Modeling extreme values with a GEV mixture probability distributions

Pascal Alain Dkengne Sielenou

September 28th, 2023

```r
# library(xfun)
```

```r
path <- ".."
```

```r
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"))
```

```r
n <- 100000
```

```r
set.seed(1122)
x <- rnorm(n = n)
```

```r
nlargest <- 1000
```

```r
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.
```

```r
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]  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
## 9  2.45628443725258 0.256440743198709  0.04256378669663031
## 10 2.47637313395835 0.258234275166653  0.02738164560120679
## 11 2.41543340306138 0.290686060651678  0.00117924257969587
## 12 2.36210335865845 0.335347611311548 -0.04123424946548924
## 13 2.39423783521682 0.321979960277751 -0.03308091180478364
## 14 2.23713203017343 0.403833891594456 -0.08456762308699213
## 15 2.52832405559420 0.237112037607216  0.04876518655428982
## 16 2.38493413827506 0.308693952100226 -0.01572077091047429
## 17 2.55394533700493 0.245046724739142  0.02928797698600330
## 18 2.50689633726953 0.239545317872480  0.04979310041600821
## 19 2.43795358324203 0.290534269252053 -0.00360412083143491
## 20 2.55804489122312 0.210755847668614  0.08137171032744203
```

```
gev_mixture_model$weighted_normalized_gev_parameters_object
```

```
##                             loc_star        scale_star           shape_star
## identic_weights     2.44263854507749 0.283184224286711  0.00851124775517518
## pessimistic_weights 2.45020786975786 0.285873506997451  0.01051224767980413
## automatic_weights   2.33315317672659 0.363324234169863 -0.08456762308699213
```

```
gev_mixture_model$automatic_weights_mw_statistics
```

```
## $function_value
## [1] 0.00249655002489225
##
## $gradient_value
## [1] 9.98927512674275e-06
##
## $function_reduction
## [1] 0.0192397897654247
##
## $number_iterations
## [1] 2405
##
## $convergence
## [1] 0
```

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

```
gev_mixture_model$automatic_weights_pw_statistics
```

```
## $function_value
## [1] 0.00225569403067106
##
## $gradient_value
## [1] 5.4161485770754e-06
##
## $function_reduction
## [1] 0.0172758264439327
##
## $number_iterations
## [1] 4496
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
```

```
gev_mixture_model$automatic_weights_mw
```

```
##                9               10               11               12
## 0.000000000000000 0.000000000000000 0.000000000000000 0.217645067134693
##               13               14               15               16
## 0.229859424904641 0.116978205638081 0.000000000000000 0.435517302322585
##               17               18               19               20
## 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000
```

```
gev_mixture_model$pessimistic_weights_pw_shape
```

```
##                 9                10                11                12
## 0.0861334227071130 0.0848356096494699 0.0826415827308796 0.0792097569225287
##                13                14                15                16
## 0.0798582207968855 0.0758506375433885 0.0866692301634263 0.0812566743152026
##                17                18                19                20
## 0.0849974886832106 0.0867583644698420 0.0822472219452161 0.0895417900728373
```

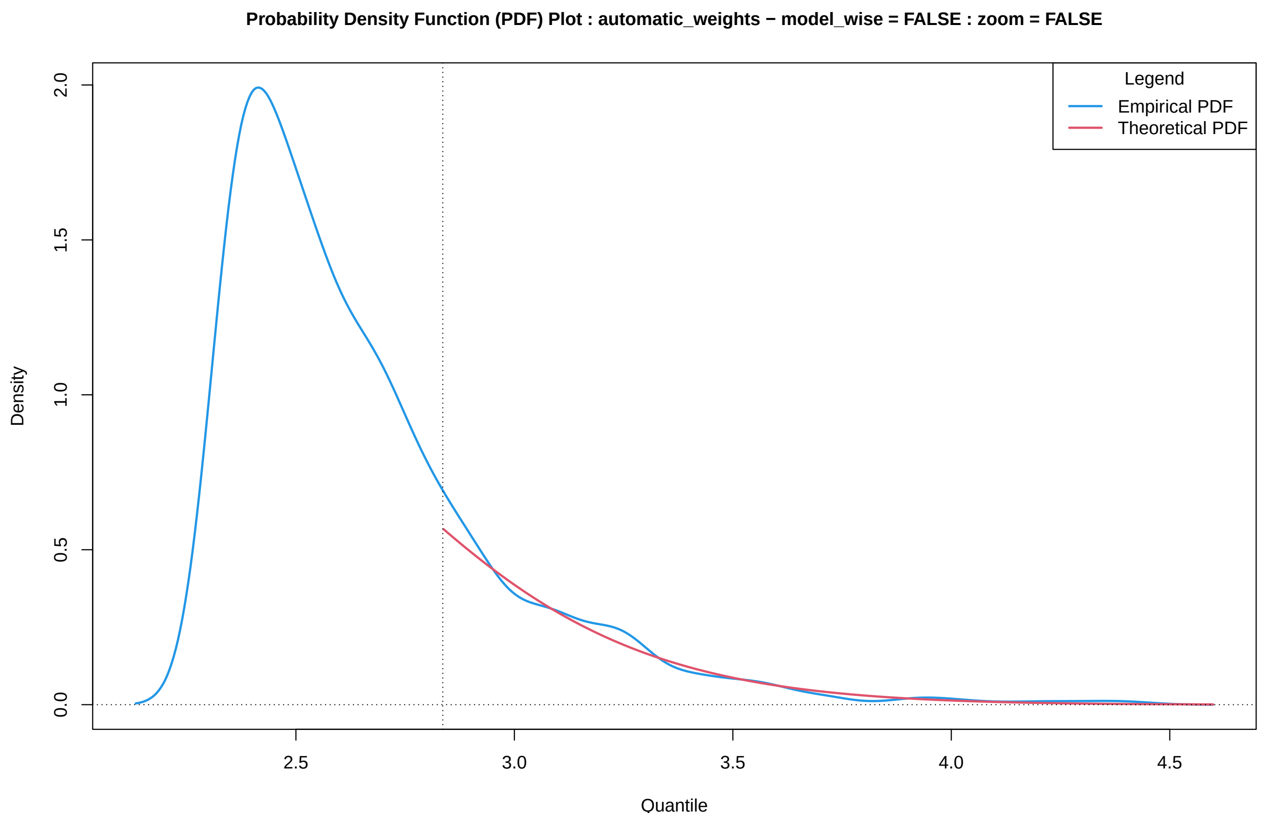
```
gev_mixture_model$pessimistic_weights_pw_scale
```

```
##                 9                10                11                12
## 0.0810259699353189 0.0811714230010026 0.0838487882886427 0.0876784888205607
##                13                14                15                16
## 0.0865142324026847 0.0938936602694034 0.0794748813743538 0.0853724055599297
##                17                18                19                20
## 0.0801079981630140 0.0796685015052749 0.0838360617296270 0.0774075889501875
```

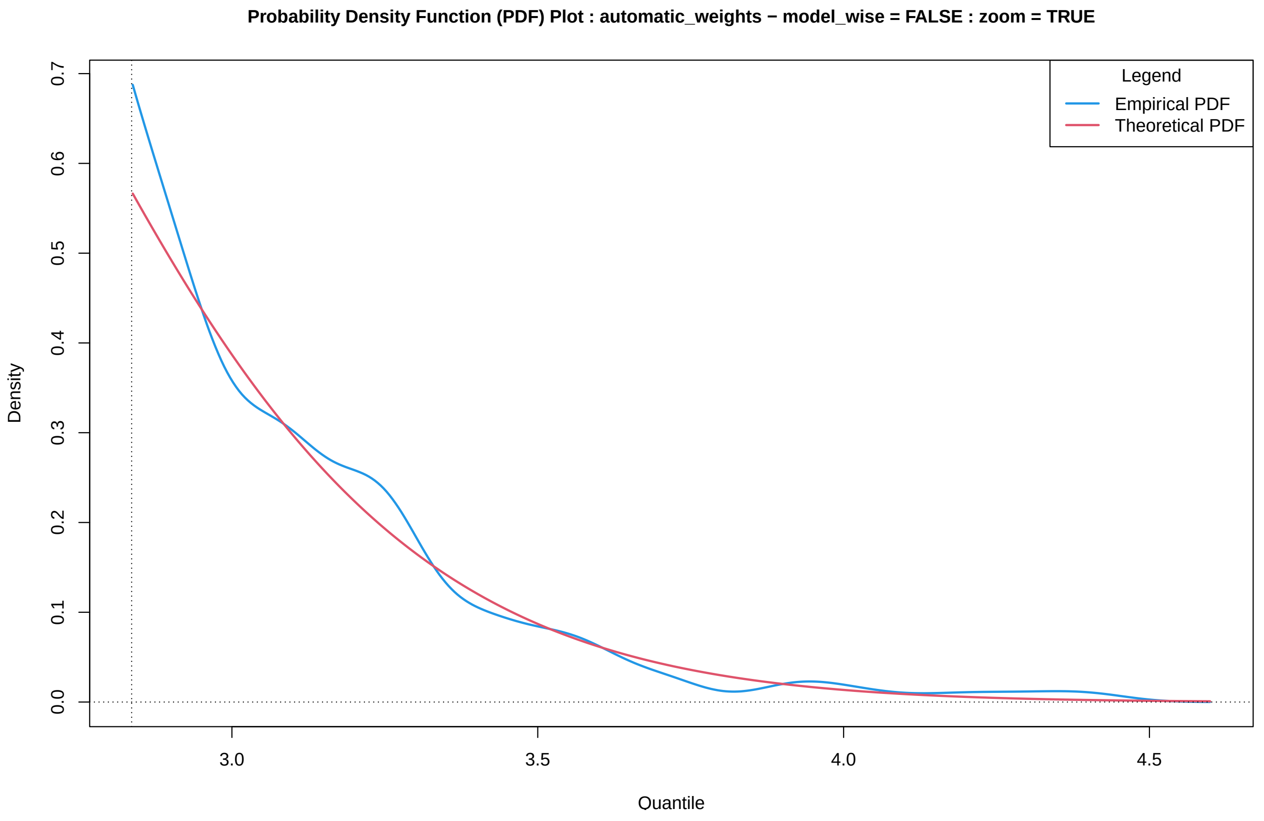
```
gev_mixture_model$pessimistic_weights_pw_loc
```

```
##                 9                10                11                12
## 0.0841558754528664 0.0858635523846430 0.0807872943642429 0.0765917723797842
##                13                14                15                16
## 0.0790929811812481 0.0675939398815008 0.0904421443865863 0.0783605365587992
##                17                18                19                20
## 0.0927893285529685 0.0885247912056398 0.0826272794241551 0.0931705042275658
```

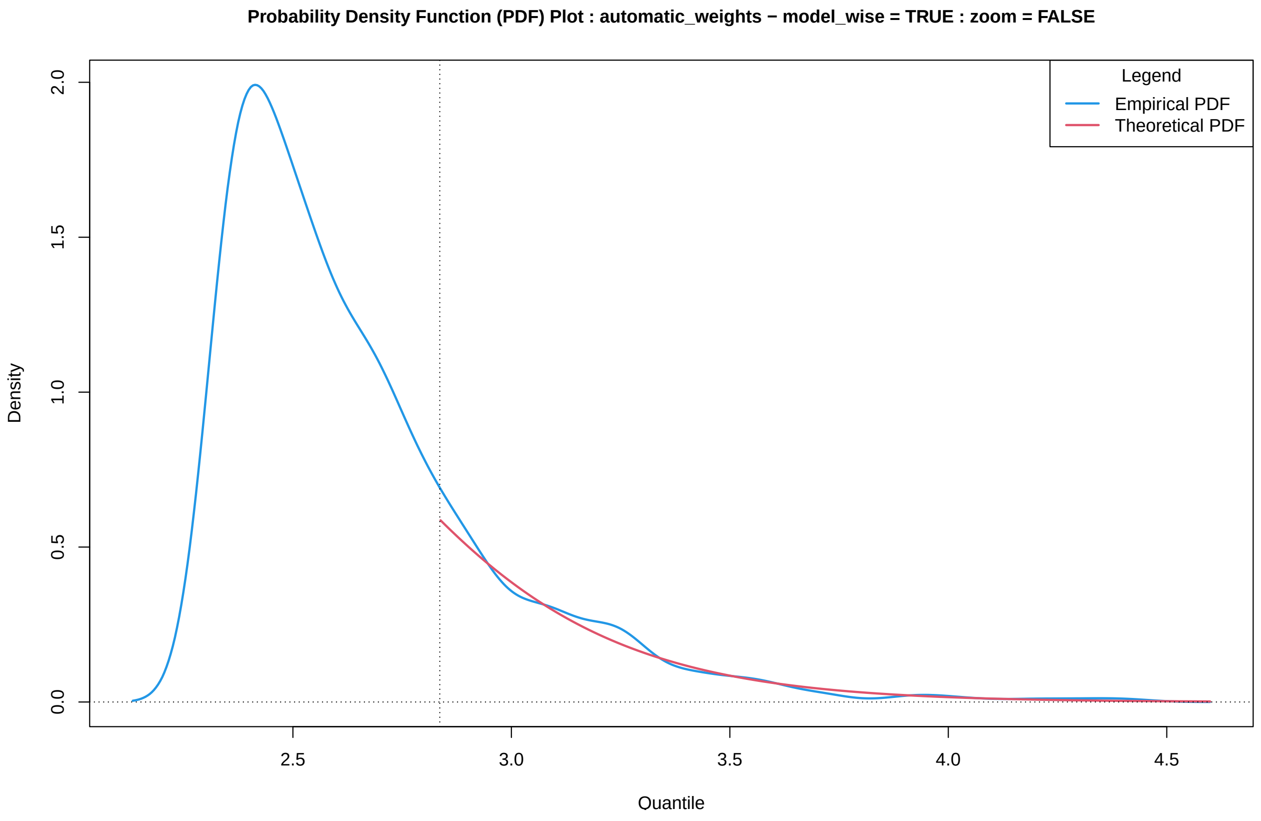
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = FALSE,
                           xlab = "Quantile",
                           ylab = "Density",
                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



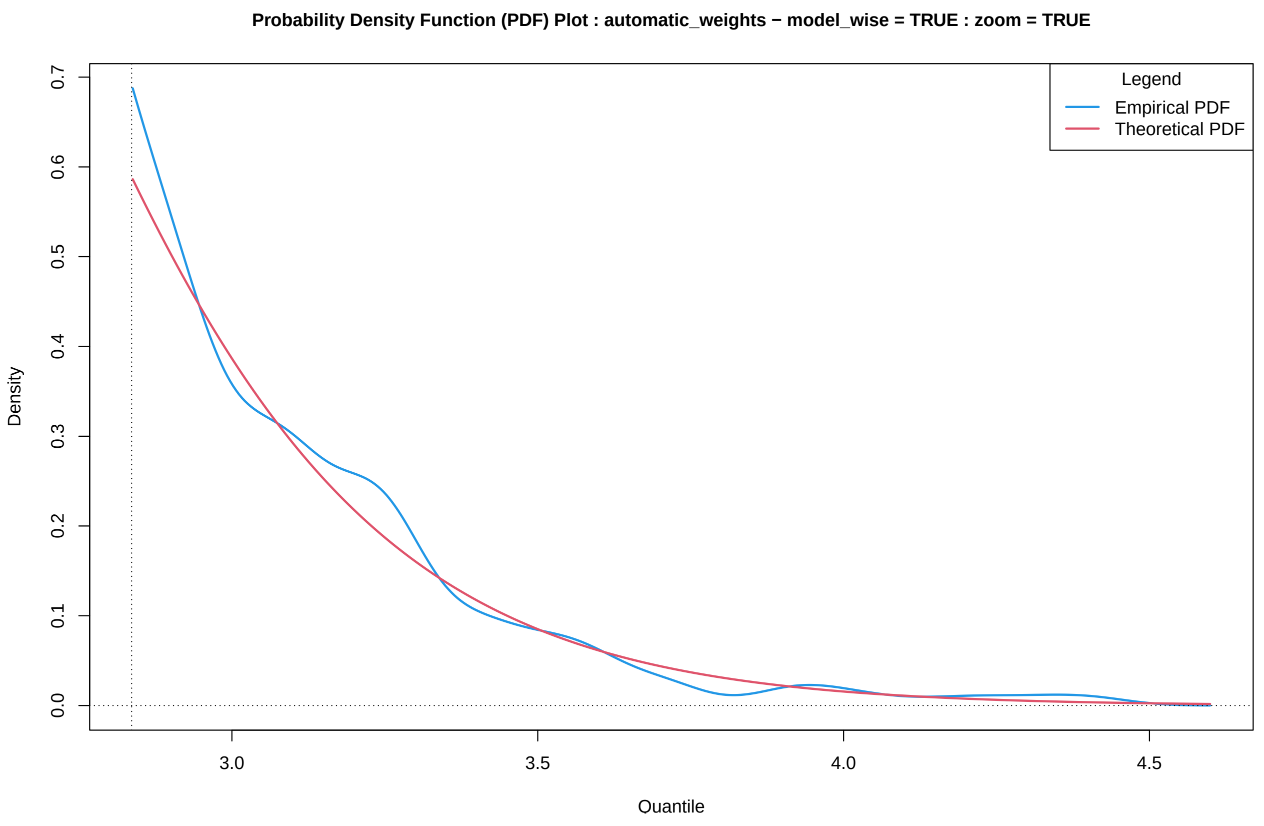
```
plot_gev_mixture_model_pdf(gev_mixture_model,
                           type = "automatic_weights",
                           model_wise = FALSE,
                           zoom = TRUE,
                           xlab = "Quantile",
                           ylab = "Density",
                 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



```r
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")
```

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

```r
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 9.1361288596493    NA
```

```r
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 6.2141903736901    NA
```

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

```
##             100%
## 4.41549649952614
```

```
true_rl <- qnorm(p = 1 - alpha)
true_rl
```

```
## [1] 7.65073090515564
```

```
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
## 9   -14.5448370657666 15.9618808542649 46.4685987742964
## 10  -7.87222655492214 13.1427696185319 34.1577657919859
## 11  -4.33187694955725 10.5798268362574 25.4915306220721
## 12  0.713285466003487 7.89190215772055 15.0705188494376
## 13 -0.306488605311648  8.2254566208075 16.7574018469266
## 14   2.37798722449619 6.55091820312705 10.7238491817579
## 15  -22.6533534491439  16.374257173082  55.401867795308
## 16  -3.40365404588741 9.30340745378741 22.0104689534622
## 17  -13.7916118860738 12.9811746142776 39.7539611146291
## 18  -27.4961046483109 16.7401148584929 60.9763343652966
## 19  -7.21233500698153 10.0791119203642 27.3705588477098
## 20  -52.8359050279607  24.505063861496 101.846032750953
```

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

```
## [1] -52.8359050279607 101.8460327509527
```

```
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
```

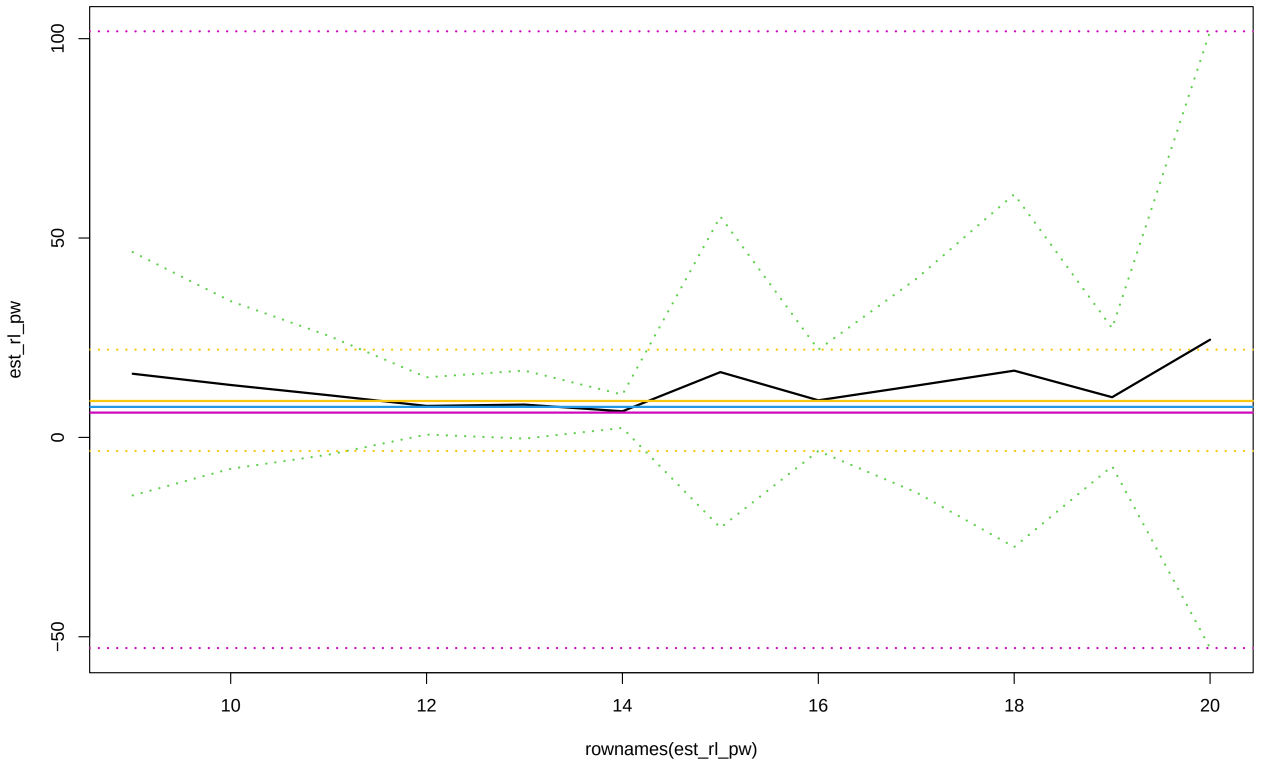
```
##                  lower         estimate            upper
## 12  0.713285466003487 7.89190215772055 15.0705188494376
## 13 -0.306488605311648  8.2254566208075 16.7574018469266
## 14   2.37798722449619 6.55091820312705 10.7238491817579
## 16  -3.40365404588741 9.30340745378741 22.0104689534622
```

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

```
## [1] -3.40365404588741 22.01046895346223
```

```r
matplot(x = rownames(est_rl_pw),
        y = est_rl_pw,
        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)
```


est_rl_pw
rownames(est_rl_pw)