

Modeling extreme values with a GEV mixture probability distributions

Application to a rain data

Pascal Alain Dkengne Sielenou

2023-10-12

```
# 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_gev_mixture_model_cdf.R"))
xfun::in_dir(dir = path, expr = source("./src/estimate_gev_mixture_model_quantile.R"))

library(readr)

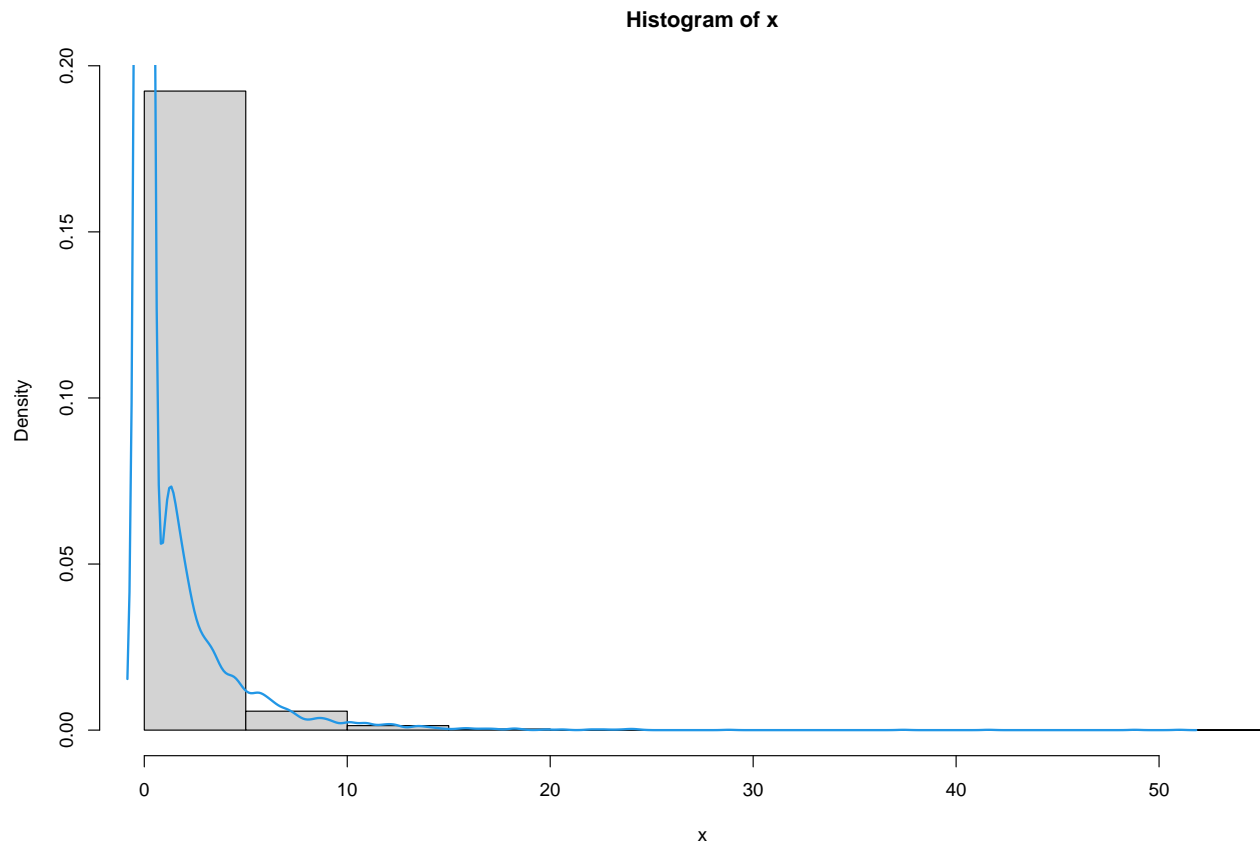
pluie <- xfun::in_dir(dir = path, expr = read_csv("./applications/pluie.csv"))

## Rows: 14623 Columns: 1
## -- Column specification -----
## Delimiter: ","
## dbl (1): x
##
## 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.
x <- pluie$x
x <- x[!is.na(x)]
n <- length(x)

n

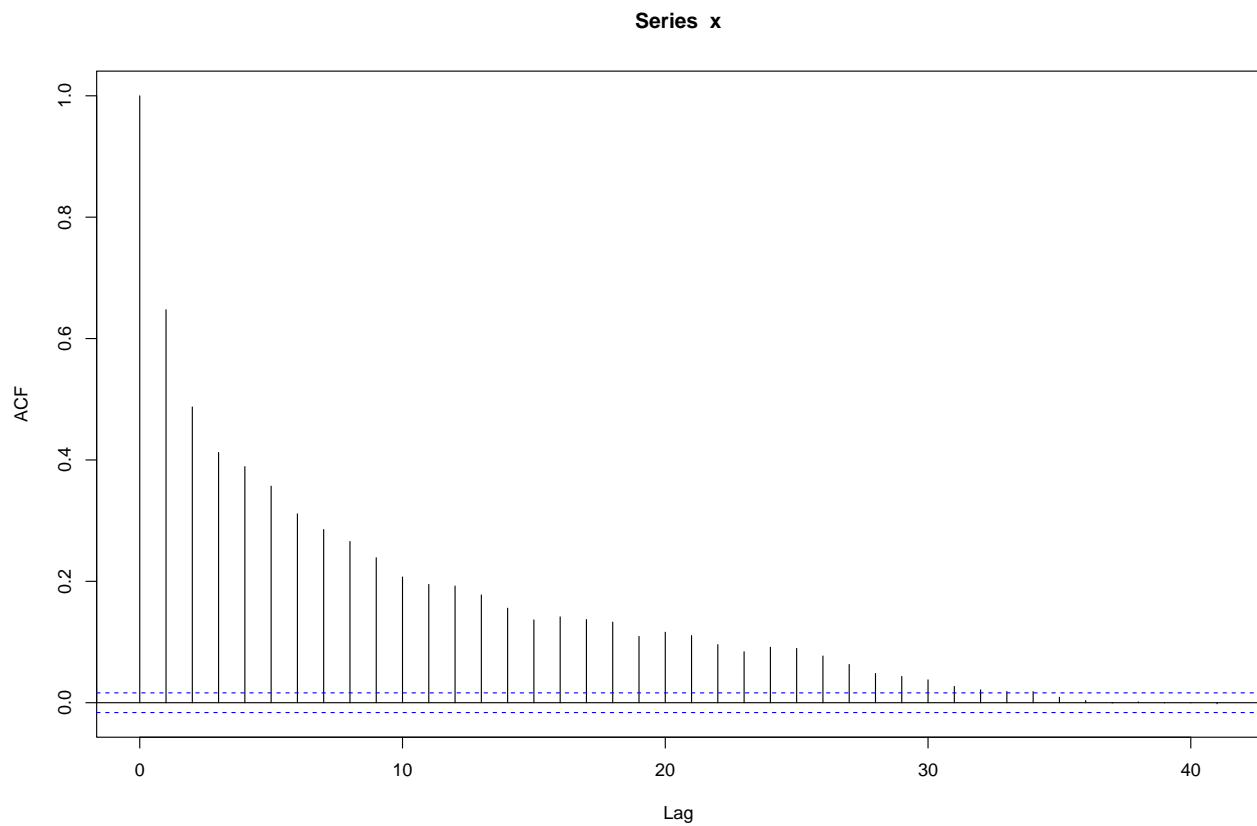
## [1] 14623
# Histogram of all data

hist(x, prob = TRUE)
lines(density(x),
      lwd = 2,
      col = 4)
```



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

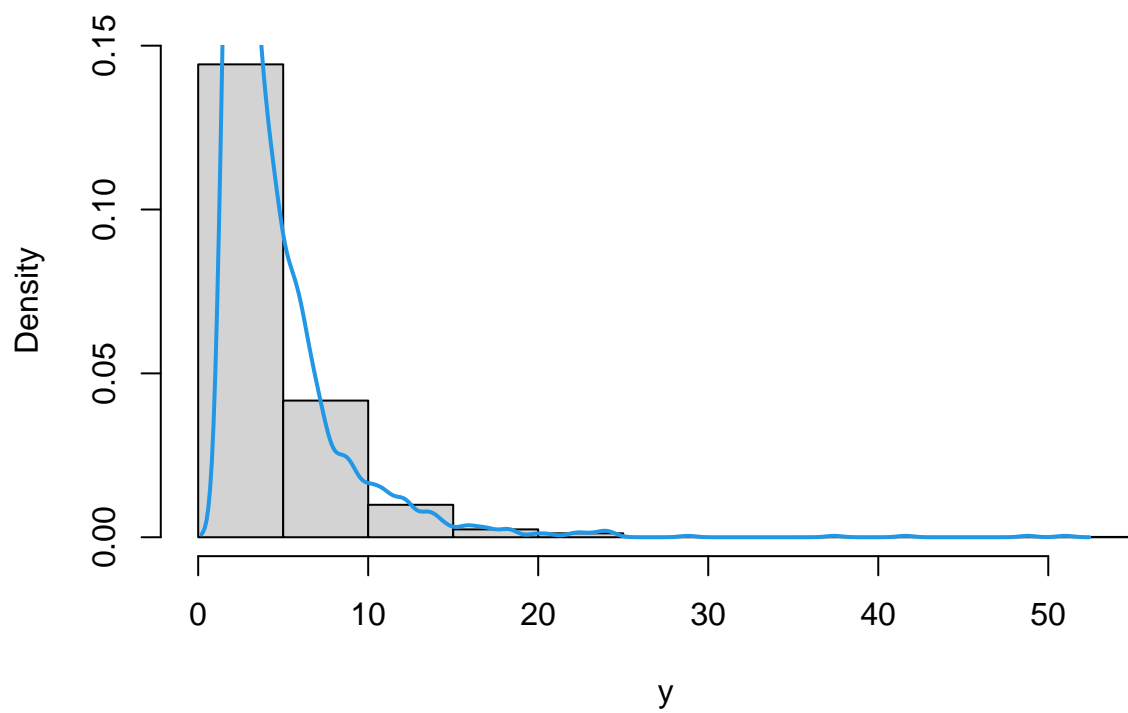
```
acf(x)
```



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

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

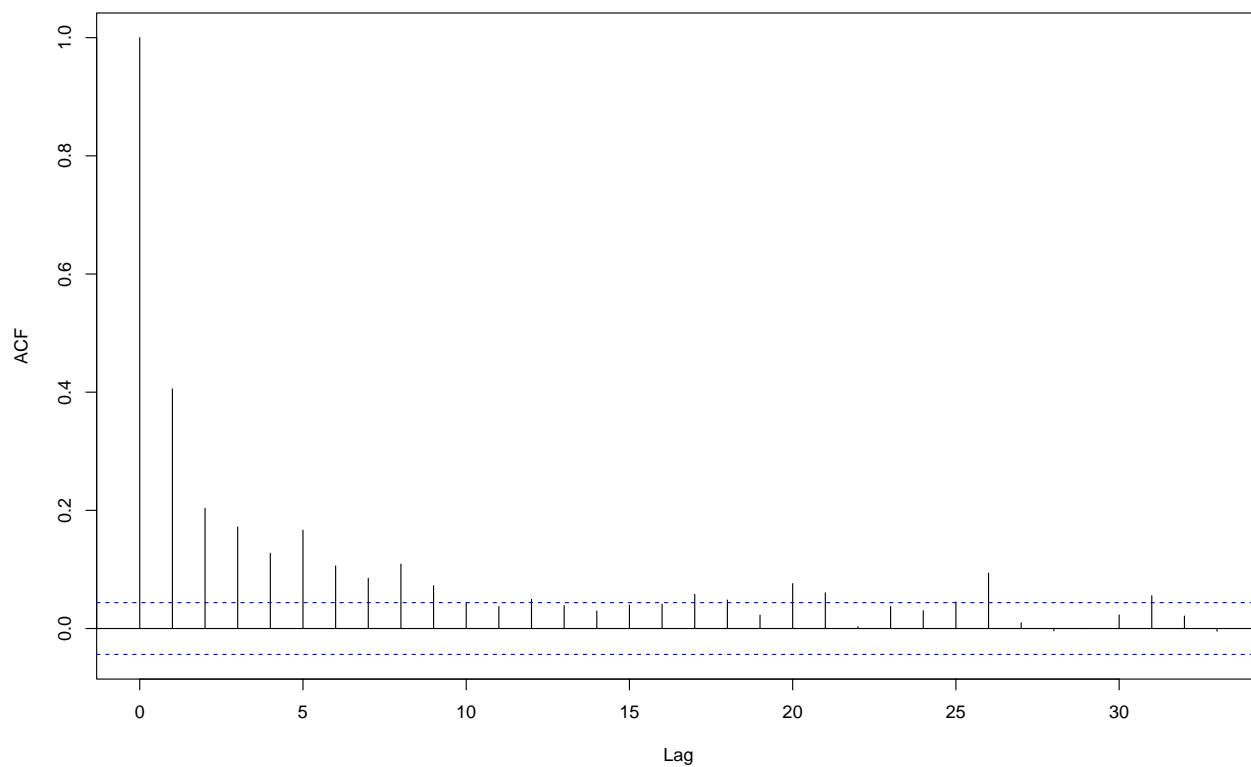
Histogram of y



Autocorrelation function of the largest data

`acf(y)`

Series y



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

```
gev_mixture_model <- estimate_gev_mixture_model_parameters(x = x,  
                                                           block_sizes = NULL,  
                                                           minimum_nblocks = 50,  
                                                           threshold = NULL,  
                                                           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
```

```
##          29          30          31          32          33          34  
## 0.3184582985 0.3184582985 0.3184582985 0.3184582985 0.3184582985 0.3079305589  
##          35          36          37          38          39          40  
## 0.2935114842 0.2938393366 0.3413828979 0.2923565005 0.3079305589 0.2923565005
```

```
gev_mixture_model$normalized_gev_parameters_object
```

```
##      loc_star  scale_star  shape_star  
## 29 -0.6555224011 1.803895374 0.2847175776  
## 30  0.8369277343 1.224717897 0.3672802508  
## 31 -1.1598963312 1.881955037 0.2782003507  
## 32 -0.9845403055 1.880675950 0.2624950585  
## 33 -1.6226212996 2.052348437 0.2418882121  
## 34  0.9388367517 1.298139788 0.3343860366  
## 35  0.2914844231 1.365960542 0.3382467378  
## 36 -0.1069973355 1.434685957 0.3254873512  
## 37 -2.8420314349 2.583474790 0.1855615569  
## 38 -0.6166176971 1.774072349 0.2748374215  
## 39 -0.2706649938 1.629244995 0.2881871931  
## 40 -0.9237856376 1.782973899 0.2744051785
```

```
gev_mixture_model$full_normalized_gev_parameters_object
```

```
##      loc_star  scale_star  shape_star  
## 29 -2.4171399460 1.3023318938 0.2847175776  
## 30 -0.3072516632 0.8044834014 0.3672802508  
## 31 -3.0042411168 1.3688576709 0.2782003507  
## 32 -2.8434072919 1.3927325516 0.2624950585  
## 33 -3.6740658265 1.5561281879 0.2418882121  
## 34 -0.3250242704 0.8755223101 0.3343860366  
## 35 -1.0792110634 0.9023272656 0.3382467378  
## 36 -1.5561174749 0.9630156808 0.3254873512  
## 37 -5.3592747712 2.1163711972 0.1855615569  
## 38 -2.4678924053 1.2652727811 0.2748374215  
## 39 -1.8979372521 1.1602859702 0.2881871931
```

```
## 40 -2.7848160474 1.2722975176 0.2744051785
gev_mixture_model$automatic_weights_pw_shape

##           29           30           31           32
## 0.000000000e+00 1.000000000e+00 0.000000000e+00 0.000000000e+00
##           33           34           35           36
## 0.000000000e+00 0.000000000e+00 7.815970093e-14 0.000000000e+00
##           37           38           39           40
## -7.105427358e-15 0.000000000e+00 0.000000000e+00 0.000000000e+00
gev_mixture_model$automatic_weights_pw_scale

##           29           30           31           32           33
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 2.557953849e-13
##           34           35           36           37           38
## 0.000000000e+00 0.000000000e+00 0.000000000e+00 1.000000000e+00 0.000000000e+00
##           39           40
## 0.000000000e+00 0.000000000e+00
gev_mixture_model$automatic_weights_pw_loc

##           29           30           31           32
## 0.000000000e+00 1.000000000e+00 0.000000000e+00 0.000000000e+00
##           33           34           35           36
## 0.000000000e+00 1.438849040e-13 0.000000000e+00 0.000000000e+00
##           37           38           39           40
## -2.842170943e-14 0.000000000e+00 0.000000000e+00 0.000000000e+00
gev_mixture_model$weighted_normalized_gev_parameters_object[3, ]

##           loc_star scale_star shape_star
## automatic_weights 0.8369277343 2.58347479 0.3672802508
gev_mixture_model$automatic_weights_mw

## 29 30 31 32 33 34 35 36 37 38 39 40
## 0 0 0 0 0 1 0 0 0 0 0 0

# Model diagnostics

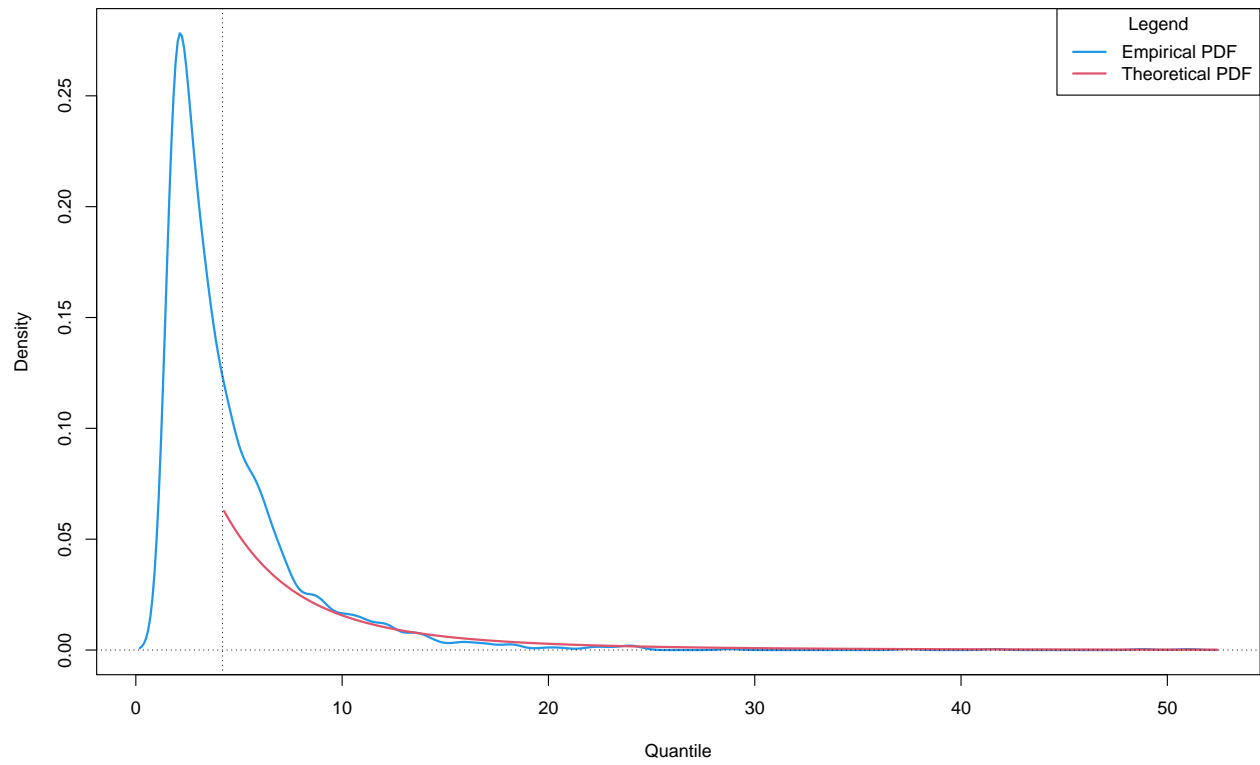
## GEV mixture model with respect to parameters

par(mfrow = c(2, 1))
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")

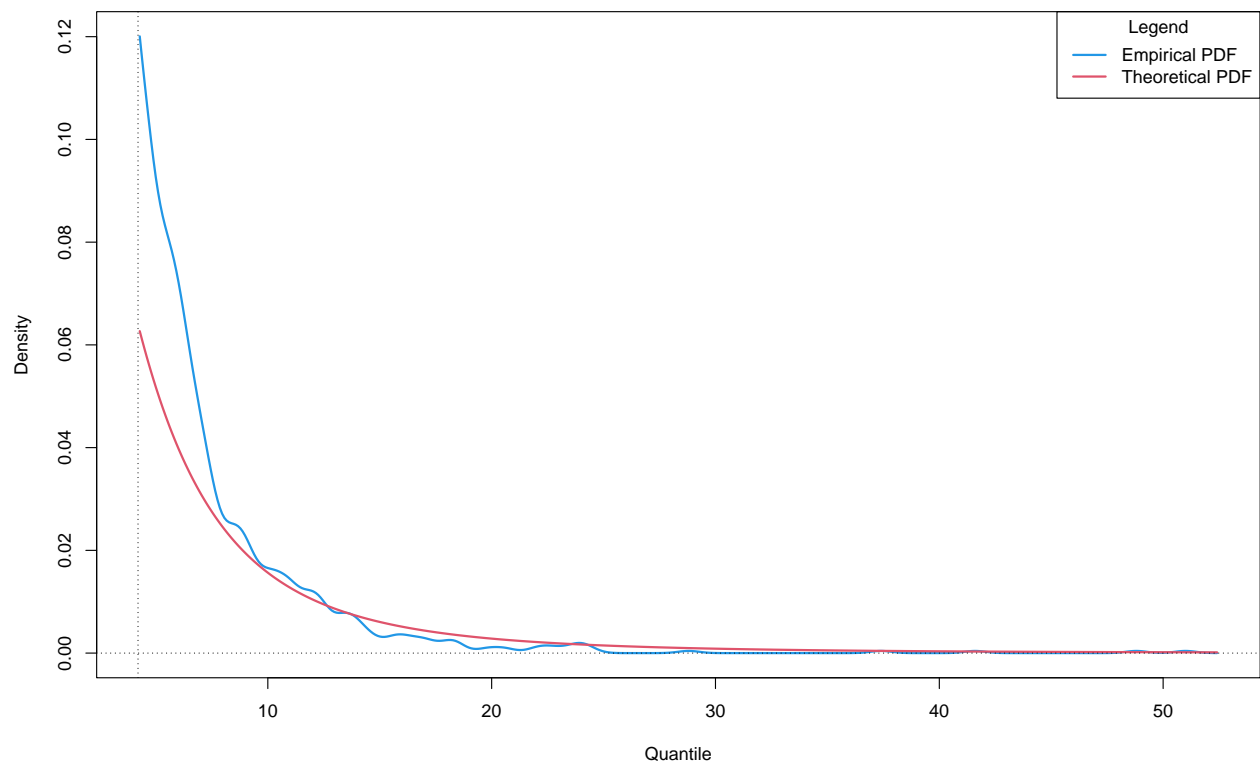
plot_gev_mixture_model_pdf(gev_mixture_model,
  type = "automatic_weights",
  model_wise = FALSE,
  zoom = TRUE,
  xlab = "Quantile",
  ylab = "Density",
```

```
main = "Probability Density Function (PDF) Plot"
```

Probability Density Function (PDF) Plot : automatic_weights – model_wise = FALSE : zoom = FALSE



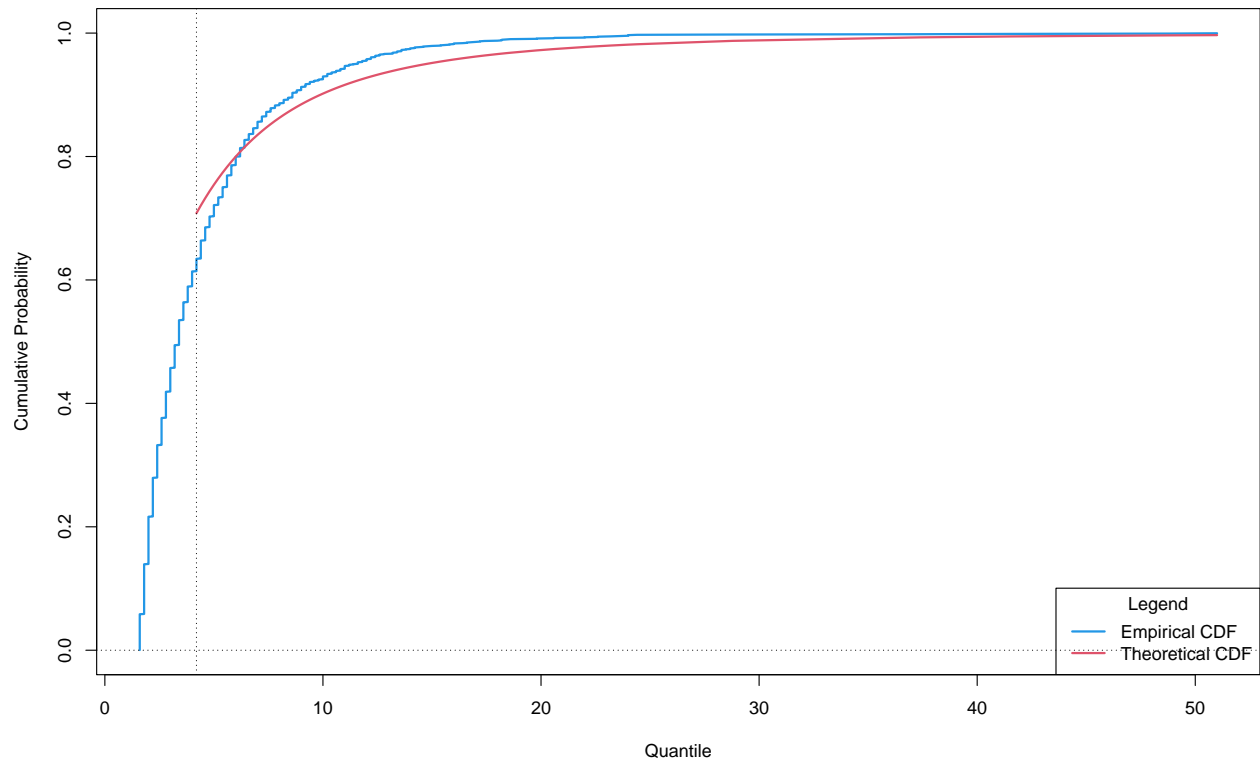
Probability Density Function (PDF) Plot : automatic_weights – model_wise = FALSE : zoom = TRUE



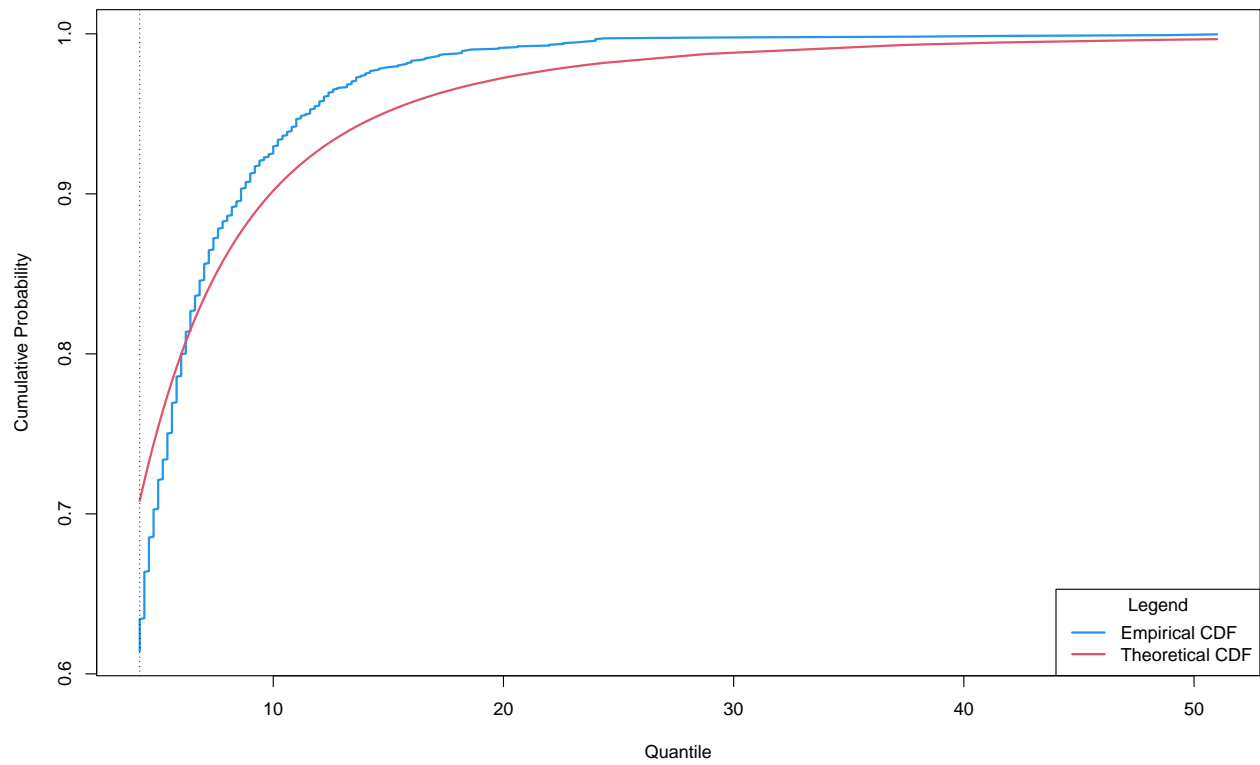

```
par(mfrow = c(2, 1))
plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Cumulative Probability",
                           main = "Cumulative Distribution Function (CDF) Plot")

plot_gev_mixture_model_cdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Cumulative Probability",
                           main = "Cumulative Distribution Function (CDF) Plot")
```

Cumulative Distribution Function (CDF) Plot : automatic_weights – model_wise = FALSE : zoom = FALSE



Cumulative Distribution Function (CDF) Plot : automatic_weights – model_wise = FALSE : zoom = TRUE



```

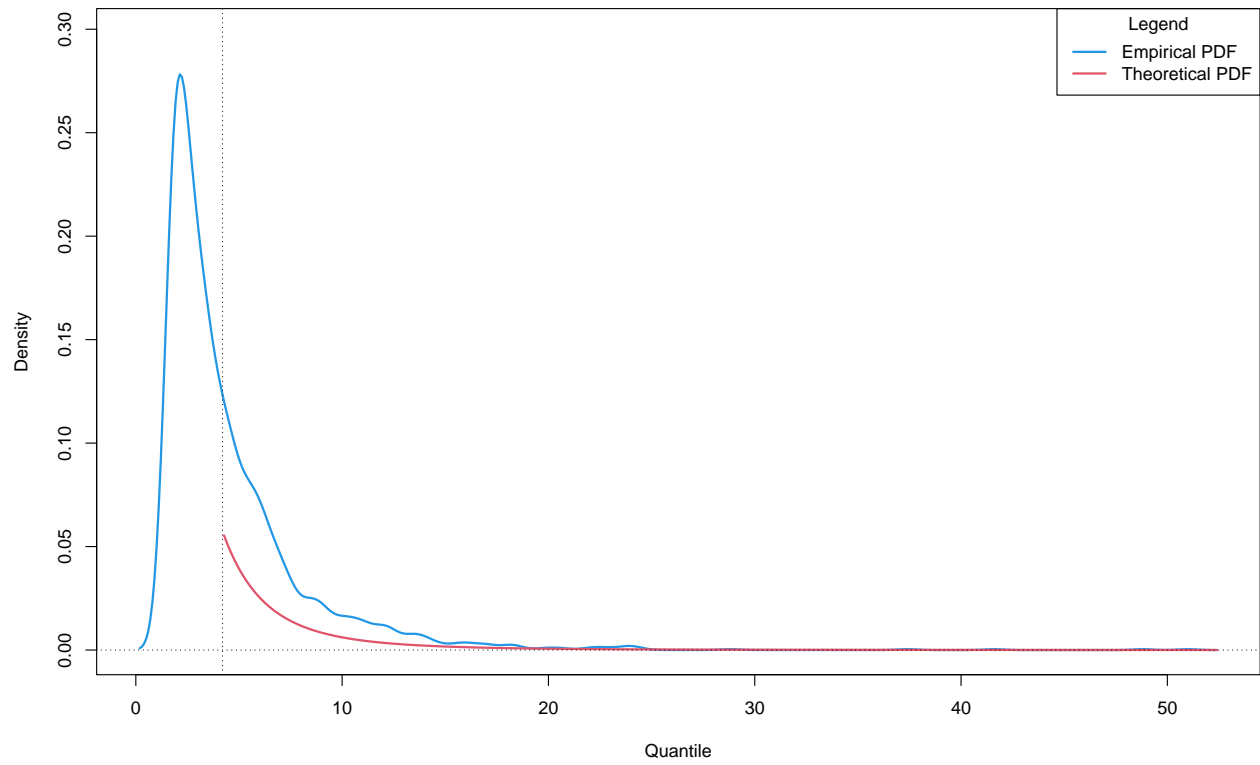
## GEV mixture model with respect to distribution functions

par(mfrow = c(2, 1))
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")

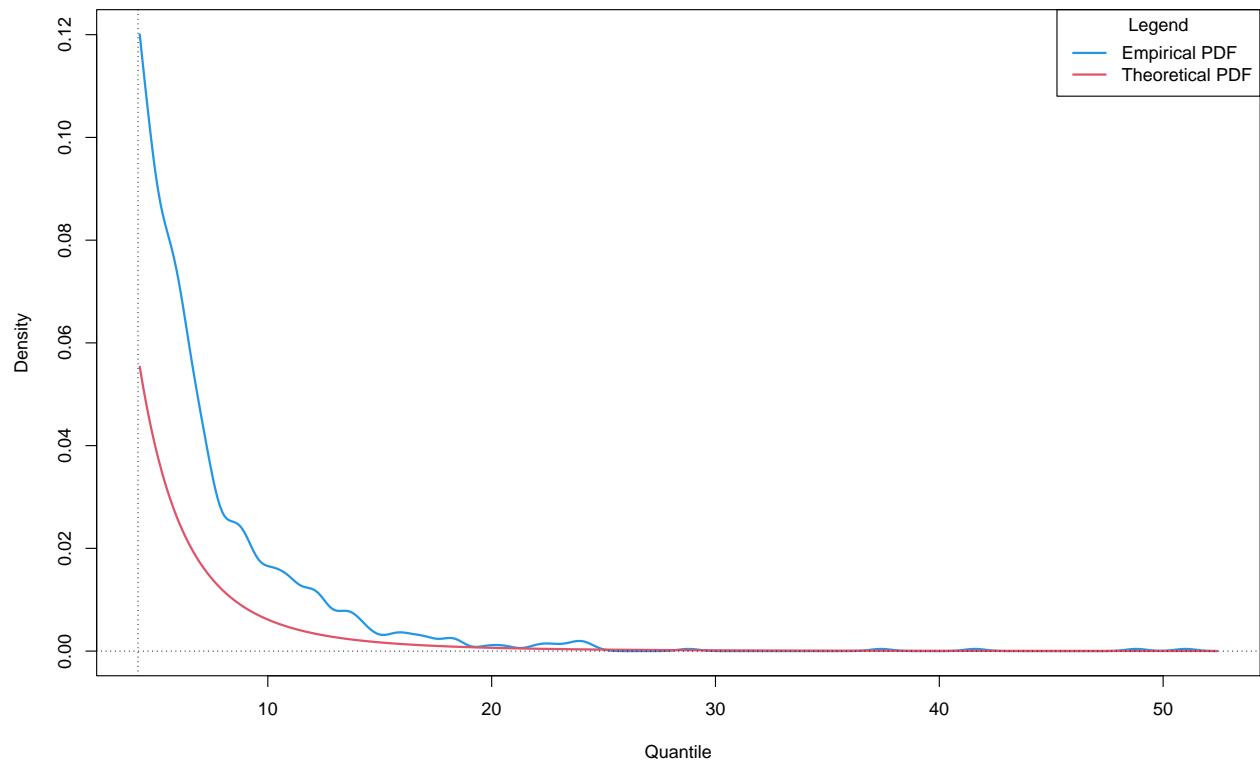
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = TRUE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                      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

```

Probability Density Function (PDF) Plot : automatic_weights – model_wise = TRUE : zoom = FALSE



Probability Density Function (PDF) Plot : automatic_weights – model_wise = TRUE : zoom = TRUE



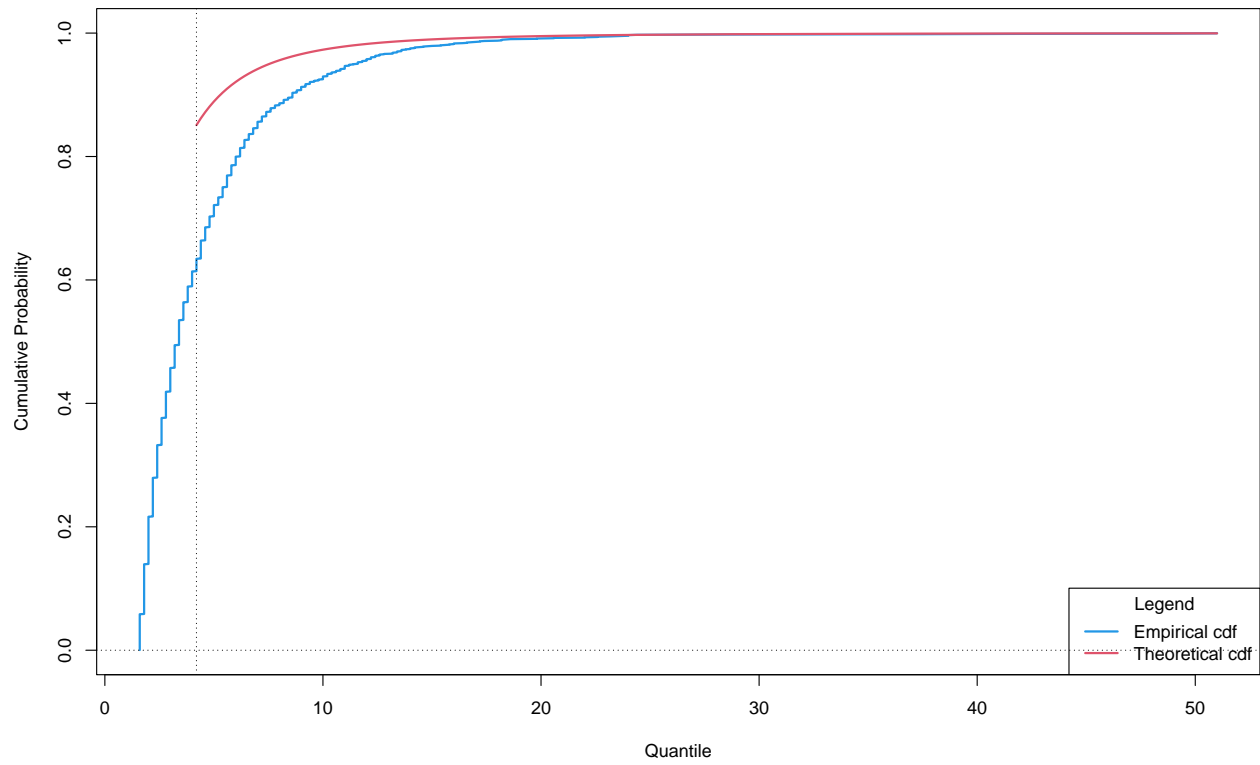
```

par(mfrow = c(2, 1))
plot_gev_mixture_model_cdf(gev_mixture_model,
                             type = "automatic_weights",
                             model_wise = TRUE,
                             zoom = FALSE,
                             xlab = "Quantile",
                             ylab = "Cumulative Probability",
                             main = "Cumulative Distribution Function (CDF) Plot")

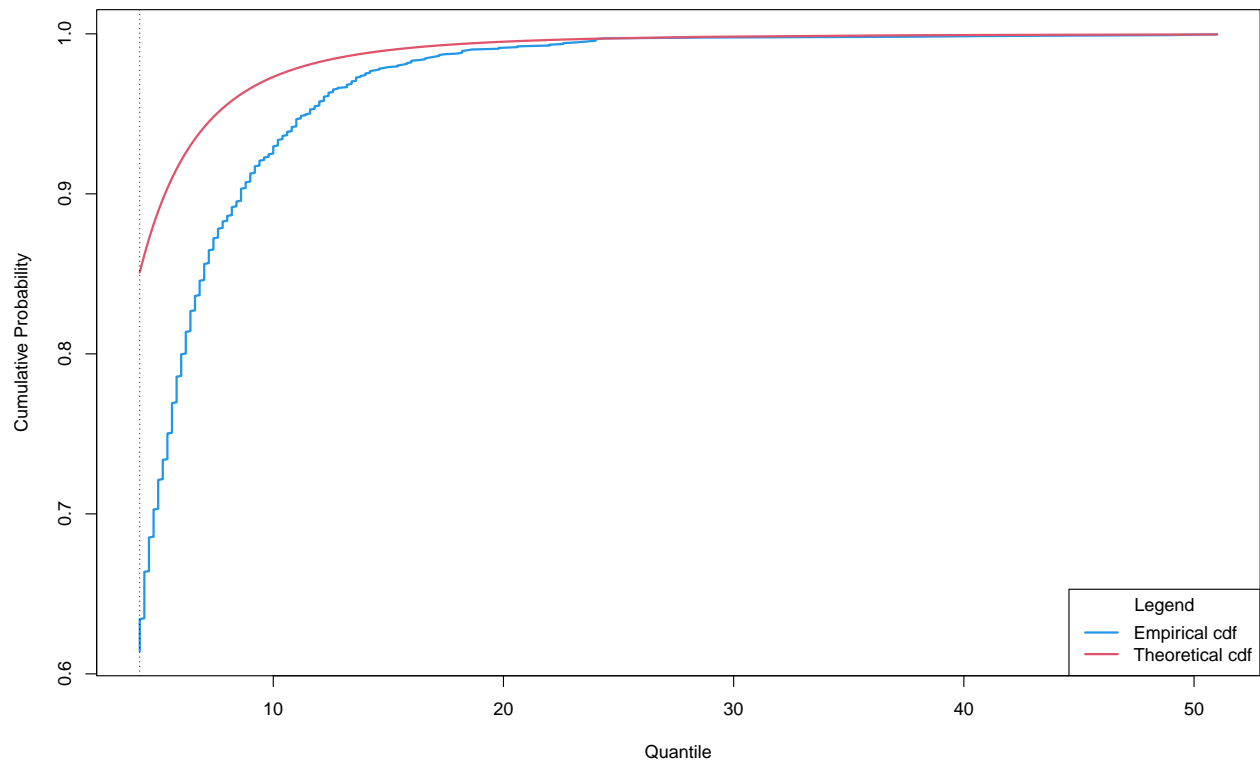
plot_gev_mixture_model_cdf(gev_mixture_model,
                             type = "automatic_weights",
                             model_wise = TRUE,
                             zoom = TRUE,
                             xlab = "Quantile",
                             ylab = "Cumulative Probability",
                             main = "Cumulative Distribution Function (CDF) Plot")

```

Cumulative Distribution Function (CDF) Plot : automatic_weights – model_wise = TRUE : zoom = FALSE



Cumulative Distribution Function (CDF) 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",  
  "empirical",  
  "confidence_interval_mw",  
  "confidence_interval_pw")
```

```
alpha <- 10^(-6)
```

```
## 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[2]
```

```
##      estimate  
## 1 535.2487405
```

```
## 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[2]
```

```
##      estimate  
## 1 199.5810581
```

```
## Quantiles from equivalent estimated GEV models
```

```
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  
## 29 -45.90262072 176.7013103 399.3052412  
## 30 -106.58112 254.1693357 614.9197913  
## 31 -48.06188659 173.6574728 395.3768322  
## 32 -36.62902345 151.568314 339.7656515  
## 33 -20.74933928 138.1308369 297.0110131  
## 34 -87.18648596 199.5728875 486.332261
```

```

## 35 -143.1068484 216.7583069 576.6234622
## 36 -108.6387905 202.4540746 513.5469397
## 37 9.16094583 108.1889565 207.2169671
## 38 -50.12077284 159.4390735 368.9989198
## 39 -49.1970273 164.8497816 378.8965904
## 40 -60.24250857 159.3334677 378.9094439

## Comparison of estimated quantiles

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

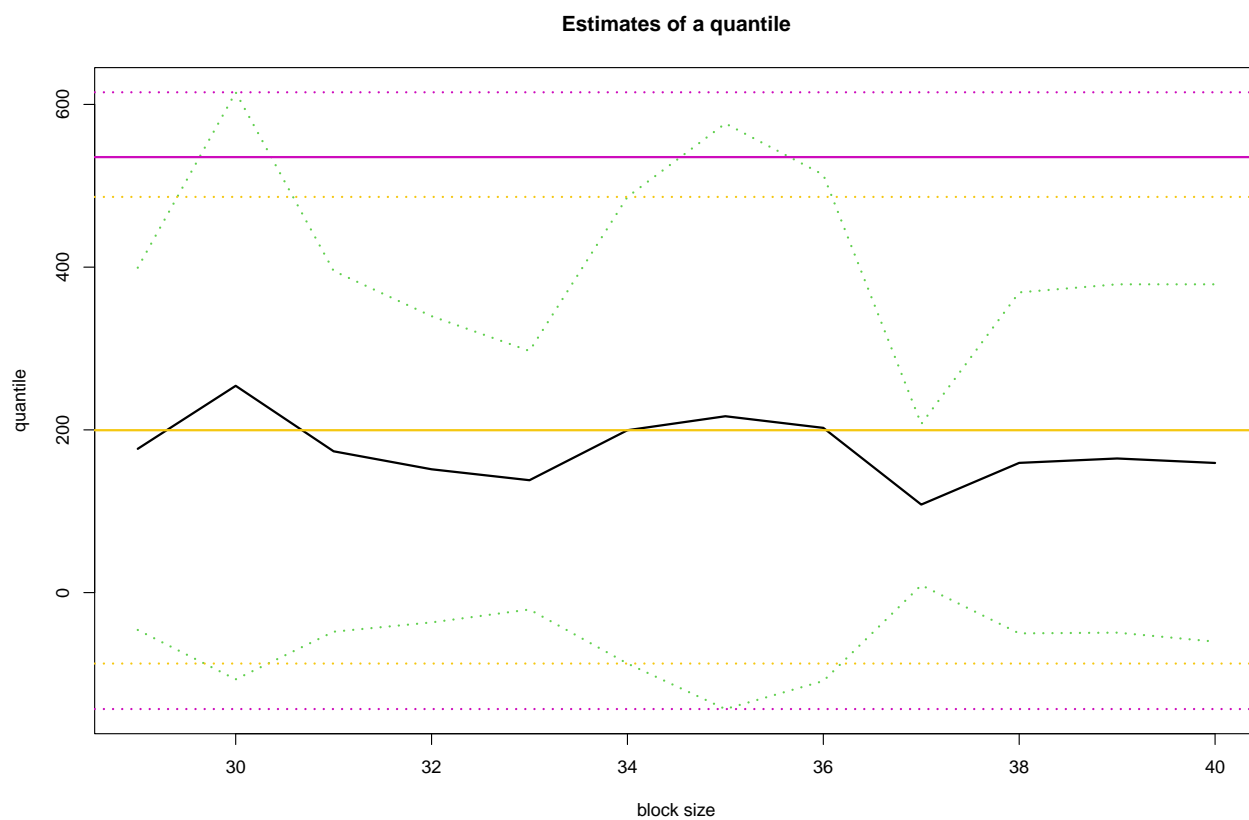
est_rl_mw <- estimate_gev_mixture_model_quantile(gev_mixture_model,
                                                  alpha = alpha,
                                                  confidence_level = 0.95,
                                                  do.ci = TRUE,
                            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

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

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)),
        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)

```

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