Computer exercise - Market and Systemic Risk Management

A. Schmidt

June, 2021

Introduction

This is the computational appendix with the analysis made for the Financial Report at the Pokè Bank Johto Region regarding the investment in the Rocket S.A. portfolio.

Dependency installation

```r
rm(list=ls())        # Clean memory

# Verify if a package is installed, if not, download and install before loading.
chooseCRANmirror(graphics = FALSE, ind = 10)
if (!require("pacman")) install.packages("pacman")
pacman::p_load(reshape2, ggplot2, latex2exp, gridExtra, readxl,
               RColorBrewer, grid, knitr, zoo, scales, evir)

# Shapper is a wrapper for the Python package with the Shap values.
# In case you don't have it on Python, you need to install running the following line
#shapper::install_shap()
```

```r
# Prevents code from getting out of the page
## Works with almost everything except urls and strings.
## Last options hold the position of figures
opts_chunk$set(tidy.opts=list(width.cutoff=50),tidy=TRUE, fig.pos = "!H", out.extra = "")

# Set a seed
set.seed(6969)

# Choosing a color palette
cores <- brewer.pal(8, "Dark2")

# Set a working directory
setwd("C:\\Users\\aisha\\OneDrive\\Documentos\\Mestrado Tinbergen\\Year 2\\Block 05\\Market and Systemi
```

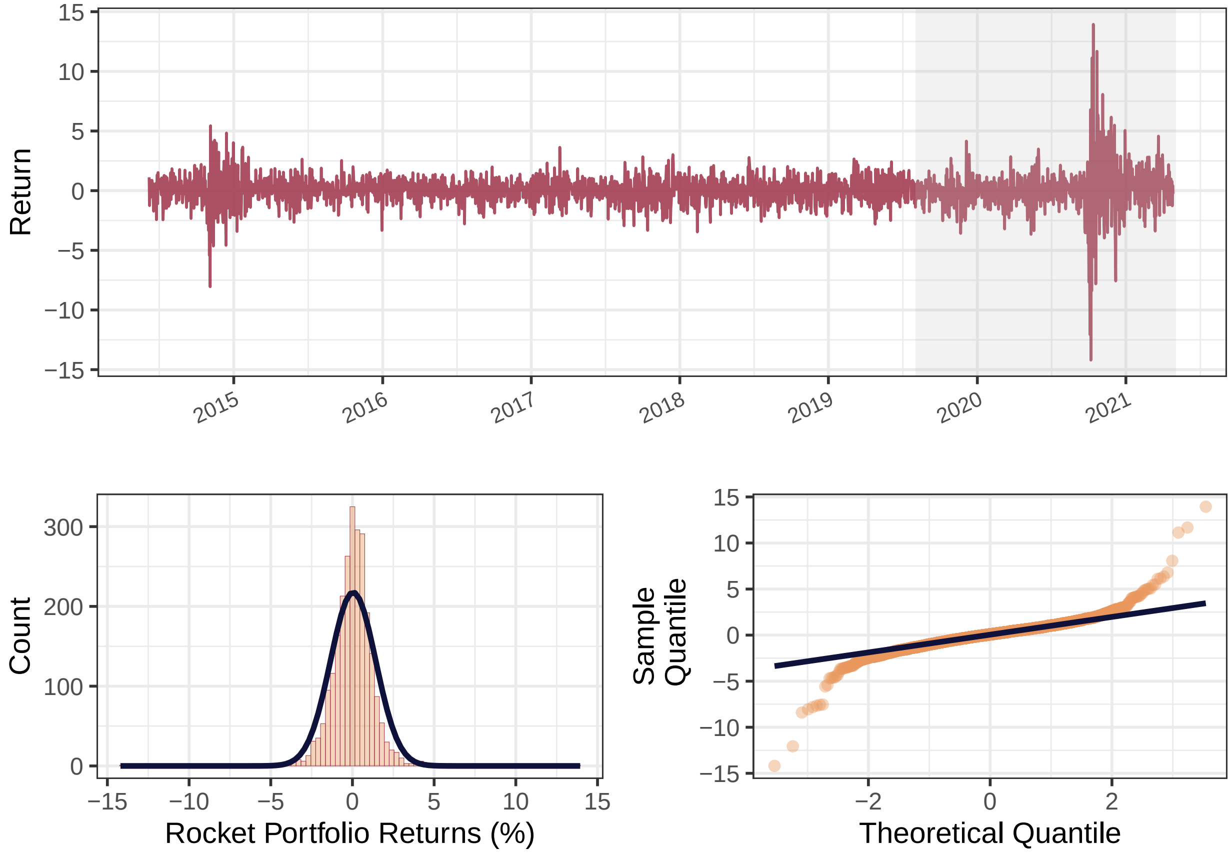
Data preparation

```
url <- "https://raw.githubusercontent.com/aishameriane/SystemicRisk/main/data.csv"
dfData01 <- read.csv2(url, sep = ",", dec = ".", header = TRUE)
dfData01 <- dfData01[, c("Market", "Schmidt")]

ending <- as.Date("2021-06-25")
# Had to cut one day to make vectors with the same
# size
starting <- as.Date(as.Date("2021-06-25") - 2516)
Date <- as.Date(starting:ending, origin = starting)


Date <- seq(starting, ending, by = "1 days")
dfData01 <- cbind(Date, dfData01)
```

Exploring data for heavy tails

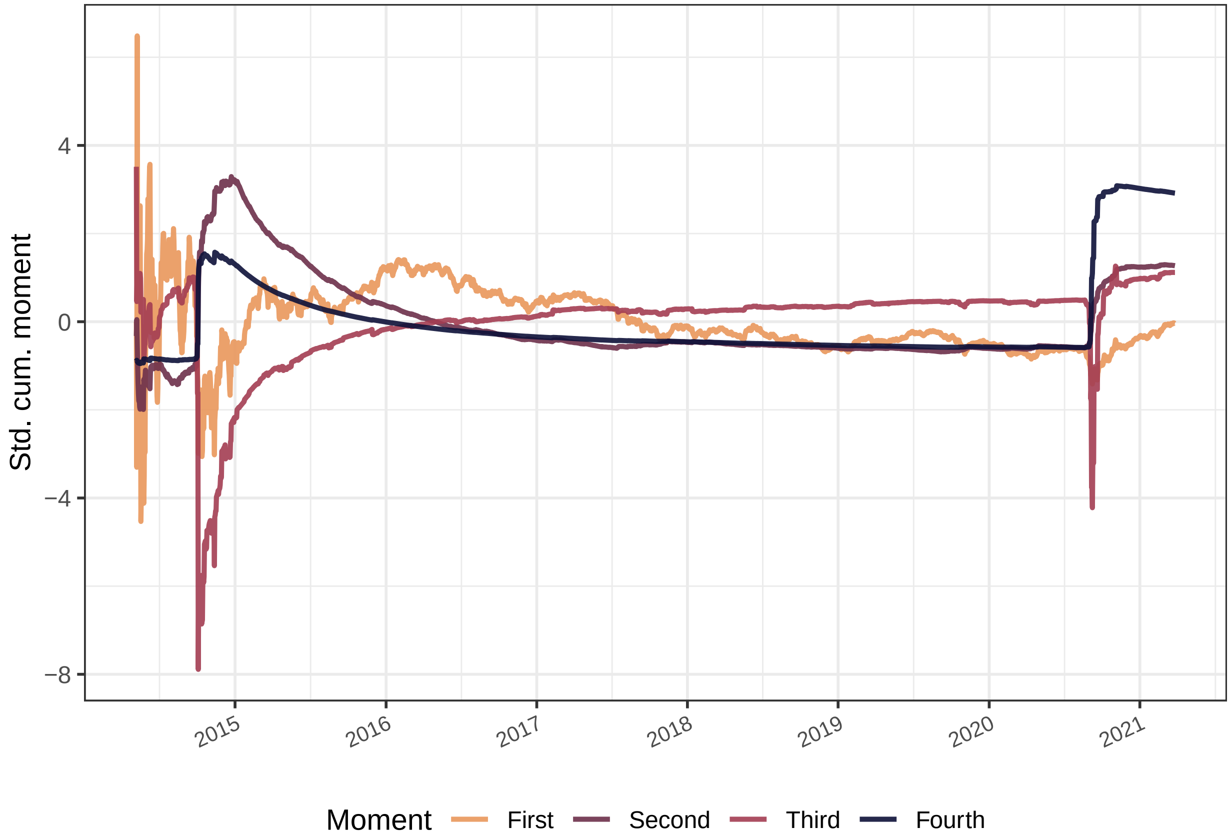
```
# Data from recessions available at:
# https://eabcn.org/dc/chronology-euro-area-business-cycles
recessions_1 <- c("2019Q4")

recessions_2 <- c("2021Q3")

recessions.trim <- data.frame(1, 1, 1, as.Date(as.yearqtr(recessions_1)),
    as.Date(as.yearqtr(recessions_2)))
names(recessions.trim) <- c("DATE", "Variable", "Value",
    "Peak", "Trough")


dfData02 <- melt(dfData01, id.vars = "Date")
names(dfData02) <- c("DATE", "Variable", "Value")

p0 <- ggplot(dfData02[which(dfData02$Variable == "Schmidt"),
    ]) + geom_line(aes(x = DATE, y = Value, color = Variable),
    alpha = 0.9) + labs(y = "Return", x = "", color = "Series") +
    scale_x_date(date_breaks = "12 months", labels = date_format("%Y")) +
    scale_colour_manual(values = c("#A43E53")) + theme_bw() +
    theme(axis.text.x = element_text(angle = 25, hjust = 1,
        size = 8), legend.position = "none", axis.title.x = element_blank(),
        plot.margin = unit(c(0.1, 0.2, 0.05, 0.2),
            "cm")) + geom_rect(data = recessions.trim,
    aes(xmin = Peak, xmax = Trough, ymin = -Inf, ymax = +Inf),
    fill = "grey", alpha = 0.2)

# Plot a histogram with a normal density
# overlapping Source:
# https://stackoverflow.com/questions/6967664/ggplot2-histogram-with-normal-curve
n <- nrow(dfData01)
mean <- mean(dfData01[, "Schmidt"])
sd <- sd(dfData01[, "Schmidt"])
binwidth <- 0.3
```

```
df <- data.frame(x = rnorm(n, mean, sd))
dfData03 <- cbind(dfData01, df)

p1 <- ggplot(dfData03, aes(x = Schmidt, mean = mean,
    sd = sd, binwidth = binwidth, n = n)) + geom_histogram(binwidth = binwidth,
    colour = "#A43E53", fill = "#E9975C", size = 0.1,
    alpha = 0.4) + stat_function(fun = function(x) dnorm(x,
    mean = mean, sd = sd) * n * binwidth, color = "#0E1139",
    size = 1) + labs(title = "", y = "Count", x = "Rocket Portfolio Returns (%)") +
    theme(axis.text.x = element_text(size = 6), axis.title.x = element_text(size = 6),
        axis.text.y = element_text(size = 6), axis.title.y = element_text(size = 6),
        plot.margin = unit(c(-1, -0.5, -2, -0.5), "cm")) +
    theme_bw()

# Make a QQplot
p2 <- ggplot(dfData01, aes(sample = Schmidt)) + stat_qq(colour = "#E9975C",
    fill = "#E9975C", size = 1.5, alpha = 0.4) + stat_qq_line(color = "#0E1139",
    size = 1) + labs(title = "", y = "Sample\n Quantile",
    x = "Theoretical Quantile") + theme(axis.text.x = element_text(size = 6),
    axis.title.x = element_text(size = 6), axis.text.y = element_text(size = 6),
    axis.title.y = element_text(size = 6), plot.margin = unit(c(-1,
        -0.5, -2, -0.5), "cm")) + theme_bw()

# png(file = 'Series_01.png', width = 450, height =
# 300)
grid.arrange(p0, grid.arrange(p1, p2, nrow = 1), nrow = 2)
```



```
# dev.off()

# Cumulative moment plot
dfMoment <- data.frame(matrix(NA, ncol = 5, nrow = nrow(dfData01)))
names(dfMoment) <- c("Date", "First", "Second", "Third",
    "Fourth")

# This will compute the cumulative moments
# (centered on zero) up to order k
for (k in 1:4) {
    for (i in 1:nrow(dfData01)) {
        dfMoment[i, k + 1] <- mean(dfData01$Schmidt[1:i]^k)
    }
}

dfMoment$Date <- dfData01$Date
dfMomentstd <- dfMoment

for (j in 2:5) {
    dfMomentstd[, j] <- (dfMoment[, j] - mean(dfMoment[,
        j]))/sd(dfMoment[, j])
}

dfMomentstd[1, 2] <- 0

dfData02 <- melt(dfMomentstd, id.vars = "Date")
```

```r
names(dfData02) <- c("DATE", "Variable", "Value")

p3 <- ggplot(dfData02) + geom_line(aes(x = DATE, y = Value,
    color = Variable), alpha = 0.9, size = 0.9) + labs(y = "Std. cum. moment",
    x = "", color = "Moment") + scale_x_date(date_breaks = "12 months",
    labels = date_format("%Y")) + scale_colour_manual(values = c("#E9975C",
    "#6D304B", "#A43E53", "#0E1139"), labels = c("First",
    "Second", "Third", "Fourth")) + theme_bw() + theme(axis.text.x = element_text(angle = 25,
    hjust = 1, size = 8), legend.position = "bottom",
    axis.title.x = element_blank(), plot.margin = unit(c(0.1,
        0.2, 0.05, 0.2), "cm"))

# png(file = 'Series_02.png', width = 450, height =
# 150)
p3
```



```r
# dev.off

sd(dfData01$Schmidt)
```

```
## [1] 1.383581
```

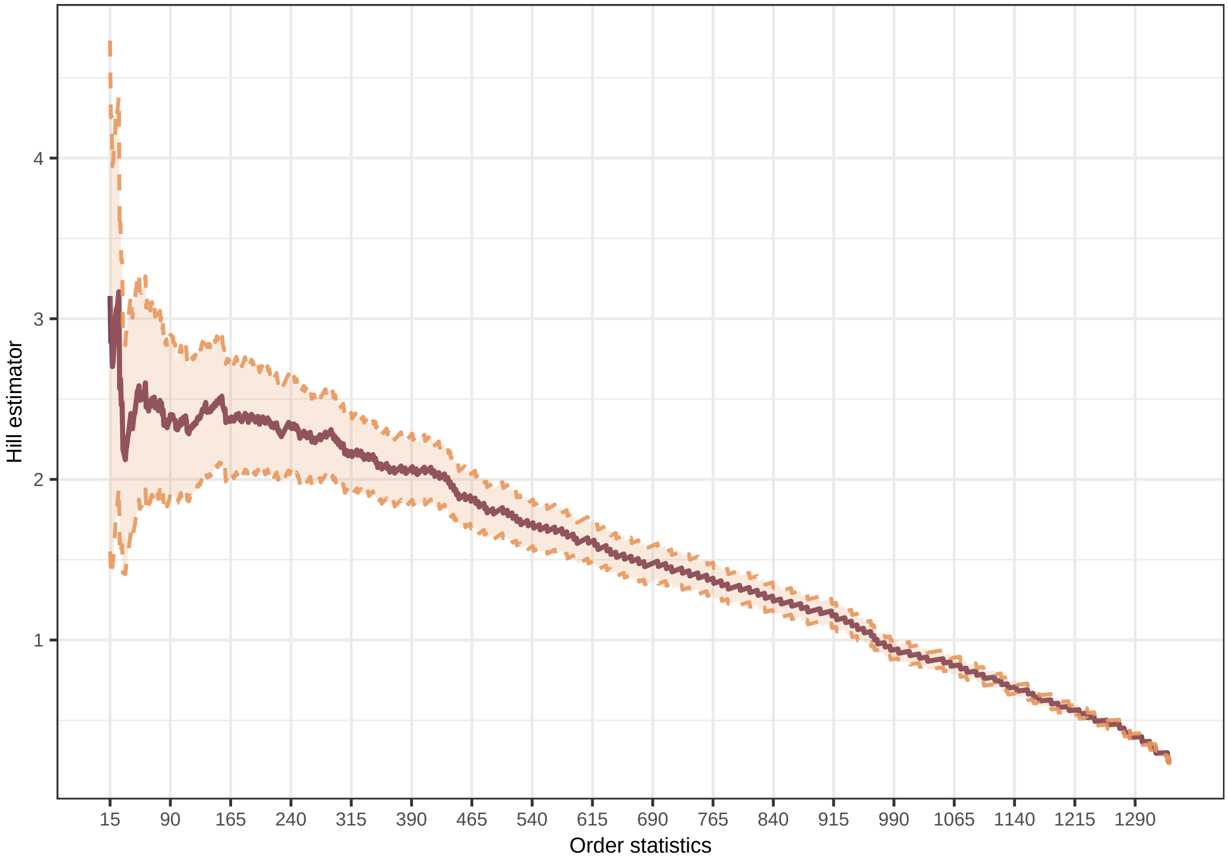
```r
mean(dfData01$Schmidt)
```

```
## [1] 0.04111641
```

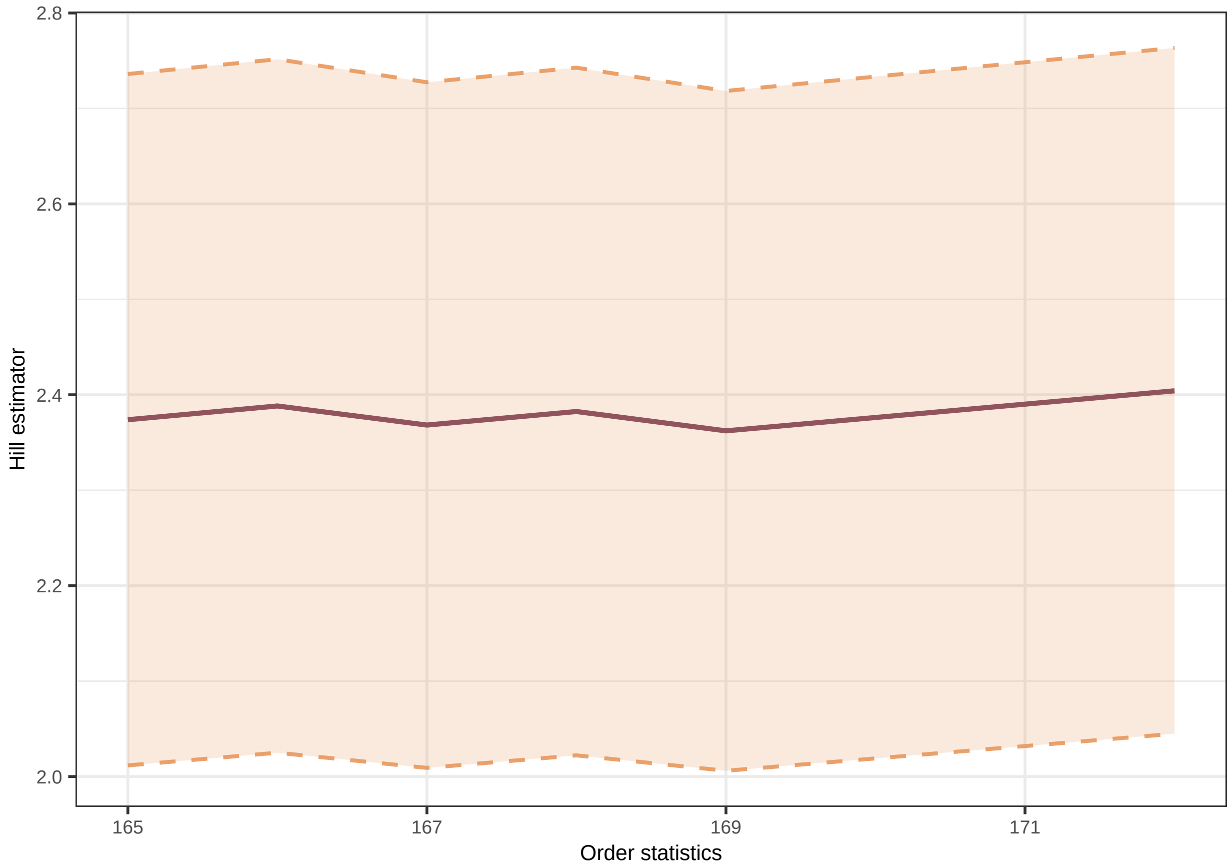
Question 1 - what is the maximum amount of asset (in mln USD) that can be invested in the risky portfolio?

- Regulator: with holding the invested portfolio for 100 days, the Value-at-Risk (VaR) at 99.9% level multiplied by a multiplier 3 has to be lower than the capital of the bank
- Capital of the bank 30 million

```r
# Computes the Hill estimator Reference:
# https://github.com/bpfaff/evir/blob/master/R/eda.R
# had to adapt so it will spit the confidence
# interval
fhill <- function(data, option = c("alpha", "xi", "quantile"),
    start = 15, end = NA, reverse = FALSE, p = NA,
    ci = 0.95, auto.scale = TRUE, labels = TRUE, plot = FALSE) {
    data <- as.numeric(data)
    ordered <- rev(sort(data))
    ordered <- ordered[ordered > 0]
    n <- length(ordered)
    option <- match.arg(option)
    if ((option == "quantile") && (is.na(p)))
        stop("Input a value for the probability p")
    if ((option == "quantile") && (p < 1 - start/n)) {
        cat("Graph may look strange !! \n\n")
        cat(paste("Suggestion 1: Increase `p' above",
            format(signif(1 - start/n, 5)), "\n"))
        cat(paste("Suggestion 2: Increase `start' above ",
            ceiling(length(data) * (1 - p)), "\n"))
    }

    k <- 1:n
    loggs <- logb(ordered)
    avesumlog <- cumsum(loggs)/(1:n)
    xihat <- c(NA, (avesumlog - loggs)[2:n])
    alphahat <- 1/xihat
    y <- switch(option, alpha = alphahat, xi = xihat,
        quantile = ordered * ((n * (1 - p))/k)^(-1/alphahat))
    ses <- y/sqrt(k)
    if (is.na(end))
        end <- n
    x <- trunc(seq(from = min(end, length(data)), to = start))
    y <- y[x]
    ylabel <- option
    yrange <- range(y)
    if (ci && (option != "quantile")) {
        qq <- qnorm(1 - (1 - ci)/2)
        u <- y + ses[x] * qq
        l <- y - ses[x] * qq
        ylabel <- paste(ylabel, " (CI, p =", ci, ")",
            sep = "")
        yrange <- range(u, l)
    }
    if (option == "quantile")
        ylabel <- paste("Quantile, p =", p)
    index <- x
    if (reverse)
        index <- -x
    if (plot == TRUE) {
        if (auto.scale)
            plot(index, y, ylim = yrange, type = "l",
                xlab = "", ylab = "", axes = FALSE,
                ...) else plot(index, y, type = "l", xlab = "",
            ylab = "", axes = FALSE, ...)
        axis(1, at = index, labels = paste(x), tick = FALSE)
        axis(2)
        threshold <- findthresh(data, x)
        axis(3, at = index, labels = paste(format(signif(threshold,
            3))), tick = FALSE)
        box()
        if (ci && (option != "quantile")) {
            lines(index, u, lty = 2, col = 2)
            lines(index, l, lty = 2, col = 2)
        }
        if (labels) {
            title(xlab = "Order Statistics", ylab = ylabel)
            mtext("Threshold", side = 3, line = 3)
        }
    }
    invisible(list(x = index, y = y, u = u, l = l))
}

lHill <- fhill(dfData01$Schmidt, option = c("alpha",
    "xi", "quantile"), start = 15, end = NA, reverse = FALSE,
    p = NA, ci = 0.95, auto.scale = TRUE, labels = TRUE,
    plot = FALSE)

dfHill <- data.frame(lHill$x, lHill$y, lHill$l, lHill$u)
names(dfHill) <- c("statistics", "Estimate", "Lower",
    "Upper")

p4 <- ggplot(dfHill, aes(x = statistics, y = Estimate)) +
    geom_line(aes(x = statistics, y = Estimate), colour = "#6D304B",
        alpha = 0.9, size = 0.9) + geom_line(aes(x = statistics,
    y = Lower), colour = "#E9975C", alpha = 0.9, size = 0.7,
    linetype = "dashed") + geom_line(aes(x = statistics,
    y = Upper), colour = "#E9975C", alpha = 0.9, size = 0.7,
    linetype = "dashed") + geom_ribbon(aes(ymin = Lower,
    ymax = Upper), fill = "#E9975C", alpha = 0.2) +
    scale_x_continuous(breaks = seq(from = min(dfHill$statistics),
        to = max(dfHill$statistics), by = 75)) + theme_bw() +
    labs(y = "Hill estimator", x = "Order statistics") +
    theme(axis.text.x = element_text(size = 7), axis.title.x = element_text(size = 8),
        axis.text.y = element_text(size = 7), axis.title.y = element_text(size = 8),
        plot.margin = unit(c(0.1, 0.2, 0.05, 0.2),
            "cm"), panel.grid.minor.x = element_blank())

# png(file = 'Series_03.png', width = 450, height =
# 150)
p4
```



```
# dev.off

lHill <- fhill(dfData01$Schmidt, option = c("alpha",
    "xi", "quantile"), start = 165, end = 172, reverse = FALSE,
    p = NA, ci = 0.95, auto.scale = TRUE, labels = TRUE,
    plot = FALSE)


dfHill <- data.frame(lHill$x, lHill$y, lHill$l, lHill$u)
names(dfHill) <- c("statistics", "Estimate", "Lower",
    "Upper")

p5 <- ggplot(dfHill, aes(x = statistics, y = Estimate)) +
    geom_line(aes(x = statistics, y = Estimate), colour = "#6D304B",
        alpha = 0.9, size = 0.9) + geom_line(aes(x = statistics,
    y = Lower), colour = "#E9975C", alpha = 0.9, size = 0.7,
    linetype = "dashed") + geom_line(aes(x = statistics,
    y = Upper), colour = "#E9975C", alpha = 0.9, size = 0.7,
    linetype = "dashed") + geom_ribbon(aes(ymin = Lower,
    ymax = Upper), fill = "#E9975C", alpha = 0.2) +
    scale_x_continuous(breaks = seq(from = min(dfHill$statistics),
        to = max(dfHill$statistics), by = 2)) + theme_bw() +
    labs(y = "Hill estimator", x = "Order statistics") +
    theme(axis.text.x = element_text(size = 7), axis.title.x = element_text(size = 8),
        axis.text.y = element_text(size = 7), axis.title.y = element_text(size = 8),
        plot.margin = unit(c(0.1, 0.2, 0.05, 0.2),
            "cm"), panel.grid.minor.x = element_blank())


# png(file = 'Series_04.png', width = 150, height =
# 75)
p5
```



```
# dev.off
```

```
kstar = 170
alphastar <- dfHill[which(dfHill$statistics == kstar),
    "Estimate"]
print(alphastar)
```

```
## [1] 2.376191
```