Modeling extreme values with a single GEV probability distribution

Pascal Alain Dkengne Sielenou

September 08th, 2023

```r
source("./src/extract_block_maxima_with_indexes.R")
source("./src/estimate_single_gev_model.R")
source("./src/generate_gev_sample.R")
source("./src/plot_gev_pdf.R")
source("./src/plot_gev_cdf.R")
source("./src/plot_gev_probability.R")
source("./src/plot_gev_quantile.R")
source("./src/plot_block_maxima.R")
```

```r
x <- rnorm(n = 10000)
```

```r
block_size <- 100
```

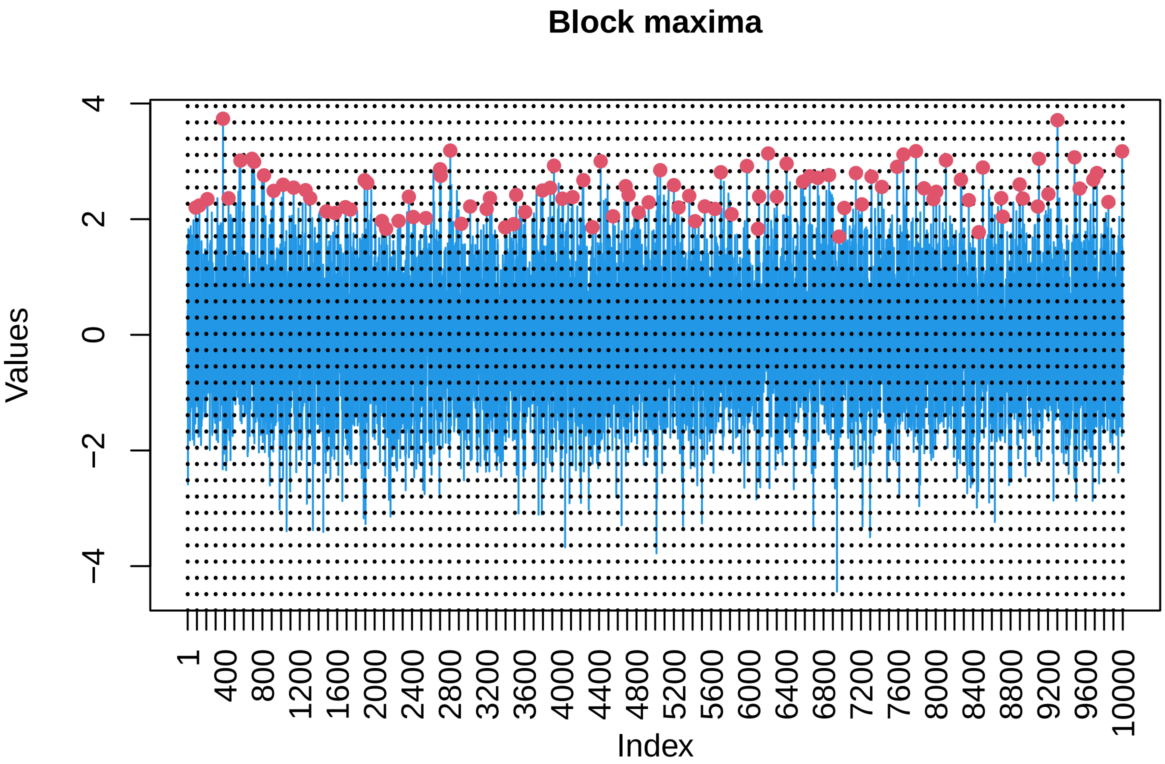
```r
extremes <- extract_block_maxima_with_indexes(x, block_size)

extremes
```

```
## $block_maxima
##   [1] 2.20028187526408 2.23568993628477 2.34688263934000 3.73665236722141
##   [5] 2.36097700069360 3.01313012079986 3.04302297062468 2.99230718761001
##   [9] 2.75787167902122 2.48983462328246 2.59505155838516 2.54500837543624
##  [13] 2.50120718407220 2.36310196486055 2.13243592491306 2.10611281789456
##  [17] 2.20692152792308 2.16705721350597 2.67302658876226 2.62696359824407
##  [21] 1.96822351014685 1.83252361425334 1.96933331774024 2.38819984691435
##  [25] 2.03746661019938 2.01831512412205 2.86316252434076 2.75001574707549
##  [29] 3.18595270568307 1.91933704821697 2.21938529421043 2.17678626478507
##  [33] 2.36471443044589 1.85750126794027 1.91428658685998 2.41966281906710
##  [37] 2.12458717151360 2.49549000080458 2.54184434091986 2.92454387880206
##  [41] 2.35471137583219 2.38096077671289 2.67391689460237 1.86060440412983
##  [45] 3.00089549729213 2.04917789689556 2.57037464757640 2.42086157105889
##  [49] 2.11523289147090 2.29149396131329 2.84698455502010 2.58782536668379
##  [53] 2.20202649003422 2.40261278691033 1.96436280546751 2.22028400832909
##  [57] 2.17618130861594 2.81102410070242 2.08560284269832 2.91943288518376
##  [61] 1.83378919833219 2.39336727117055 3.13558002368626 2.38771317847590
##  [65] 2.96085123813813 2.64904180286711 2.74137199417495 2.71479446429908
##  [69] 2.76142472140487 1.69782851054344 2.19393494704852 2.79759112017256
##  [73] 2.25548105908212 2.73611656401008 2.55755221156983 2.90522468830336
##  [77] 3.11860570647491 3.17584320699073 2.53400936509325 2.34577524994492
##  [81] 2.47288063083092 3.01869417644990 2.68328541844602 2.32883170989367
##  [85] 1.77429179004985 2.89334615726424 2.36436513586572 2.04182432856071
##  [89] 2.60065679941427 2.35629343015435 2.21872257629870 3.04420542204399
##  [93] 2.43444378270389 3.71206383267352 3.06909532052141 2.52878782472923
##  [97] 2.68338988878601 2.79755665646339 2.29646580262054 3.17288896776477
##
```

```
## $block_maxima_indexes
##   [1]   87  126  212  379  438  565  689  711  817  920 1022 1134 1262 1311 1484
##  [16] 1575 1688 1742 1893 1922 2078 2123 2257 2366 2413 2548 2700 2707 2809 2926
##  [31] 3022 3198 3233 3396 3487 3514 3610 3796 3878 3918 4008 4117 4232 4329 4417
##  [46] 4550 4684 4715 4823 4930 5054 5200 5250 5364 5427 5530 5638 5702 5817 5981
##  [61] 6099 6111 6208 6301 6405 6581 6653 6739 6859 6966 7022 7146 7210 7313 7424
##  [76] 7587 7655 7789 7875 7975 8005 8109 8269 8353 8460 8504 8700 8717 8897 8931
##  [91] 9091 9103 9205 9302 9484 9538 9685 9723 9846 9993
```

```r
plot_block_maxima(x, block_size, xlab = "Index", ylab = "Values", main = "Block maxima")
```



```r
model <- estimate_single_gev_model(x, block_size, nsloc = NULL)
```

```r
names(model)
```

```
## [1] "data"                        "block_maxima_indexes"
## [3] "gev_model"                   "block_size"
## [5] "extremal_index"              "normalized_gev_parameters"
```

```r
model$gev_model
```

```
##
## Call: evd::fgev(x = x, nsloc = nsloc, prob = NULL, std.err = std.err,      corr = FALSE, method = "BF
## Deviance: 99.1838942180778
##
## Estimates
##             loc           scale           shape
##  2.324755335705   0.365327420989  -0.131426487585
##
## Optimization Information
##   Convergence: successful
##   Function Evaluations: 42
##   Gradient Evaluations: 13
```

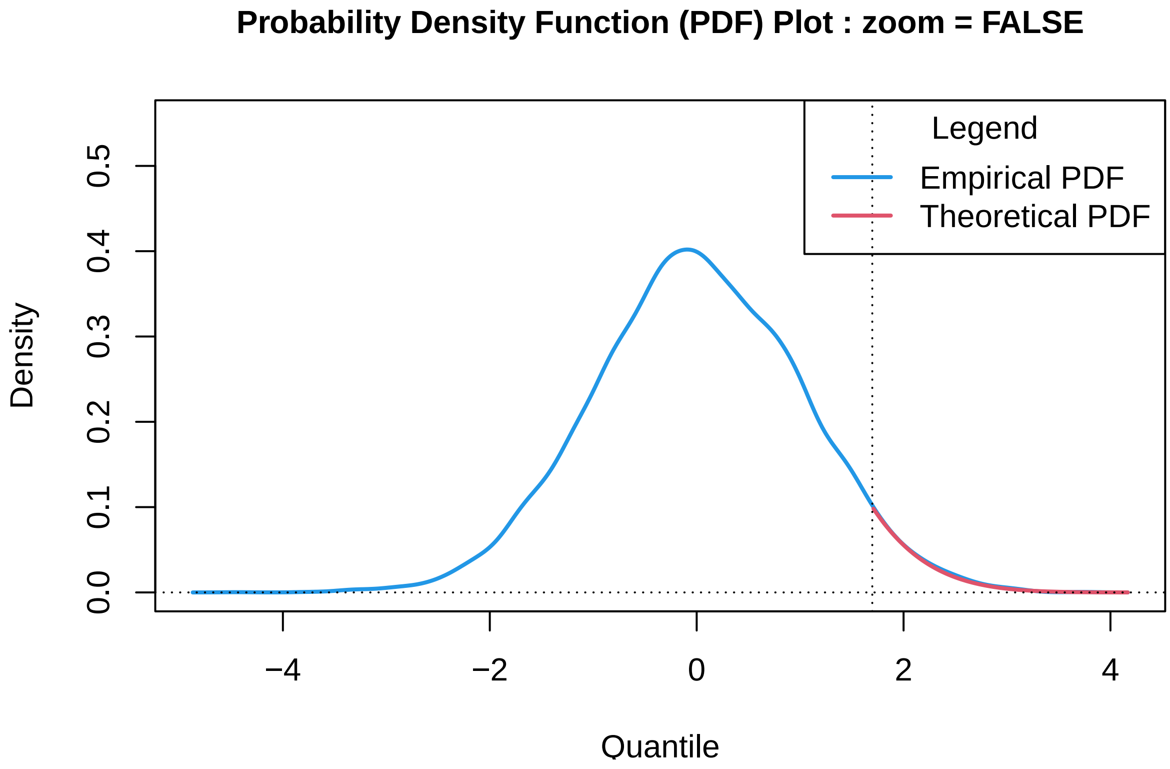
```r
names(model$gev_model)
```

```
##  [1] "estimate"    "std.err"     "fixed"       "param"       "deviance"
##  [6] "corr"        "var.cov"     "convergence" "counts"      "message"
## [11] "data"        "tdata"       "nsloc"       "n"           "prob"
## [16] "loc"         "call"
```

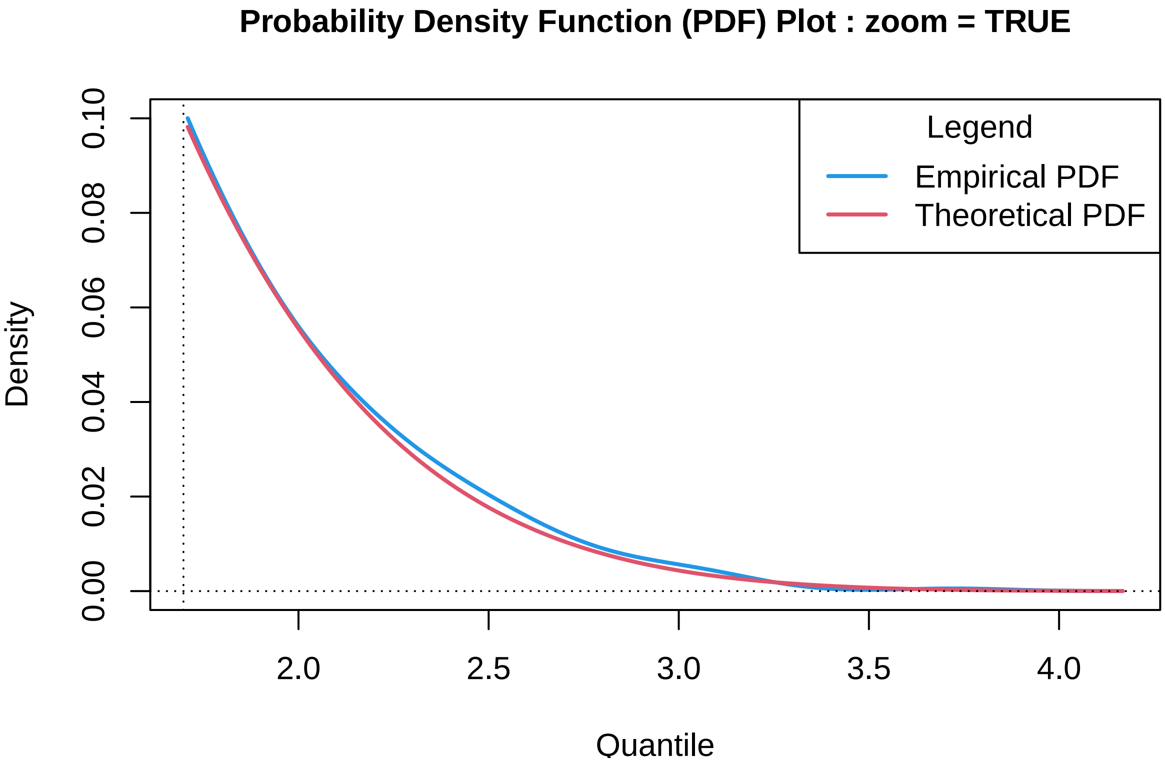
```r
model$normalized_gev_parameters
```

```
##           loc_star         scale_star         shape_star
##  0.012887231197042  0.669168125724828 -0.131426487585311
```

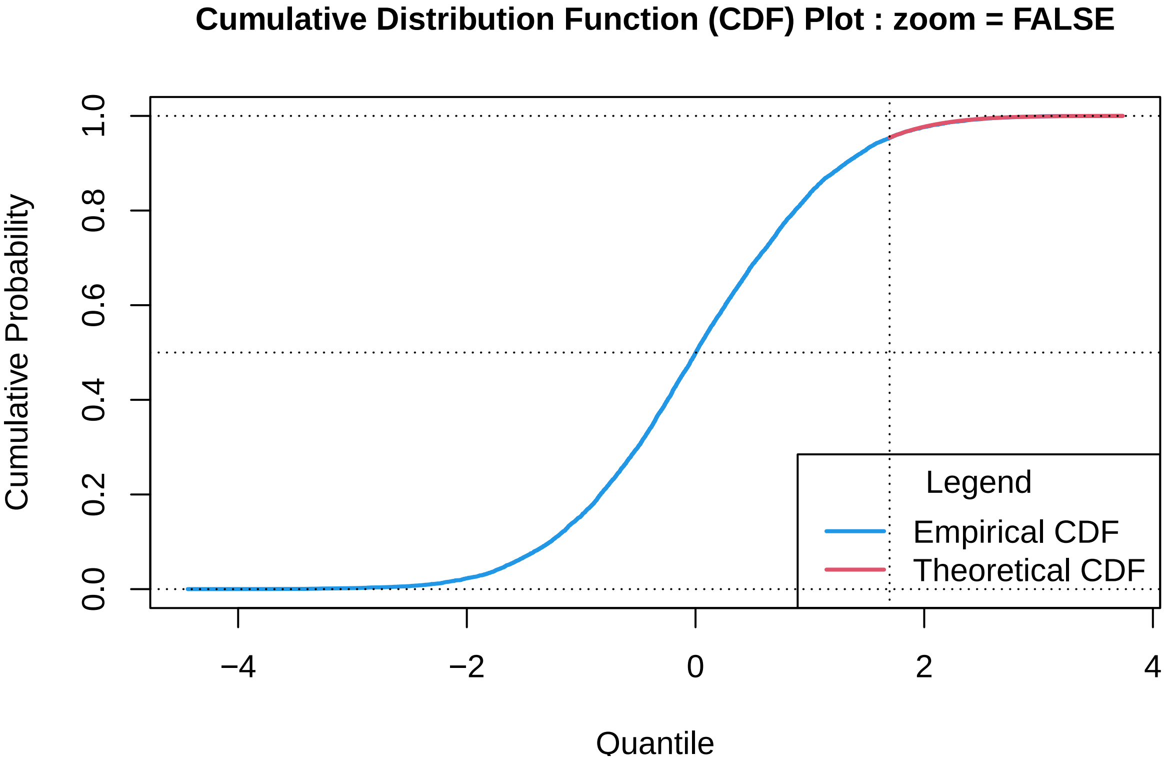
```r
plot_gev_pdf(model,
             zoom = FALSE,
             xlab = "Quantile",
             ylab = "Density",
             main = "Probability Density Function (PDF) Plot")
```



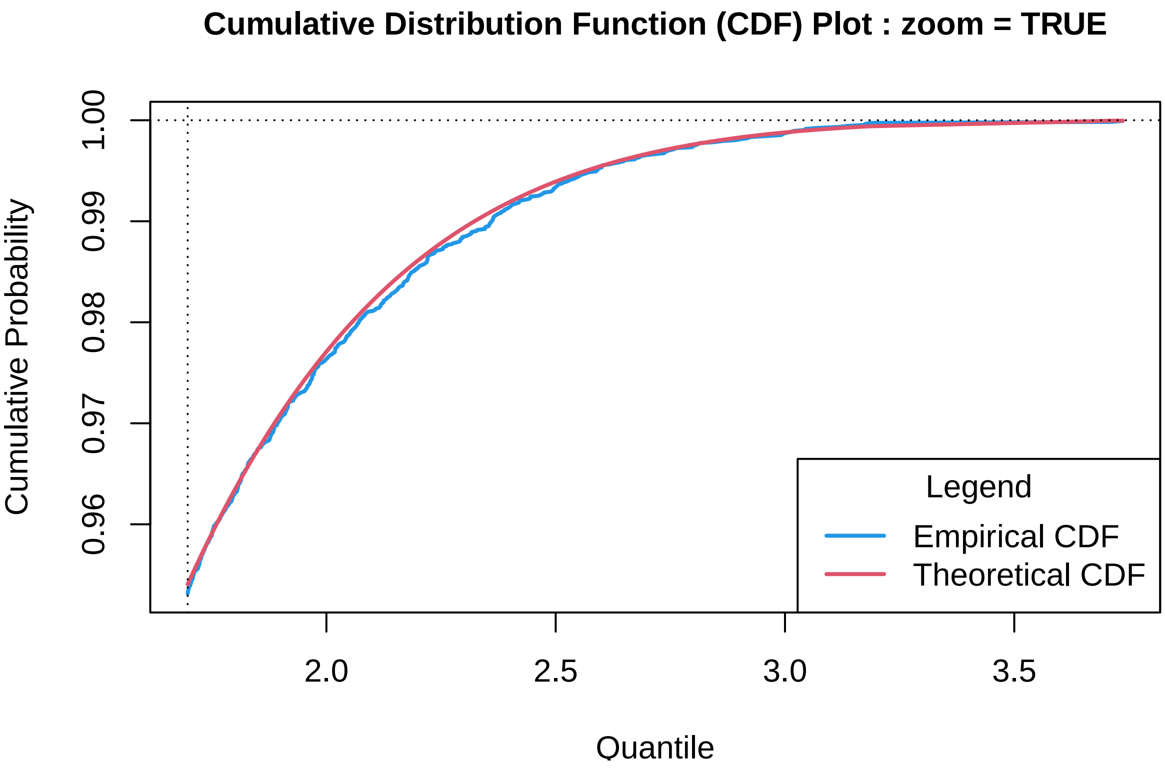
```r
plot_gev_pdf(model,
             zoom = TRUE,
             xlab = "Quantile",
             ylab = "Density",
             main = "Probability Density Function (PDF) Plot")
```



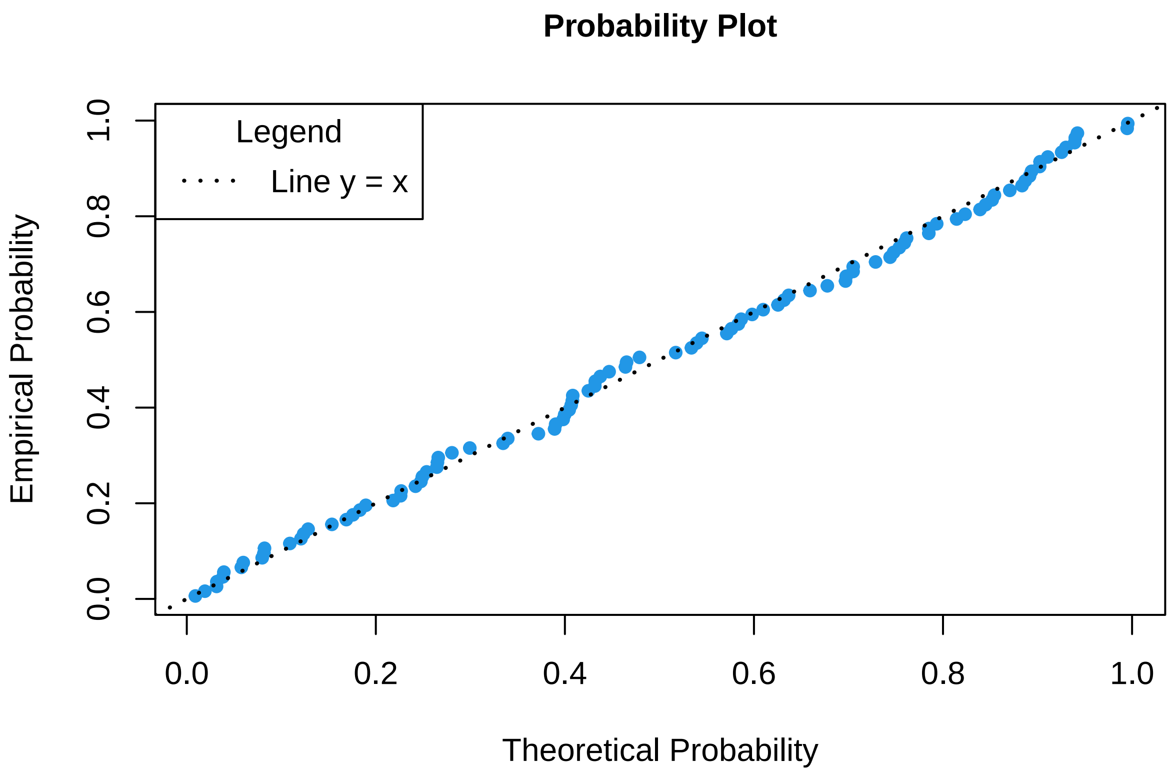
```r
plot_gev_cdf(model,
             zoom = FALSE,
             xlab = "Quantile",
             ylab = "Cumulative Probability",
             main = "Cumulative Distribution Function (CDF) Plot")
```



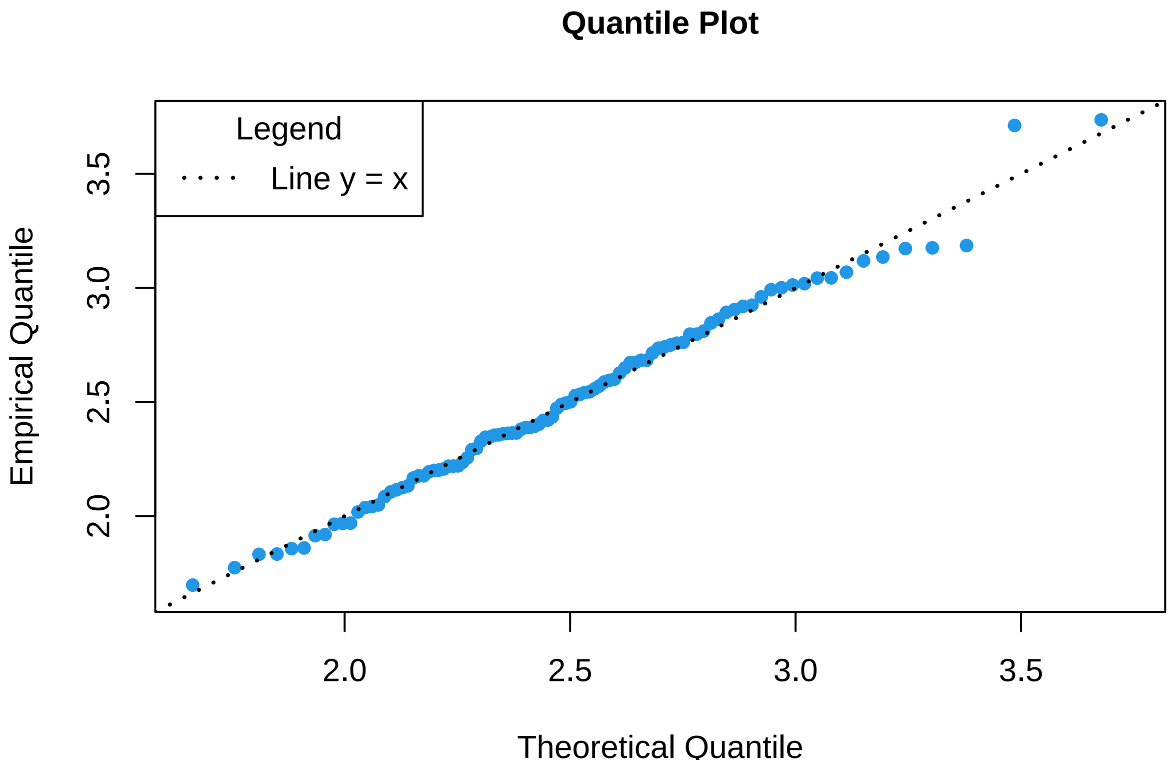
```
plot_gev_cdf(model,
             zoom = TRUE,
             xlab = "Quantile",
             ylab = "Cumulative Probability",
             main = "Cumulative Distribution Function (CDF) Plot")
```



```
plot_gev_probability(model,
                     xlab = "Theoretical Probability",
                     ylab = "Empirical Probability",
                     main = "Probability Plot")
```



```r
plot_gev_quantile(model,
                  xlab = "Theoretical Quantile",
                  ylab = "Empirical Quantile",
                  main = "Quantile Plot")
```


Quantile Plot
Legend
Line y = x
Empirical Quantile
Theoretical Quantile