EPFL, under the supervision of Pr. Stephan Morgenthaler

Master Thesis Project : Extreme Value Theory
– Introductory Task –

Killian Martin–Horgassan

February 22, 2015

1 Summary - tasks assigned

Let us consider a sequence of i.i.d. random variables $X_1, X_2, \ldots \sim \mathcal{N}(0,1)$[1]. Let us define the maximum at stage n as $M_n = \max(X_1, \ldots, X_n)$.
Let us consider the following example :

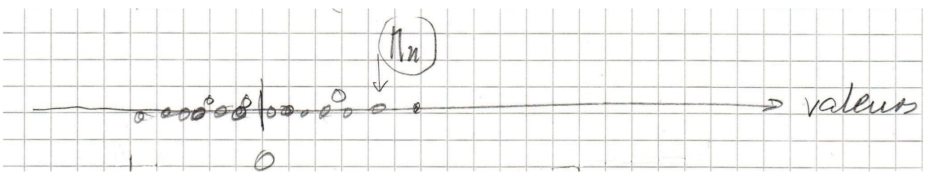
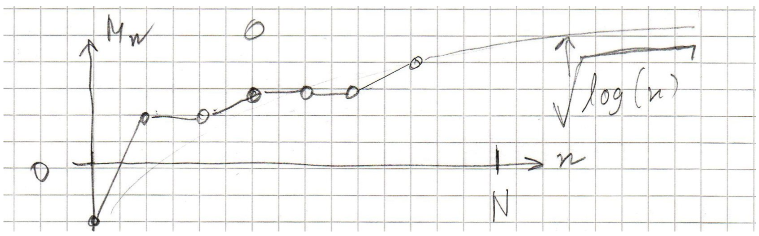


Figure 1.1: Distribution of the $x_1, \ldots, x_n$ around 0.

Let us draw the plotting of M_n against n :



(M_n) is a non-decreasing sequence with growth is of order $\sqrt{\log(n)}$.

[1] could be extended to $\mathcal{N}(\mu, \sigma^2)$

1.1 Sub-task 1

Write a R program that will generate the sequence of the maxima $M_1,...,M_n$ and make the plot of M_n against n. The signature is the following : NormMax(N,m,s,DIST), where :

- N is the number of stages
- m is the mean
- s is the standard deviation
- DIST is the common distribution of the X_i[2]

1.1.1 Sub-task 2

Let us make the following observation, if F_n denotes the cumulated distribution function of M_n then : $F_n(t) = \Pr(\{M_n \leq t\}) = \Pr(\{\max(X_1,...,X_n) \leq t\}) = \Pr(\{X_1 \leq t\} \cap ... \cap \{X_n \leq t\})$.
By independence of the X_i, it boils down to $F_n(t) = \Phi(t)^n$.

- What is the limit, if any, when n goes to infinity ?
- Is it possible to find two sequences (a_n) and (b_n) such that $\Pr(a_n + b_n M_n \leq t)$ has a limit distribution ?

2 Results

2.1 R programming

```
# Master Thesis Project - Extreme Value Theory
# Introductory Task
# Killian Martin--Horgassan
# 19-02-2015

# Clear the environment
rm(list=ls())

# Close all already open graphic windows
graphics.off()

# Elementary test
#x <- 5
#print(x)

# sourcing works
```

[2] gaussian, exponential or Cauchy for instance.

```r
# Loading function 'NormMax'
source("/Users/kimartin/Desktop/R_files_pdm/NormMax.r")
source("/Users/kimartin/Desktop/R_files_pdm/InputIntroductoryTask.r")

# Calling function NormMax
#listMax <- NormMax()

# Input from keyboard
DIST <- InputIntroductoryTask()

# Calling function NormMax
listMax <- NormMax(length(DIST),DIST)
```

⇒ **script**

```r
# Function 'NormMax'
# Killian Martin--Horgassan
# 19-02-2015

# Generates the list of maxima [M_1, ..., M_n] of the list of
# i.i.d random variables [X_1,...,X_n]. Plots the M_i against
# the i.

# Arguments :
# - N       : number of r.v. X_i
# - DIST    : a vector of N pseudo-random numbers from a certain
#                       distribution.
# RETURNS the list of maxima

NormMax <- function(N = 10000, DIST = rnorm(10000,0,1)) {
        ListX   <- DIST
        ListMax <- rep(0,N)

        # Computes the list of the M_i
        for (i in 1:length(ListMax)) {
                ListMax[i] <- max(ListX[1:i])
        }

        # Plots the 1-D scatter plot of the M_i
        title_1 <- "1-D scatter plot of the maxima"
        xlabel_1 <- expression(M[i])
        stripchart(ListMax, xlab = xlabel_1, main = title_1)

        # Opens a new graphic window
```

```
        quartz()

        # Plots the M_i against the i
        title_2 <- "Maxima␣as␣a␣function␣of␣the␣time␣steps"
        xlabel_2 <- "Time␣step"
        ylabel_2 <- "Maximum"
        plot(ListMax, xlab = xlabel_2, ylab = ylabel_2, main =
                        title_2)

        # Opens a new graphic windows
        quartz()

        # Plot the 1-D scatter plot of the M_i and the M_i against the
        # i together in a grid.
        par(mfrow = c(1,2))

        stripchart(ListMax, xlab = xlabel_1, main = title_1)
    plot(ListMax, xlab = xlabel_2, ylab = ylabel_2, main =
                        title_2, pch = 4)
        points(1:N, sqrt(log(1:N)), col="cyan", pch = 3)
        legend(x="bottomright", y=NULL, c("Maxima",
                expression("sqrt(log(n))")), col = c("black","cyan"),
                bty="o", pch = c(4,3))

        # Exports the plots to a PDF file
        pdf(file = "graphIntroTask.pdf", width = 10, height = 8)
        par(mfrow = c(1,2))
        stripchart(ListMax, xlab = xlabel_1, main = title_1)
    plot(ListMax, xlab = xlabel_2, ylab = ylabel_2, main =
                        title_2, pch = 4)
        points(1:N, sqrt(log(1:N)), col="cyan", pch = 3)
        legend(x="bottomright", y=NULL, c("Maxima",
                expression("sqrt(log(n))")), col = c("black","cyan"),
                bty="o", pch = c(4,3))
        dev.off()

        return(ListMax)
}
```

⇒ **Core function**

```
# Function 'InputIntroductoryTask'
# Killian Martin--Horgassan
# 19-02-2015
```

```r
# Manages the input for the sample size and distribution chosen
# USED IN script IntroductoryTask.r

InputIntroductoryTask <- function() {
cat("Enter the sample size")
Nb <- as.numeric(readline("Sample size :\n"))
cat("Choose a distribution :\n","- [1] for a gaussian distribution\n",
        "- [2] for an exponential distribution\n",
        "- [3] for a Cauchy distribution\n")
choice_dist <- as.character(readline("Your choice ? :\n"))
if (choice_dist == "1") {
        m  <- as.numeric(readline("\nExpectation ? :\n"))
        s  <- as.numeric(readline("\nStandard deviation ? :\n"))
        DIST <-rnorm(Nb,m,s)
} else if (choice_dist == "2") {
        lambda  <- as.numeric(readline("\nRate ? :\n"))
        DIST <- rexp(Nb,lambda)
} else if (choice_dist == "3") {
        l  <- as.numeric(readline("\nLocation ? :\n"))
        s  <- as.numeric(readline("\nScale ? :\n"))
        DIST <- rcauchy(Nb,l,s)
} else {
        stop("\n!! Invalid choice, try again !!\n")
}
output <- DIST
}
```

⇒ **Encapsulates input management in the script** See appendix for an example of plot (1000 i.i.d standard normal random variables).

2.2 THEORETICAL PART

WHAT IS THE LIMIT, IF ANY, WHEN N GOES TO INFINITY ?

$\forall t \in \mathbb{R}$, $0 < \Phi(t) < 1$. Hence, $\forall t \in \mathbb{R}$, $\lim_{n\to\infty} \Phi(t) = 0$. Obviously, there is no convergence in distribution[3] !

Here, the issue is particularly acute as the upper-bound limit of Ω which I hereafter denote by t_+[4] is infinity. If t_+ is finite there is a limiting cumulative distribution function but it is degenerate. That raises an issue as a degenerate distribution is not something it is convenient to work with. It is possible to circumvent that issue by considering a linear renormalization of M_n namely $M_n^* = \frac{M_n - b_n}{a_n}$[5]. That brings us to the next issue : is it possible to find two sequences

[3] Of course there is no convergence in distribution : the limiting function is not a cdf. Indeed, a cdf must take value 0 at $-\infty$ and value 1 at ∞ and here the limiting function is constant.

[4] that is the smallest t such that $F(t) = 1$

[5] I use here the same notation as Coles in his book.

(a_n) and (b_n) such that (M_n^*) admits a limiting distribution ?

IS IT POSSIBLE TO FIND TWO SEQUENCES (a_n) AND (b_n) SUCH THAT $\Pr(a_n + b_n M_n \leq t)$ HAS A LIMIT DISTRIBUTION ?

The extremal types theorem[6] provides a related result. It states that **if** there exists $((a_n),(b_n)) \in {\mathbb{R}_+^*}^{\mathbb{N}} \times \mathbb{R}^{\mathbb{N}}$ such that $\lim_{n\to\infty} \Pr(\{\frac{M_n - b_n}{a_n} \leq t\}) = F(t)$ **then** F is non-degenerate and either a Grumbel, a Fréchet or a Weibull distribution. These distributions can be regrouped into the Generalized Extreme Value Distribution (**GEV**).
The aforementioned theorem does not allow us to conclude as to the existence of a limiting cdf. There exists sufficient conditions, for instance the Von Mises condition that guarantee under additional assumptions that the theorem can be applied.

3 POINTERS

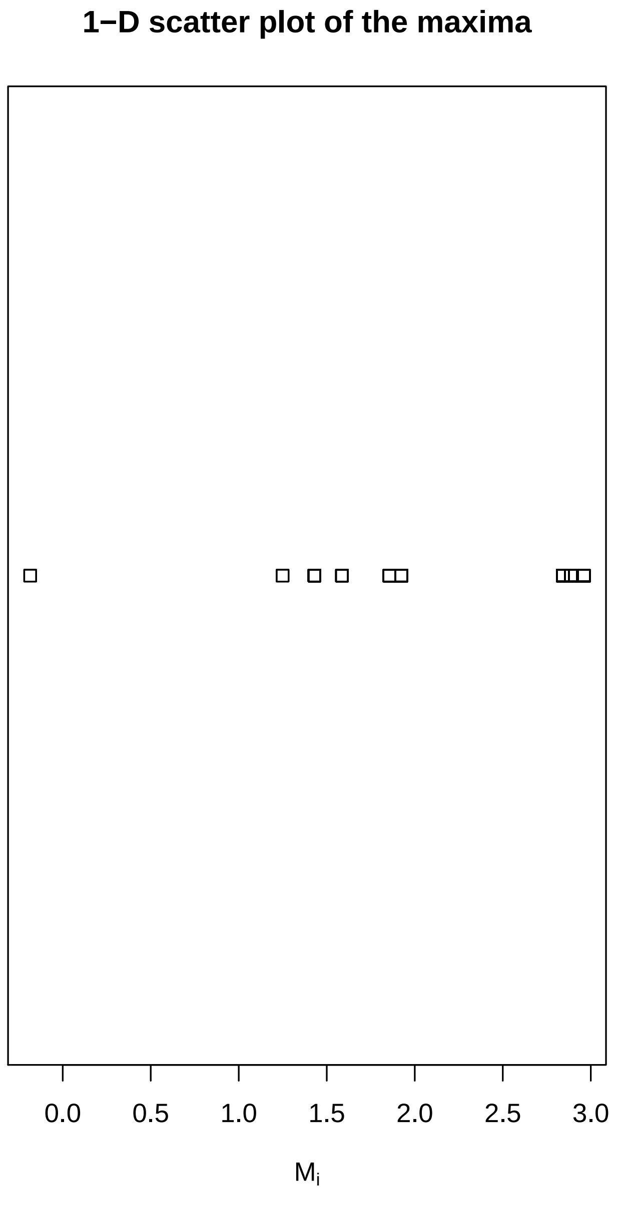
- re-read lecture notes
- study more in-depth properties of the order statistics
- Now an interesting thing would be to estimate the parameters of the 'GEV distribution generated' : theoretical aspects + estimation of parameters in R

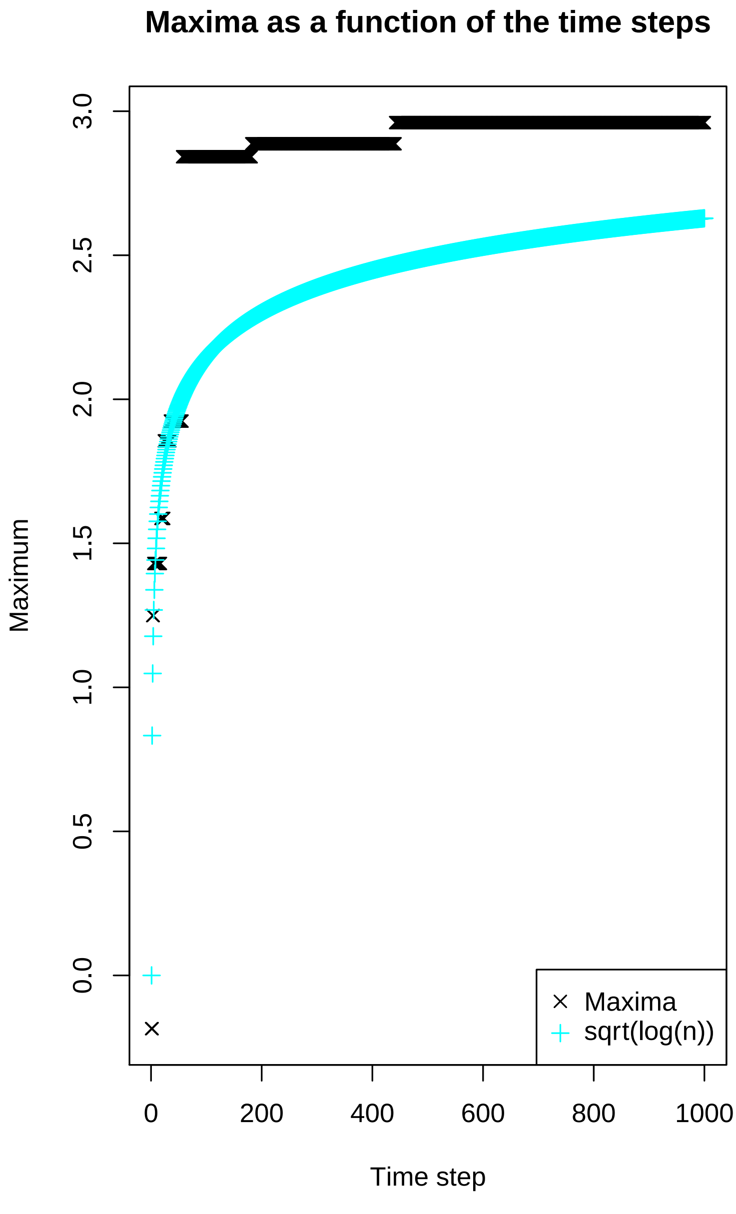
4 RECAP : WORK DONE

Please report to file 'JournalDeBord.pdf'[7], section *Week I.*

5 APPENDIX

[6] also referred to as Fisher-Tippett-Gnedenko theorem in the literature.
[7] available on `https://github.com/CillianMH/pdmExtremeValueTheory`


1−D scatter plot of the maxima
0.0
0.5
1.0
1.5
2.0
2.5
3.0
M_i


Maxima as a function of the time steps
Maximum
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
200
400
600
800
1000
Time step
Maxima
sqrt(log(n))