Modeling extreme values with a GEV mixture probability distributions

Application to a rain data

Pascal Alain Dkengne Sielenou

2023-09-27

```
# 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)
```

```
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)]
x <- x[x > 0]
```

```
n <- length(x)
n
```

```
## [1] 2855
```

```
nlargest <- 2000
```

```
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] 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
## 29 -0.655640718032394 1.80391601375939 0.284717010733928
## 30  0.836893071751833 1.22473565081861 0.367274055914226
## 31 -1.159927741737757 1.88197348197433 0.278198427848032
## 32 -0.984617626707811 1.88066835388259 0.262499126697637
## 33 -1.624513509547997 2.05294670274297 0.241852132607498
## 34  0.938735637430488 1.29818853350414 0.334373802081873
## 35  0.291182672423455 1.36607148342613 0.338235540141251
## 36 -0.107174774672256 1.43475253390137 0.325479792339282
## 37 -2.842123019326126 2.58348135175971 0.185562408644337
## 38 -0.624393195832429 1.77628551752773 0.274611622078795
## 39 -0.271143333327229 1.62928706592735 0.288182618118868
## 40 -0.923839349827874 1.78299022509863 0.274404155525876
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                               loc_star       scale_star        shape_star
## identic_weights     -0.593880157283841 1.72627474286025 0.287949224394300
## pessimistic_weights  0.210595017526111 1.87316825039926 0.290107555070016
## automatic_weights    0.938735637430488 2.58348135175971 0.365014733067874
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 16.6557695818259
##
## $gradient_value
## [1] 7.8159700933611e-14
##
## $function_reduction
## [1] 2.70134842895282
##
## $number_iterations
## [1] 1
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.98479055443101
##
## $gradient_value
## [1] 3.40362658692549e-05
##
## $function_reduction
## [1] 18.2162960904662
##
## $number_iterations
## [1] 1562
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                29                30                31                32
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
##                33                34                35                36
## 0.000000000000000 0.999999999999979 0.000000000000000 0.000000000000000
##                37                38                39                40
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                 29                 30                 31                 32
## 0.0829745856527353 0.0901154304523819 0.0824354679893330 0.0811513946523827
##                 33                 34                 35                 36
## 0.0794930412058550 0.0871988510940937 0.0875362412530145 0.0864267423240778
##                 37                 38                 39                 40
## 0.0751420078967423 0.0821403176171181 0.0832626418460646 0.0821232780162011
```

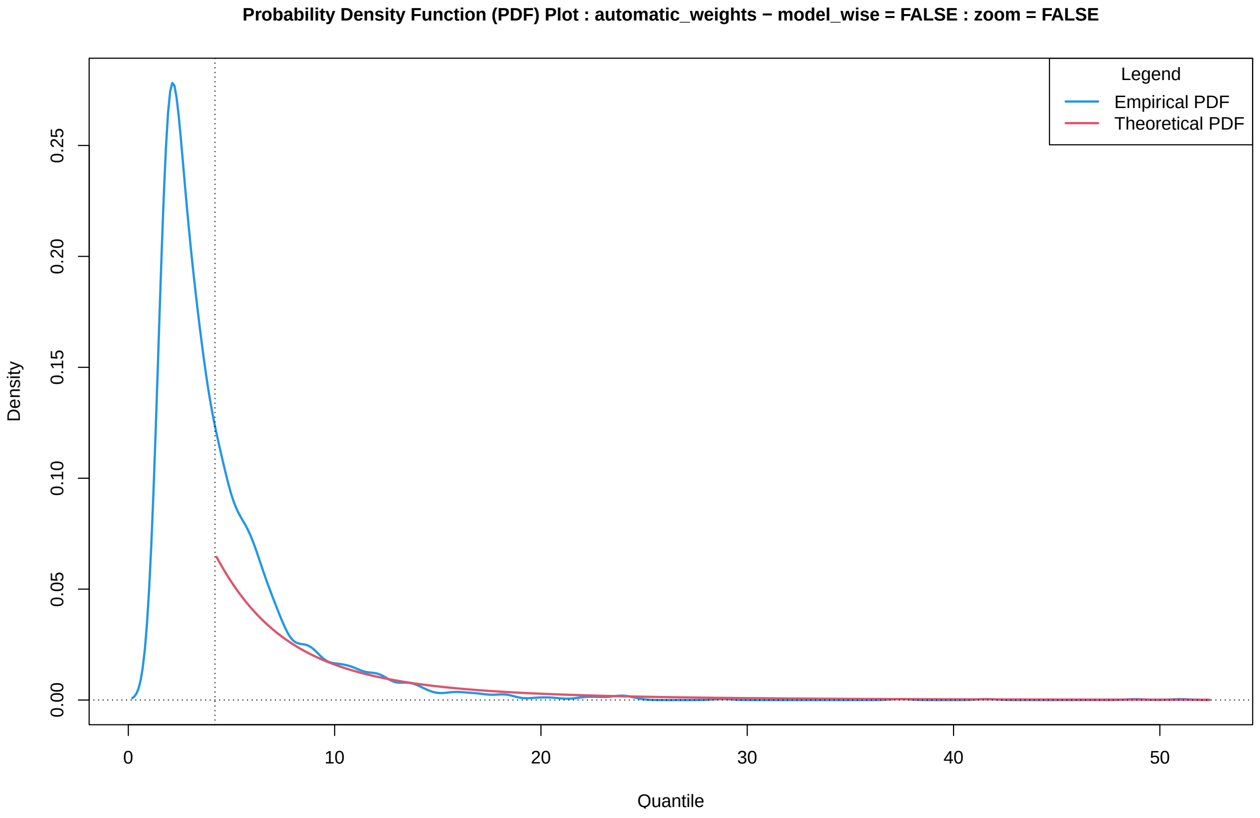
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                 29                 30                 31                 32
## 0.0839191189486136 0.0470247050224727 0.0907320750515446 0.0906137353126300
##                 33                 34                 35                 36
## 0.1076498848684291 0.0506088258538240 0.0541635912731382 0.0580143263145670
##                 37                 38                 39                 40
## 0.1829876754539849 0.0816321328264840 0.0704726377524737 0.0821812913218383
```

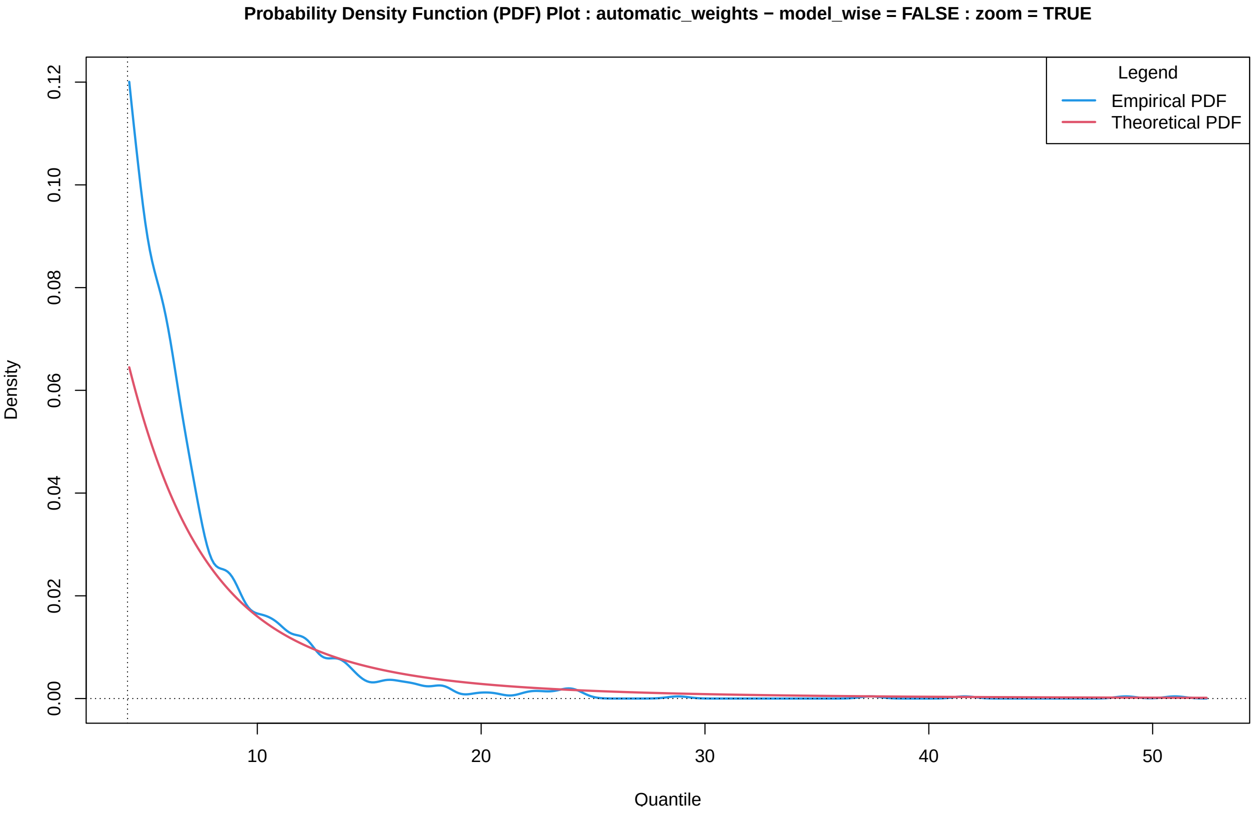
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                  29                  30                  31                  32
## 0.05060087545557751 0.22509052788075459 0.03055969102876763 0.03641540753318159
##                  33                  34                  35                  36
## 0.01920356782754391 0.24922229098669796 0.13042420691208495 0.08756968014795259
##                  37                  38                  39                  40
## 0.00568304156577689 0.05220699030633261 0.07432640426955990 0.03869731608576988
```

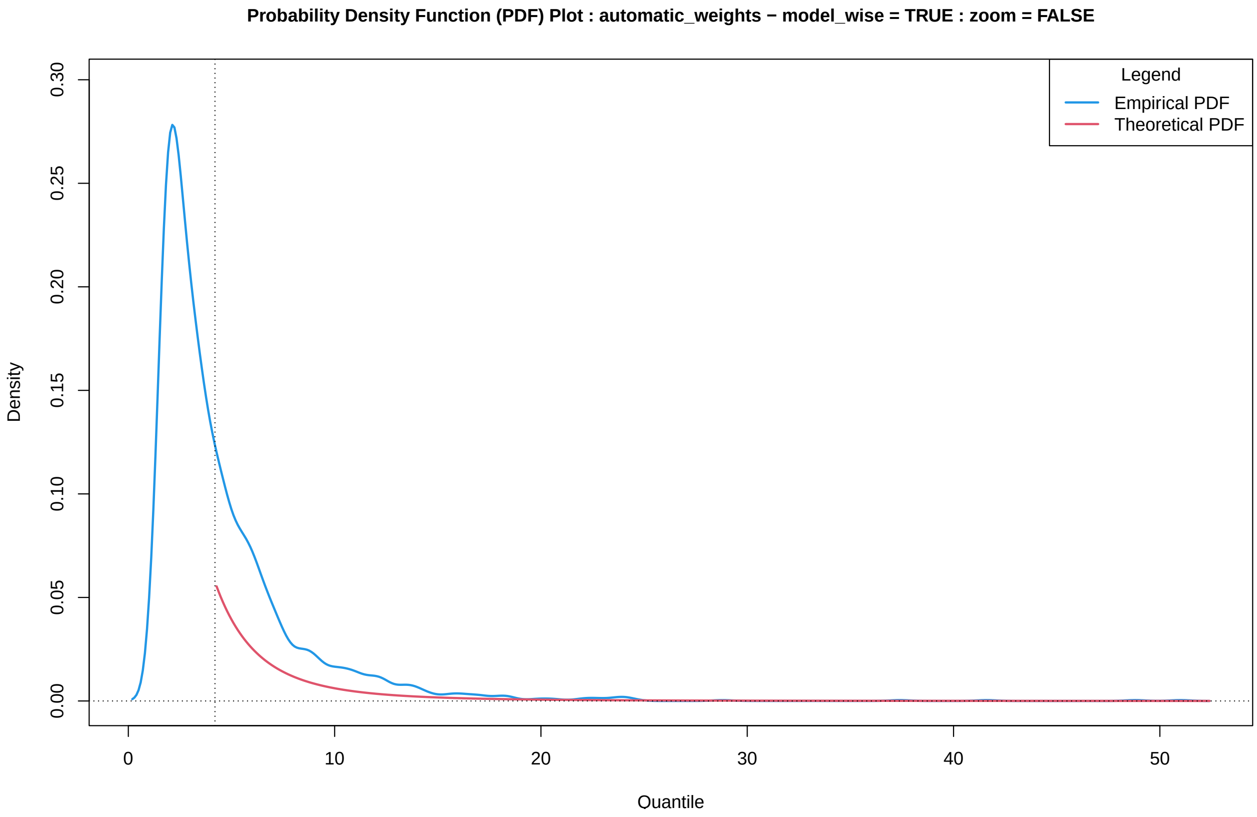
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



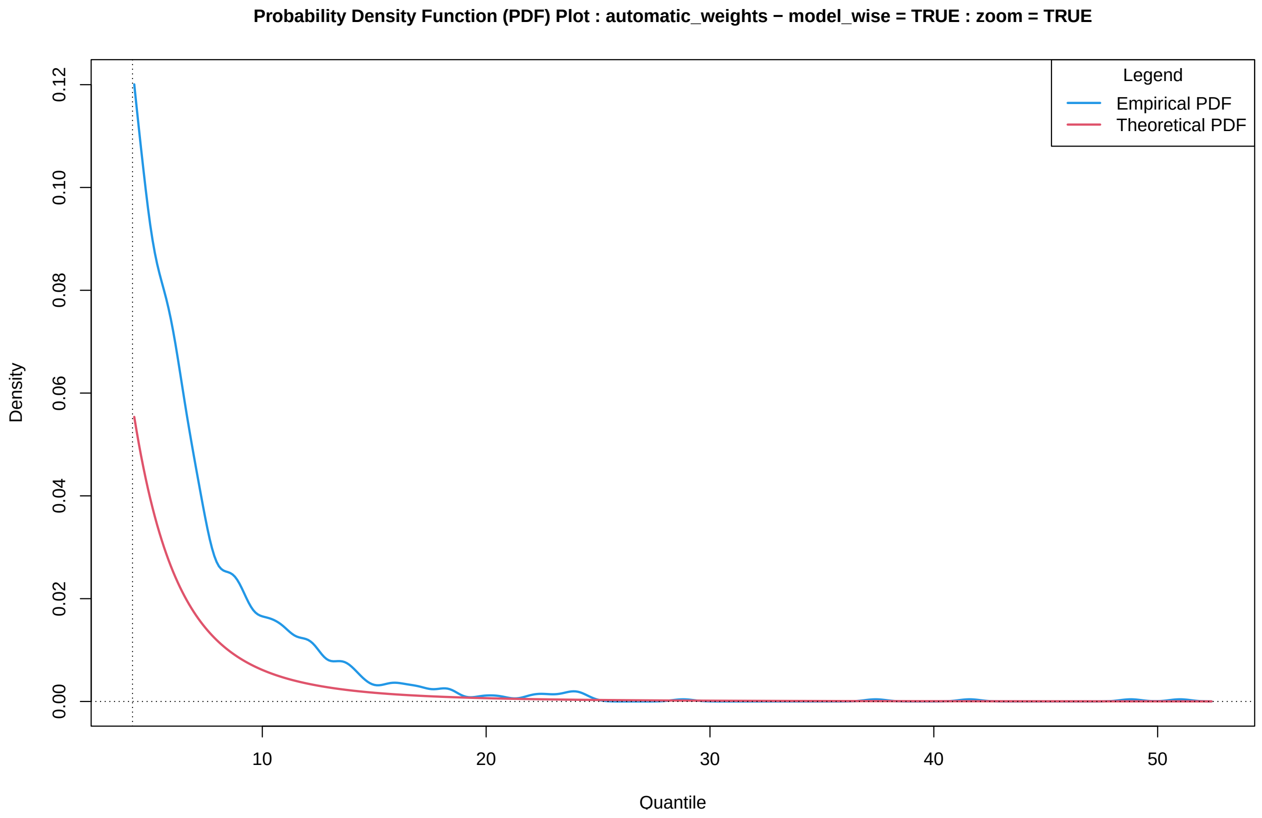
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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



```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")
```



```
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)
```

```
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 346.729232942052    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 956.715484844315    NA
```

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

```
##         99.9999%
## 50.9937211999993
```

```
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 -170.651560727064 285.484202159205 741.619965045474
## 30 -366.993068559081 465.177010252148 1297.34708906338
## 31 -173.146487118434 278.129254863372 729.404996845177
## 32 -138.031535808542 237.087350467054 612.206236742651
## 33 -97.5353318795926 209.968865165355 517.473062210302
## 34 -287.130679100386 346.758525941091 980.647730982567
## 35 -425.305353629991 379.427477380411 1184.16030839081
## 36 -334.266849163395 347.718675416519 1029.70419999643
## 37 -24.7153154739967 152.435379888069 329.586075250135
## 38 -173.213748550616 253.803988518762 680.821725588141
## 39 -176.289618997808 267.530069635196 711.349758268201
## 40 -194.890322493338 253.652763596775 702.195849686888
```

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

```
## [1] -425.305353629991 1297.347089063376
```

```
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
```

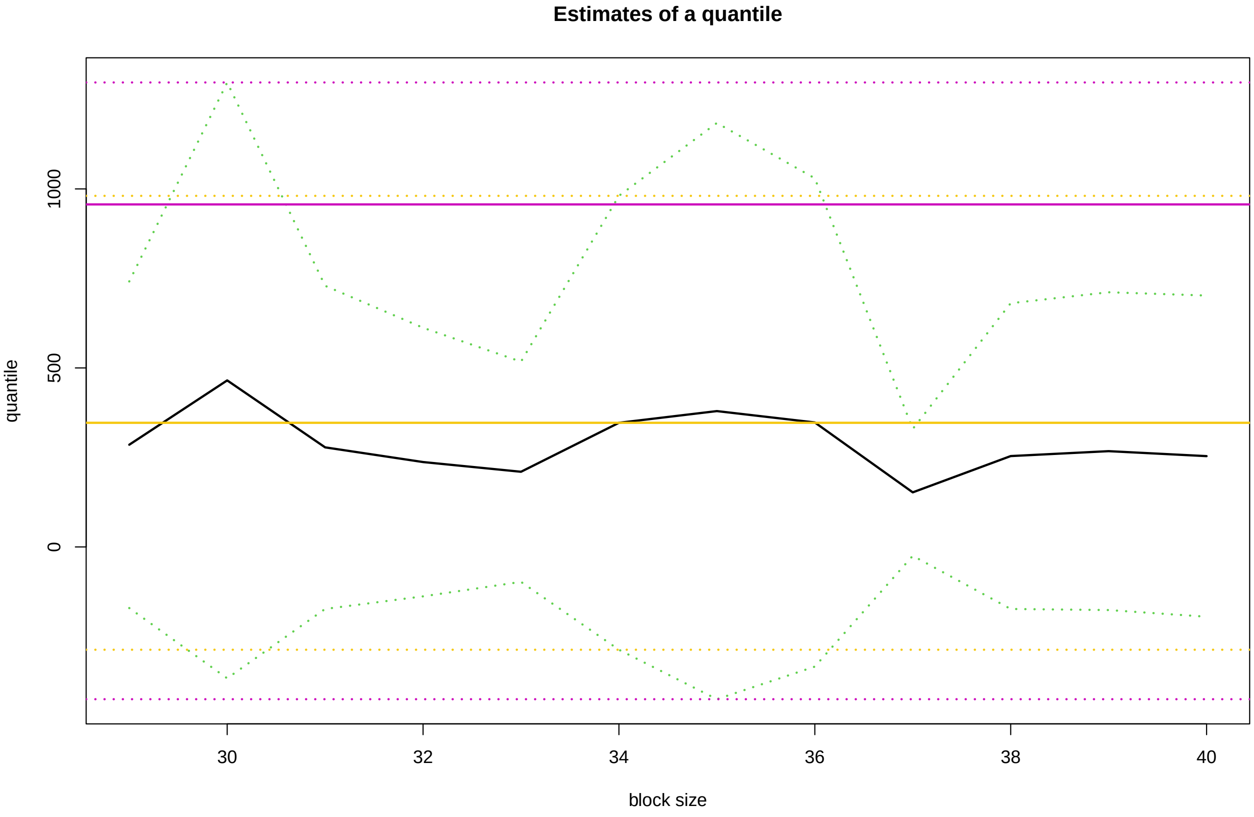
```
##                  lower         estimate            upper
## 34 -287.130679100386 346.758525941091 980.647730982567
```

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

```
## [1] -287.130679100386  980.647730982567
```

```
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 = 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)
```


Estimates of a quantile
quantile
block size