

# Modeling extreme values with a GEV mixture probability distributions

Application to localisation w.r.t. latitude

Pascal Alain Dkengne Sielenou

2023-10-26

```
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_gev_mixture_model_cdf.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$latitude[-1] - Gnss_map_matching$latitude

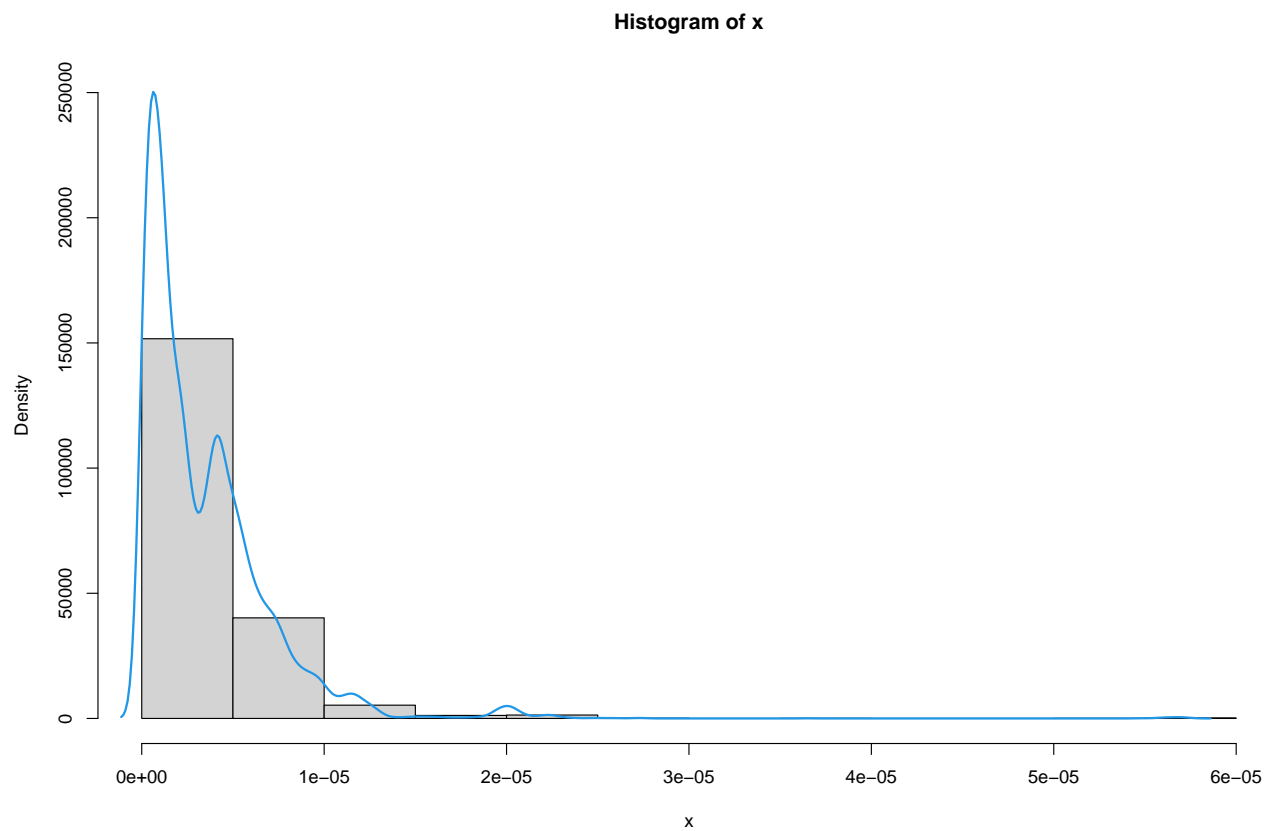
x <- abs(latitude_Gnss_map_matching_errors)
n <- length(x)

n
```

```
## [1] 20001
```

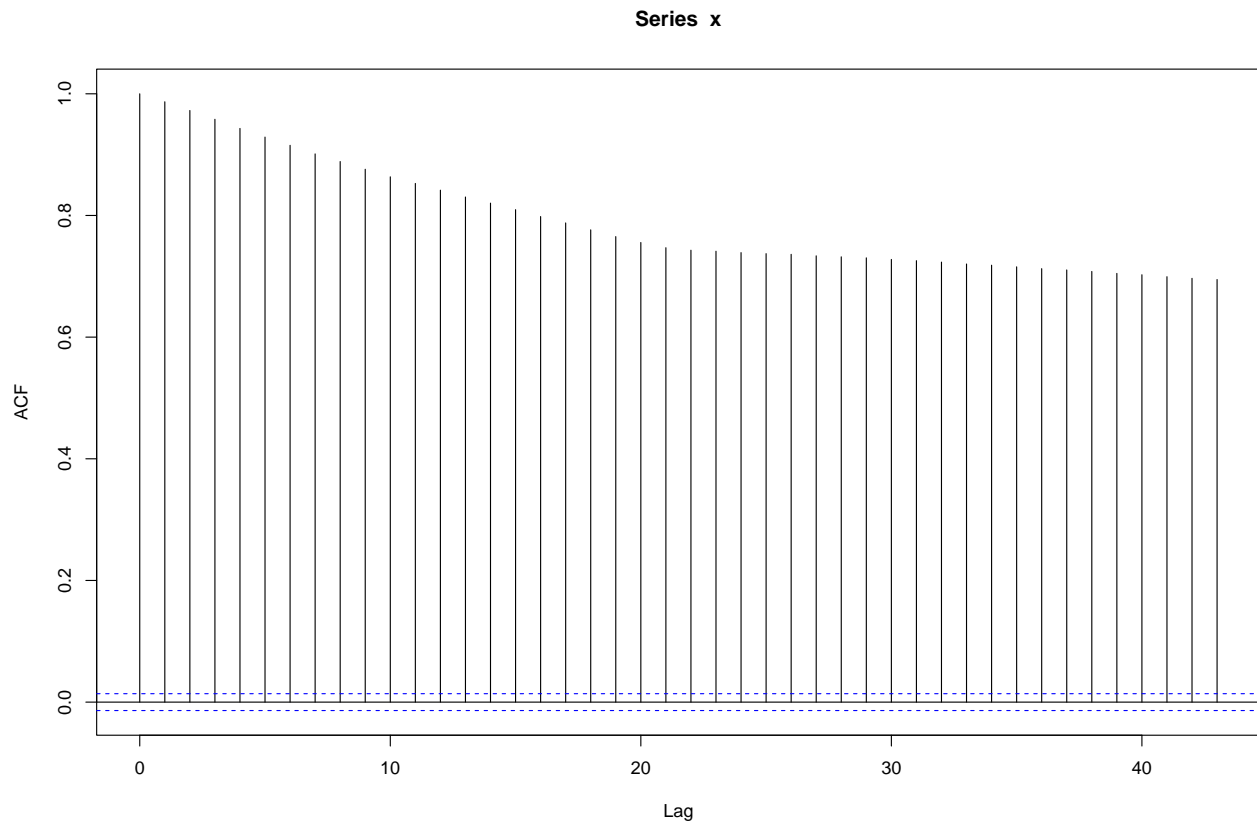
```
# Histogram of all data
```

```
dens_x <- density(x)  
hist(x, prob = TRUE, ylim = range(dens_x$y))  
lines(dens_x, lwd = 2, col = 4)
```



```
# Autocorrelation function of all data
```

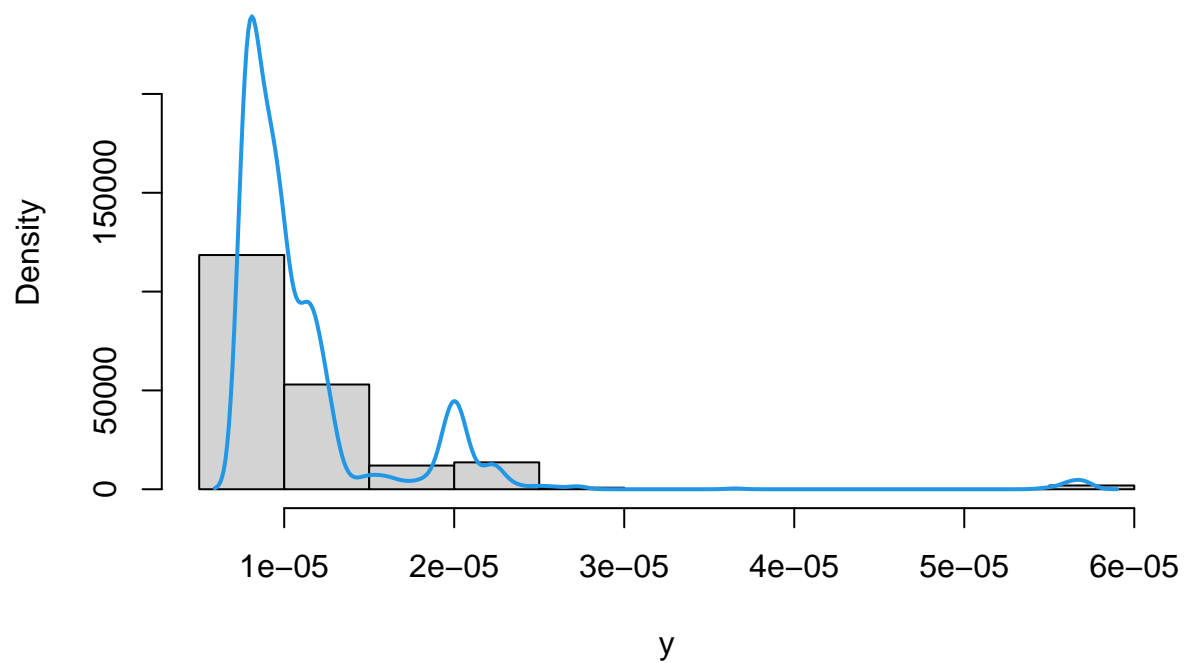
```
acf(x)
```



```
# Histogram of the largest data
```

```
nlargest <- 2000  
y <- extract_nlargest_sample(x, n = nlargest)  
dens_y <- density(y)  
hist(y, prob = TRUE, ylim = range(dens_y$y))  
lines(density(y), lwd = 2, col = 4)
```

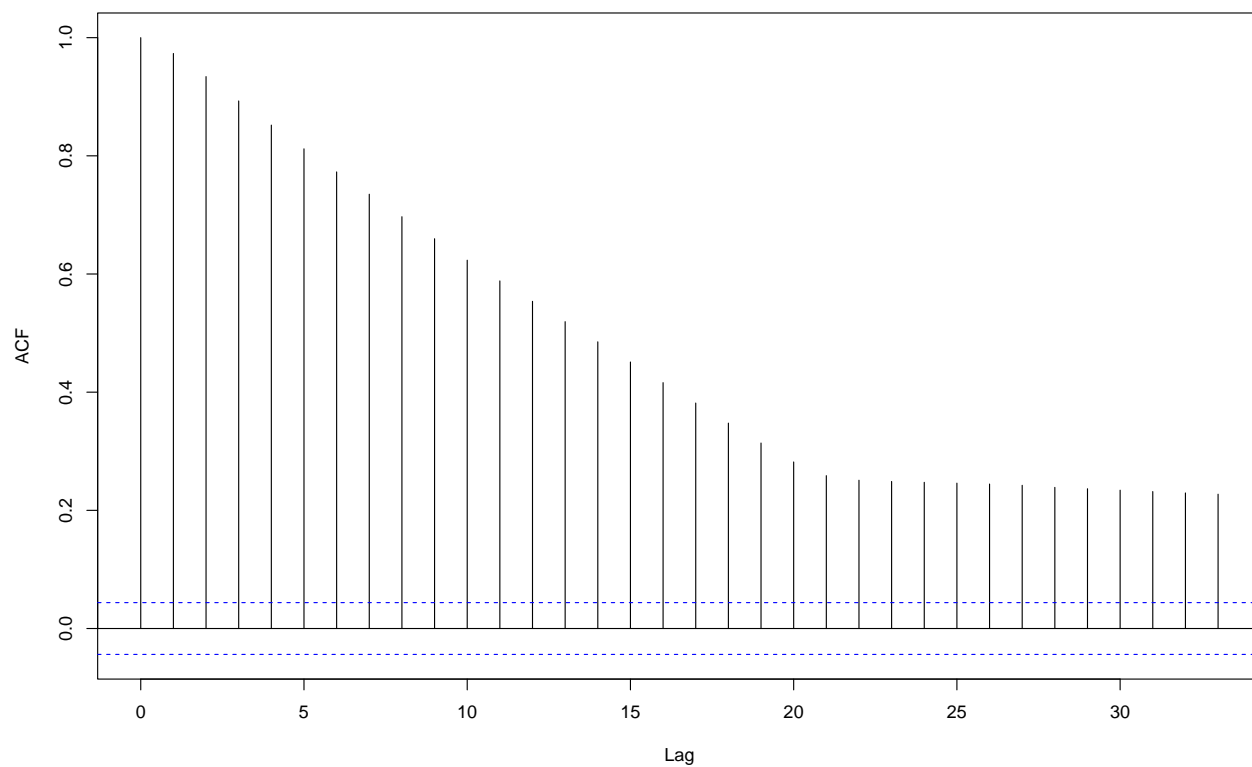
# Histogram of y



*# Autocorrelation function of the largest data*

`acf(y)`

## Series y



```
# Estimation of gev mixture models
```

```
gev_mixture_model <- suppressWarnings(estimate_gev_mixture_model_parameters(x = x,
                                                                              block_sizes = 10:40,
                                                                              minimum_nblocks = 50,
                                                                              threshold = min(y),
                                                                              nlargest = nlargest,
                                                                              confidence_level = 0.95,
                                                                              use_extremal_index = TRUE,
                                                                              use_lower_threshold = FALSE,
                                                                              maximum_iterations = 1500,
                                                                              log_mv = TRUE,
                                                                              log_pw = TRUE,
                                                                              trace = FALSE,
                                                                              method = "MLE"))
```

```
## Successful convergence.
```

```
## Successful convergence.
```

```
gev_mixture_model$extremal_indexes
```

```
##          10          11          12          13          14
## 0.05455837083 0.07814452754 0.05455837083 0.05170740715 0.05170740715
##          15          16          17          18          19
## 0.05170740715 0.05170740715 0.05170740715 0.05455837083 0.05170740715
##          20          21          22          23          24
## 0.04721881682 0.05170740715 0.05170740715 0.05170740715 0.04721881682
##          25          26          27          28          29
## 0.04721881682 0.03721457476 0.05170740715 0.04721881682 0.03721457476
##          30          31          32          33          34
## 0.03721457476 0.03082972874 0.03082972874 0.03721457476 0.03721457476
##          35          36          37          38          39
## 0.04721881682 0.02995362671 0.02995362671 0.03721457476 0.02995362671
##          40
## 0.02775272812
```

```
gev_mixture_model$normalized_gev_parameters_object
```

```
##          loc_star      scale_star  shape_star
## 10 7.204883247e-06 4.472925028e-07 0.6227929265
## 11 7.199328856e-06 3.991244189e-07 0.6488347948
## 12 7.049453809e-06 4.653051601e-07 0.5848421745
## 13 7.210542915e-06 3.583251096e-07 0.6619029664
## 14 7.035958749e-06 4.321059457e-07 0.5886015736
## 15 7.052150238e-06 3.803194929e-07 0.6212062314
## 16 6.998935048e-06 3.795404892e-07 0.6174885041
## 17 7.034634239e-06 3.616063896e-07 0.6175609129
## 18 6.926231854e-06 3.953970469e-07 0.5915645418
## 19 6.915102392e-06 3.875266157e-07 0.5925876895
## 20 6.956024919e-06 3.483896094e-07 0.6186291191
## 21 6.746074051e-06 4.221587212e-07 0.5623529452
## 22 6.768368190e-06 3.831641773e-07 0.5910739106
## 23 6.497294032e-06 4.667094802e-07 0.5342340206
## 24 6.845391846e-06 3.655446258e-07 0.5800666620
## 25 6.684087138e-06 3.888379136e-07 0.5906472328
## 26 6.850306268e-06 3.359389676e-07 0.6126085895
```

```
## 27 6.599860056e-06 3.815307227e-07 0.5760601330
## 28 6.732053052e-06 3.464419870e-07 0.5848394960
## 29 6.606543135e-06 3.614551140e-07 0.5926173297
## 30 6.501050648e-06 3.837188865e-07 0.5724774309
## 31 6.584695638e-06 3.797454021e-07 0.5637761030
## 32 6.810320570e-06 2.816404155e-07 0.6429576876
## 33 6.483719112e-06 3.521260931e-07 0.6115755204
## 34 6.530733755e-06 3.455700361e-07 0.5997915888
## 35 6.198620224e-06 4.518002761e-07 0.5332420582
## 36 6.557165154e-06 3.651417884e-07 0.5527048038
## 37 6.291181366e-06 4.038476400e-07 0.5354792219
## 38 6.236669796e-06 4.045296029e-07 0.5514645919
## 39 6.803787469e-06 2.517348439e-07 0.6745742644
## 40 6.609191895e-06 2.833765563e-07 0.6162270417
```

```
gev_mixture_model$full_normalized_gev_parameters_object
```

```
##          loc_star      scale_star  shape_star
## 10 6.604053644e-06 7.310007617e-08 0.6227929265
## 11 6.701853828e-06 7.634531124e-08 0.6488347948
## 12 6.399044337e-06 8.491827028e-08 0.5848421745
## 13 6.745391298e-06 5.043987465e-08 0.6619029664
## 14 6.430235446e-06 7.557625589e-08 0.5886015736
## 15 6.537144630e-06 6.039479952e-08 0.6212062314
## 16 6.482970798e-06 6.093849605e-08 0.6174885041
## 17 6.543087878e-06 5.804657030e-08 0.6175609129
## 18 6.377459643e-06 7.076286537e-08 0.5915645418
## 19 6.374181643e-06 6.698363912e-08 0.5925876895
## 20 6.478053967e-06 5.270286037e-08 0.6186291191
## 21 6.137288760e-06 7.980651968e-08 0.5623529452
## 22 6.232670483e-06 6.652723885e-08 0.5910739106
## 23 5.803183614e-06 9.589208063e-08 0.5342340206
## 24 6.322455924e-06 6.220693097e-08 0.5800666620
## 25 6.134232148e-06 6.406758552e-08 0.5906472328
## 26 6.374958770e-06 4.473700719e-08 0.6126085895
## 27 6.057773349e-06 6.925618225e-08 0.5760601330
## 28 6.239031108e-06 5.810328200e-08 0.5848394960
## 29 6.083361160e-06 5.140840911e-08 0.5926173297
## 30 5.932636792e-06 5.831478254e-08 0.5724774309
## 31 6.005854183e-06 5.340842269e-08 0.5637761030
## 32 6.419053563e-06 3.007228516e-08 0.6429576876
## 33 5.984886613e-06 4.705234845e-08 0.6115755204
## 34 6.034615154e-06 4.800227208e-08 0.5997915888
## 35 5.517692693e-06 8.870107829e-08 0.5332420582
## 36 5.991557336e-06 5.252763048e-08 0.5527048038
## 37 5.652252658e-06 6.171459264e-08 0.5354792219
## 38 5.622577289e-06 6.587932936e-08 0.5514645919
## 39 6.465619449e-06 2.361540055e-08 0.6745742644
## 40 6.199840865e-06 3.112338201e-08 0.6162270417
```

```
gev_mixture_model$automatic_weights_pw_shape
```

```
##          10          11          12          13          14
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 1.376676551e-14 0.000000000e+00
##          15          16          17          18          19
```

```
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          20          21          22          23          24
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          25          26          27          28          29
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          30          31          32          33          34
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          35          36          37          38          39
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 1.000000000e+00
##          40
## 0.000000000e+00
```

```
gev_mixture_model$automatic_weights_pw_scale
```

```
##          10          11          12          13
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          14          15          16          17
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          18          19          20          21
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          22          23          24          25
## 0.000000000e+00 1.000000000e+00 0.000000000e+00 0.000000000e+00
##          26          27          28          29
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 1.110223025e-16
##          30          31          32          33
## 1.110223025e-16 0.000000000e+00 0.000000000e+00 1.110223025e-16
##          34          35          36          37
## 0.000000000e+00 0.000000000e+00 1.110223025e-16 0.000000000e+00
##          38          39          40
## 0.000000000e+00 3.885780586e-16 -5.551115123e-17
```

```
gev_mixture_model$automatic_weights_pw_loc
```

```
## 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
## 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 36 37 38 39 40
## 0 0 0 0 0
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object[3, ]
```

```
##          loc_star      scale_star  shape_star
## automatic_weights 7.210542915e-06 4.667094802e-07 0.6745742644
```

```
gev_mixture_model$automatic_weights_mw
```

```
##          10          11          12          13
## 0.000000000e+00 1.000000000e+00 0.000000000e+00 0.000000000e+00
##          14          15          16          17
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00
##          18          19          20          21
## 0.000000000e+00 0.000000000e+00 -8.881784197e-16 0.000000000e+00
##          22          23          24          25
## 0.000000000e+00 0.000000000e+00 -8.881784197e-16 -2.664535259e-15
##          26          27          28          29
## 0.000000000e+00 -8.881784197e-16 0.000000000e+00 0.000000000e+00
##          30          31          32          33
## 1.776356839e-15 8.881784197e-16 -1.776356839e-15 8.881784197e-16
```

```

##          34          35          36          37
## 0.000000000e+00 8.881784197e-16 4.440892099e-16 0.000000000e+00
##          38          39          40
## -2.220446049e-15 2.664535259e-15 0.000000000e+00

# Model diagnostics

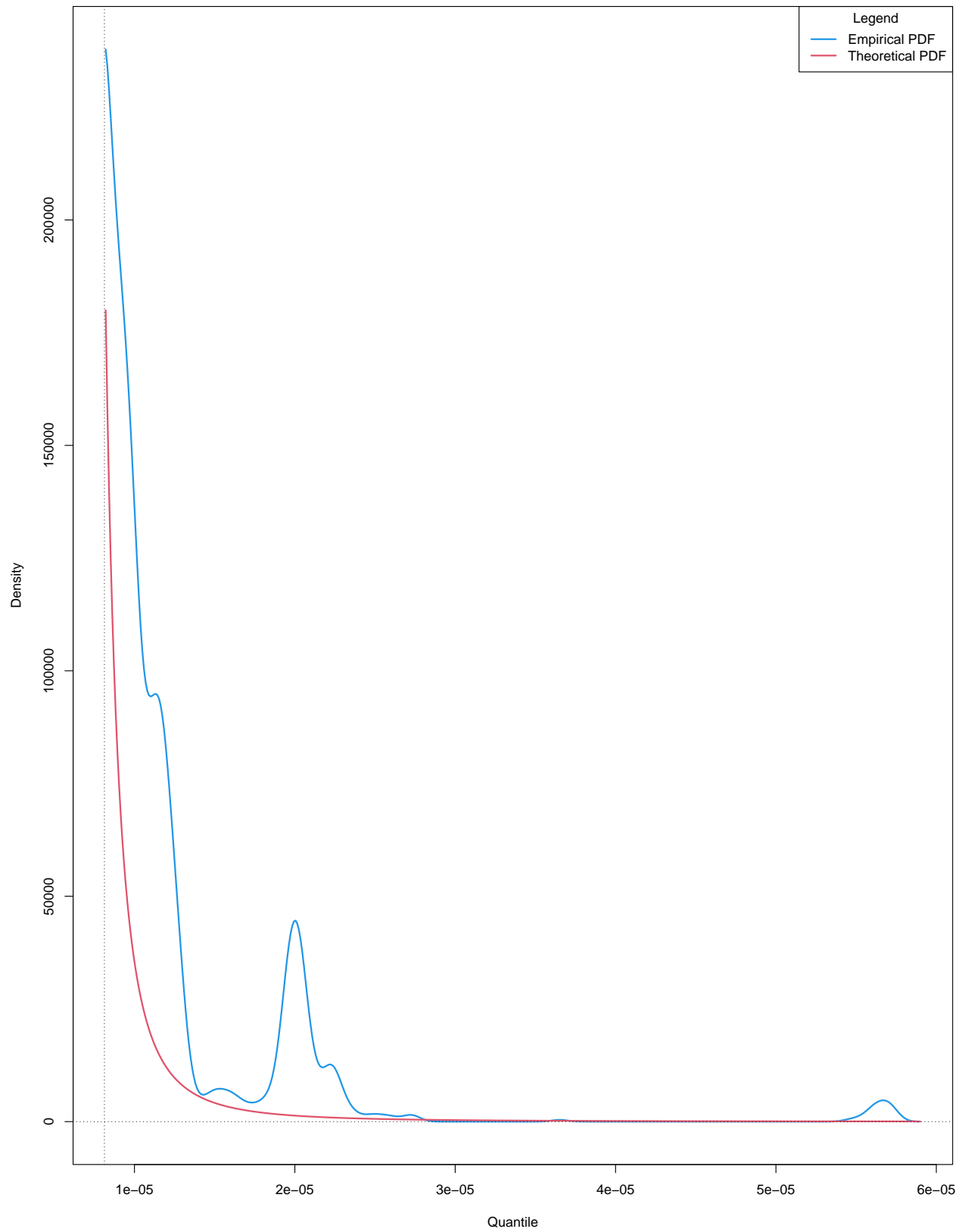
## GEV mixture model with respect to parameters

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")

```



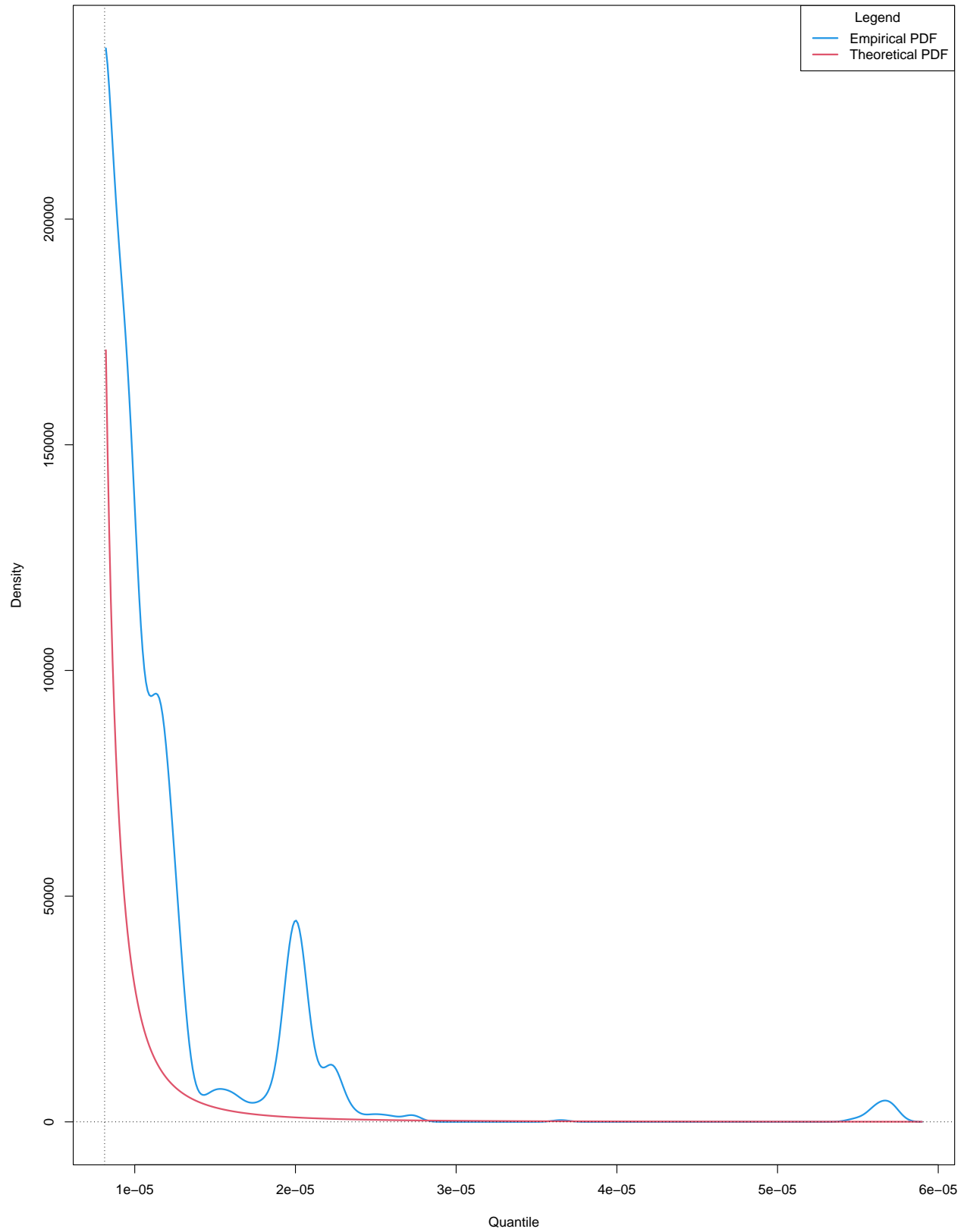
Probability Density Function (PDF) Plot : automatic\_weights – model\_wise = FALSE : zoom = TRUE



```
## GEV mixture model with respect to distribution functions
```

```
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



```
# Estimation of an extreme quantile
```

```
estimator_types <- c("automatic_weights_mw",  
  "pessimistic_weights_mw",  
  "identic_weights_mw",  
  "automatic_weights_pw",  
  "pessimistic_weights_pw",  
  "identic_weights_pw",  
  "model_wise",  
  "parameter_wise",  
  "empirical")
```

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

```
## Quantile from GEV mixture model with respect to parameters
```

```
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
```

```
## [1] 33.78749721
```

```
## Quantile from GEV mixture model with respect to distribution functions
```

```
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
```

```
## [1] 2.92420721
```

```
## Quantiles from equivalent estimated distributions in GEV mixture model with respect to parameters
```

```
est_rl_pw <- suppressWarnings(estimate_gev_mixture_model_quantile(gev_mixture_model,  
  alpha = alpha,  
  confidence_level = 0.95,  
  do.ci = TRUE,  
  estimator_type = estimator_types[8]))
```

```
est_rl_pw
```

```
##           lower      quantile      upper  
## 10 -3.4741673077 1.49585757162 6.4658824510  
## 11 -7.0563791613 2.90126504620 12.8589092537  
## 12 -1.7624041914 0.67638197074 3.1151681329  
## 13 -7.2790791682 2.71727643824 12.7136320447  
## 14 -1.8523146248 0.65463950110 3.1615936270  
## 15 -3.5607695763 1.16841481198 5.8975992003  
## 16 -3.4403060556 1.07646442189 5.5932348994  
## 17 -3.3435834431 1.02193095574 5.3874453545
```

```
## 18 -2.0887329519 0.66185721968 3.4124473912
## 19 -1.7096574326 0.51646759135 2.7425926153
## 20 -3.4310004640 0.94300425948 5.3170089829
## 21 -1.2711601137 0.36468986948 2.0005398527
## 22 -2.2835912292 0.62936532248 3.5423218741
## 23 -0.7782683887 0.20197815947 1.1822247076
## 24 -1.9280597378 0.45388347533 2.8358266885
## 25 -2.5549127323 0.61062103233 3.7761547970
## 26 -3.0737867471 0.71519388118 4.5041745095
## 27 -1.8734754994 0.44491271629 2.7633009319
## 28 -1.9961465497 0.42760435710 2.8513552639
## 29 -2.3233776336 0.49229570237 3.3079690383
## 30 -1.6334191434 0.34273516426 2.3188894719
## 31 -1.1594801620 0.25563261434 1.6707453906
## 32 -5.1134977833 1.03050499171 7.1745077667
## 33 -3.7822112098 0.71811165533 5.2184345205
## 34 -0.1737466045 0.03443761848 0.2426218414
## 35 -0.9981324131 0.19916054101 1.3964534951
## 36 -1.0575465161 0.17823490873 1.4140163335
## 37 -0.9174848112 0.14756405741 1.2126129260
## 38 -1.2236804726 0.22936847660 1.6824174258
## 39 -10.8038080227 1.81608368022 14.4359753832
## 40 -3.5244292924 0.53063448804 4.5856982685
```

```
## Comparison of estimated quantiles
```

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

```
## Quantiles from equivalent estimated GEV distributions in GEV mixture model respect to distribution f
```

```
est_rl_mw <- suppressWarnings(estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                                    alpha = alpha,
                                                                    confidence_level = 0.95,
                                                                    do.ci = TRUE,
                                                                    estimator_type = estimator_types[7]))
est_rl_mw
```

```
##          lower      quantile      upper
## 11 -7.0563791613 2.9012650462 12.858909254
## 30 -1.6334191434 0.3427351643 2.318889472
## 31 -1.1594801620 0.2556326143 1.670745391
## 33 -3.7822112098 0.7181116553 5.218434520
## 35 -0.9981324131 0.1991605410 1.396453495
## 36 -1.0575465161 0.1782349087 1.414016334
## 39 -10.8038080227 1.8160836802 14.435975383
```

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

```
est_rl_mw_range
```

```
## [1] -10.80380802 14.43597538
```

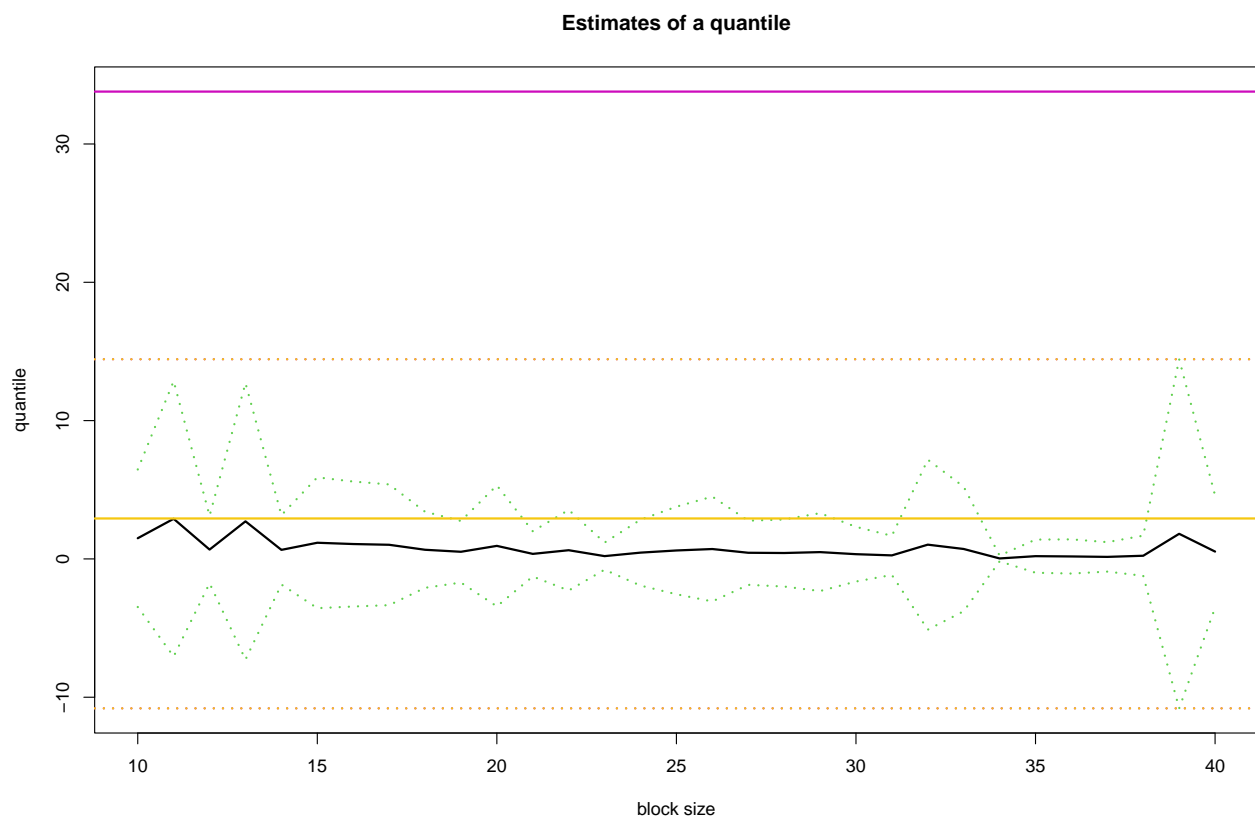
```
matplot(x = rownames(est_rl_pw),
        y = est_rl_pw,
        xlab = "block size",
```

```

ylab = "quantile",
main = "Estimates of a quantile",
ylim = range(c(est_rl_pw_range, rl_pw)),
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, col = 7, lwd = 2)
abline(h = rl_pw, 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)

```



*# Legend:*  
*# yellow: Quantile from GEV mixture model with respect to distribution functions*  
*# pink: Quantile from GEV mixture model with respect to parameters*