimistic_weights_pw_loc
```

```
##           2           3           4           5
## 0.0258874498921343 0.0258242859223406 0.0257869276665854 0.0257611028700010
##           6           7           8           9
## 0.0257310421501279 0.0257188256673104 0.0257102610490749 0.0256942107178940
##          10          11          12          13
## 0.0256869449171603 0.0256806671638694 0.0256598364248780 0.0256617769600079
##          14          15          16          17
```

```
## 0.0256393927226566 0.0256571466402853 0.0256646843787970 0.0256149060438828
##          18          19          20          21
## 0.0256309525878315 0.0256036434119145 0.0255993393170748 0.0256303463031502
##          22          23          24          25
## 0.0256072783242448 0.0256128670436124 0.0256251358480762 0.0255929714223001
##          26          27          28          29
## 0.0256116215649193 0.0256240392246537 0.0255731060637082 0.0255596072734794
##          30          31          32          33
## 0.0255556732863177 0.0256108226480216 0.0256368101109842 0.0255814852024833
##          34          35          36          37
## 0.0254857081731923 0.0255564336108144 0.0255078032910183 0.0255615854753792
##          38          39          40
## 0.0256306562731426 0.0256387298594860 0.0255839224971899
```

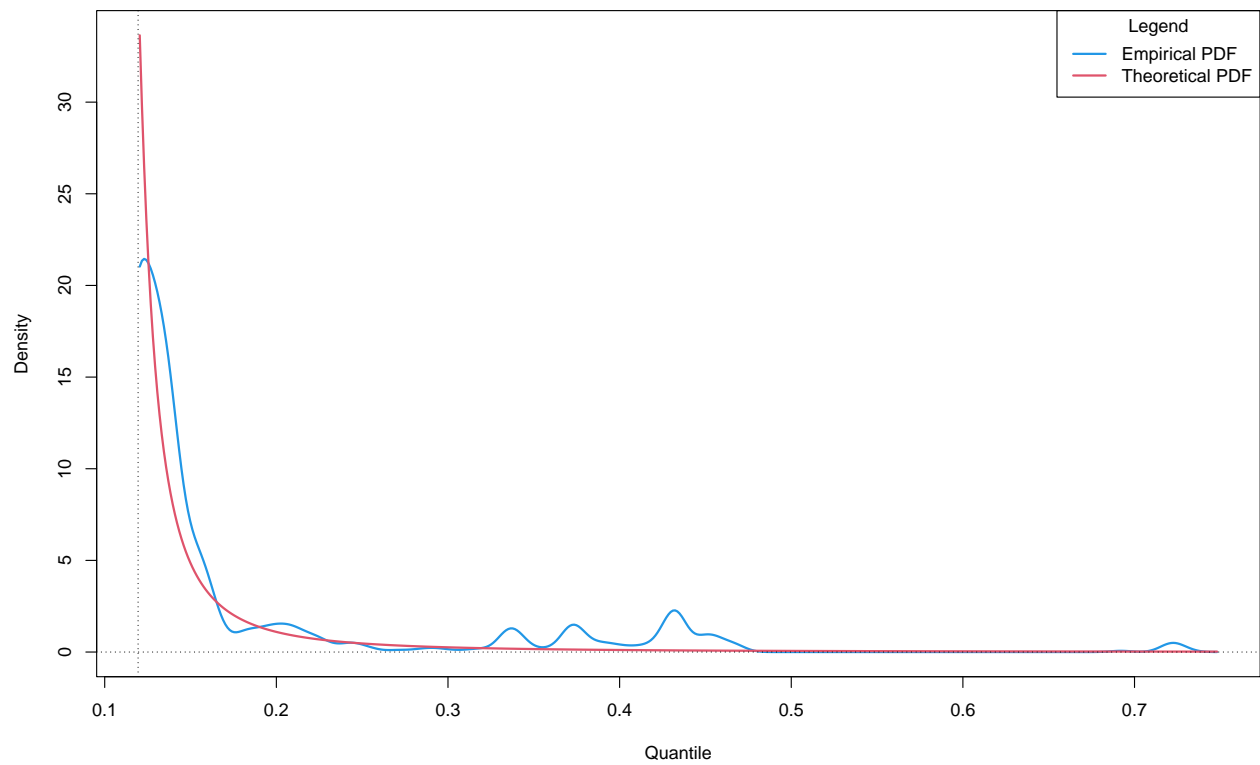
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                             type = "automatic_weights",
                             model_wise = FALSE,
                             zoom = FALSE,
                             xlab = "Quantile",
                             ylab = "Density",
      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

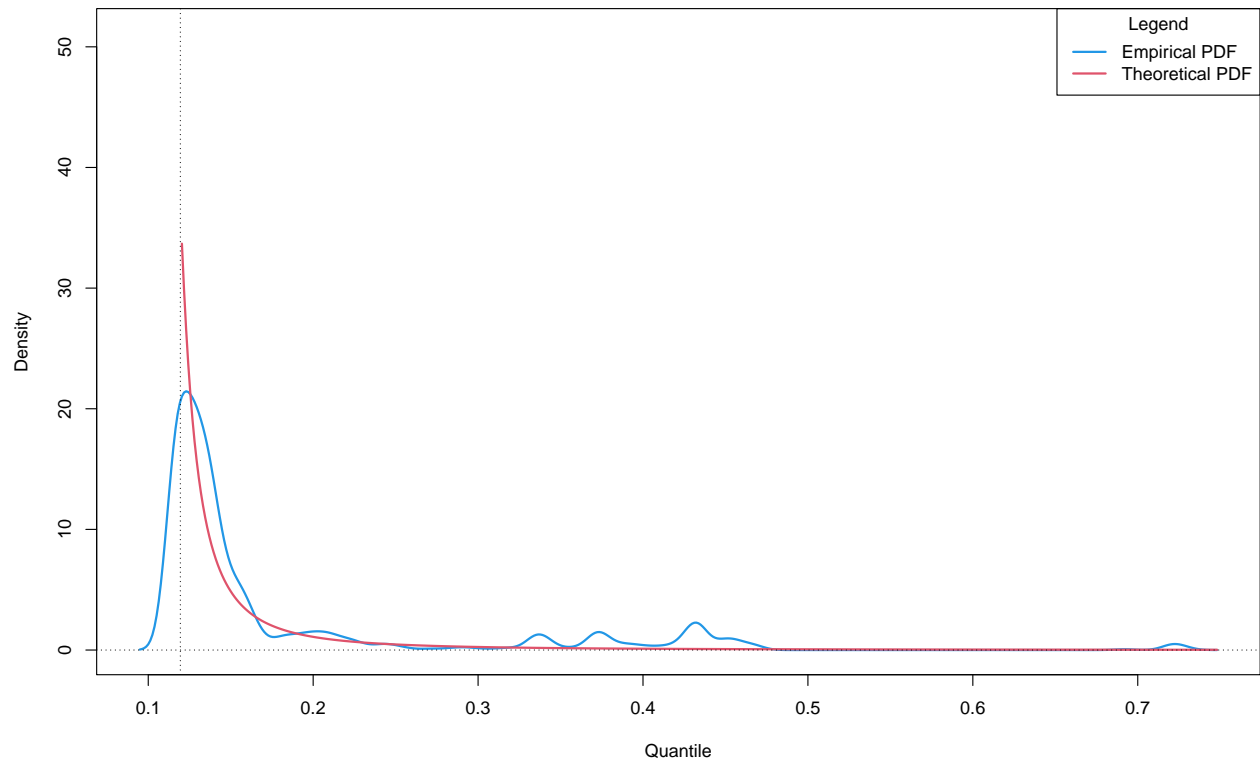
Probability Density Function (PDF) Plot : automatic\_weights – model\_wise = FALSE : zoom = TRUE



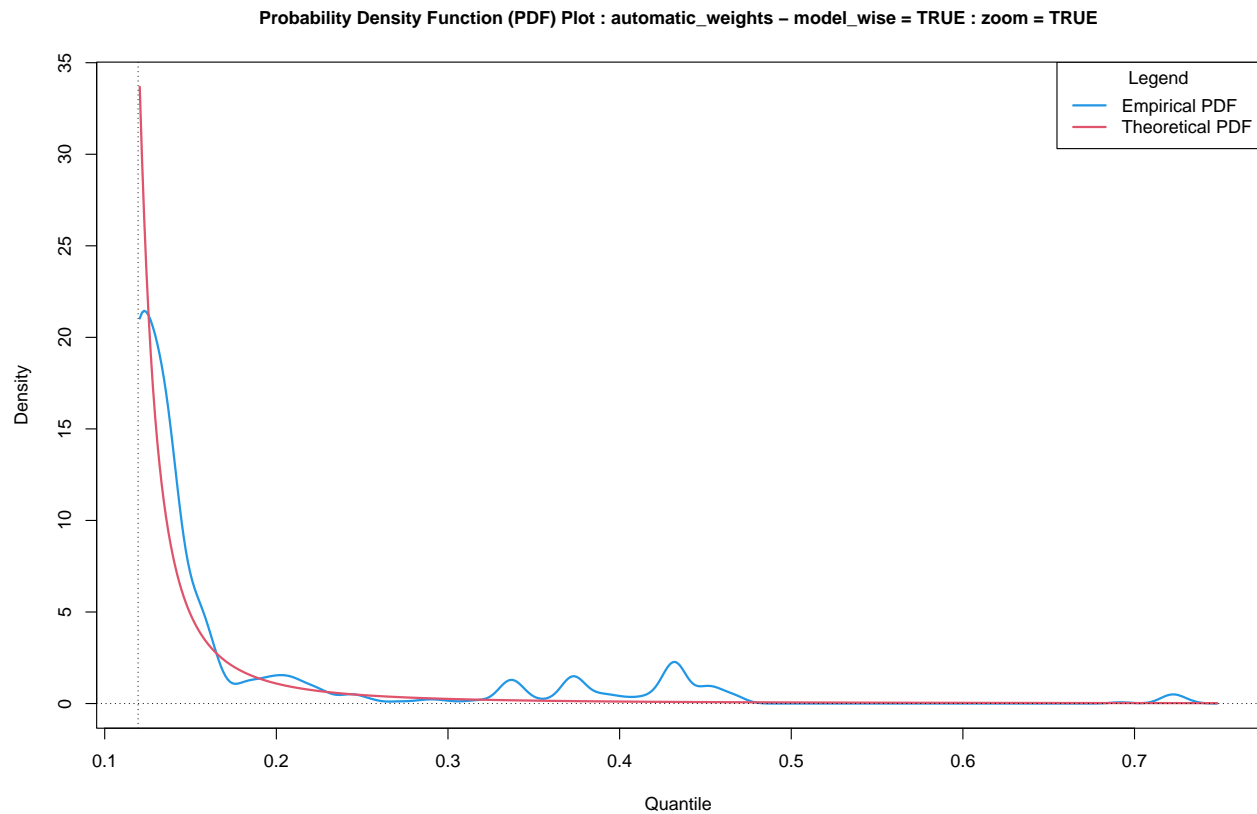
```
plot_gev_mixture_model_pdf(gev_mixture_model,  
    type = "automatic_weights",  
    model_wise = TRUE,  
    zoom = FALSE,  
    xlab = "Quantile",  
    ylab = "Density",  
    main = "Probability Density Function (PDF) Plot")
```



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



```
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)
```

```
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 29013289495.9469 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 45939402389.0617    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 -25016264978.7777 17792703633.234 60601672245.2457
## 3 -8874326969.54842 5090515174.05142 19055357317.6513
## 4 -5708688131.9936 2743481191.254 11195650514.5016
## 5 -4466009328.35952 1907580072.56867 8281169473.49686
## 6 -4387756477.95035 1650233982.98988 7688224443.9301
## 7 -3869833170.42972 1358396323.83421 6586625818.09814
## 8 -6303634984.98874 1975133959.52238 10253902904.0335
## 9 -4469449489.09579 1321726360.14128 7112902209.37835
## 10 -3828332222.64209 1075090213.29472 5978512649.23154
## 11 -4393932420.56366 1152063933.57937 6698060287.7224
## 12 -3216193658.66695 821732657.255303 4859658973.17756
## 13 -4031685978.29264 983883333.701793 5999452645.69623
## 14 -3834049475.74453 889923810.903353 5613897097.55123
## 15 -4367334582.63498 948690538.403368 6264715659.44172
## 16 -10122844497.6394 2096814138.78948 14316472775.2183
## 17 -1572927179.718 332261386.419768 2237449952.55753
## 18 -4914144685.27403 982297067.885058 6878738821.04414
## 19 -4017021645.96175 783611104.336821 5584243854.63539
## 20 -1422783123.08766 268365138.086222 1959513399.26011
## 21 -7929408820.17814 1446745992.15493 10822900804.488
## 22 -3317160814.48014 570913324.790236 4458987464.06061
## 23 -5053951700.22338 848325364.802819 6750602429.82902
## 24 -19066950130.5083 3021110302.71886 25109170735.9461
## 25 -2931952039.63256 482727227.83872 3897406495.31
## 26 -7478508735.98614 1193328782.07896 9865166300.14405
## 27 -17722162795.8151 2651959224.34359 23026081244.5022
## 28 -11974562100.6781 1743160025.79463 15460882152.2673
## 29 -7004409369.67438 1019558855.66578 9043527081.00595
## 30 -4205341367.58964 613502176.582848 5432345720.75533
## 31 -14142459303.7481 1986369416.20009 18115198136.1483

```

```
## 32 -13485451484.1498 1794179889.44417 17073811263.0381
## 33 -4384395292.05501 603403910.66553 5591203113.38607
## 34 -634376675.059219 85194583.0647062 804765841.188631
## 35 -14504183544.5879 1822386430.43647 18148956405.4608
## 36 -2969082568.50822 393718181.911467 3756518932.33116
## 37 -15838199034.3311 1903835109.80814 19645869253.9474
## 38 -49501854638.1993 5448191709.13608 60398238056.4715
## 39 -12416333189.1975 1524676919.00096 15465687027.1994
## 40 -16908165669.2011 1829525455.44511 20567216580.0913
```

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

```
## [1] -49501854638.1993 60601672245.2457
```

```
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 -25016264978.7777 17792703633.234 60601672245.2457
```

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

```
## [1] -25016264978.7777 60601672245.2457
```

```
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)
```

