Exploratory Data Analysis (EDA)

Statsomat.com

19 April 2021

Basic Information

Automatic statistics for the file:

File
Finance.csv

Your selection for the encoding: Auto
Your selection for the decimal character: Auto
Observations (rows with at least one non-missing value): 1662
Variables (columns with at least one non-missing value): 3
Variables considered continuous: 3

Variables considered continuous
return
size
volatility

Variables considered categorical: 0

Results for Numerical Variables

Descriptive Statistics

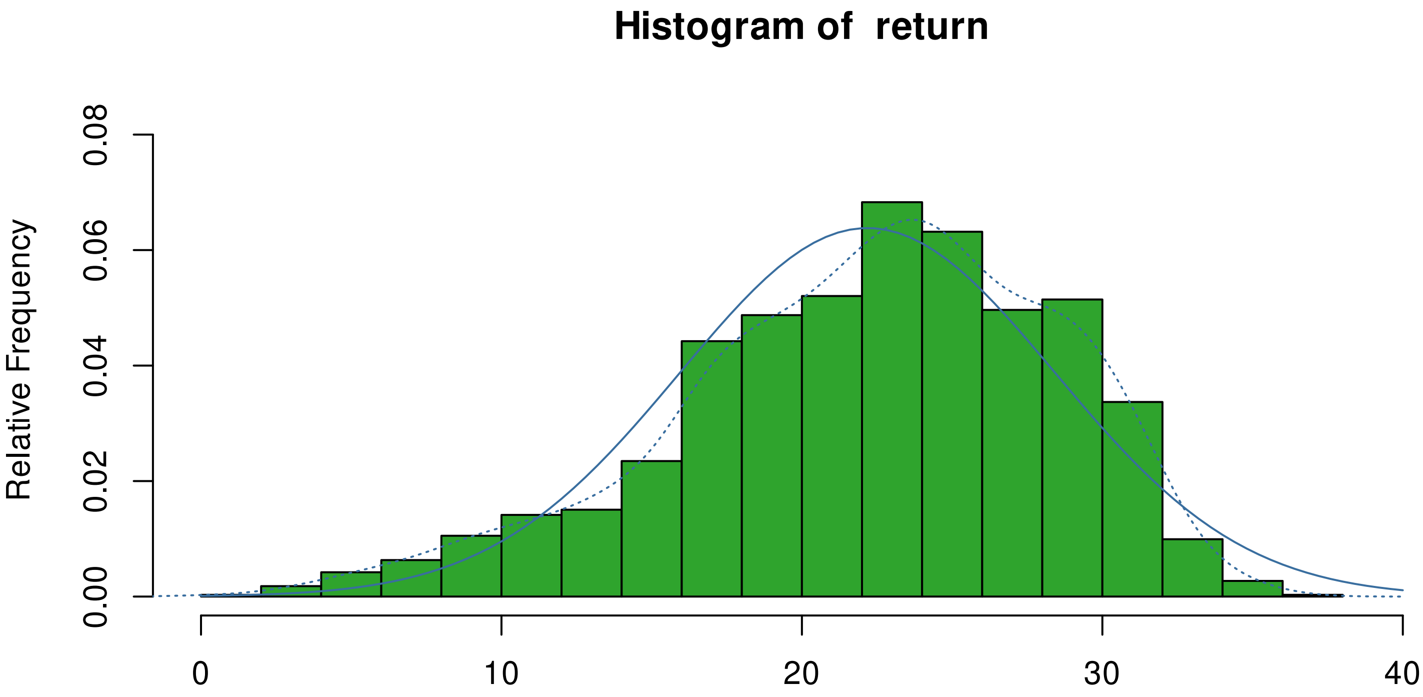
Variables are sorted alphabetically. Missings are omitted in the stats. CV only for positive variables.

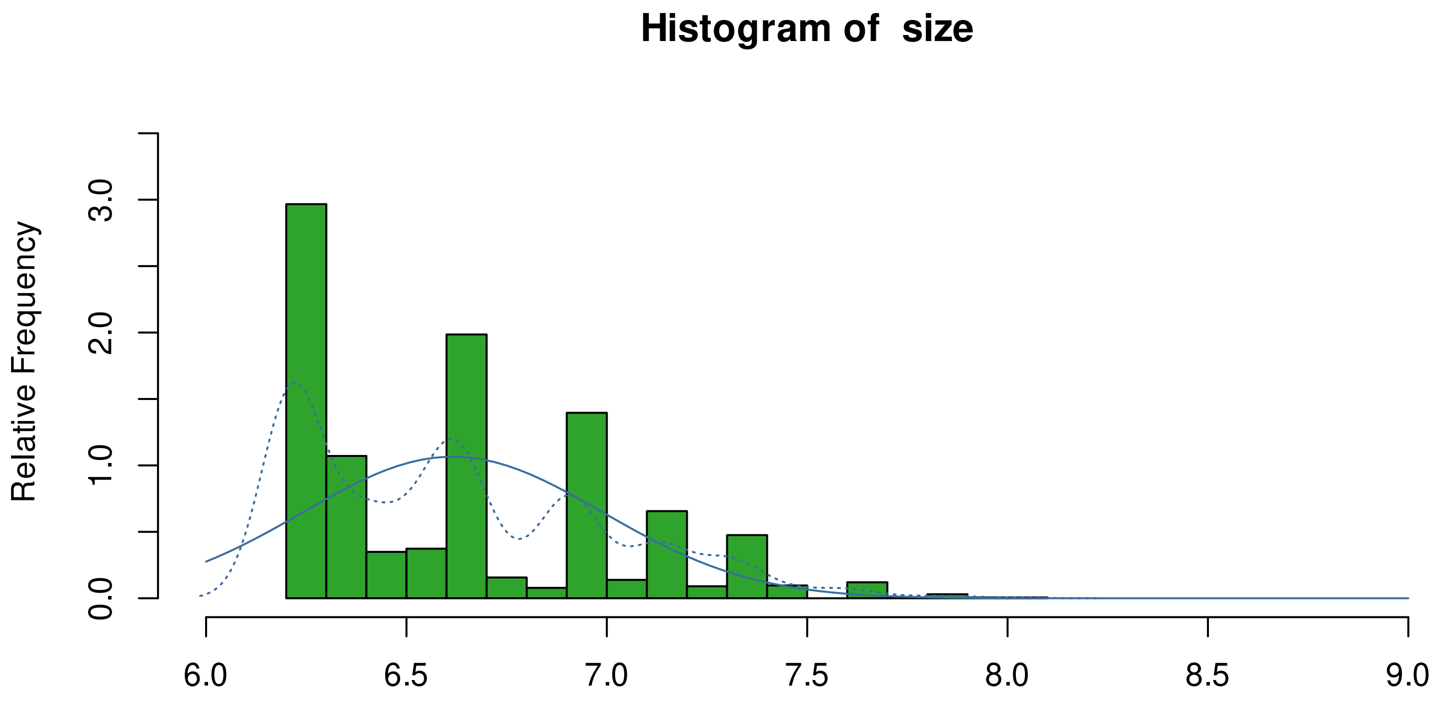
Variable	N Obs	N Missing	N Valid	% Complete	N Unique	Mean	SD	Median	MAD	MIN	MAX	Skewness	Kurtosis	CV
return	1662	0	1662	100	1662	22.18	6.25	22.82	6.38	0.31	36.14	-0.55	0.00	0.28
size	1662	0	1662	100	57	6.62	0.37	6.62	0.43	6.21	8.01	0.73	-0.21	0.06
volatility	1662	0	1662	100	1662	0.27	0.28	0.18	0.20	0.00	1.32	1.58	2.29	1.01

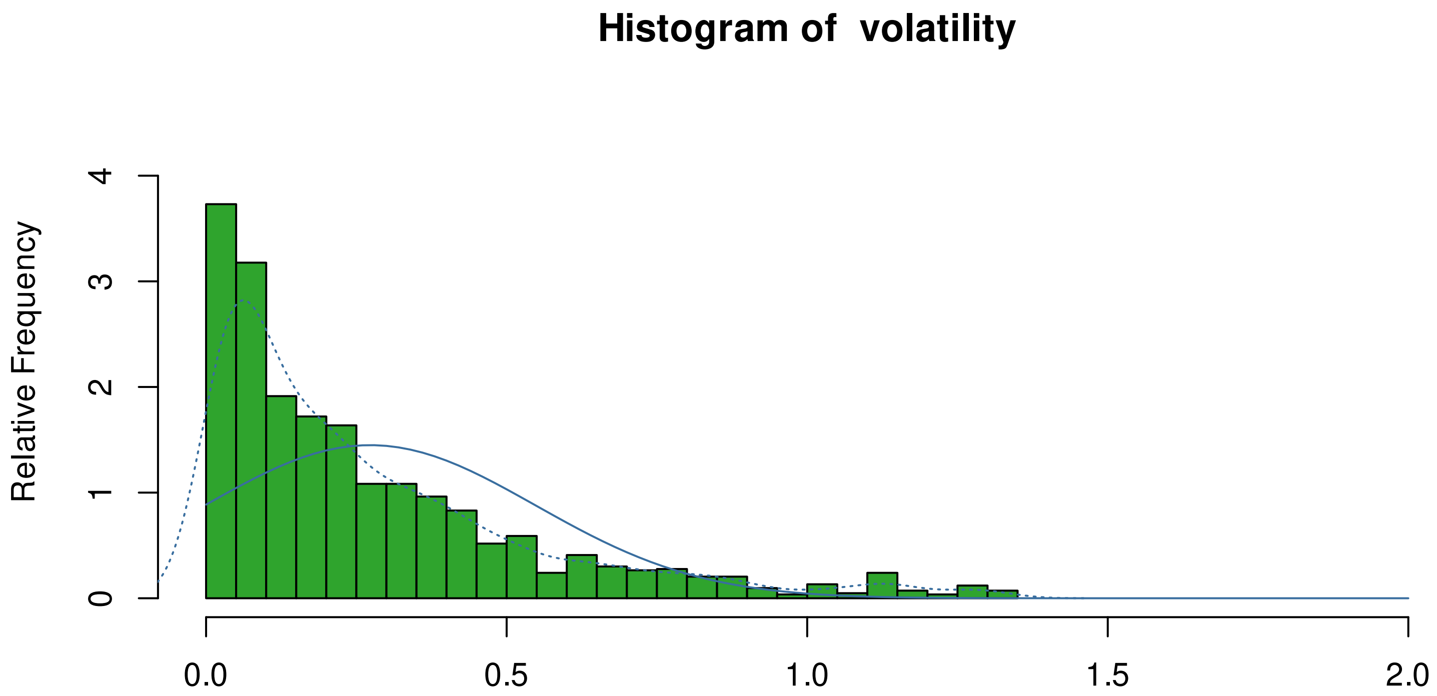
Graphics

Histograms

One Relative Frequency Histogram per page for each variable. Variables are sorted alphabetically. The blue line represents the normal density approximation. The blue dotted line represents a special kernel density approximation.

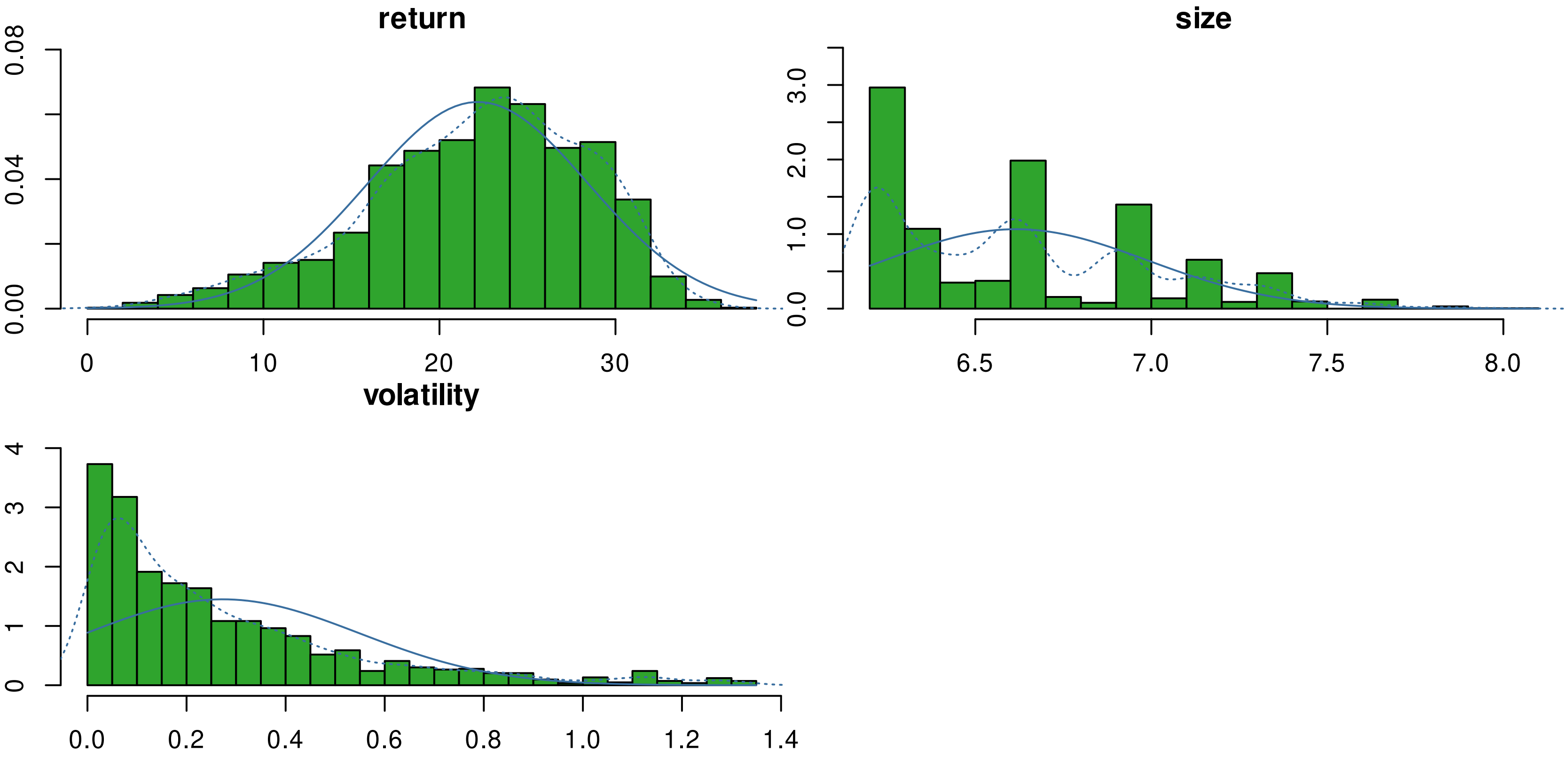



Histogram of size
Relative Frequency
0.0
1.0
2.0
3.0
6.0
6.5
7.0
7.5
8.0
8.5
9.0


Histogram of volatility
Relative Frequency
0
1
2
3
4
0.0
0.5
1.0
1.5
2.0

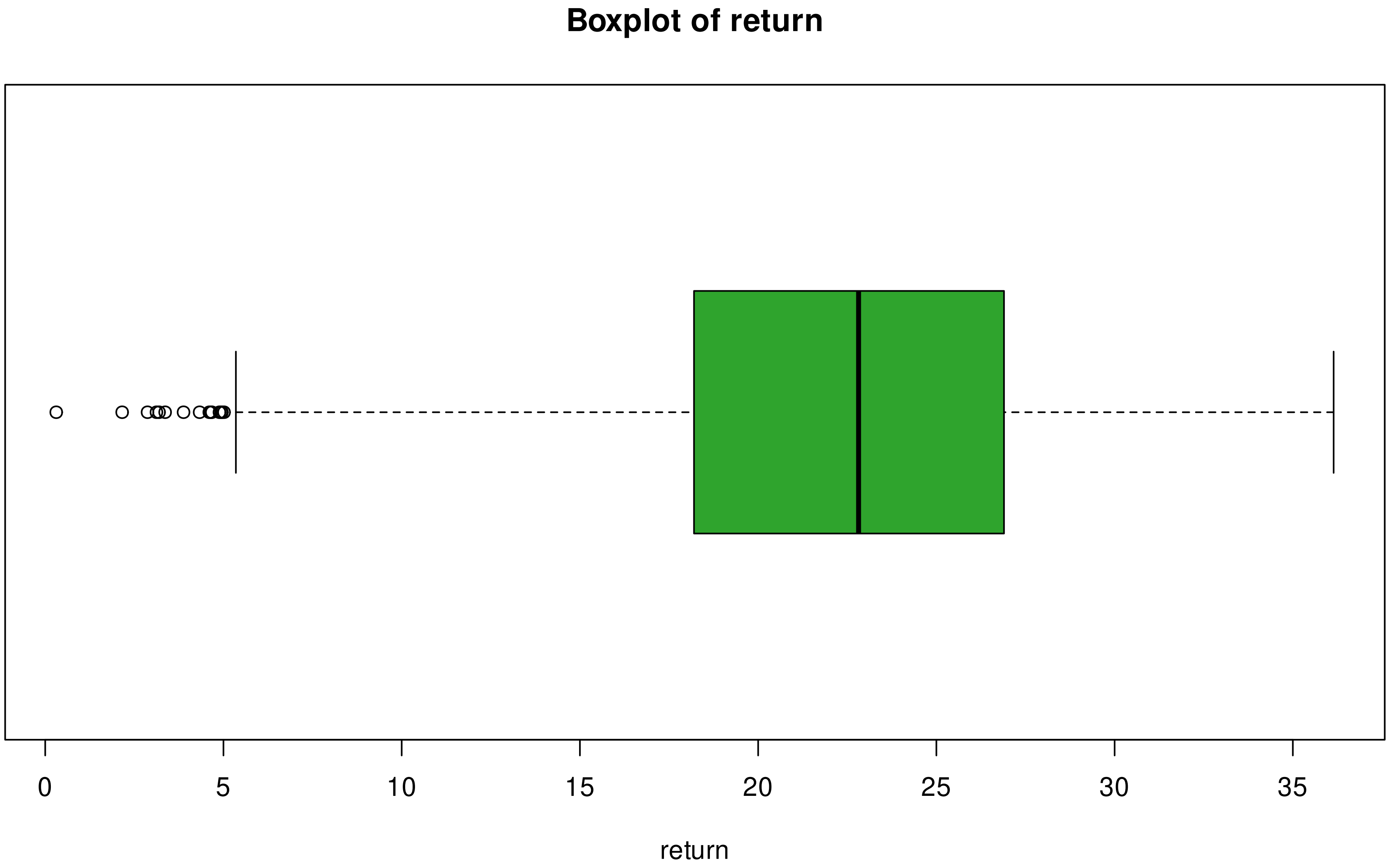
Histograms Summary

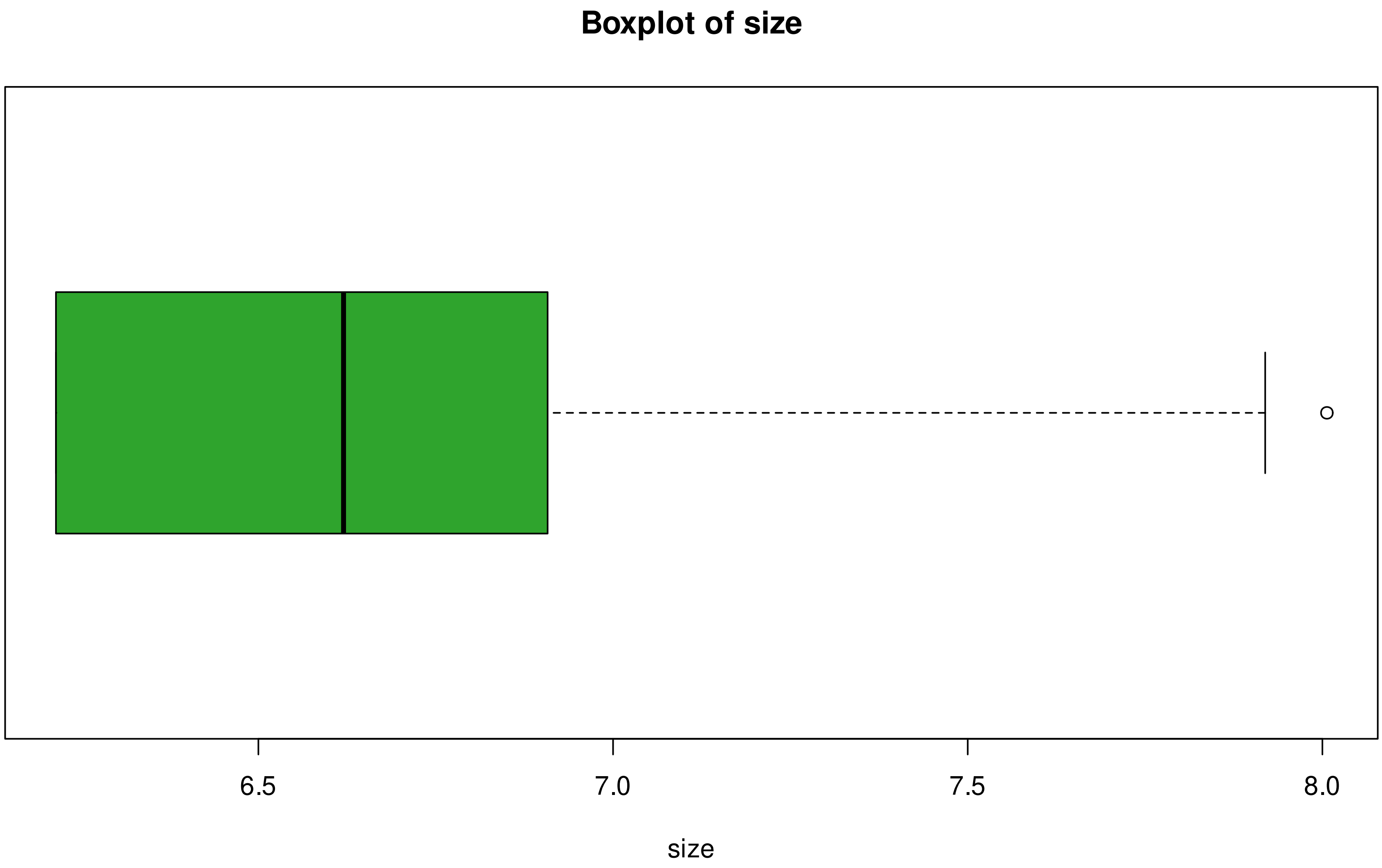
Multiple Relative Frequency Histogram in one figure. Variables are sorted alphabetically. The blue line represents the normal density approximation. The blue dotted line represents a special kernel density approximation.

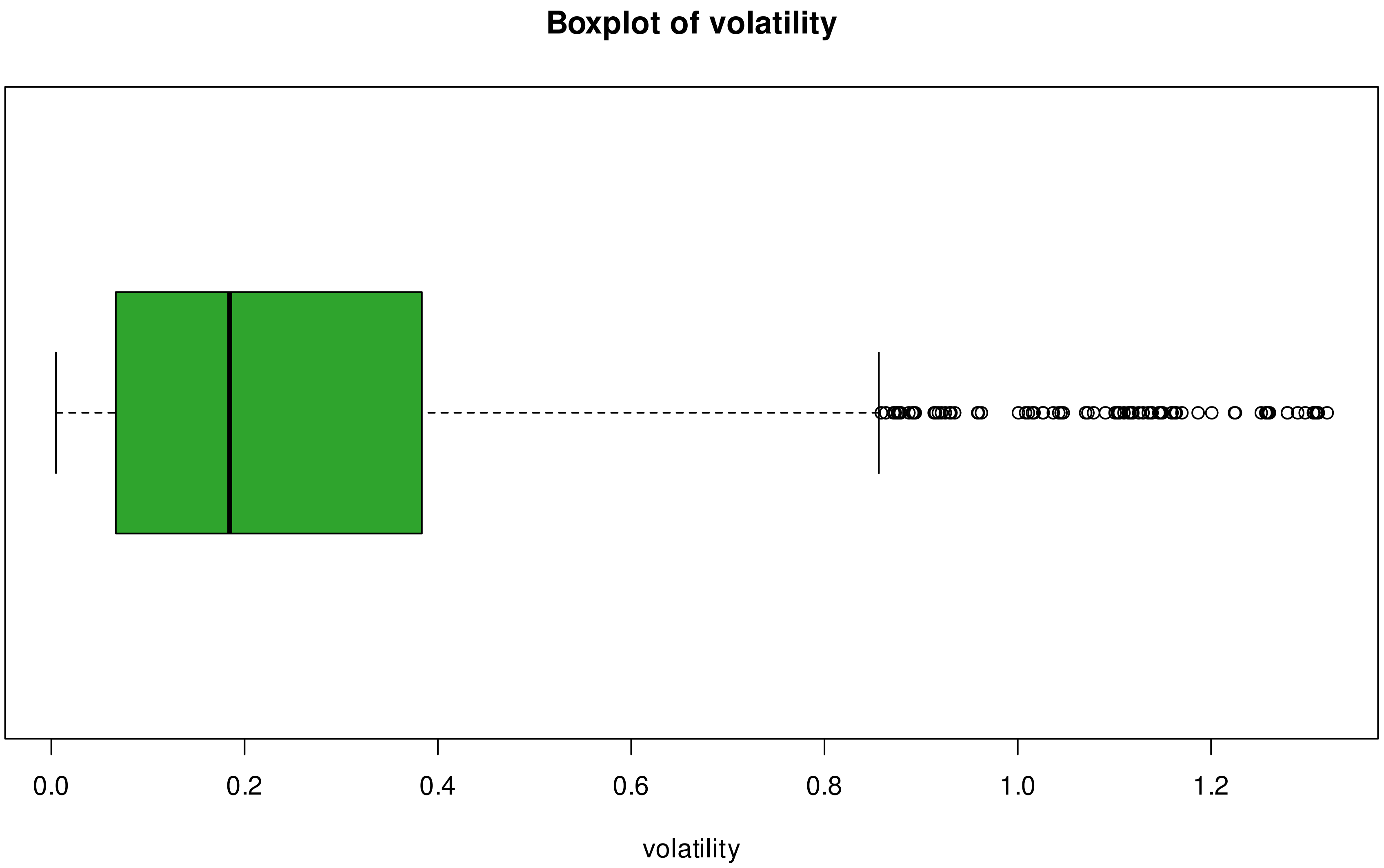


Box-Plots

One Box-Plot per page for each variable. Variables are sorted alphabetically.

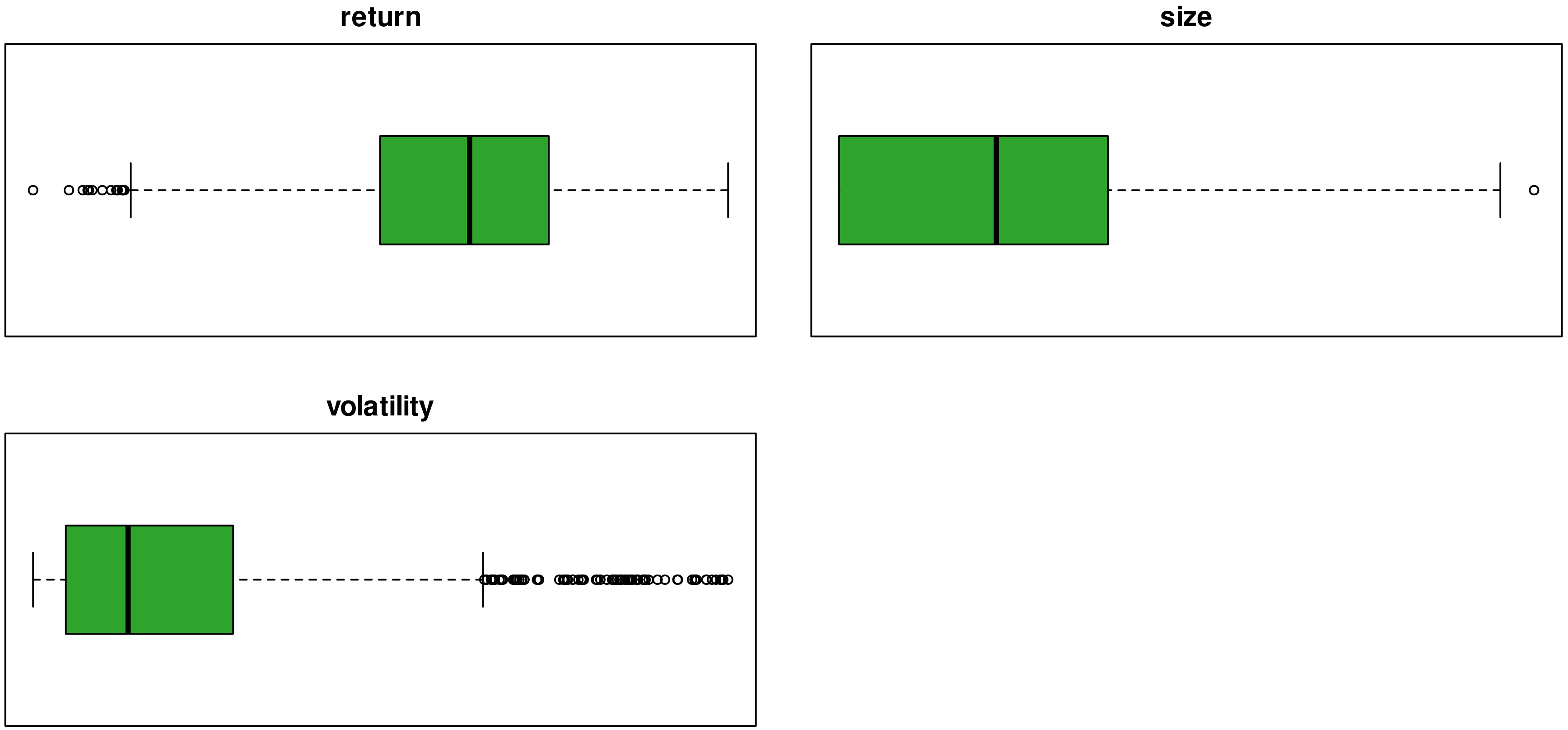



Boxplot of size
6.5
7.0
7.5
8.0
size


Boxplot of volatility
0.0
0.2
0.4
0.6
0.8
1.0
1.2
volatility

Box-Plots Summary

Multiple Box-Plots of variables in one figure. Variables are sorted alphabetically.



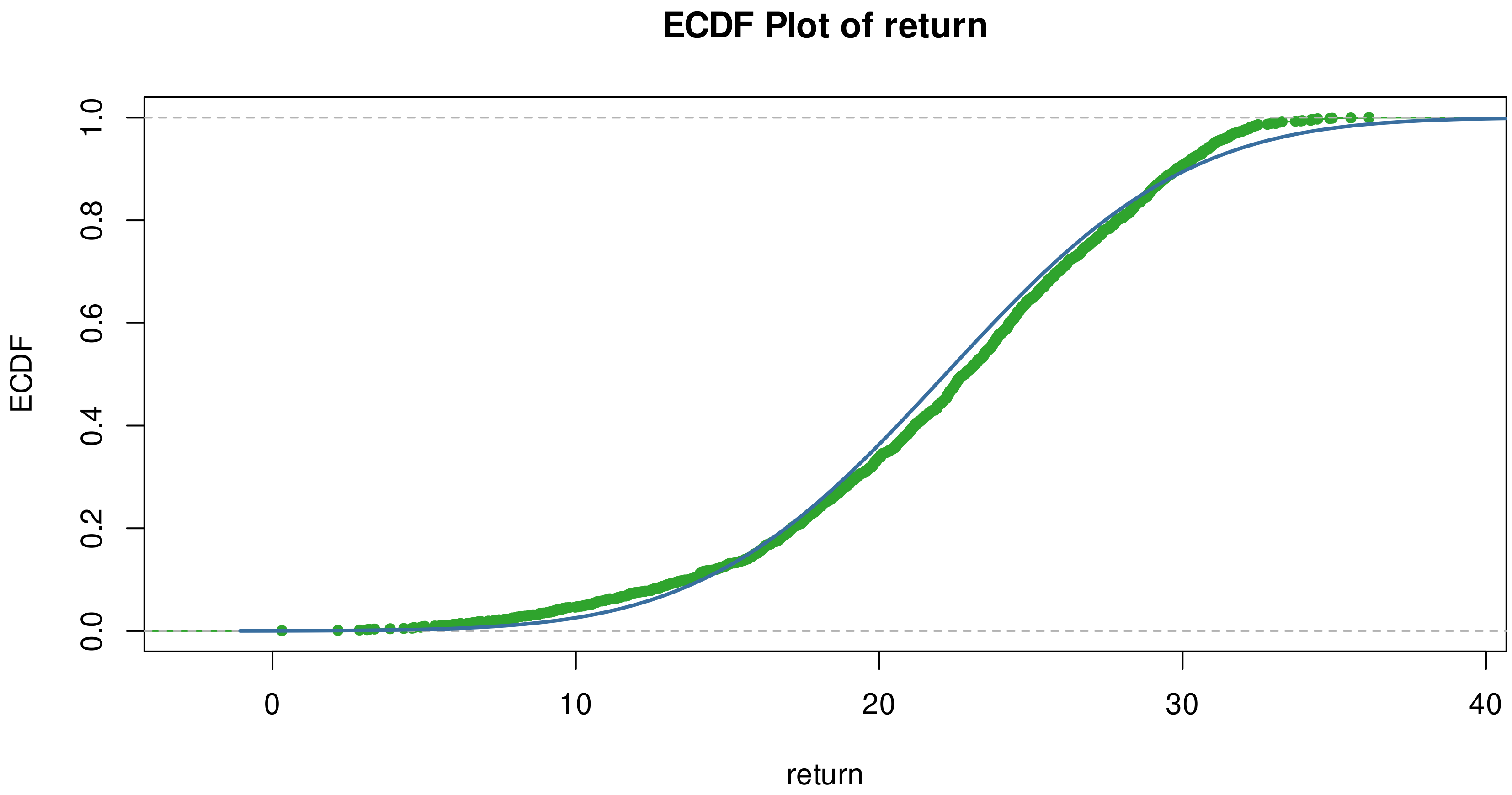
return

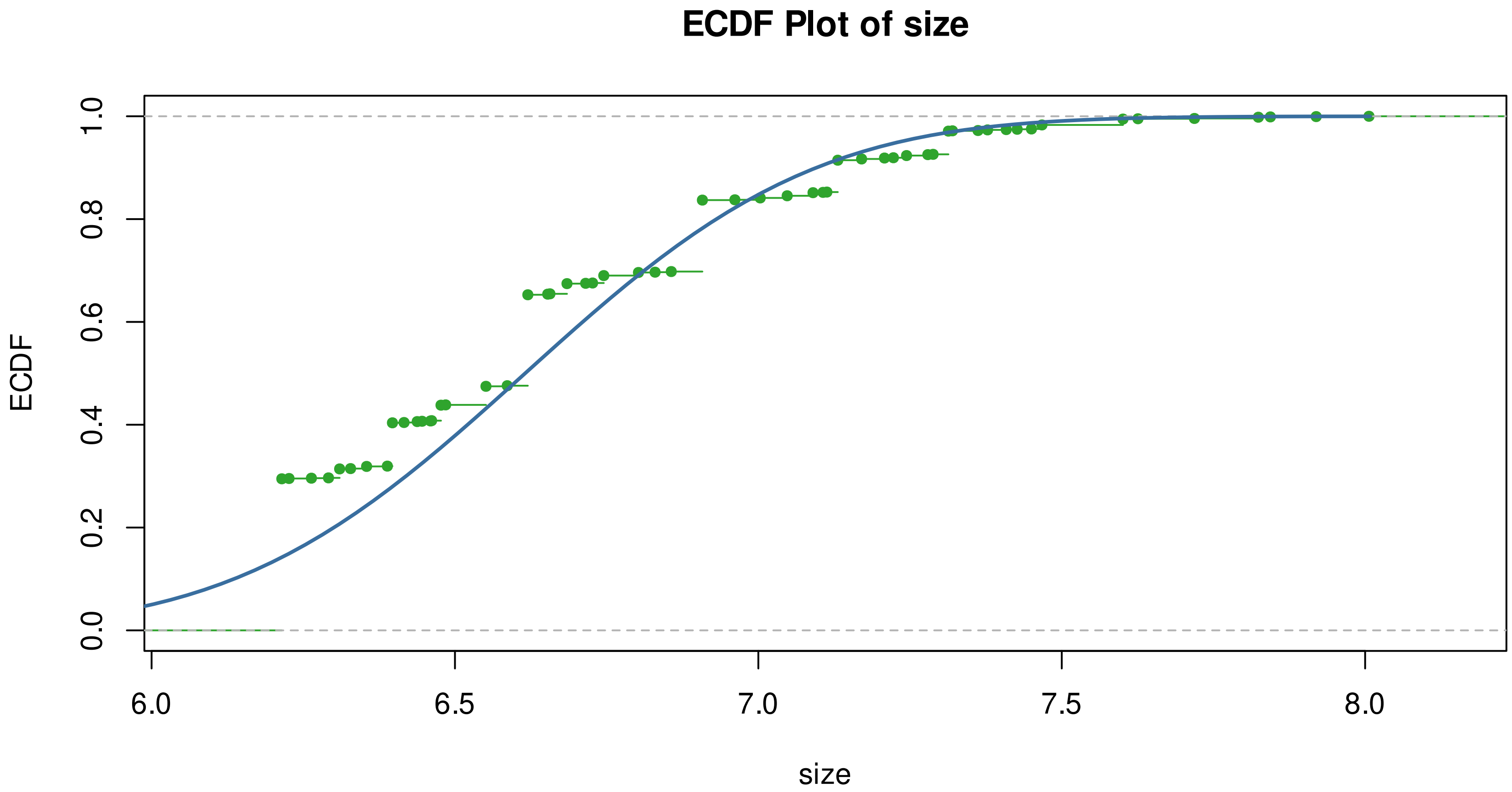
size

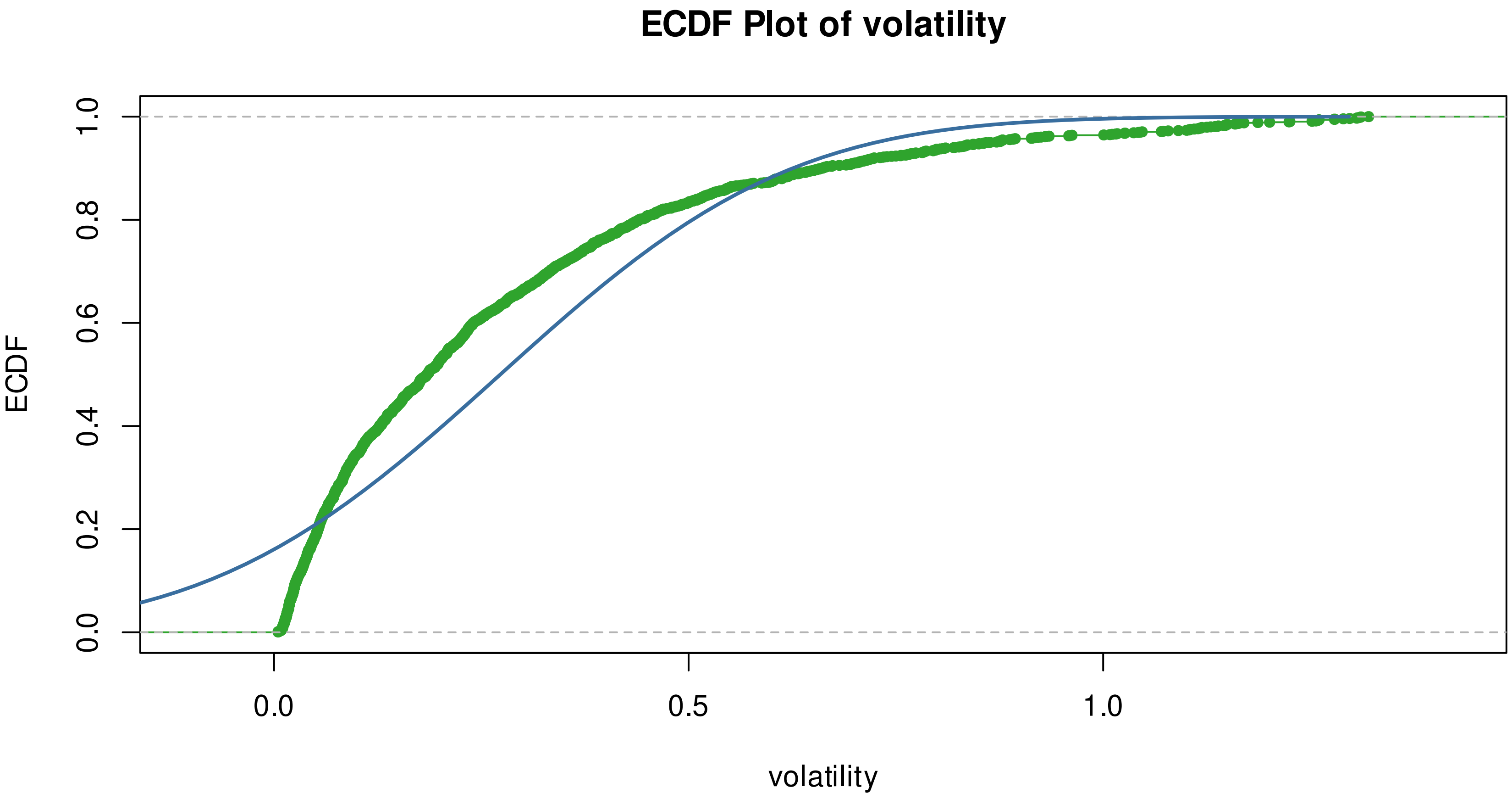
volatility

ECDF Plots

One ECDF (Empirical Cumulative Distribution Function) Plot per page for each variable. Variables are sorted alphabetically. The blue line represents the CDF of a normal distribution. If the variable is normally distributed, the blue line approximates well the ECDF.

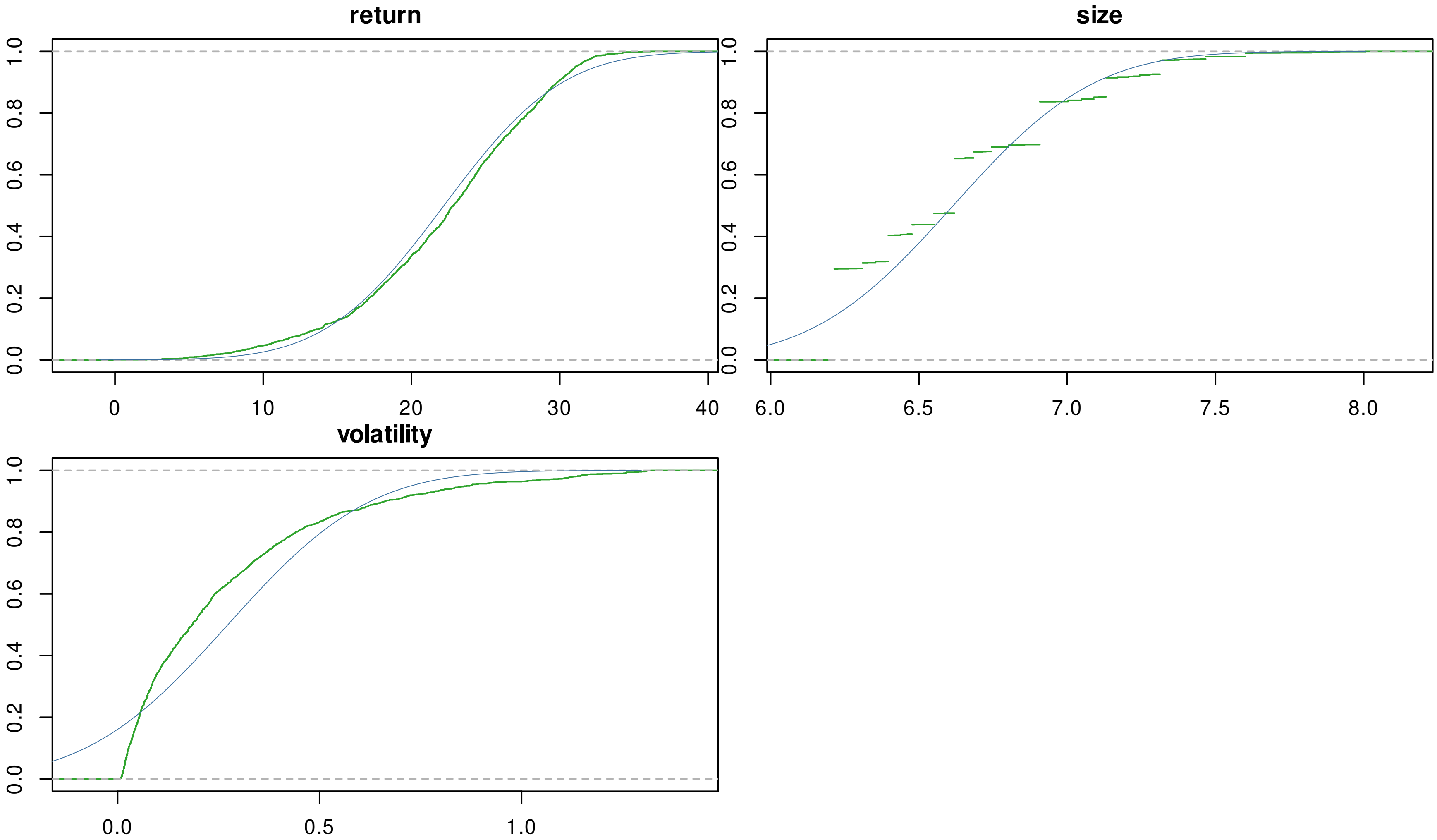



ECDF Plot of size
ECDF
size
0.0
0.2
0.4
0.6
0.8
1.0
6.0
6.5
7.0
7.5
8.0


ECDF Plot of volatility
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
volatility

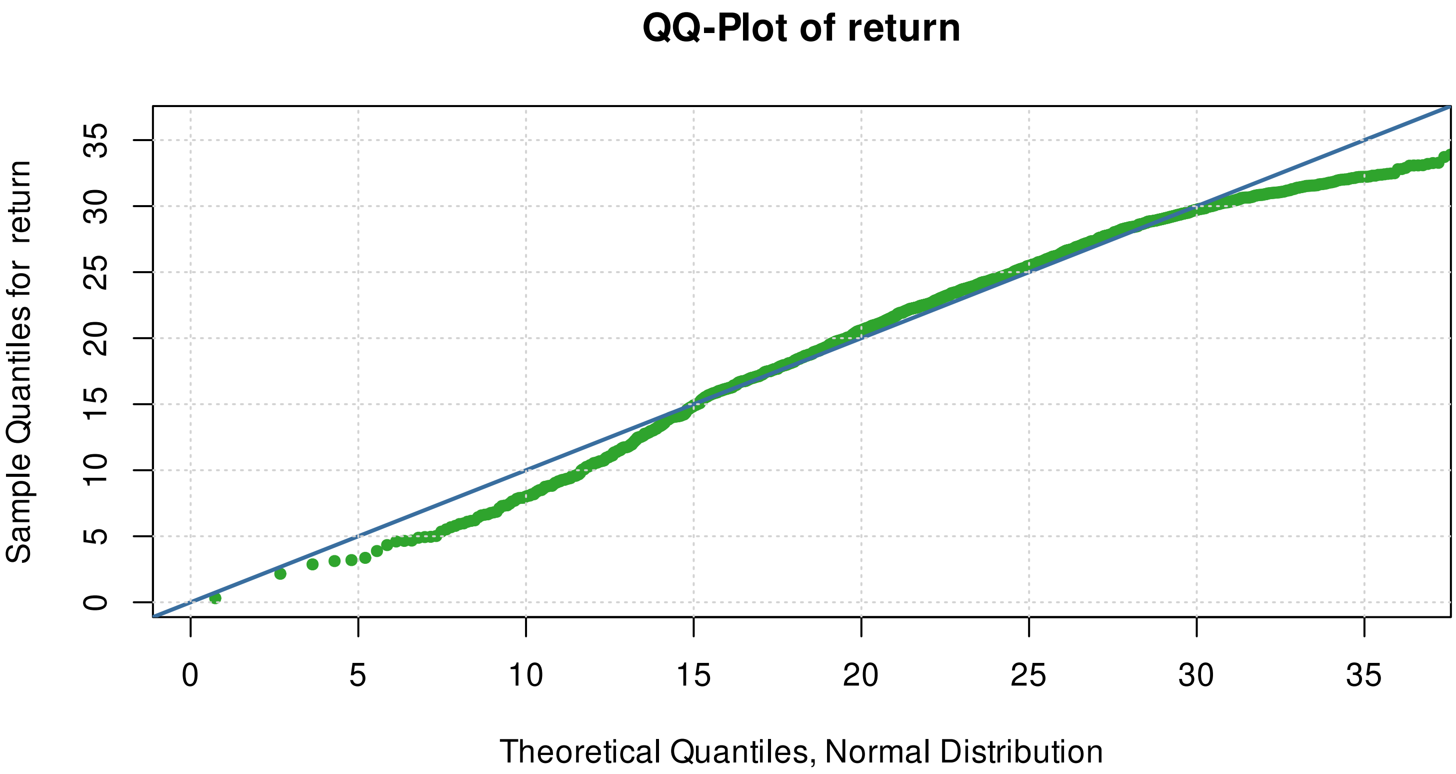
ECDF Plots Summary

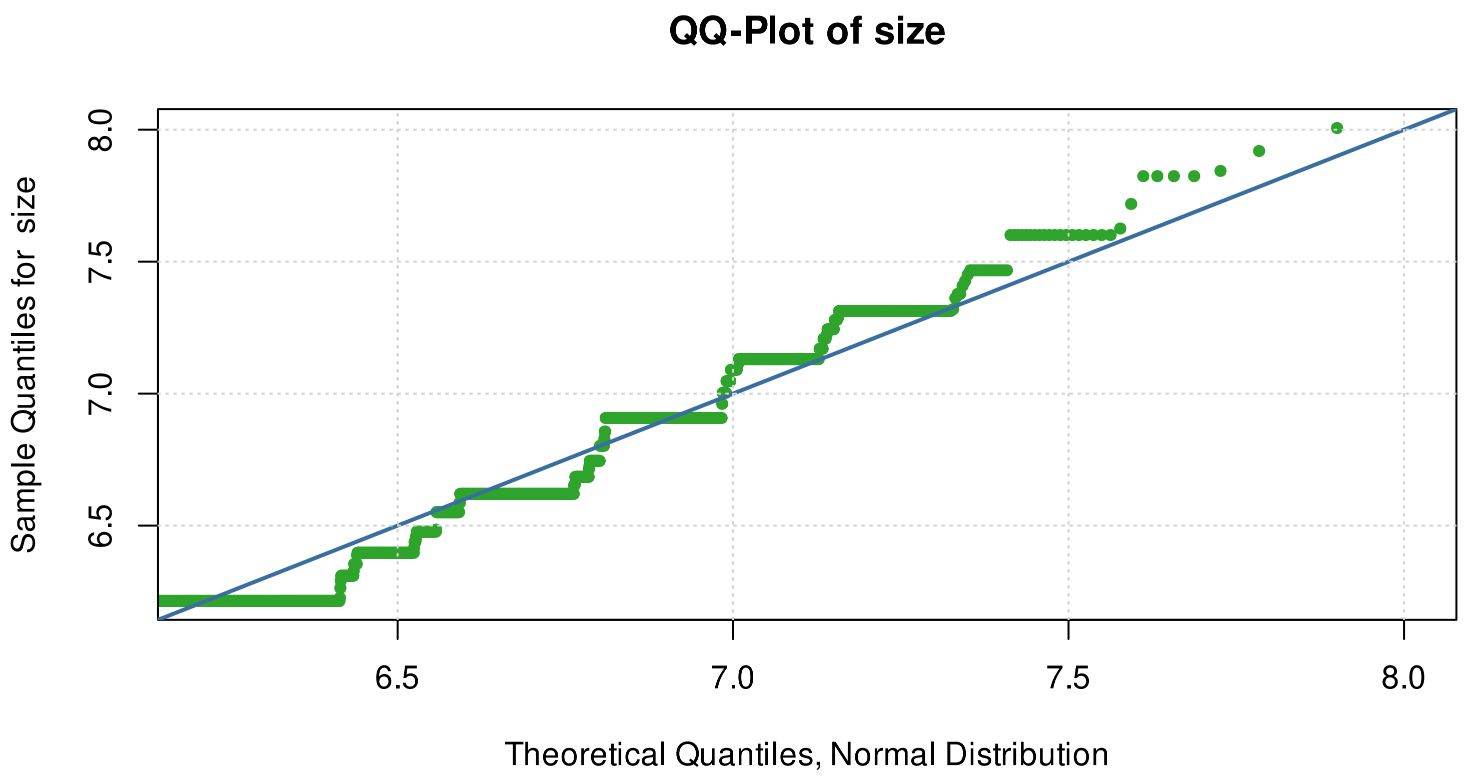
Multiple ECDF Plots of variables in one figure. Variables are sorted alphabetically.

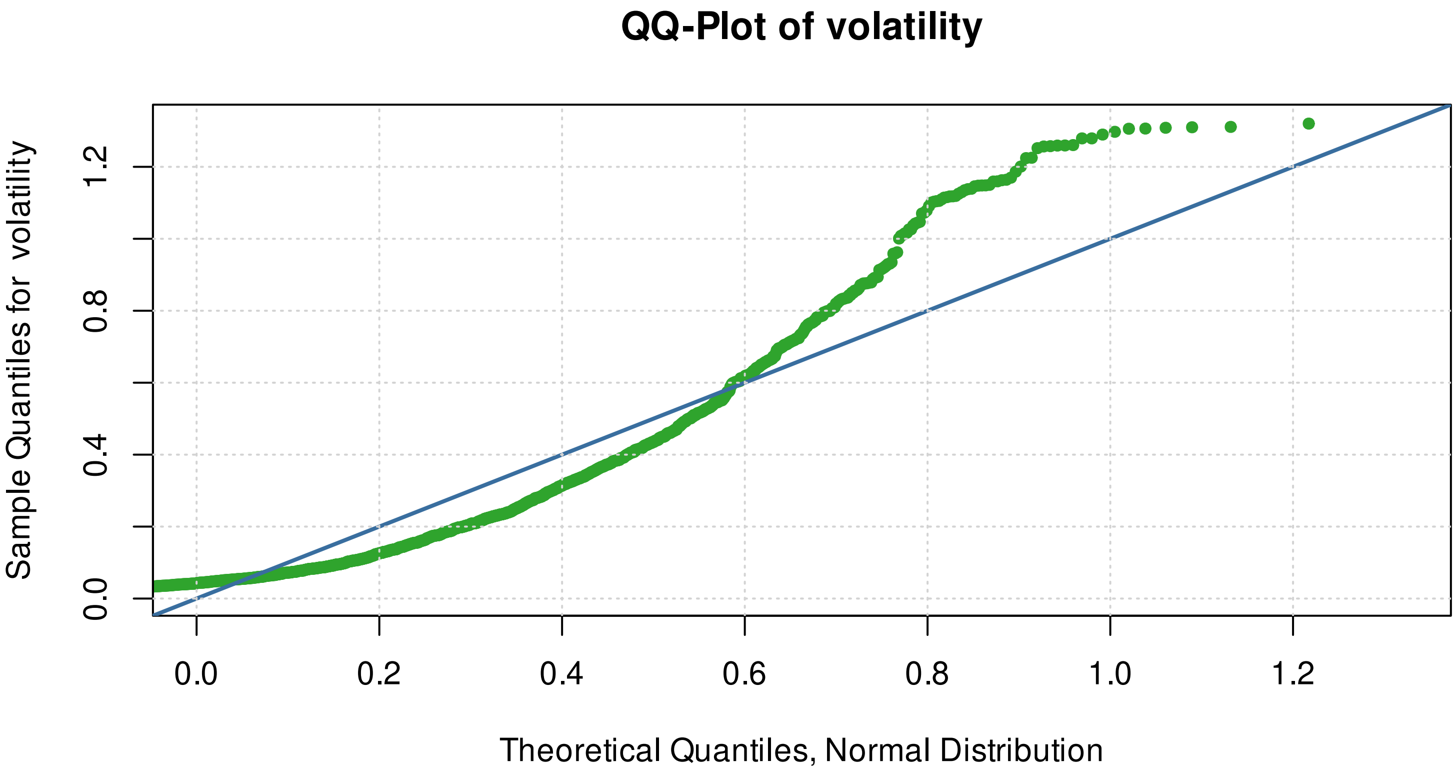


QQ-Plots

One QQ-Plot per page for each variable. Variables are sorted alphabetically.

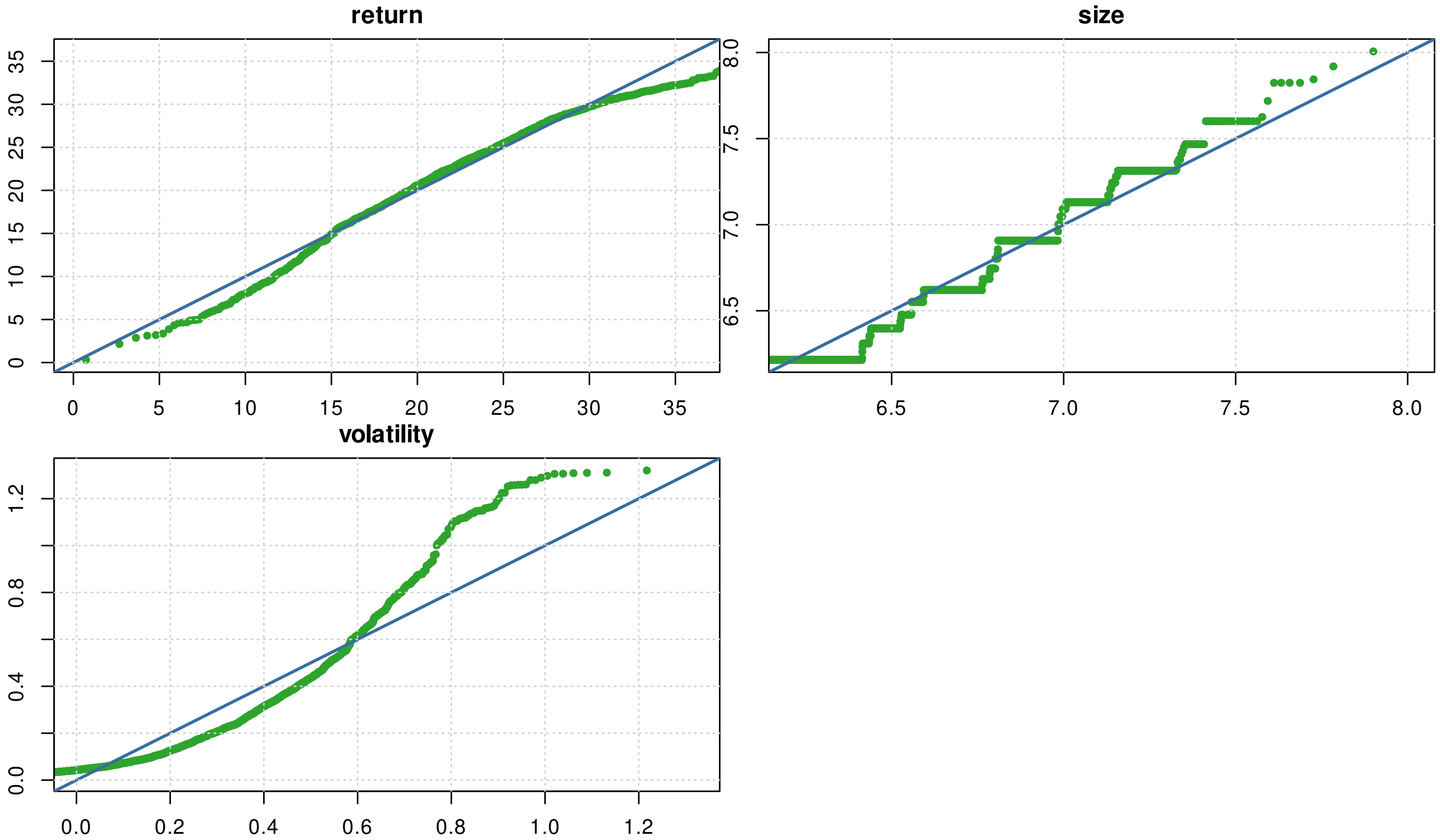



QQ-Plot of size
Sample Quantiles for size
6.5
7.0
7.5
8.0
Theoretical Quantiles, Normal Distribution
6.5
7.0
7.5
8.0


QQ-Plot of volatility
Sample Quantiles for volatility
0.0
0.4
0.8
1.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Theoretical Quantiles, Normal Distribution

QQ-Plots Summary

QQ-Plots of variables in one figure. Theoretical Quantiles of the Normal Distribution.



R Packages

To run the code you need to install following R packages:

R version: 4.0.3
Package car, version: 3.0.10
Package data.table, version: 1.12.8
Package ggplot2, version: 3.3.3
Package gridExtra, version: 2.3
Package Hmisc, version: 4.4.2
Package knitr, version: 1.31
Package PerformanceAnalytics, version: 2.0.4
Package psych, version: 2.0.12
Package reshape2, version: 1.4.4

R Code

Instructions
If not already available, please install R, RStudio and the requiered packages listed on the previous page. Copy the code below, paste it carefully in a new R Script within RStudio. For a seamless copy & paste process, open the PDF report in a browser. Change the path to your data in the line `filepath <- ...` For Windows users, avoid using paths containing spaces. Run the code. Explore the results (numerical results in the Console, plots in the Plots tab).

```
# Import required libraries
suppressPackageStartupMessages(library(data.table))
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(psych))
suppressPackageStartupMessages(library(Hmisc))
suppressPackageStartupMessages(library(reshape2))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(PerformanceAnalytics))
suppressPackageStartupMessages(library(gridExtra))
suppressPackageStartupMessages(library(car))


# Make a copy of current graphical settings
opar <- par(no.readonly = TRUE)

# Define the path to your data (please remark the forward slash)
filepath <-"C:/Finance.csv"

# Upload the data
df <- fread(filepath, header ="auto", sep ="auto", dec =",", encoding ="unknown", data.table = FALSE, na.strings = "")

# Convert characters to UTF-8 encoding
## Depending on your local R settings
## you could try to ignore and skip the next 4 lines
colnames(df) <- iconv(colnames(df),"ASCII","UTF-8")
col_names <- sapply(df, is.character)
df[ ,col_names] <- sapply(df[, col_names], function(col) iconv(col, "ASCII", "UTF-8"))

# Column names of selected continuous variables
colnames_continuous = c(1,2,3)

# Data frame of the continuous variables
df_num <- df[ ,colnames_continuous, drop=FALSE]

# Continuous variables
## Descriptive statistics


### Take over summary from psych package and add new stats
stats_new <- psych::describe(df_num)
```

```
### Drop some stats which we do not need
stats_new <- as.data.frame(stats_new)
stats_new <- stats_new[c(-1,-6,-10,-13)]

### Add new stats
stats_new$Variable <- colnames(df_num)
stats_new$ntotal <- nrow(df_num)

### Missings
stats_new$miss <- sapply(df_num, function(col) sum(is.na(col)))

### Complete rate
stats_new$complete <- sapply(df_num, function(col) (1-(sum(is.na(col)) / nrow(df_num)))*100)

### N Unique
stats_new$N_Unique <- sapply(df_num, function(col) length(unique(na.omit(col))))

### CV
stats_new$CV <- sapply(df_num, function(col) {
  ifelse(any(col <= 0, na.rm=TRUE), "-", round((sd(col, na.rm=TRUE) / mean(col, na.rm=TRUE)),2))
  })

### Reorder columns
stats_new <- stats_new[,c(10,11,12,1,13,14,2:9,15)]

### Column names
colnames(stats_new) <- c("Variable", "N Obs", "N Missing", "N Valid", "% Complete", "N Unique", "Mean",
                         "SD", "Median", "MAD", "MIN", "MAX", "Skewness", "Kurtosis", "CV")

### Order by variable name
stats_new <- stats_new[order(stats_new$Variable),]

### Output
knitr::kable(stats_new, digits=2, row.names = FALSE, format="simple")
```

```
# Continuous variables
## Descriptive graphics: Histograms One Per Page

### Order by variable name
df_num_order <- df_num[,order(colnames(df_num)),drop=FALSE]

### Function to plot histogram for each variable
```

```
single_hist <- function(x, main = "Histogram",
                    ylab="Relative Frequency", xlab=NULL, freq=FALSE, bcol="#2fa42d",
                    dcol=c("#396e9f","#396e9f"), dlty=c("dotted", "solid"),
                    breaks=21) {

  h <-  hist(x, plot=FALSE, breaks=breaks)
  m <- mean(x, na.rm=TRUE)
  s <- sd(x, na.rm=TRUE)
  d <- density(x, na.rm=TRUE)

  # Set nice x and y axis limits
  xlims <- pretty(c(floor(h$breaks[1]),ceiling(last(h$breaks))))
  ymax <- max(h$density)
  dmax <- max(d$y)
  ymax <- max(ymax,dmax)

  # Plots
  plot(h, freq=freq, ylim=c(0, ymax*1.2), ylab=ylab, xlab=xlab,
       main=main, col=bcol,  xlim = c(min(xlims), max(xlims)))
  lines(d, lty=dlty[1], col=dcol[1])
  curve(dnorm(x,m,s), add=TRUE, lty=dlty[2], col=dcol[2])

}


### Loop over variables
for (i in 1:ncol(df_num)){
  single_hist(df_num_order[,i], main = paste("Histogram of ", colnames(df_num_order[i])))
}
```

```
# Continuous variables
## Descriptive graphics: Histograms Summary
k <- ceiling(ncol(df_num)/20)-1
for (i in 0:k){
  m <- 20*i+1
  n <- min(20*(i+1),ncol(df_num))
  multi.hist(df_num_order[,m:n], dcol=c("#396e9f","#396e9f"),
             bcol= "#2fa42d",
             dlty=c("dotted", "solid"),
             main = colnames(df_num_order[,m:n]))
}
```

```
# Continuous variables
```

```
## Descriptive graphics: Box-Plot One Per Page

### Loop over variables
for (i in 1:ncol(df_num)){
  boxplot(df_num_order[,c(i)], col = "#2fa42d",
       main = paste("Boxplot of",colnames(df_num_order[i])),
       xlab=paste(colnames(df_num_order[i])), horizontal = TRUE)
}

# Continuous variables
## Descriptive graphics: Box-Plots Summary

### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_num_order))), ceiling(sqrt(length(df_num_order)))),
    mar=c(1.5,1,2,1), oma=c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_num)){
  boxplot(df_num_order[,c(i)], col = "#2fa42d", main = colnames(df_num_order[i]),
          xlab=paste(colnames(df_num_order[i])), xaxt="n", horizontal = TRUE)
}

### Restore original graphical settings
par(opar)

# Continuous variables
## Descriptive graphics: ECDF Plots One Per Page

### Loop over variables
for (i in 1:ncol(df_num)){

  data <- as.data.frame(df_num_order[,c(i)])
  colnames(data) <- "variable"

  # Plot ECDF
  step_function <- ecdf(data$variable)
  plot(step_function,
      main=paste("ECDF Plot of", colnames(df_num_order[i])),
      xlab=colnames(df_num_order[i]), ylab="ECDF",
      cex=0.7, col="#2fa42d", do.points=TRUE)

  # Plot CDF of normal distribution
  data_mean<- mean(data$variable, na.rm=TRUE)
  data_sd<- sd(data$variable, na.rm=TRUE)
  curve(pnorm(x, data_mean,data_sd),
        from=qnorm(0.0001, mean=data_mean, sd=data_sd),
        to=qnorm(0.9999, mean=data_mean, sd=data_sd),
        add=TRUE, col="#396e9f", lwd=2)
}

# Continuous variables
## Graphics: ECDF Plots Summary

### ECDF function
ecdf_plot <- function(i){

  data <- as.data.frame(df_num_order[,c(i)])
  colnames(data)<-"variable"

  # Plot ECDF
  step_function <- ecdf(data$variable)
  ecdf_plot <- plot(step_function,
                  main = colnames(df_num_order[i]),
                  xlab = colnames(df_num_order[i]), ylab = "ECDF",
                  cex = 0.7, col="#2fa42d", do.points = FALSE)

  # Plot CDF of normal distribution
  data_mean <- mean(data$variable, na.rm=TRUE)
  data_sd <- sd(data$variable, na.rm=TRUE)
  curve(pnorm(x, data_mean,data_sd),
        from = qnorm(0.0001, mean = data_mean, sd = data_sd),
        to = qnorm(0.9999, mean = data_mean, sd = data_sd),
        add = TRUE, col="#396e9f", lwd=0.5,pch=1)
}


### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_num_order))), ceiling(sqrt(length(df_num_order)))),
    mar=c(1.5,1,2,1), oma=c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_num)) ecdf_plot(i)

### Restore original graphical settings
par(opar)
```

```
# Continuous variables
## Graphics: QQ Plots One Per Page

### Define function for the QQ-Plot
qq_plot <- function(i, main, xlab, ylab){
    var <- df_num_order[,i]
    qqplot(x = qnorm(ppoints(var), mean = mean(var, na.rm = TRUE),
                     sd = sd(var, na.rm = TRUE)),
         y = var,
         xlim = c(min(var, na.rm = TRUE), max(var, na.rm = TRUE)),
         ylim = c(min(var, na.rm = TRUE), max(var, na.rm = TRUE)),
         main = main,
         xlab = xlab,
         ylab = ylab,
         col = "#2fa42d", cex=0.7, pch=19
         )
    abline(a = 0, b = 1, col = "#396e9f", lwd = 2)
    grid()
}

### Loop over variables
for (i in 1:ncol(df_num)){
  qq_plot(i, main = paste("QQ-Plot of", colnames(df_num_order[i])),
            xlab = "Theoretical Quantiles, Normal Distribution",
            ylab = paste("Sample Quantiles for ", colnames(df_num_order[i]))
          )
}
# Continuous variables
## Graphics: QQ Plots Summary
### Set graphical parameters
par(mfrow=c(ceiling(sqrt(length(df_num_order))),
            ceiling(sqrt(length(df_num_order)))),
    mar=c(1.5,1,2,1), oma=c(1,1,1,1))

### Loop over variables
for(i in 1:ncol(df_num)){
  qq_plot(i, colnames(df_num_order[i]), "", "")
}

### Restore original graphical settings
par(opar)
```

R Code License