

# Modeling extreme values with a single GEV probability distribution

Pascal Alain Dkengne Sielenou

September 08th, 2023

```
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")

x <- rnorm(n = 1000)

block_size <- 40

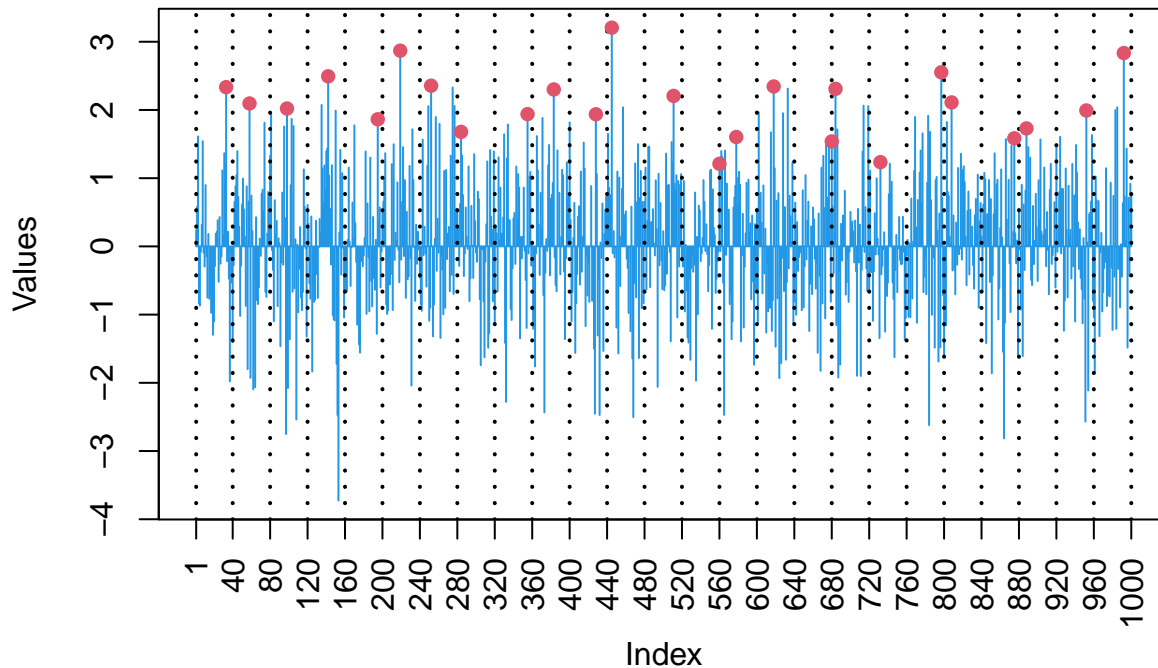
extremes <- extract_block_maxima_with_indexes(x, block_size)

extremes

## $block_maxima
## [1] 2.33461936912460 2.09551296632651 2.02099131958727 2.49332339366973
## [5] 1.86303174965445 2.86944241507028 2.35564071413536 1.67885399588604
## [9] 1.93878078743715 2.30069222303712 1.93707209685423 3.20617959341111
## [13] 2.20583638476356 1.21138009705697 1.60337158923168 2.34451386883541
## [17] 1.53891610826960 2.30935654848828 1.23484687092620 2.55203515462720
## [21] 2.11065360055698 1.58498656664887 1.72958250254334 1.99254313217680
## [25] 2.83450496675310
##
## $block_maxima_indexes
## [1] 33 58 98 142 195 219 252 284 355 383 428 445 511 560 578 618 680 684 732
## [20] 797 808 875 888 952 992

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

## Block maxima



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

```
names(model)
```

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

```
model$gev_model
```

```
##
## Call: evd::fgev(x = x, nsloc = nsloc, prob = NULL, std.err = std.err,      corr = FALSE, method = "B
## Deviance: 34.0140517582797
##
## Estimates
##           loc           scale           shape
## 1.911271301122 0.462079442585 -0.222093182348
##
## Optimization Information
## Convergence: successful
## Function Evaluations: 40
## Gradient Evaluations: 13
```

```
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"
```

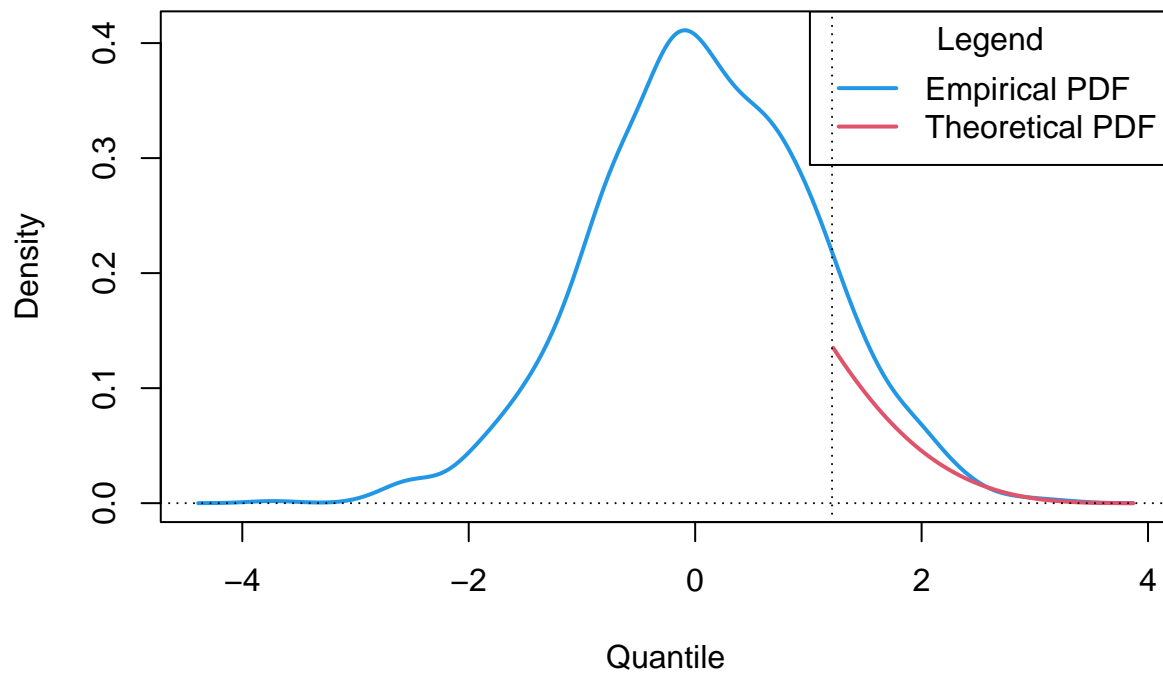
```
model$normalized_gev_parameters
```

```
##           loc_star           scale_star           shape_star
```

```
## -0.728663286992686 1.048390916450072 -0.222093182347938
```

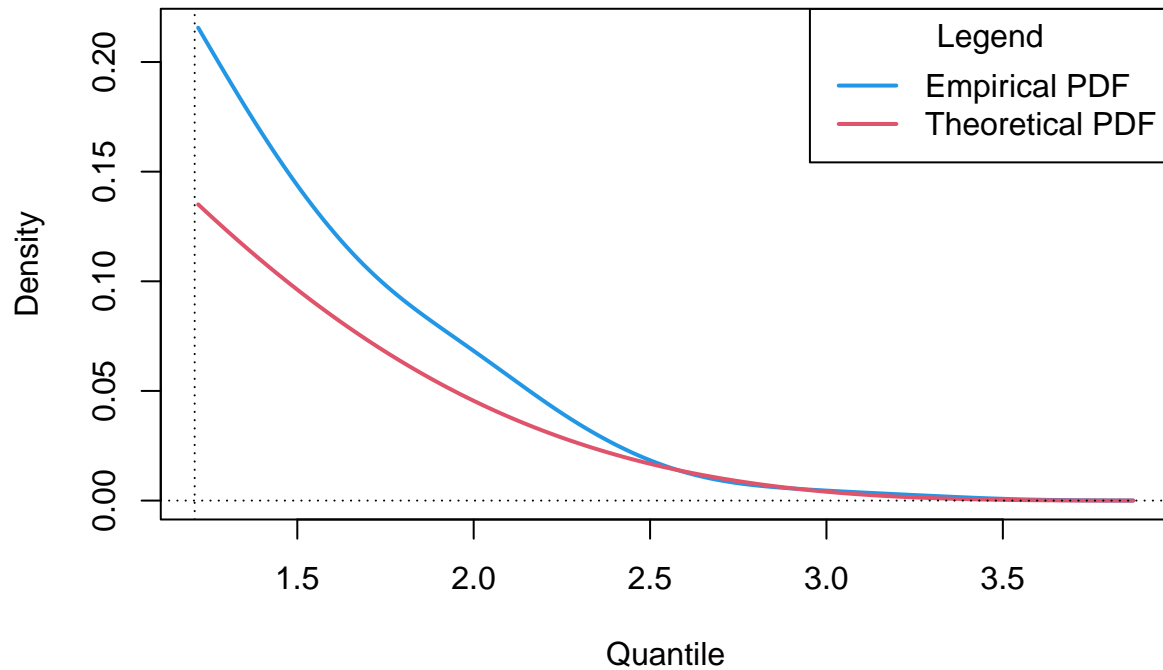
```
plot_gev_pdf(model,  
  zoom = FALSE,  
  xlab = "Quantile",  
  ylab = "Density",  
  main = "Probability Density Function (PDF) Plot")
```

**Probability Density Function (PDF) Plot : zoom = FALSE**



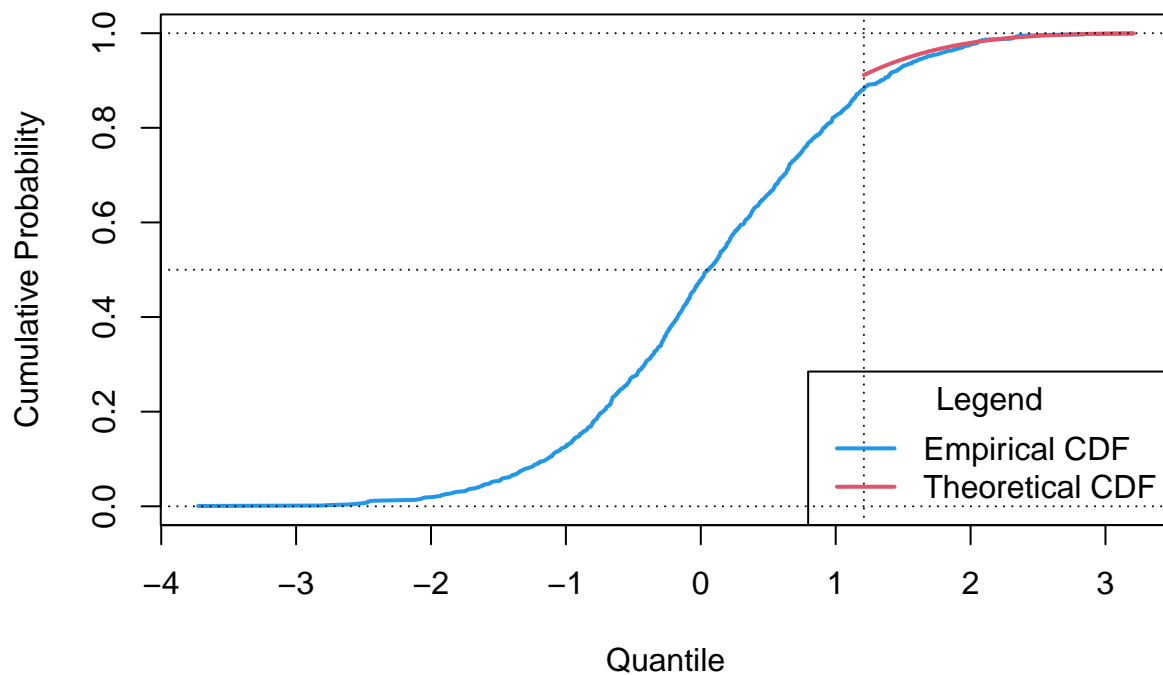
```
plot_gev_pdf(model,  
  zoom = TRUE,  
  xlab = "Quantile",  
  ylab = "Density",  
  main = "Probability Density Function (PDF) Plot")
```

**Probability Density Function (PDF) Plot : zoom = TRUE**



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

**Cumulative Distribution Function (CDF) Plot : zoom = FALSE**



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

**Cumulative Distribution Function (CDF) Plot : zoom = TRUE**

