

# Analiza techniczna funduszu opartego na ETF notowanych na GPW z wykorzystaniem długoterminowej strategii

## Pakiety

ten dokument wymaga poniższych pakietów

```
#install.packages('devtools') # jeżeli konieczne

# przed zainstalowaniem "blotter/quantstrat" należy zainstalować
# obydwie poniższe pakiety

#install.packages('PerformanceAnalytics')
#install.packages('FinancialInstrument')

# devtools::install_github('braverock/blotter') # jeżeli konieczne
# devtools::install_github('braverock/quantstrat') # jeżeli konieczne

suppressPackageStartupMessages({
  library(quantmod)
  library(FinancialInstrument)
  library(PerformanceAnalytics)
  library(foreach)
  library(blotter)
  library(quantstrat)

#devtools::install_github('IlyaKipnis/IKTrading')
  library(IKTrading)
})
```

### Suppresses warnings

```
options("getSymbols.warning4.0" = FALSE)
```

### Ustawienia strefy czasowej oraz waluty

```
# czyszczenie procesora z ewentualnych strategii

# rm(list = ls(.blotter), envir = .blotter)

# Ustawienie waluty oraz strefy czasowej
currency("PLN")
```

```
## [1] "PLN"
```

```
Sys.setenv(TZ = "UTC")
```

Pobieranie danych ze stooq.com i przekształcenie w format xts. Musimy je zapisywać na dysku przez funkcję `write.zoo()` ze względu na specyfikę dalszych analiz.

```
W20L <- read.csv("f:/R_nauka/etfw201_pl_d.csv")
# W20L <- select(W20L, -OpenInt) # dla kontraktów terminowych lepiej usunąć, psuje wykresy w "quantmod"
W20L <- tibble::add_column(W20L, Adjusted = W20L$Close) # to jest format dodawny przez yahoo.com.
# W polskich reliach = się cenie "Close"
head(W20L, 4)
```

```
W20L <- xts(W20L[, -1], order.by = as.Date(W20L[, 1]))
names(W20L) = c("W20L.Open", "W20L.High", "W20L.Low", "W20L.Close", "W20L.Volume", "W20L.Adjusted")
chartSeries(W20L)
```



```
write.zoo(W20L, file = "f:/R_nauka/W20L.csv", sep = ",")
```

```
DAX <- read.csv("f:/R_nauka/etfdax_pl_d.csv")
DAX <- tibble::add_column(DAX, Adjusted = DAX$Close)
head(DAX, 4)
```

```
tail(DAX, 4)
```

```
DAX <- xts(DAX[,-1], order.by = as.Date(DAX[,1]))
names(DAX) = c("DAX.Open", "DAX.High", "DAX.Low", "DAX.Close", "DAX.Volume", "DAX.Adjusted")
chartSeries(DAX)
```



```
write.zoo(DAX, file = "f:/R_nauka/DAX.csv", sep = ",")

SPY_1 <- read.csv("f:/R_nauka/etf500_pl_d.csv")
SPY_1 <- tibble::add_column(SPY_1, Adjusted = SPY_1$Close)
head(SPY_1, 4)
```

```
tail(SPY_1)
```

```
SPY_1 <- xts(SPY_1[, -1], order.by = as.Date(SPY_1[, 1]))
names(SPY_1) = c("SPY_1.Open", "SPY_1.High", "SPY_1.Low", "SPY_1.Close", "SPY_1.Volume", "SPY_1.Adjusted")
chartSeries(SPY_1)
```



```
addSMA(200)
```



```
write.zoo(SPY_1, file = "f:/R_nauka/SPY_1.csv", sep = ",")

#setDefault(getSymbols, src = 'csv') # trzeba prze źródło 'csv'
```

## Określenie aktywów przeznaczonych do analizy

```
symbols <- c("DAX", #ETF DAX
            "SPY_1", #ETF SP500
            "W20L") #ETF Wig20 Long
```

## ETF z GPW ponowne zaciągnięcie danych już wcześniej przekształconych i zapisanych na dysku

```
suppressMessages(getSymbols(symbols, from = '2000-01-01', dir = "f:/R_nauka/",
                             to = Sys.Date(), src = "csv", adjust = TRUE))
```

```
## [1] "DAX" "SPY_1" "W20L"
```

## Nadanie nazw, wykorzystywanych w dalszej pracy

```
stock(symbols, currency = "PLN", multiplier = 1)
```

```
## [1] "DAX" "SPY_1" "W20L"
```

```
str(W20L)
```

```
## An 'xts' object on 2000-02-12/2020-04-14 containing:
## Data: num [1:2385, 1:6] 212 248 249 247 250 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : chr [1:6] "W20L.Open" "W20L.High" "W20L.Low" "W20L.Close" ...
## Indexed by objects of class: [Date] TZ: UTC
## xts Attributes:
## List of 2
## $ src : chr "csv"
## $ updated: POSIXct[1:1], format: "2020-06-16 20:14:18"
```

```
str(DAX)
```

```
## An 'xts' object on 2000-04-26/2020-04-14 containing:
## Data: num [1:2093, 1:6] 429 282 300 299 276 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : chr [1:6] "DAX.Open" "DAX.High" "DAX.Low" "DAX.Close" ...
## Indexed by objects of class: [Date] TZ: UTC
## xts Attributes:
## List of 2
## $ src : chr "csv"
## $ updated: POSIXct[1:1], format: "2020-06-16 20:14:18"
```

```
str(SPY_1)
```

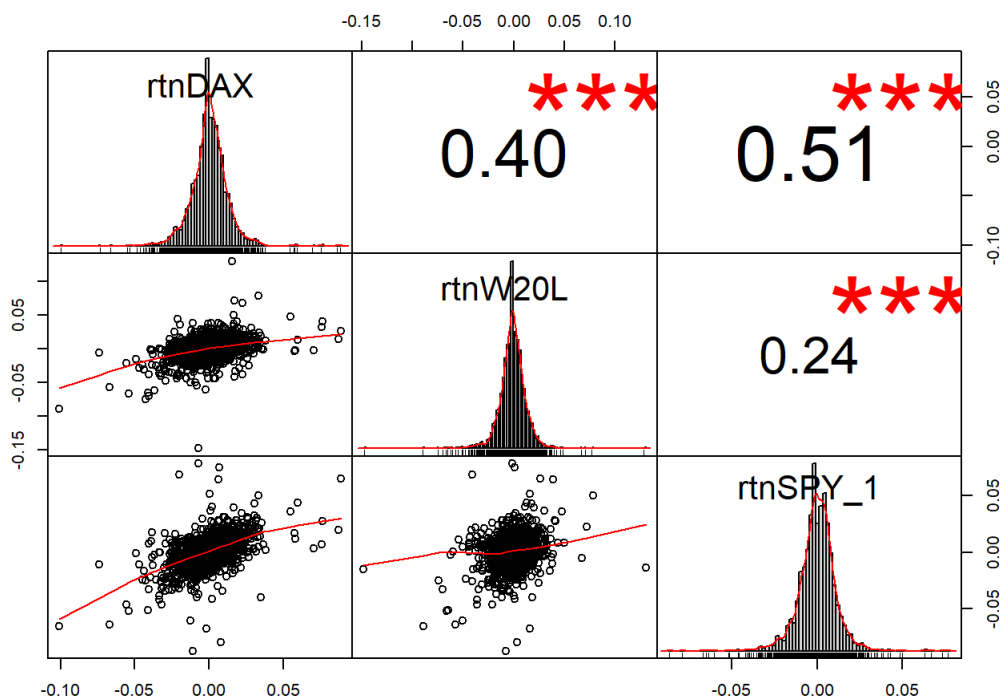
```
## An 'xts' object on 2011-05-31/2020-04-14 containing:
## Data: num [1:2142, 1:6] 34.8 35.7 34.5 31.7 31.3 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : chr [1:6] "SPY_1.Open" "SPY_1.High" "SPY_1.Low" "SPY_1.Close" ...
## Indexed by objects of class: [Date] TZ: UTC
## xts Attributes:
## List of 2
## $ src : chr "csv"
## $ updated: POSIXct[1:1], format: "2020-06-16 20:14:18"
```

## wykres korelacji logarytmowanych stóp zwrotu oraz tabela korelacji

```
data<-cbind( diff(log(Cl(DAX))), diff(log(Cl(W20L))), diff(log(Cl(SPY_1))) )
data <- na.omit(data)
names(data) = c("rtnDAX", "rtnW20L", "rtnSPY_1")

#str(data)
#View(data)

chart.Correlation(data)
```



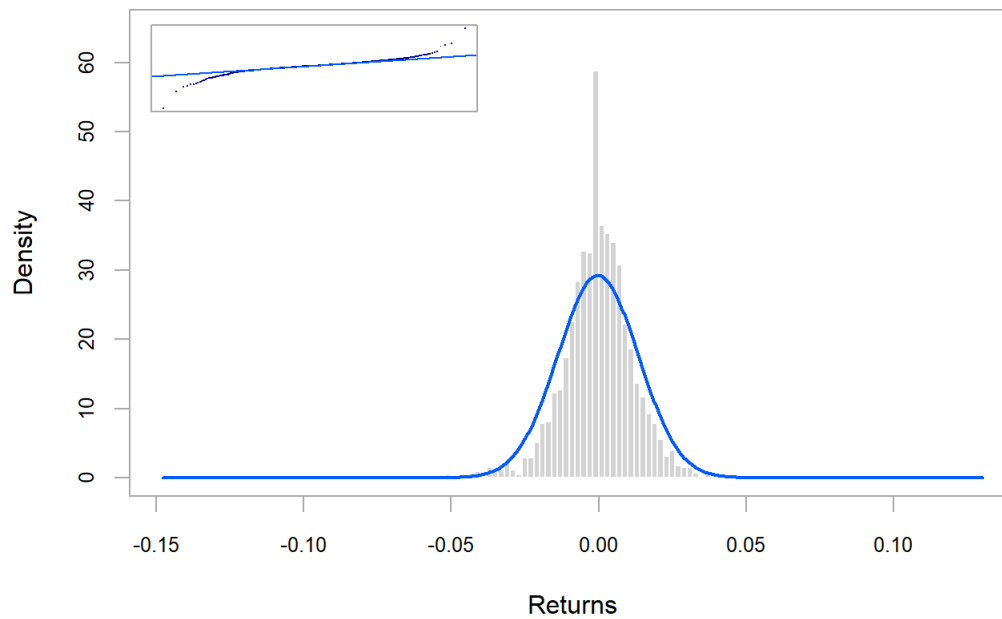
```
table.cor <- table.Correlation(data$rtnW20L, data)
knitr::kable(table.cor)
```

	Correlation	p-value	Lower CI	Upper CI
rtnW20L to rtnDAX	0.4018678	0	0.3648502	0.4376170
rtnW20L to rtnW20L	1.0000000	0	1.0000000	1.0000000
rtnW20L to rtnSPY_1	0.2414406	0	0.2001591	0.2818663

histogramy zlogarytmowanych stóp zwrotu badanych aktywów

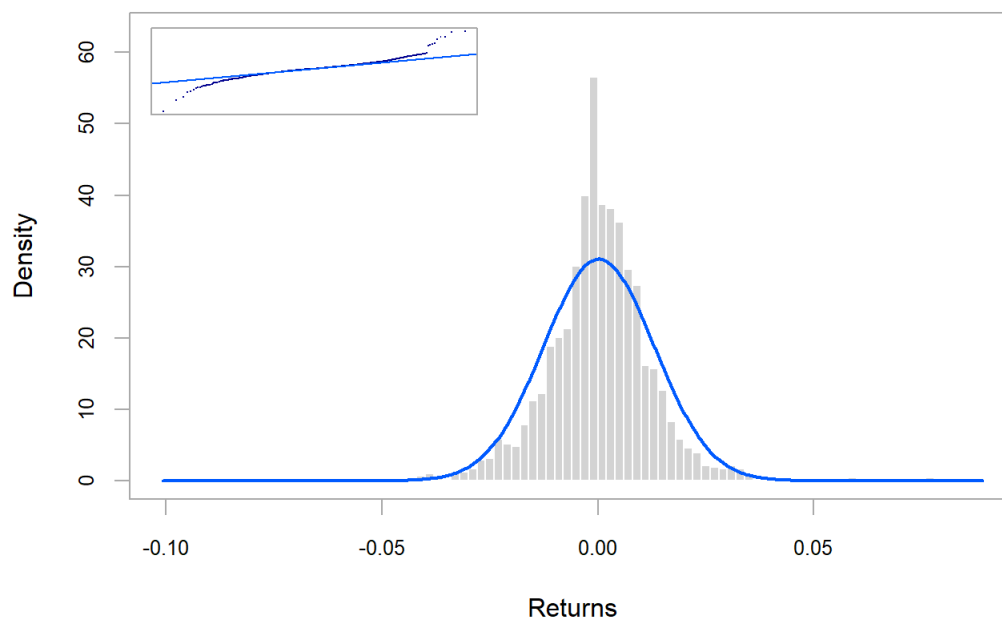
```
chart.Histogram(data$rtnW20L, probability = TRUE, main = "histogram dla stóp zwrotu W20L",
  methods = c("add.normal", "add.qqplot"))
```

histogram dla stóp zwrotu W20L



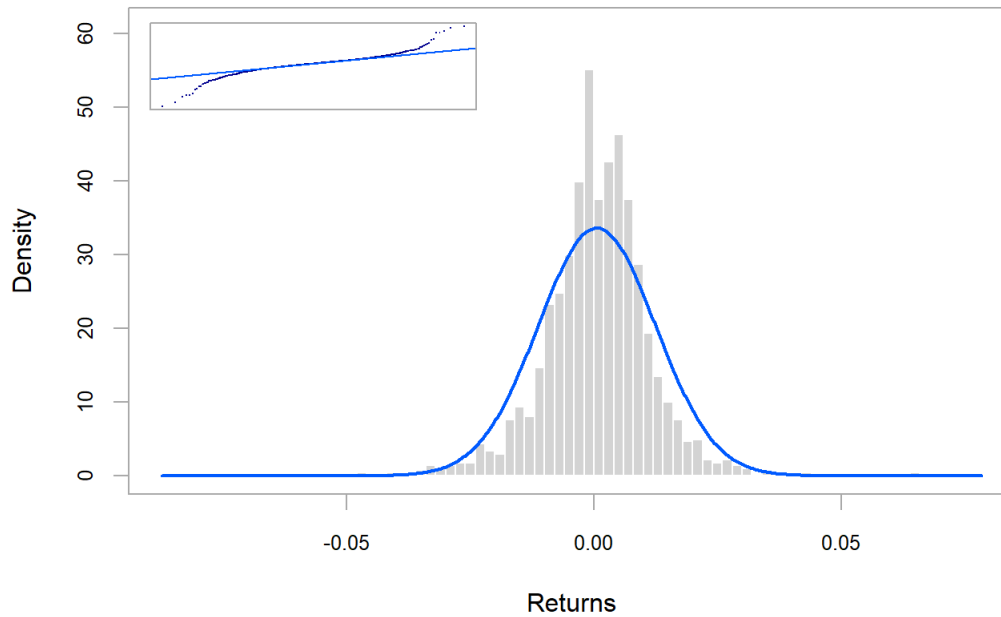
```
chart.Histogram(data$rtuDAX, probability = TRUE, main = "histogram dla stóp zwrotu DAX",  
  methods = c("add.normal", "add.qqplot"))
```

histogram dla stóp zwrotu DAX



```
chart.Histogram(data$rtuSPY_1, probability = TRUE, main = "histogram dla stóp zwrotu SPY_1",  
  methods = c("add.normal", "add.qqplot"))
```

### histogram dla stóp zwrotu SPY\_1

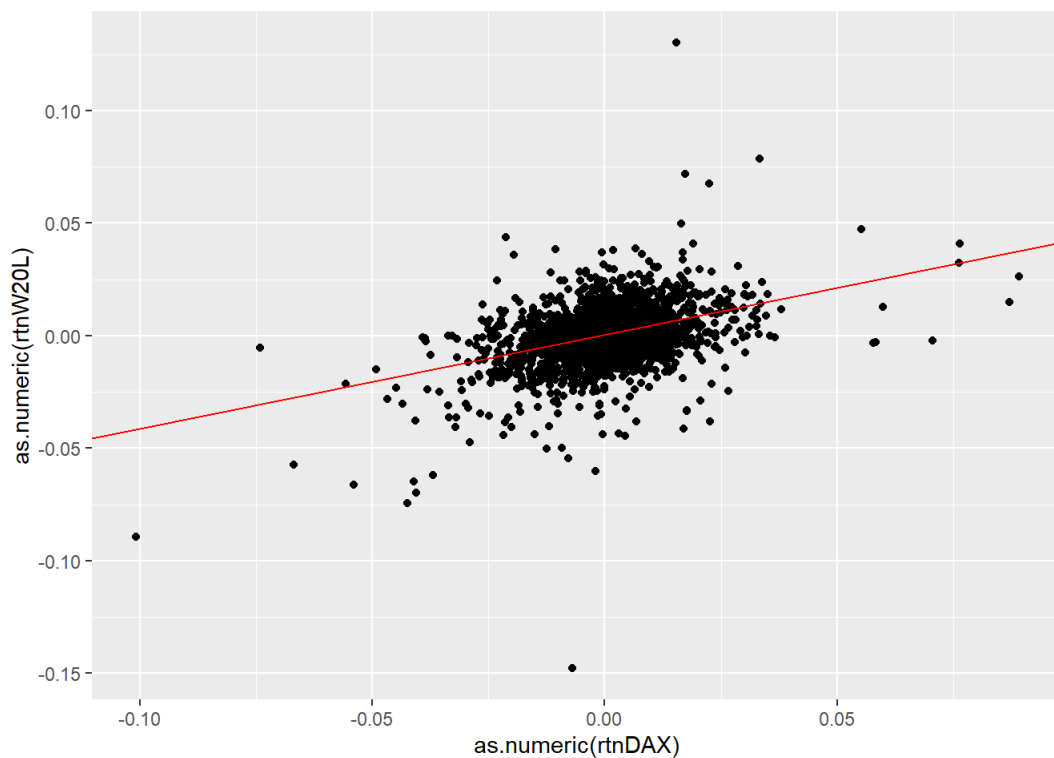


modele regresji pomiędzy zlogarytmowanymi stopami zwrotów

```
library(ggplot2)
fit_1 <- lm(rtnW20L ~ rtnDAX, data = data)
summary(fit_1)
```

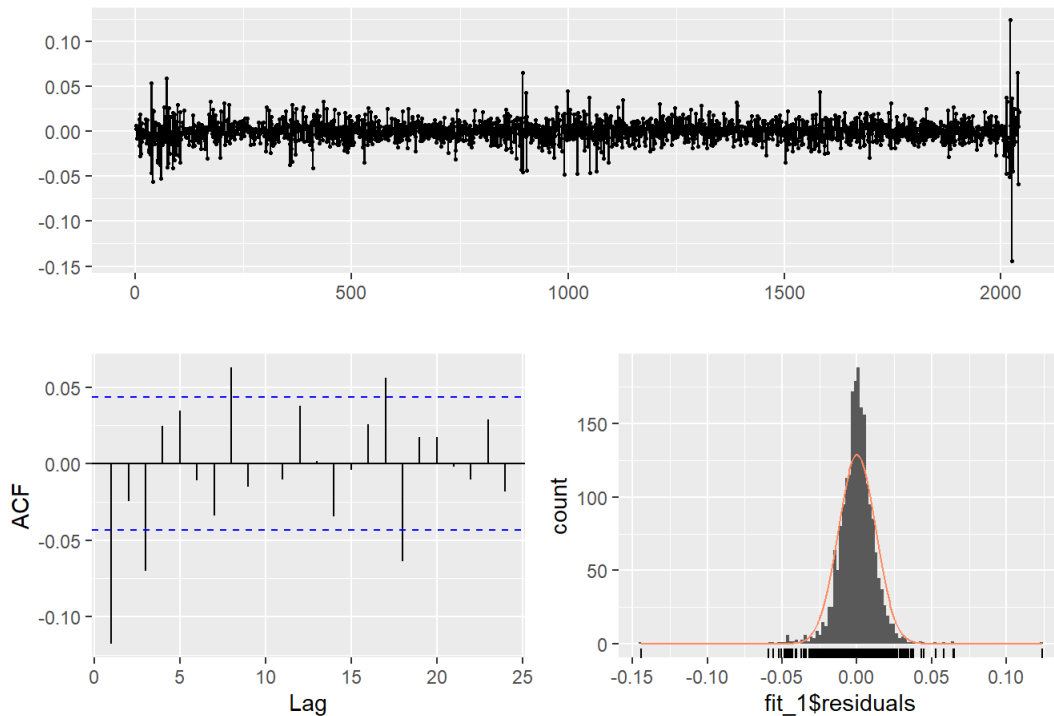
```
##
## Call:
## lm(formula = rtnW20L ~ rtnDAX, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.144523 -0.006179  0.000233  0.006355  0.123854
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0003393  0.0002765  -1.227    0.22
## rtnDAX       0.4264330  0.0215132  19.822 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01249 on 2040 degrees of freedom
## Multiple R-squared:  0.1615, Adjusted R-squared:  0.1611
## F-statistic: 392.9 on 1 and 2040 DF,  p-value: < 2.2e-16
```

```
ggplot(data = data, aes( x = as.numeric(rtnDAX), y = as.numeric(rtnW20L))) +
  geom_point() +
  geom_abline(intercept = 0.0004, slope = 0.418, colour = "red")
```



```
# ggplot(data = data, aes( x = as.numeric(rtnDAX), y = as.numeric(rtnW20L))) +
#   geom_point() +
#   geom_smooth()
forecast::ggtsdisplay(fit_1$residuals, plot.type = "histogram", lag.max = 24,
                      main = "Wykres reszt pozostałych z modelu regresji liniowej fit_1") # dopasowanie reszt
rtnDAX vs. rtnW20L
```

Wykres reszt pozostałych z modelu regresji liniowej fit\_1

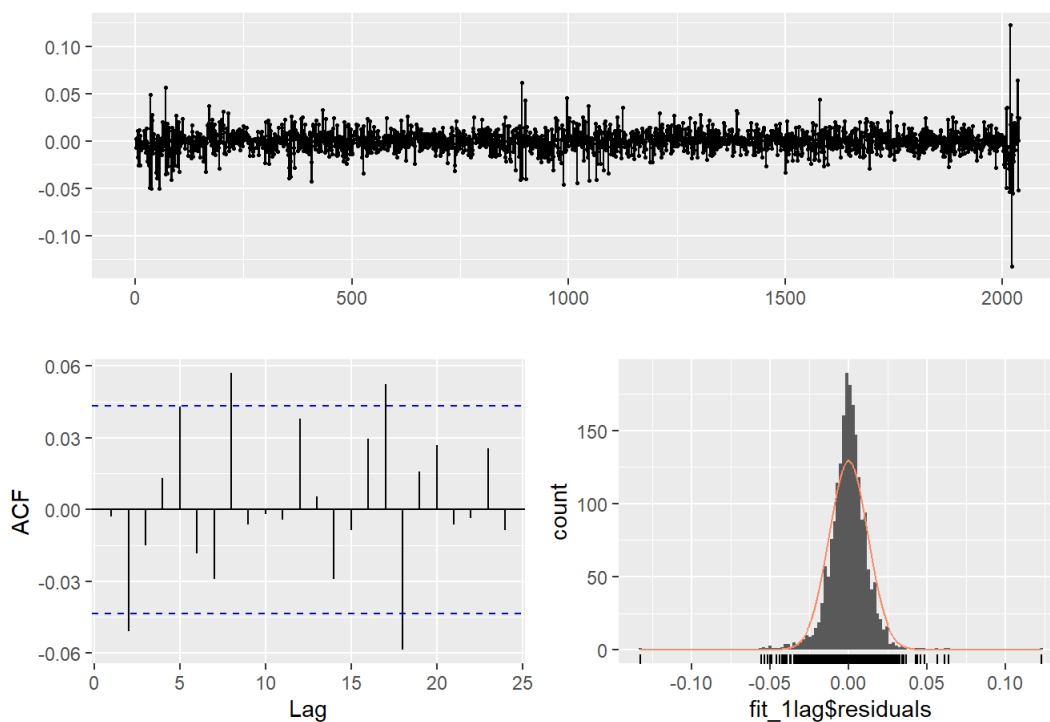


```
data_lag <- data
data_lag$rtnW20L_lag1 <- lag(data$rtnW20L)
data_lag$rtnDAX_lag1 <- lag(data$rtnDAX)
data_lag$rtnW20L_lag3 <- lag(data$rtnW20L, 3)
fit_lag <- lm(rtnW20L ~ rtnDAX + rtnW20L_lag1 + rtnDAX_lag1 + rtnW20L_lag3, data = data_lag)
summary(fit_lag)
```



```
##
## Call:
## lm(formula = rtnW20L ~ rtnDAX + rtnW20L_lag1 + rtnDAX_lag1 +
##     rtnW20L_lag3, data = data_lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.132618 -0.006095  0.000229  0.006388  0.122747
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.0004210  0.0002742  -1.535  0.12483
## rtnDAX        0.4425828  0.0214393  20.644 < 2e-16 ***
## rtnW20L_lag1 -0.1215196  0.0220403  -5.514 3.97e-08 ***
## rtnDAX_lag1   0.1110502  0.0233353   4.759 2.08e-06 ***
## rtnW20L_lag3 -0.0627899  0.0202297  -3.104  0.00194 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01237 on 2034 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.1804, Adjusted R-squared:  0.1788
## F-statistic: 112 on 4 and 2034 DF, p-value: < 2.2e-16
```

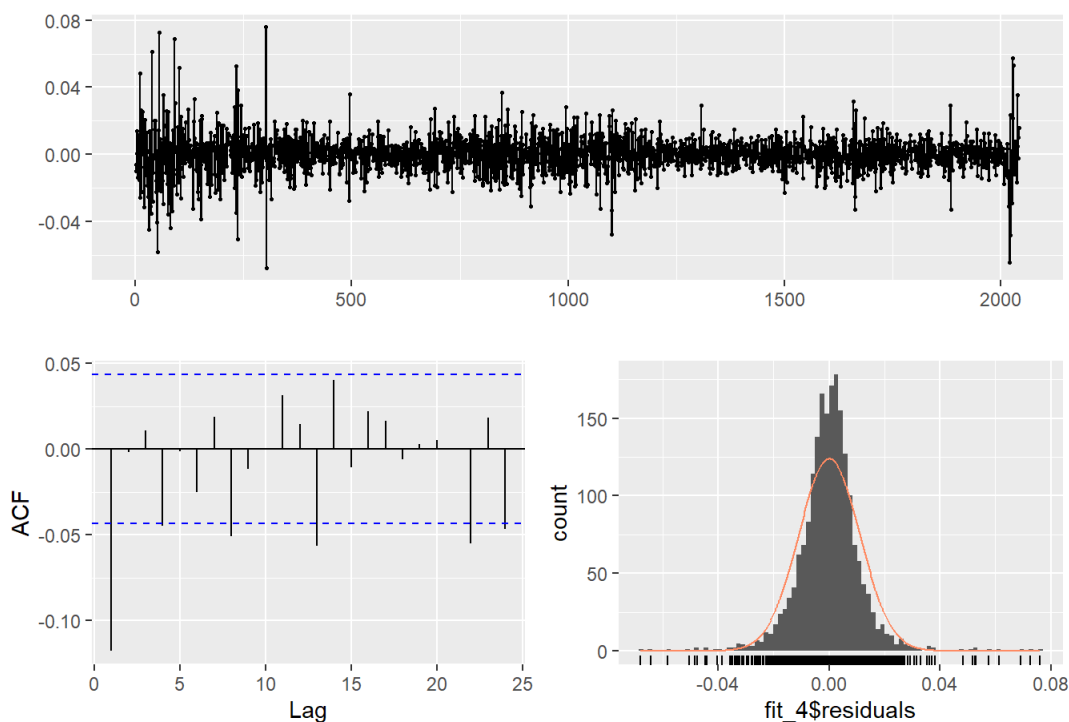
```
forecast::ggtsdisplay(fit_lag$residuals, plot.type = "histogram", lag.max = 24 )
```



```
fit_4 <- lm(rtnDAX ~ rtnSPY_1 , data = data)
summary(fit_4)
```

```
##
## Call:
## lm(formula = rtnDAX ~ rtnSPY_1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.067961 -0.005332  0.000243  0.005379  0.076162
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001041  0.0002442  -0.426   0.67
## rtnSPY_1      0.5573020  0.0205465  27.124 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01102 on 2040 degrees of freedom
## Multiple R-squared:  0.2651, Adjusted R-squared:  0.2647
## F-statistic: 735.7 on 1 and 2040 DF, p-value: < 2.2e-16
```

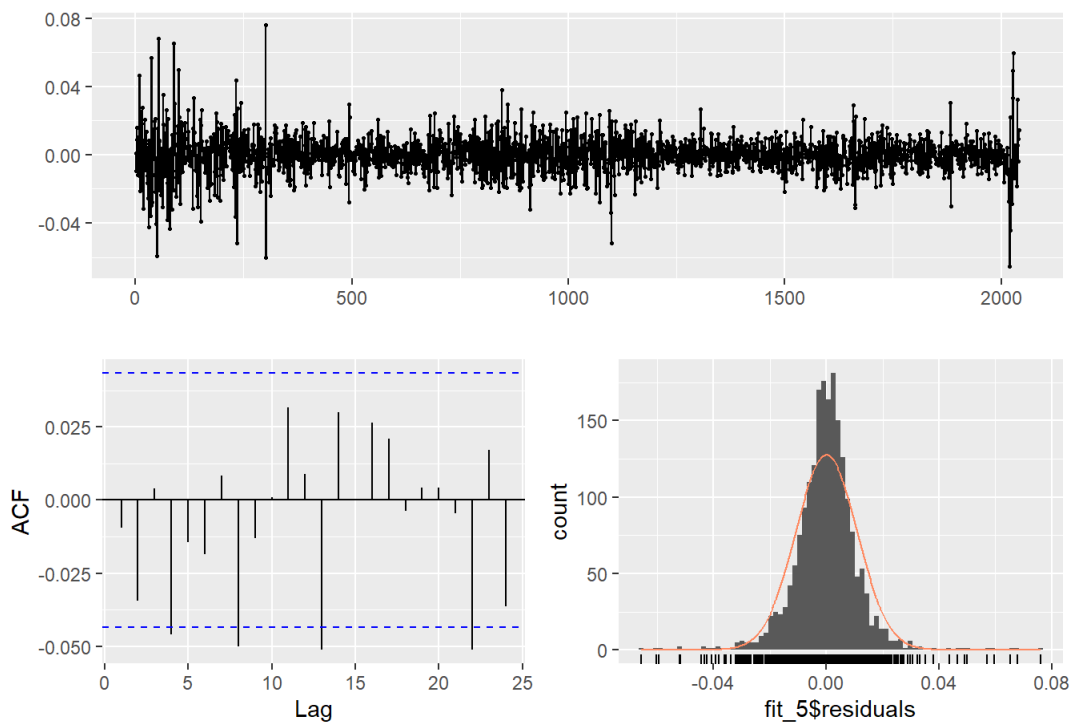
```
forecast::ggtsdisplay(fit_4$residuals, plot.type = "histogram", lag.max = 24) # dopasowanie reszt rtnDAX vs.
rtnSPY_1
```



```
data_lag$rtnSPY_1_lag1 <- lag(data$rtnSPY_1)
fit_5 <- lm(rtnDAX ~ rtnSPY_1 + rtnDAX_lag1 + rtnSPY_1_lag1, data = data_lag)
summary(fit_5)
```

```
##
## Call:
## lm(formula = rtnDAX ~ rtnSPY_1 + rtnDAX_lag1 + rtnSPY_1_lag1,
##     data = data_lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.065523 -0.005423  0.000168  0.005489  0.076143
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.0001775  0.0002420  -0.733    0.463
## rtnSPY_1       0.5781669  0.0205608  28.120 < 2e-16 ***
## rtnDAX_lag1    -0.1199322  0.0220160  -5.448 5.73e-08 ***
## rtnSPY_1_lag1  0.1551912  0.0239745   6.473 1.20e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0109 on 2037 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.2818, Adjusted R-squared:  0.2807
## F-statistic: 266.4 on 3 and 2037 DF, p-value: < 2.2e-16
```

```
forecast::ggtsdisplay(fit_5$residuals, plot.type = "histogram", lag.max = 24)
```



##strategia podążania za trendem “CLENOW SIMPLE”

Podstawowe parametry: zakres dat, ważne aby *initDate*= była wcześniej niż *from*=

```
initDate = '2010-01-01'
from = '2011-01-01'
to = Sys.Date()
options(width = 70)
```

wartosc i wielkosc początkowej pozycji

```
tradeSize <- 100000
initEq <- tradeSize * length(symbols)

strategy.st <- "Clenow_Simple"
portfolio.st <- "Clenow_Simple"
account.st <- "Clenow_Simple"
```

Należy pamiętać aby wyczyścić procesor jeżeli może mieć w pamięci poprzednio analizowaną strategię. Równocześnie ponownie wprowadzić ponownie środowisko pakietu **blotter**

```
#rm.strat(portfolio.st)
#rm.strat(strategy.st)
#if(!exists(".blotter")) .blotter <- new.env()
```

Ustawienie parametrów początkowych

```
initPortf(portfolio.st, symbols = symbols,
          initDate = initDate, currency = 'PLN')
```

```
## [1] "Clenow_Simple"
```

```
initAcct(account.st, portfolios = portfolio.st,
          initDate = initDate, currency = 'PLN', initEq = initEq)
```

```
## [1] "Clenow_Simple"
```

```
initOrders(portfolio.st, initDate = initDate)
strategy(strategy.st, store=TRUE)
```

## Backtest CLENOW SIMPLE

indykatory

```
nLag = 252
pctATR = 0.02
period = 10

namedLag <- function(x, k = 1, na.pad = TRUE, ...) { # oblicza wartość łączącą warunki -252dniowe opóźnienie
                                                    # -utworzy wynik w dodatkowej kolumnie

  out <- lag(x, k = k, na.pad = na.pad, ...)
  out[is.na(out)] <- x[is.na(out)]
  colnames(out) <- "namedLag"
  return(out)
}
```

Dodawanie wskaźników do strategii

```
add.indicator(strategy.st, name = "namedLag",
              arguments = list(x = quote(Cl(mktdata)), k = nLag),
              label = "ind")
```

```
## [1] "Clenow_Simple"
```

```
add.indicator(strategy.st, name = "lagATR", # z pakietu IKTrading zwraca wartość opóźnionego ATR(n=period)
              arguments = list(HLC = quote(HLC(mktdata)), n = period),
              label = "atrX")
```

```
## [1] "Clenow_Simple"
```

```
test <- applyIndicators(strategy.st, mktdata = OHLC(W20L))
head(round(test, 2), 6)
```

```
##          W20L.Open W20L.High W20L.Low W20L.Close namedLag.ind
## 2000-02-12    212.00    211.46    212.00     290.00     290.00
## 2010-09-22    248.40    248.59    246.99     248.13     248.13
## 2010-09-23    249.35    250.48    244.54     246.40     246.40
## 2010-09-24    246.61    249.25    245.34     248.78     248.78
## 2010-09-27    250.19    250.38    248.78     248.78     248.78
## 2010-09-28    247.27    248.12    246.33     246.33     246.33
##          atr.atrX
## 2000-02-12         NA
## 2010-09-22         NA
## 2010-09-23         NA
## 2010-09-24         NA
## 2010-09-27         NA
## 2010-09-28         NA
```

## dodawanie sygnałów do strategii

```
add.signal(strategy.st, name = "sigCrossover",
            arguments = list(columns = c("Close", "namedLag.ind"),
                              relationship = "gt"),
            label = "coverOrBuy")
```

```
## [1] "Clenow_Simple"
```

```
add.signal(strategy.st, name = "sigCrossover",
            arguments = list(columns = c("Close", "namedLag.ind"),
                              relationship = "lt"),
            label = "sellOrShort")
```

```
## [1] "Clenow_Simple"
```

## dodawanie warunków dla pozycji długiej(kupno)

```
add.rule(strategy.st, name = "ruleSignal",
          arguments = list(sigcol = "coverOrBuy",
                          sigval = TRUE, ordertype = "market",
                          orderside = "long", replace = FALSE,
                          prefer = "Open", osFUN = osDollarATR,
                          tradeSize = tradeSize, pctATR = pctATR,
                          atrMod = "X"), type = "enter", path.dep = TRUE)
```

```
## [1] "Clenow_Simple"
```

```
add.rule(strategy.st, name = "ruleSignal",
          arguments = list(sigcol = "sellOrShort",
                          sigval = TRUE, orderqty = "all",
                          ordertype = "market", orderside = "long",
                          replace = FALSE, prefer = "Open"),
          type = "exit", path.dep = TRUE)
```

```
## [1] "Clenow_Simple"
```

## dodawanie warunków dla pozycji krótkiej(sprzedaż)

```
add.rule(strategy.st, name = "ruleSignal",
          arguments = list(sigcol = "sellOrShort",
                          sigval = TRUE, ordertype = "market",
                          orderside = "short", replace = FALSE,
                          prefer = "Open", osFUN = osDollarATR,
                          tradeSize = -tradeSize, pctATR = pctATR,
                          atrMod = "X"), type = "enter", path.dep = TRUE)
```

```
## [1] "Clenow_Simple"
```

```
add.rule(strategy.st, name = "ruleSignal",
  arguments = list(sigcol = "coverOrBuy",
    signal = TRUE, orderqty = "all",
    ordertype = "market", orderside = "short",
    replace = FALSE, prefer = "Open"),
  type = "exit", path.dep = TRUE)
```

```
## [1] "Clenow_Simple"
```

## uruchomienie analizy

```
t1 <- Sys.time()
out <- applyStrategy(strategy = strategy.st,
  portfolios = portfolio.st)
```

```
## [1] "2012-09-21 00:00:00 DAX -709 @ 298.5"
## [1] "2012-09-25 00:00:00 DAX 709 @ 299.8"
## [1] "2012-09-25 00:00:00 DAX 755 @ 299.8"
## [1] "2014-10-16 00:00:00 DAX -755 @ 349.43"
## [1] "2014-10-16 00:00:00 DAX -299 @ 349.43"
## [1] "2014-10-20 00:00:00 DAX 299 @ 360"
## [1] "2014-10-20 00:00:00 DAX 279 @ 360"
## [1] "2015-09-23 00:00:00 DAX -279 @ 390.35"
## [1] "2015-09-23 00:00:00 DAX -275 @ 390.35"
## [1] "2015-09-28 00:00:00 DAX 275 @ 395"
## [1] "2015-09-28 00:00:00 DAX 246 @ 395"
## [1] "2015-09-29 00:00:00 DAX -246 @ 382.52"
## [1] "2015-09-29 00:00:00 DAX -234 @ 382.52"
## [1] "2015-10-01 00:00:00 DAX 234 @ 399.97"
## [1] "2015-10-01 00:00:00 DAX 240 @ 399.97"
## [1] "2016-01-18 00:00:00 DAX -240 @ 400"
## [1] "2016-01-18 00:00:00 DAX -183 @ 400"
## [1] "2016-08-24 00:00:00 DAX 183 @ 436.24"
## [1] "2016-08-24 00:00:00 DAX 331 @ 436.24"
## [1] "2016-11-02 00:00:00 DAX -331 @ 443.9"
## [1] "2016-11-02 00:00:00 DAX -455 @ 443.9"
## [1] "2016-11-14 00:00:00 DAX 455 @ 454.89"
## [1] "2016-11-14 00:00:00 DAX 278 @ 454.89"
## [1] "2016-11-28 00:00:00 DAX -278 @ 449.87"
## [1] "2016-11-28 00:00:00 DAX -379 @ 449.87"
## [1] "2016-12-01 00:00:00 DAX 379 @ 451.69"
## [1] "2016-12-01 00:00:00 DAX 395 @ 451.69"
## [1] "2016-12-02 00:00:00 DAX -395 @ 450.09"
## [1] "2016-12-02 00:00:00 DAX -404 @ 450.09"
## [1] "2016-12-09 00:00:00 DAX 404 @ 477.77"
## [1] "2016-12-09 00:00:00 DAX 416 @ 477.77"
## [1] "2018-03-02 00:00:00 DAX -416 @ 482.25"
## [1] "2018-03-02 00:00:00 DAX -350 @ 482.25"
## [1] "2018-03-12 00:00:00 DAX 350 @ 498"
## [1] "2018-03-12 00:00:00 DAX 294 @ 498"
## [1] "2018-03-14 00:00:00 DAX -294 @ 491.45"
## [1] "2018-03-14 00:00:00 DAX -334 @ 491.45"
## [1] "2018-03-19 00:00:00 DAX 334 @ 497.5"
## [1] "2018-03-19 00:00:00 DAX 345 @ 497.5"
## [1] "2018-03-20 00:00:00 DAX -345 @ 491.3"
## [1] "2018-03-20 00:00:00 DAX -344 @ 491.3"
## [1] "2018-04-16 00:00:00 DAX 344 @ 493.75"
## [1] "2018-04-16 00:00:00 DAX 296 @ 493.75"
## [1] "2018-04-17 00:00:00 DAX -296 @ 492"
## [1] "2018-04-17 00:00:00 DAX -311 @ 492"
## [1] "2018-04-18 00:00:00 DAX 311 @ 500"
## [1] "2018-04-18 00:00:00 DAX 323 @ 500"
## [1] "2018-06-22 00:00:00 DAX -323 @ 514.2"
## [1] "2018-06-22 00:00:00 DAX -313 @ 514.2"
## [1] "2018-07-02 00:00:00 DAX 313 @ 507.3"
## [1] "2018-07-02 00:00:00 DAX 296 @ 507.3"
## [1] "2018-09-07 00:00:00 DAX -296 @ 491.2"
## [1] "2018-09-07 00:00:00 DAX -458 @ 491.2"
## [1] "2018-09-11 00:00:00 DAX 458 @ 492.3"
```

```
## [1] "2018-09-11 00:00:00 DAX 455 @ 492.3"
## [1] "2018-09-12 00:00:00 DAX -455 @ 491.4"
## [1] "2018-09-12 00:00:00 DAX -467 @ 491.4"
## [1] "2018-09-27 00:00:00 DAX 467 @ 510.1"
## [1] "2018-09-27 00:00:00 DAX 272 @ 510.1"
## [1] "2018-09-28 00:00:00 DAX -272 @ 506.1"
## [1] "2018-09-28 00:00:00 DAX -260 @ 506.1"
## [1] "2019-04-03 00:00:00 DAX 260 @ 482.95"
## [1] "2019-04-03 00:00:00 DAX 456 @ 482.95"
## [1] "2019-04-11 00:00:00 DAX -456 @ 482.55"
## [1] "2019-04-11 00:00:00 DAX -493 @ 482.55"
## [1] "2019-04-24 00:00:00 DAX 493 @ 496.7"
## [1] "2019-04-24 00:00:00 DAX 509 @ 496.7"
## [1] "2019-05-02 00:00:00 DAX -509 @ 500.9"
## [1] "2019-05-02 00:00:00 DAX -543 @ 500.9"
## [1] "2019-07-05 00:00:00 DAX 543 @ 506.8"
## [1] "2019-07-05 00:00:00 DAX 311 @ 506.8"
## [1] "2019-07-09 00:00:00 DAX -311 @ 500.8"
## [1] "2019-07-09 00:00:00 DAX -353 @ 500.8"
## [1] "2019-09-12 00:00:00 DAX 353 @ 507"
## [1] "2019-09-12 00:00:00 DAX 336 @ 507"
## [1] "2019-10-02 00:00:00 DAX -336 @ 506.4"
## [1] "2019-10-02 00:00:00 DAX -522 @ 506.4"
## [1] "2019-10-14 00:00:00 DAX 522 @ 513.8"
## [1] "2019-10-14 00:00:00 DAX 377 @ 513.8"
## [1] "2020-03-09 00:00:00 DAX -377 @ 437"
## [1] "2020-03-09 00:00:00 DAX -118 @ 437"
## [1] "2012-07-20 00:00:00 SPY_1 3534 @ 41.438"
## [1] "2016-02-10 00:00:00 SPY_1 -3534 @ 70.627"
## [1] "2016-02-10 00:00:00 SPY_1 -1393 @ 70.627"
## [1] "2016-02-19 00:00:00 SPY_1 1393 @ 72.04"
## [1] "2016-02-19 00:00:00 SPY_1 1145 @ 72.04"
## [1] "2016-02-22 00:00:00 SPY_1 -1145 @ 72.124"
## [1] "2016-02-22 00:00:00 SPY_1 -1202 @ 72.124"
## [1] "2016-02-23 00:00:00 SPY_1 1202 @ 72.719"
## [1] "2016-02-23 00:00:00 SPY_1 1266 @ 72.719"
## [1] "2016-02-25 00:00:00 SPY_1 -1266 @ 72.358"
## [1] "2016-02-25 00:00:00 SPY_1 -1329 @ 72.358"
## [1] "2016-02-26 00:00:00 SPY_1 1329 @ 74.262"
## [1] "2016-02-26 00:00:00 SPY_1 1316 @ 74.262"
## [1] "2016-03-09 00:00:00 SPY_1 -1316 @ 73.436"
## [1] "2016-03-09 00:00:00 SPY_1 -1713 @ 73.436"
## [1] "2016-03-10 00:00:00 SPY_1 1713 @ 74.55"
## [1] "2016-03-10 00:00:00 SPY_1 1634 @ 74.55"
## [1] "2016-03-14 00:00:00 SPY_1 -1634 @ 74.244"
## [1] "2016-03-14 00:00:00 SPY_1 -1655 @ 74.244"
## [1] "2016-03-30 00:00:00 SPY_1 1655 @ 73.668"
## [1] "2016-03-30 00:00:00 SPY_1 1812 @ 73.668"
## [1] "2016-04-04 00:00:00 SPY_1 -1812 @ 73.899"
## [1] "2016-04-04 00:00:00 SPY_1 -2041 @ 73.899"
## [1] "2016-04-05 00:00:00 SPY_1 2041 @ 73.37"
## [1] "2016-04-05 00:00:00 SPY_1 1940 @ 73.37"
## [1] "2016-04-12 00:00:00 SPY_1 -1940 @ 73.082"
## [1] "2016-04-12 00:00:00 SPY_1 -2169 @ 73.082"
## [1] "2016-04-14 00:00:00 SPY_1 2169 @ 75.126"
## [1] "2016-04-14 00:00:00 SPY_1 2124 @ 75.126"
## [1] "2017-12-29 00:00:00 SPY_1 -2124 @ 92.659"
## [1] "2017-12-29 00:00:00 SPY_1 -2561 @ 92.659"
## [1] "2018-01-04 00:00:00 SPY_1 2561 @ 94.804"
## [1] "2018-01-04 00:00:00 SPY_1 2403 @ 94.804"
## [1] "2018-02-12 00:00:00 SPY_1 -2403 @ 88.362"
## [1] "2018-02-12 00:00:00 SPY_1 -1121 @ 88.362"
## [1] "2018-03-01 00:00:00 SPY_1 1121 @ 92.274"
## [1] "2018-03-01 00:00:00 SPY_1 1704 @ 92.274"
## [1] "2018-03-02 00:00:00 SPY_1 -1704 @ 91.742"
## [1] "2018-03-02 00:00:00 SPY_1 -1756 @ 91.742"
## [1] "2018-03-13 00:00:00 SPY_1 1756 @ 93.867"
## [1] "2018-03-13 00:00:00 SPY_1 1392 @ 93.867"
## [1] "2018-03-15 00:00:00 SPY_1 -1392 @ 92.708"
## [1] "2018-03-15 00:00:00 SPY_1 -1575 @ 92.708"
## [1] "2018-03-19 00:00:00 SPY_1 1575 @ 93.538"
## [1] "2018-03-19 00:00:00 SPY_1 1691 @ 93.538"
## [1] "2018-03-20 00:00:00 SPY_1 -1691 @ 91.751"
```

```
## [1] "2018-03-20 00:00:00 SPY_1 -1796 @ 91.751"
## [1] "2018-03-21 00:00:00 SPY_1 1796 @ 92.805"
## [1] "2018-03-21 00:00:00 SPY_1 1692 @ 92.805"
## [1] "2018-03-23 00:00:00 SPY_1 -1692 @ 89.502"
## [1] "2018-03-23 00:00:00 SPY_1 -1826 @ 89.502"
## [1] "2018-04-03 00:00:00 SPY_1 1826 @ 86.924"
## [1] "2018-04-03 00:00:00 SPY_1 1205 @ 86.924"
## [1] "2018-04-04 00:00:00 SPY_1 -1205 @ 88.266"
## [1] "2018-04-04 00:00:00 SPY_1 -1193 @ 88.266"
## [1] "2018-04-23 00:00:00 SPY_1 1193 @ 90.198"
## [1] "2018-04-23 00:00:00 SPY_1 1695 @ 90.198"
## [1] "2018-04-26 00:00:00 SPY_1 -1695 @ 91.742"
## [1] "2018-04-26 00:00:00 SPY_1 -1899 @ 91.742"
## [1] "2018-04-27 00:00:00 SPY_1 1899 @ 91.964"
## [1] "2018-04-27 00:00:00 SPY_1 1761 @ 91.964"
## [1] "2018-12-27 00:00:00 SPY_1 -1761 @ 92.406"
## [1] "2018-12-27 00:00:00 SPY_1 -1039 @ 92.406"
## [1] "2019-01-02 00:00:00 SPY_1 1039 @ 93.035"
## [1] "2019-01-02 00:00:00 SPY_1 849 @ 93.035"
## [1] "2020-03-11 00:00:00 SPY_1 -849 @ 109.6"
## [1] "2020-03-11 00:00:00 SPY_1 -394 @ 109.6"
## [1] "2020-03-12 00:00:00 SPY_1 394 @ 102.94"
## [1] "2020-03-12 00:00:00 SPY_1 387 @ 102.94"
## [1] "2020-03-13 00:00:00 SPY_1 -387 @ 98.12"
## [1] "2020-03-13 00:00:00 SPY_1 -397 @ 98.12"
## [1] "2020-03-20 00:00:00 SPY_1 397 @ 111.9"
## [1] "2020-03-20 00:00:00 SPY_1 347 @ 111.9"
## [1] "2020-03-23 00:00:00 SPY_1 -347 @ 100.98"
## [1] "2020-03-23 00:00:00 SPY_1 -334 @ 100.98"
## [1] "2020-03-27 00:00:00 SPY_1 334 @ 108.6"
## [1] "2020-03-27 00:00:00 SPY_1 313 @ 108.6"
## [1] "2020-03-30 00:00:00 SPY_1 -313 @ 106"
## [1] "2020-03-30 00:00:00 SPY_1 -313 @ 106"
## [1] "2020-03-31 00:00:00 SPY_1 313 @ 111.92"
## [1] "2020-03-31 00:00:00 SPY_1 324 @ 111.92"
## [1] "2020-04-02 00:00:00 SPY_1 -324 @ 107.86"
## [1] "2020-04-02 00:00:00 SPY_1 -322 @ 107.86"
## [1] "2020-04-03 00:00:00 SPY_1 322 @ 108.56"
## [1] "2020-04-03 00:00:00 SPY_1 331 @ 108.56"
## [1] "2011-09-22 00:00:00 W20L -269 @ 220.47"
## [1] "2012-08-09 00:00:00 W20L 269 @ 235.82"
## [1] "2012-08-09 00:00:00 W20L 598 @ 235.82"
## [1] "2012-08-31 00:00:00 W20L -598 @ 233"
## [1] "2012-08-31 00:00:00 W20L -706 @ 233"
## [1] "2012-09-04 00:00:00 W20L 706 @ 238.71"
## [1] "2012-09-04 00:00:00 W20L 635 @ 238.71"
## [1] "2012-10-30 00:00:00 W20L -635 @ 243.5"
## [1] "2012-10-30 00:00:00 W20L -685 @ 243.5"
## [1] "2012-10-31 00:00:00 W20L 685 @ 245"
## [1] "2012-10-31 00:00:00 W20L 748 @ 245"
## [1] "2013-12-20 00:00:00 W20L -748 @ 264.9"
## [1] "2013-12-20 00:00:00 W20L -370 @ 264.9"
## [1] "2013-12-27 00:00:00 W20L 370 @ 267"
## [1] "2013-12-27 00:00:00 W20L 324 @ 267"
## [1] "2013-12-30 00:00:00 W20L -324 @ 265.25"
## [1] "2013-12-30 00:00:00 W20L -320 @ 265.25"
## [1] "2014-02-07 00:00:00 W20L 320 @ 264.4"
## [1] "2014-02-07 00:00:00 W20L 366 @ 264.4"
## [1] "2014-03-14 00:00:00 W20L -366 @ 255.4"
## [1] "2014-03-14 00:00:00 W20L -468 @ 255.4"
## [1] "2014-03-18 00:00:00 W20L 468 @ 260.72"
## [1] "2014-03-18 00:00:00 W20L 418 @ 260.72"
## [1] "2014-03-19 00:00:00 W20L -418 @ 259.55"
## [1] "2014-03-19 00:00:00 W20L -393 @ 259.55"
## [1] "2014-03-21 00:00:00 W20L 393 @ 261"
## [1] "2014-03-21 00:00:00 W20L 433 @ 261"
## [1] "2014-03-25 00:00:00 W20L -433 @ 263.77"
## [1] "2014-03-25 00:00:00 W20L -442 @ 263.77"
## [1] "2014-03-26 00:00:00 W20L 442 @ 263.95"
## [1] "2014-03-26 00:00:00 W20L 419 @ 263.95"
## [1] "2014-08-11 00:00:00 W20L -419 @ 257.1"
## [1] "2014-08-11 00:00:00 W20L -660 @ 257.1"
## [1] "2014-08-13 00:00:00 W20L -660 @ 260"
```



```
## [1] "2014-08-12 00:00:00 W20L 660 @ 262"
## [1] "2014-08-12 00:00:00 W20L 624 @ 262"
## [1] "2014-10-17 00:00:00 W20L -624 @ 266.77"
## [1] "2014-10-17 00:00:00 W20L -442 @ 266.77"
## [1] "2014-10-22 00:00:00 W20L 442 @ 275.89"
## [1] "2014-10-22 00:00:00 W20L 464 @ 275.89"
## [1] "2014-10-28 00:00:00 W20L -464 @ 272.03"
## [1] "2014-10-28 00:00:00 W20L -521 @ 272.03"
## [1] "2014-11-03 00:00:00 W20L 521 @ 278.5"
## [1] "2014-11-03 00:00:00 W20L 566 @ 278.5"
## [1] "2014-11-04 00:00:00 W20L -566 @ 278.6"
## [1] "2014-11-04 00:00:00 W20L -581 @ 278.6"
## [1] "2014-11-19 00:00:00 W20L 581 @ 274.8"
## [1] "2014-11-19 00:00:00 W20L 484 @ 274.8"
## [1] "2014-11-20 00:00:00 W20L -484 @ 272.2"
## [1] "2014-11-20 00:00:00 W20L -495 @ 272.2"
## [1] "2014-12-09 00:00:00 W20L 495 @ 277.65"
## [1] "2014-12-09 00:00:00 W20L 619 @ 277.65"
## [1] "2014-12-10 00:00:00 W20L -619 @ 271"
## [1] "2014-12-10 00:00:00 W20L -587 @ 271"
## [1] "2014-12-30 00:00:00 W20L 587 @ 260"
## [1] "2014-12-30 00:00:00 W20L 412 @ 260"
## [1] "2015-01-05 00:00:00 W20L -412 @ 259.53"
## [1] "2015-01-05 00:00:00 W20L -439 @ 259.53"
## [1] "2015-01-13 00:00:00 W20L 439 @ 261.75"
## [1] "2015-01-13 00:00:00 W20L 411 @ 261.75"
## [1] "2015-01-19 00:00:00 W20L -411 @ 254.3"
## [1] "2015-01-19 00:00:00 W20L -388 @ 254.3"
## [1] "2015-01-30 00:00:00 W20L 388 @ 263.7"
## [1] "2015-01-30 00:00:00 W20L 435 @ 263.7"
## [1] "2015-02-12 00:00:00 W20L -435 @ 261.2"
## [1] "2015-02-12 00:00:00 W20L -671 @ 261.2"
## [1] "2015-03-09 00:00:00 W20L 671 @ 263.68"
## [1] "2015-03-09 00:00:00 W20L 835 @ 263.68"
## [1] "2015-03-10 00:00:00 W20L -835 @ 263.95"
## [1] "2015-03-10 00:00:00 W20L -886 @ 263.95"
## [1] "2015-03-18 00:00:00 W20L 886 @ 263.05"
## [1] "2015-03-18 00:00:00 W20L 900 @ 263.05"
## [1] "2015-06-09 00:00:00 W20L -900 @ 267.57"
## [1] "2015-06-09 00:00:00 W20L -562 @ 267.57"
## [1] "2015-06-11 00:00:00 W20L 562 @ 269.8"
## [1] "2015-06-11 00:00:00 W20L 538 @ 269.8"
## [1] "2015-06-12 00:00:00 W20L -538 @ 269.12"
## [1] "2015-06-12 00:00:00 W20L -521 @ 269.12"
## [1] "2015-07-06 00:00:00 W20L 521 @ 266"
## [1] "2015-07-06 00:00:00 W20L 547 @ 266"
## [1] "2015-07-07 00:00:00 W20L -547 @ 260.81"
## [1] "2015-07-07 00:00:00 W20L -475 @ 260.81"
## [1] "2016-12-07 00:00:00 W20L 475 @ 221.11"
## [1] "2016-12-07 00:00:00 W20L 644 @ 221.11"
## [1] "2018-04-30 00:00:00 W20L -644 @ 273"
## [1] "2018-04-30 00:00:00 W20L -534 @ 273"
## [1] "2018-06-11 00:00:00 W20L 534 @ 271.2"
## [1] "2018-06-11 00:00:00 W20L 407 @ 271.2"
## [1] "2018-06-12 00:00:00 W20L -407 @ 267.95"
## [1] "2018-06-12 00:00:00 W20L -433 @ 267.95"
## [1] "2018-08-28 00:00:00 W20L 433 @ 286.05"
## [1] "2018-08-28 00:00:00 W20L 579 @ 286.05"
## [1] "2018-08-29 00:00:00 W20L -579 @ 287.1"
## [1] "2018-08-29 00:00:00 W20L -535 @ 287.1"
## [1] "2019-02-13 00:00:00 W20L 535 @ 288.8"
## [1] "2019-02-13 00:00:00 W20L 511 @ 288.8"
## [1] "2019-02-14 00:00:00 W20L -511 @ 283.5"
## [1] "2019-02-14 00:00:00 W20L -532 @ 283.5"
## [1] "2019-03-08 00:00:00 W20L 532 @ 278.85"
## [1] "2019-03-08 00:00:00 W20L 589 @ 278.85"
## [1] "2019-03-14 00:00:00 W20L -589 @ 281"
## [1] "2019-03-14 00:00:00 W20L -621 @ 281"
## [1] "2019-03-15 00:00:00 W20L 621 @ 281.5"
## [1] "2019-03-15 00:00:00 W20L 647 @ 281.5"
## [1] "2019-05-08 00:00:00 W20L -647 @ 274.5"
## [1] "2019-05-08 00:00:00 W20L -666 @ 274.5"
## [1] "2019-05-29 00:00:00 W20L 666 @ 265"
```

```
## [1] "2019-05-29 00:00:00 W20L 482 @ 265"
## [1] "2019-05-30 00:00:00 W20L -482 @ 267.65"
## [1] "2019-05-30 00:00:00 W20L -498 @ 267.65"
## [1] "2019-05-31 00:00:00 W20L 498 @ 268.1"
## [1] "2019-05-31 00:00:00 W20L 467 @ 268.1"
## [1] "2019-08-02 00:00:00 W20L -467 @ 275.7"
## [1] "2019-08-02 00:00:00 W20L -718 @ 275.7"
## [1] "2019-09-16 00:00:00 W20L 718 @ 272.75"
## [1] "2019-09-16 00:00:00 W20L 526 @ 272.75"
## [1] "2019-09-18 00:00:00 W20L -526 @ 272.3"
## [1] "2019-09-18 00:00:00 W20L -556 @ 272.3"
## [1] "2019-09-19 00:00:00 W20L 556 @ 271.45"
## [1] "2019-09-19 00:00:00 W20L 558 @ 271.45"
## [1] "2019-09-20 00:00:00 W20L -558 @ 272"
## [1] "2019-09-20 00:00:00 W20L -598 @ 272"
## [1] "2019-10-17 00:00:00 W20L 598 @ 267"
## [1] "2019-10-17 00:00:00 W20L 569 @ 267"
## [1] "2019-10-21 00:00:00 W20L -569 @ 266.7"
## [1] "2019-10-21 00:00:00 W20L -618 @ 266.7"
## [1] "2019-10-22 00:00:00 W20L 618 @ 271.95"
## [1] "2019-10-22 00:00:00 W20L 615 @ 271.95"
## [1] "2019-11-15 00:00:00 W20L -615 @ 275.8"
## [1] "2019-11-15 00:00:00 W20L -750 @ 275.8"
## [1] "2019-11-19 00:00:00 W20L 750 @ 277.25"
## [1] "2019-11-19 00:00:00 W20L 786 @ 277.25"
## [1] "2019-11-21 00:00:00 W20L -786 @ 271.25"
## [1] "2019-11-21 00:00:00 W20L -699 @ 271.25"
## [1] "2019-11-25 00:00:00 W20L 699 @ 272"
## [1] "2019-11-25 00:00:00 W20L 708 @ 272"
## [1] "2019-11-29 00:00:00 W20L -708 @ 268.3"
## [1] "2019-11-29 00:00:00 W20L -749 @ 268.3"
```

Czas obliczeń. Przy rozbudowanym portfelu może trwać kilkanaście minut

```
t2 <- Sys.time()
print(t2 - t1)
```

```
## Time difference of 3.006224 secs
```

Ocena wydajności strategii

```
updatePortf(portfolio.st)
```

```
## [1] "Clenow_Simple"
```

```
dateRange <- time(getPortfolio(portfolio.st)$summary)[-1]
updateAcct(portfolio.st, dateRange)
```

```
## [1] "Clenow_Simple"
```

```
updateEndEq(account.st)
```

```
## [1] "Clenow_Simple"
```

```
tStats <- tradeStats(Portfolios = portfolio.st, use = "trades",
                     inclZeroDays = FALSE)
tStats[, 4:ncol(tStats)] <- round(tStats[, 4:ncol(tStats)], 2)

knitr::kable(data.frame(t(tStats[, -c(1, 2)])))
```

	DAX	SPY_1	W20L
Num.Txns	81.00	85.00	137.00
Num.Trades	41.00	43.00	69.00
Net.Trading.PL	-54088.80	94331.97	73430.24

	DAX	SPY_1	W20L
Avg.Trade.PL	-1319.24	2193.77	1064.21
Med.Trade.PL	-1395.56	-906.53	-489.06
Largest.Winner	37470.65	103153.93	33417.16
Largest.Loser	-28953.60	-15480.13	-7837.30
Gross.Profits	65247.74	173169.18	161468.02
Gross.Losses	-119336.54	-78837.21	-88037.78
Std.Dev.Trade.PL	8360.17	17271.29	7918.20
Std.Err.Trade.PL	1305.64	2633.85	953.24
Percent.Positive	21.95	30.23	39.13
Percent.Negative	78.05	69.77	60.87
Profit.Factor	0.55	2.20	1.83
Avg.Win.Trade	7249.75	13320.71	5980.30
Med.Win.Trade	2535.46	2617.08	2207.80
Avg.Losing.Trade	-3729.27	-2627.91	-2096.14
Med.Losing.Trade	-2163.80	-1879.16	-1675.22
Avg.Daily.PL	-1279.50	2155.05	393.64
Med.Daily.PL	-1337.15	-996.77	-506.66
Std.Dev.Daily.PL	8462.75	17478.76	5669.58
Std.Err.Daily.PL	1338.08	2697.03	687.54
Ann.Sharpe	-2.40	1.96	1.10
Max.Drawdown	-132019.30	-52885.79	-46142.19
Profit.To.Max.Draw	-0.41	1.78	1.59
Avg.WinLoss.Ratio	1.94	5.07	2.85
Med.WinLoss.Ratio	1.17	1.39	1.32
Max.Equity	77930.50	137649.30	96574.34
Min.Equity	-54088.80	-8308.43	-23935.13
End.Equity	-54088.80	94331.97	73430.24

```
aggPF <- sum(tStats$Gross.Profits) / -sum(tStats$Gross.Losses)
aggCorrect <- mean(tStats$Percent.Positive)
numTrades <- sum(tStats$Num.Trades)
meanAvgWLR <- mean(tStats$Avg.WinLoss.Ratio[tStats$Avg.WinLoss.Ratio < Inf], na.rm = TRUE)
aggEnd.Equity <- sum(tStats$End.Equity) + 100000

table_1 <- data.frame(Items = c("współczynnik zysku/straty ", "średnia z zyskowych",
                                "ilość transakcji", "średnia wartość zysk/strata",
                                "wartość końcowa portfela"),
                      Features = c(aggPF, aggCorrect, numTrades,
                                   meanAvgWLR, aggEnd.Equity))
knitr::kable(table_1, digits = 2)
```

Items	Features
współczynnik zysku/straty	1.40
średnia z zyskowych	30.44
ilość transakcji	153.00
średnia wartość zysk/strata	3.29

Items	Features
wartość końcowa portfela	213673.41

Uśrednianie stopy zwrotu z aktywów portfela oraz konwertowanie na format `xts`

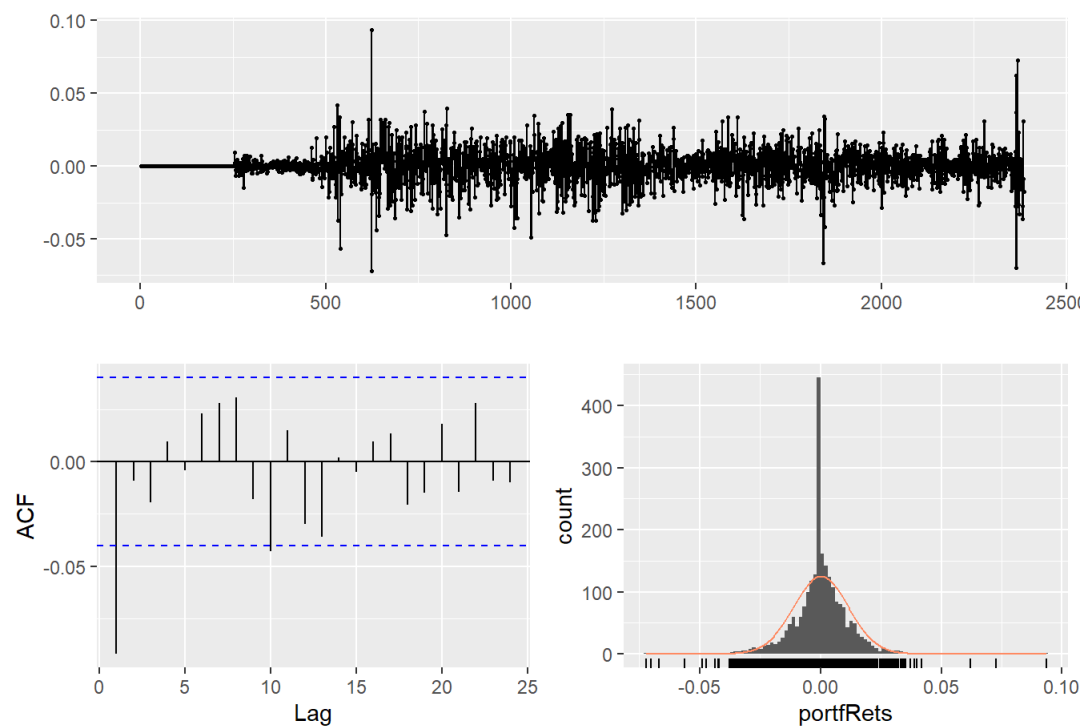
```
instRets <- PortfReturns(account.st)
portfRets <- xts(rowMeans(instRets) * ncol(instRets),
                 order.by = index(instRets))
portfRets <- portfRets[!is.na(portfRets)]
```

Skumulowana stopa zwrotu oraz określenie pierwszego dnia, w którym wartość  $> 0$  w celu porównania w dalszych analizach z wykresem odniesienia

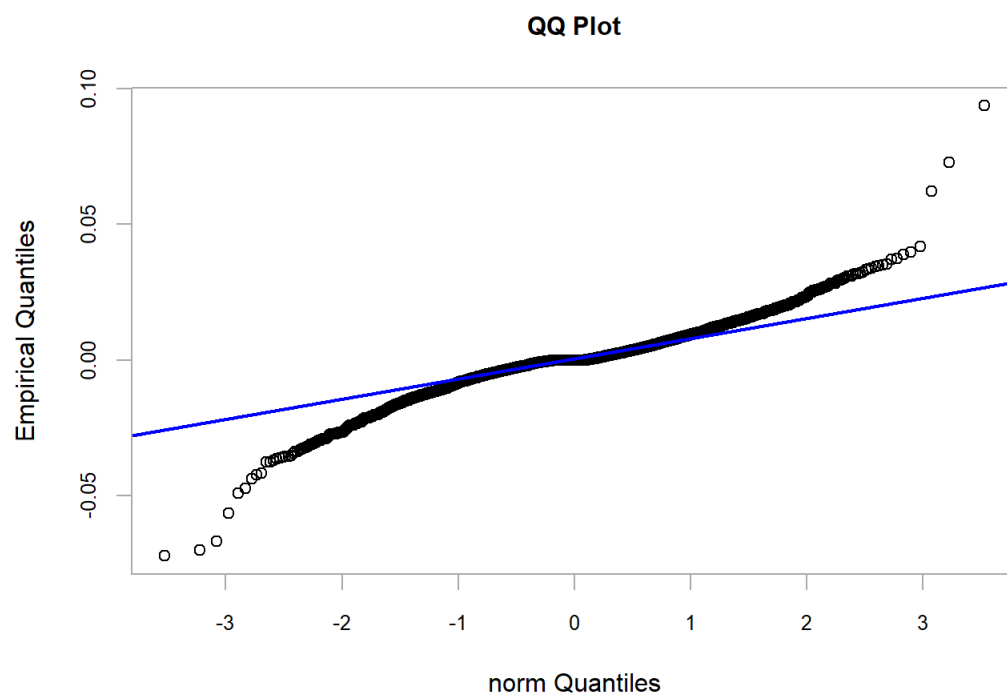
```
cumPortfRets <- cumprod(1 + portfRets)
firstNonZeroDay <- as.character(index(portfRets)[
  min(which(portfRets != 0))])
```

wykresy zlogarytmowanych stóp zwrotu portfela: histogram, wykres reszt, autokorelacji

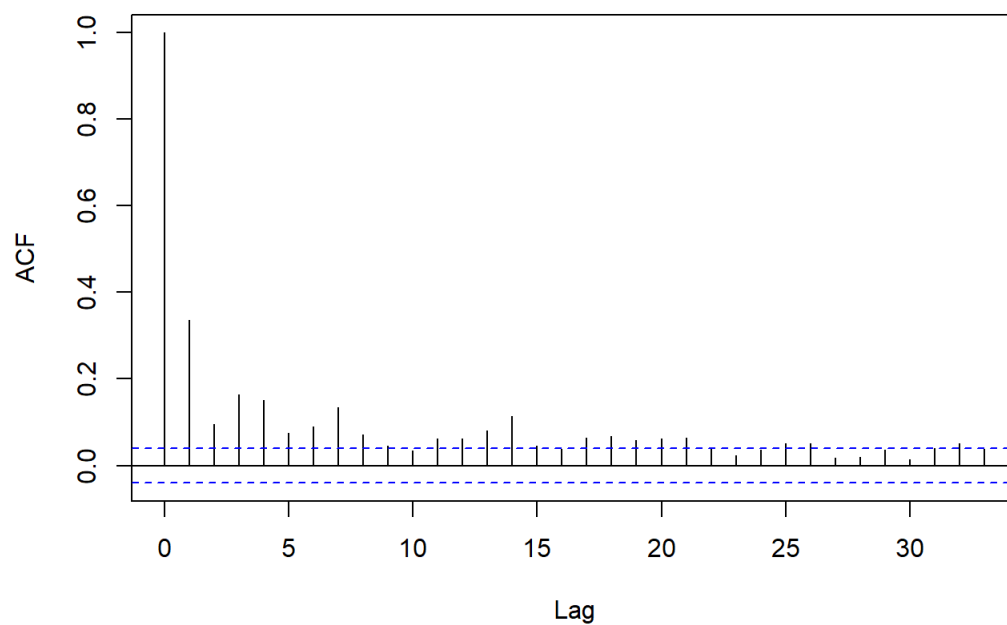
```
forecast::ggtsdisplay(portfRets, plot.type = "histogram", lag.max = 24)
```



```
chart.QQPlot(portfRets, distribution = "norm")
```



```
acf(portfRets^2, main = "")
```



#Model GARCH

Dopasowania modelu

```
library(fGarch)
```

```
## Loading required package: timeDate
```

```
##  
## Attaching package: 'timeDate'
```

```
## The following objects are masked from 'package:PerformanceAnalytics':  
##  
##      kurtosis, skewness
```

```
## Loading required package: timeSeries
```

```
##  
## Attaching package: 'timeSeries'
```

```
## The following object is masked from 'package:zoo':  
##  
##      time<-
```

```
## Loading required package: fBasics
```

```
##  
## Attaching package: 'fBasics'
```

```
## The following object is masked from 'package:TTR':  
##  
##      volatility
```

```
fit_4 <- garchFit(~ 1 + garch(1 , 1), data = portfRets, trace = FALSE) #dla rozkładu normalnego  
summary(fit_4)
```

```
##
## Title:
##   GARCH Modelling
##
## Call:
##   garchFit(formula = ~1 + garch(1, 1), data = portfRets, trace = FALSE)
##
## Mean and Variance Equation:
##   data ~ 1 + garch(1, 1)
##   <environment: 0x0000000019619100>
##   [data = portfRets]
##
## Conditional Distribution:
##   norm
##
## Coefficient(s):
##           mu      omega      alphas      betas
## 5.0427e-05  5.7658e-08  1.8349e-01  8.5317e-01
##
## Std. Errors:
##   based on Hessian
##
## Error Analysis:
##           Estimate Std. Error t value Pr(>|t|)
## mu      5.043e-05  4.042e-05   1.248 0.212156
## omega  5.766e-08  1.752e-08   3.291 0.000999 ***
## alphas 1.835e-01  1.442e-02  12.724 < 2e-16 ***
## betas  8.532e-01  9.013e-03  94.659 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##   8175.347      normalized:  3.427819
##
## Description:
##   Tue Jun 16 20:14:32 2020 by user: AdamMatuszczyk
##
##
## Standardised Residuals Tests:
##
##           Statistic p-Value
## Jarque-Bera Test   R      Chi^2 49276.37 0
## Shapiro-Wilk Test  R      W      0.9098188 0
## Ljung-Box Test     R      Q(10) 16.32085 0.09080871
## Ljung-Box Test     R      Q(15) 22.31279 0.09986071
## Ljung-Box Test     R      Q(20) 25.68114 0.176603
## Ljung-Box Test     R^2 Q(10) 0.8640526 0.9999123
## Ljung-Box Test     R^2 Q(15) 1.186824 0.9999992
## Ljung-Box Test     R^2 Q(20) 1.633192 1
## LM Arch Test       R      TR^2 1.146844 0.9999697
##
## Information Criterion Statistics:
##           AIC      BIC      SIC      HQIC
## -6.852283 -6.842594 -6.852288 -6.848757
```

```
fit_5 <- garchFit(~ 1 + garch(1, 1), data= portfRets, trace= FALSE, cond.dist="std") # dla warunku rozkladu
t-studenta
summary(fit_5)
```

```
##
## Title:
##   GARCH Modelling
##
## Call:
##   garchFit(formula = ~1 + garch(1, 1), data = portfRets, cond.dist = "std",
##     trace = FALSE)
##
## Mean and Variance Equation:
##   data ~ 1 + garch(1, 1)
##   <environment: 0x000000001df419c8>
##   [data = portfRets]
##
## Conditional Distribution:
##   std
##
## Coefficient(s):
##           mu      omega      alphal      betal      shape
## 1.6718e-09 1.2973e-10 1.0000e+00 6.8253e-01 2.4086e+00
##
## Std. Errors:
##   based on Hessian
##
## Error Analysis:
##           Estimate Std. Error t value Pr(>|t|)
## mu      1.672e-09   8.504e-07   0.002   0.998
## omega  1.297e-10   6.276e-09   0.021   0.984
## alphal 1.000e+00   9.489e-02  10.538 <2e-16 ***
## betal  6.825e-01   1.744e-02  39.146 <2e-16 ***
## shape  2.409e+00   6.607e-02  36.453 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##   9247.903      normalized:  3.877527
##
## Description:
##   Tue Jun 16 20:14:32 2020 by user: AdamMatuszczyk
##
## Standardised Residuals Tests:
##
##           Statistic  p-Value
## Jarque-Bera Test   R    Chi^2 552084385 0
## Shapiro-Wilk Test  R     W    0.0211072 0
## Ljung-Box Test     R    Q(10) 0.04146748 1
## Ljung-Box Test     R    Q(15) 0.08549899 1
## Ljung-Box Test     R    Q(20) 0.09442017 1
## Ljung-Box Test     R^2  Q(10) 0.004293759 1
## Ljung-Box Test     R^2  Q(15) 0.006448868 1
## Ljung-Box Test     R^2  Q(20) 0.008634696 1
## LM Arch Test       R     TR^2 0.00514782 1
##
## Information Criterion Statistics:
##           AIC      BIC      SIC      HQIC
## -7.750862 -7.738751 -7.750871 -7.746455
```

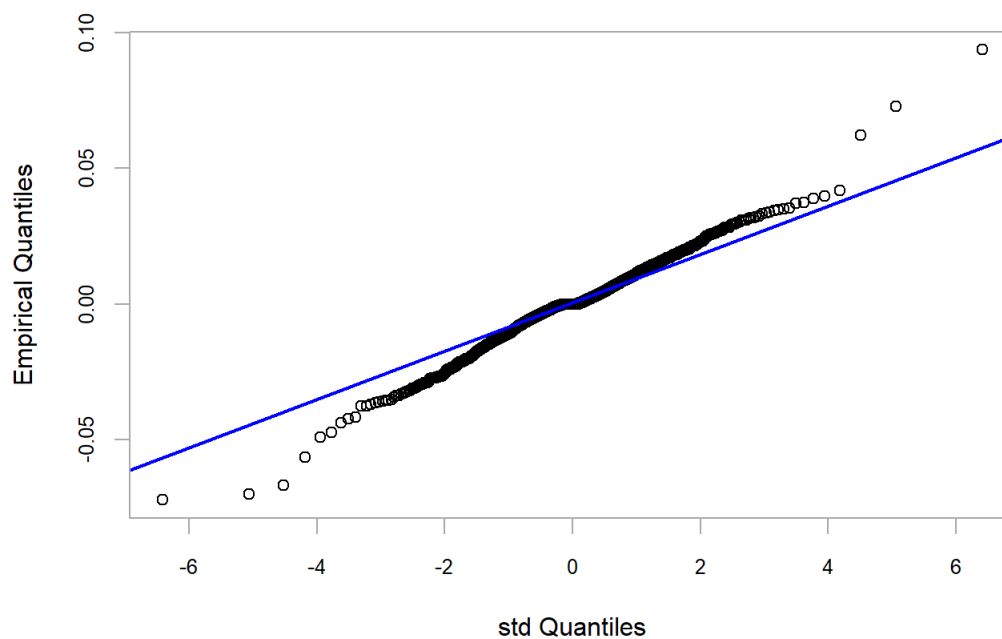
```
fit_6 <- garchFit(~ 1 + garch(1 , 0),
                  data = portfRets, trace = FALSE, cond.dist = "sstd") # dla warunku rozkładu skośnego t-st
udenta
summary(fit_6)
```



```
##
## Title:
##   GARCH Modelling
##
## Call:
##   garchFit(formula = ~1 + garch(1, 0), data = portfRets, cond.dist = "sstd",
##     trace = FALSE)
##
## Mean and Variance Equation:
##   data ~ 1 + garch(1, 0)
## <environment: 0x000000001d0d3808>
##   [data = portfRets]
##
## Conditional Distribution:
##   sstd
##
## Coefficient(s):
##           mu      omega      alphal      skew      shape
## 0.00027316 0.00007629 0.99999999 1.00034873 2.89950247
##
## Std. Errors:
##   based on Hessian
##
## Error Analysis:
##           Estimate Std. Error t value Pr(>|t|)
## mu      2.732e-04  1.737e-04   1.572   0.116
## omega   7.629e-05  8.680e-06   8.789 < 2e-16 ***
## alphal  1.000e+00  1.530e-01   6.537 6.27e-11 ***
## skew    1.000e+00  2.099e-02  47.662 < 2e-16 ***
## shape   2.900e+00  1.681e-01  17.250 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##   7662.46      normalized:  3.212772
##
## Description:
##   Tue Jun 16 20:14:33 2020 by user: AdamMatuszczyk
##
## Standardised Residuals Tests:
##
##           Statistic p-Value
## Jarque-Bera Test   R   Chi^2  9760.041  0
## Shapiro-Wilk Test  R   W      0.9226927  0
## Ljung-Box Test     R   Q(10)  11.87432  0.2935559
## Ljung-Box Test     R   Q(15)  18.60493  0.2321977
## Ljung-Box Test     R   Q(20)  24.06147  0.2397172
## Ljung-Box Test     R^2 Q(10)  40.62395  1.315154e-05
## Ljung-Box Test     R^2 Q(15)  76.85594  2.611682e-10
## Ljung-Box Test     R^2 Q(20)  84.24465  7.386517e-10
## LM Arch Test       R   TR^2   43.50328  1.855119e-05
##
## Information Criterion Statistics:
##           AIC      BIC      SIC      HQIC
## -6.421350 -6.409239 -6.421359 -6.416943
```

Wykres reszt po uwzględnieniu poprawki wynikającej z z uwzględnienia warunku t-studenta

```
chart.QQPlot(portfRets, distribution = "std", main = "")
```

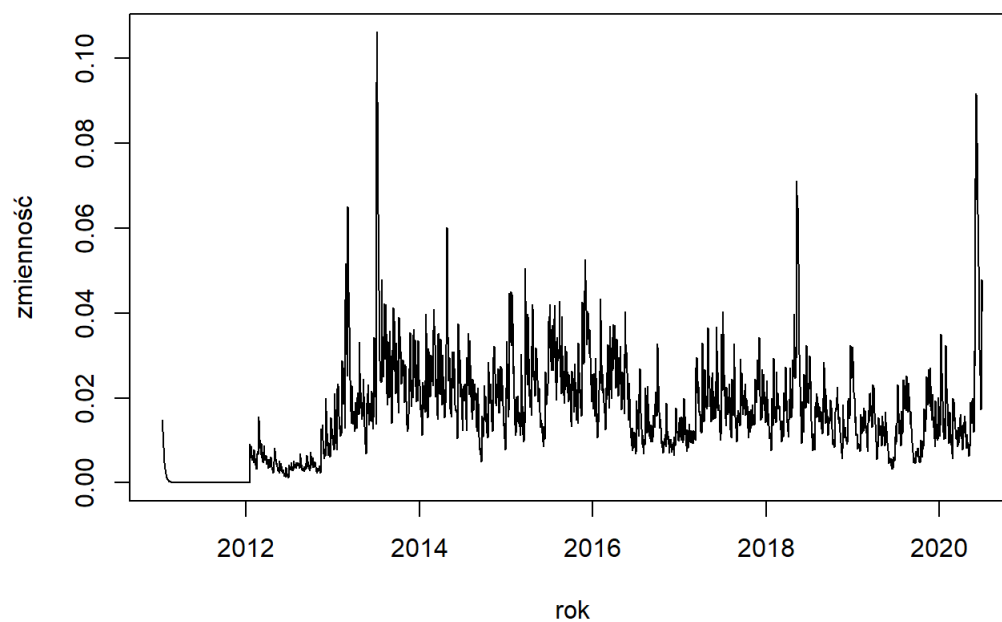


Zmienność dla modelu (fit\_5) uwzględniającego poprawkę wynikającą z warunku t-studenta modelu GARCH(1,1)

```
vol_5 <- volatility(fit_5) #zmienność dla modelu fit_5
res_fit_5 <- residuals(fit_5, standardize = TRUE) #standaryzowanie reszt
vol <- ts(vol_5, frequency = 252, start = c(2011, 10))
res <- ts(res_fit_5, frequency = 252, start = c(2011, 10))
```

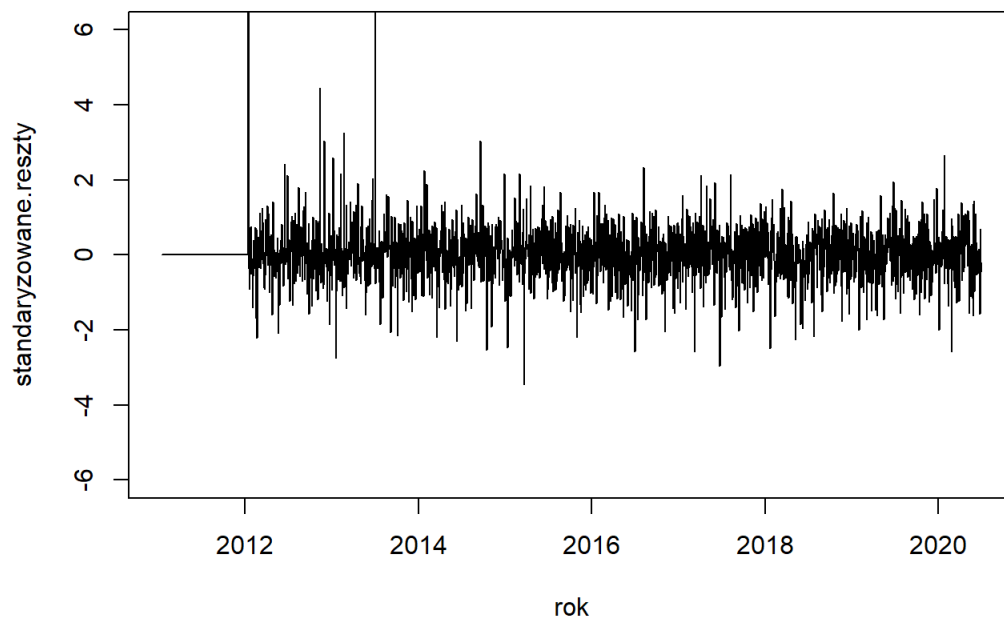
Wykres zmienności zlogarytmowanych stóp zwrotu w badanym przedziale czasowym

```
plot(vol, xlab = "rok", ylab = "zmienność", type = "l")
```



wykres zestandaryzowanych reszt po uwzględnieniu warunku t-studenta modelu GARCH(1,1)

```
plot(res, xlab = "rok", ylab = "standaryzowane.reszty", type = "l", ylim = c(-6, 6))
```

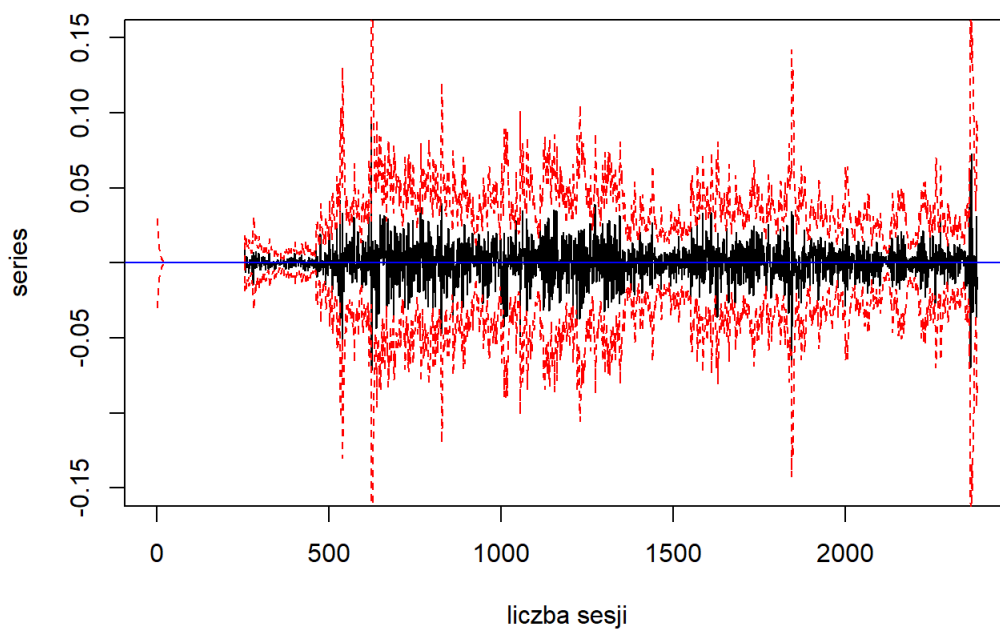


Predykcja ze zmiennością w tle jako przedział zaufania w 95%

```

upp <- 1.672e-09 + 2 * vol_5
low <- 1.672e-09 - 2 * vol_5
tdx <- c(1:2385)
plot(tdx, portfRets, xlab = "liczba sesji", ylab = "series", type = "l", ylim = c(-0.15, 0.15))
lines(tdx, upp, lty = 2, col = "red")
lines(tdx, low, lty = 2, col = "red")
abline(h = c(1.672e-09), col = "blue")

```



#Pobieranie danych S&P500 ze strony www.yahoo.com

```
getSymbols("SPY", from = firstNonZeroDay, to = to)
```

```
## [1] "SPY"
```

zlogarytmowanie oraz zmiana na format xts, zlogarytmowanie i przeliczenie na skumulowaną stopę zwrotu

```
SPYrets <- diff(log(Cl(SPY)))[-1]
cumSPYrets <- cumprod(1 + SPYrets)
```

Połączenie portfela wraz z porównywanymi danymi *cumSPYrets*

```
comparison <- cbind(cumPortfRets, cumSPYrets)
colnames(comparison) <- c("strategy", "SPY")
```

Wykres porównania strategii oraz S&P500

```
chart.TimeSeries(na.omit(comparison), legend.loc = "topleft",
  element.color =c("green", "red"))
```



Podsumowanie portfela

```
SharpeRatio.annualized(portfRets)
```

```
##                                [,1]
## Annualized Sharpe Ratio (Rf=0%) 0.1325025
```

```
Return.annualized(portfRets)
```

```
##                                [,1]
## Annualized Return 0.02395746
```

```
maxDrawdown(portfRets)
```

```
## [1] 0.3284446
```

```
table_3 <- data.frame(Items = c("Wskaźnik Sharpe annualizowany", "Roczna stopa annualizowana", "maxSpadek ka  
pitału"),
  Features = c(SharpeRatio.annualized(portfRets) * 100, Return.annualized(portfRets) *  
100, maxDrawdown(portfRets) *100))
knitr::kable(table_3, digits = 2)
```

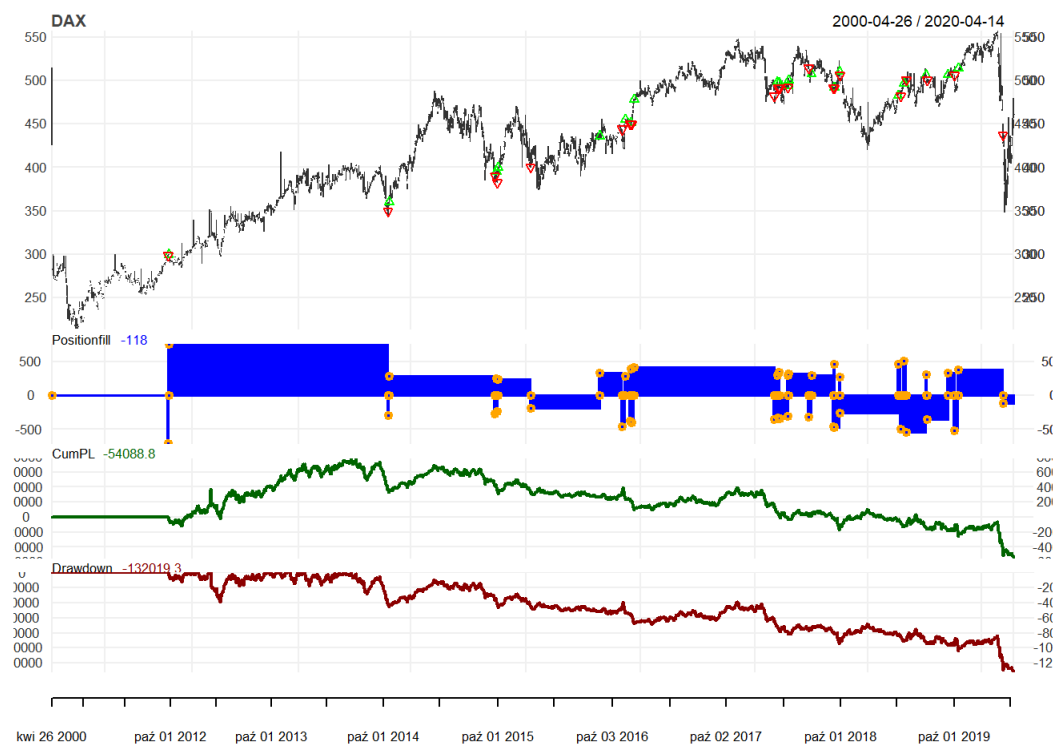
Items

Features

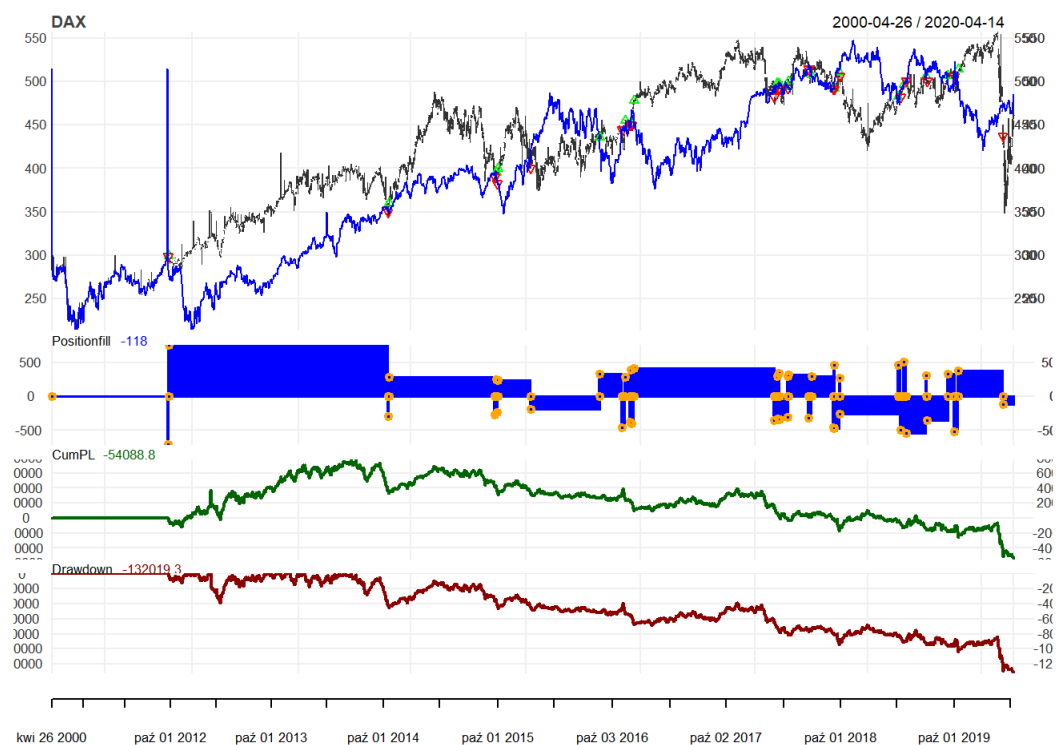
Items	Features
Wskaźnik Sharpe annualizowany	13.25
Roczna stopa annualizowana	2.40
maxSpadek kapitału	32.84

#Wykres dla ETF\_DAX

```
chart.Posn(portfolio.st, "DAX")
```

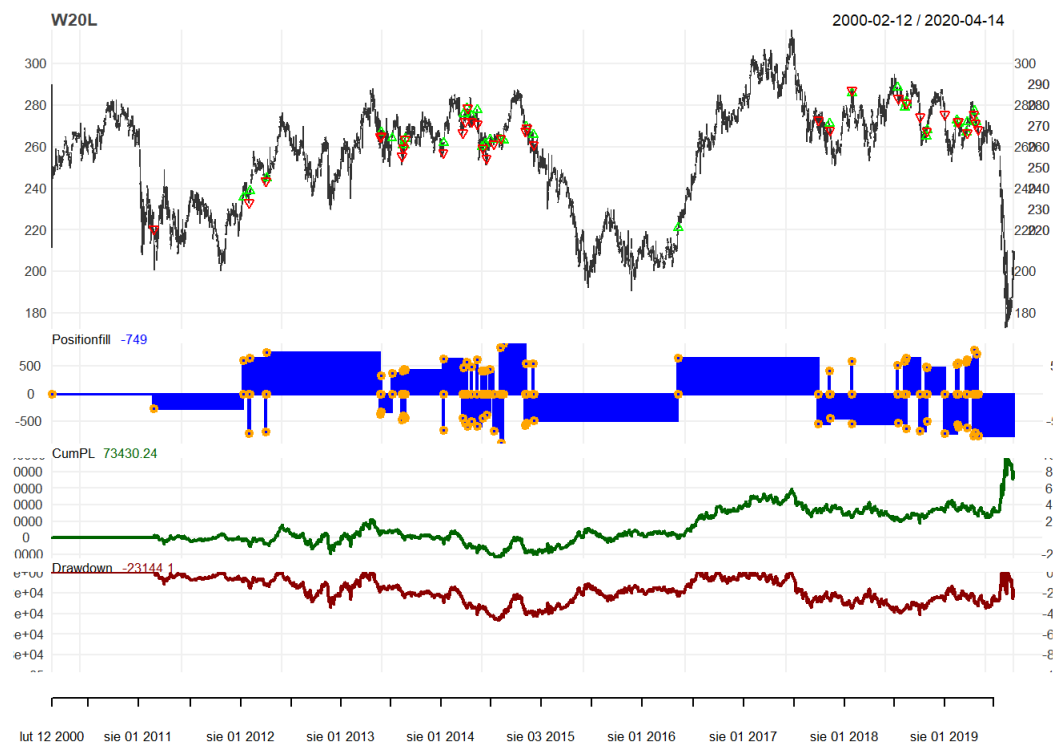


```
tmp <- namedLag(Cl(DAX), k = nLag)
add_TA(tmp$namedLag, col = "blue", on = 1)
```

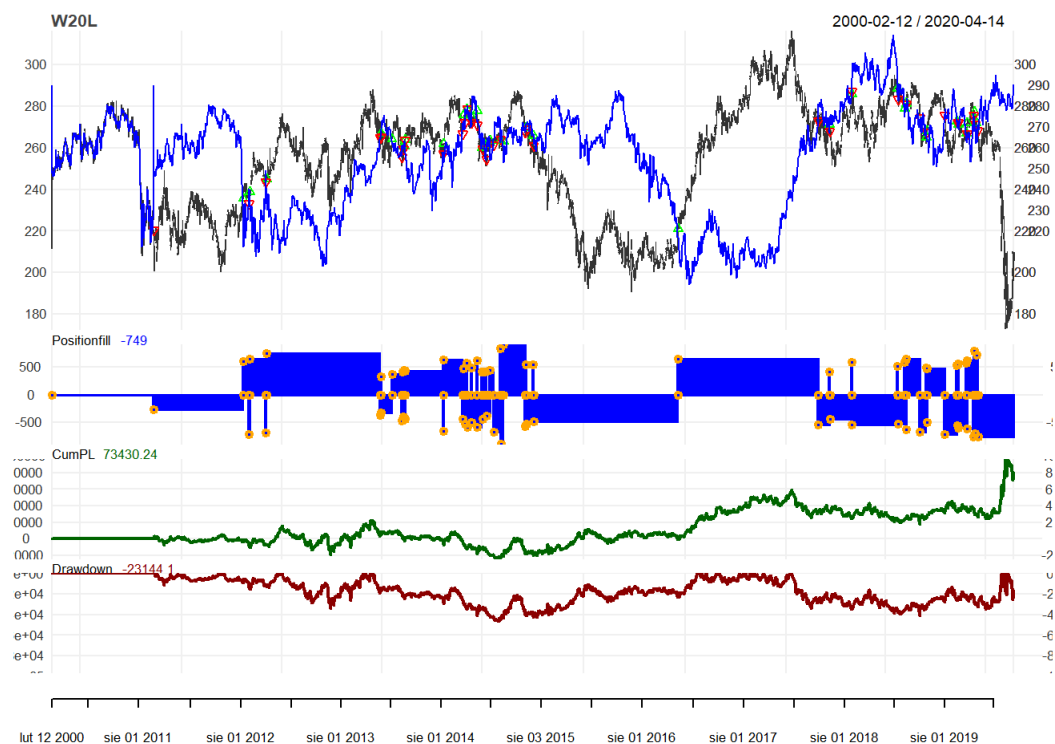


#Wykres dla ETF\_WIG20

```
chart.Posn(portfolio.st, "W20L")
```

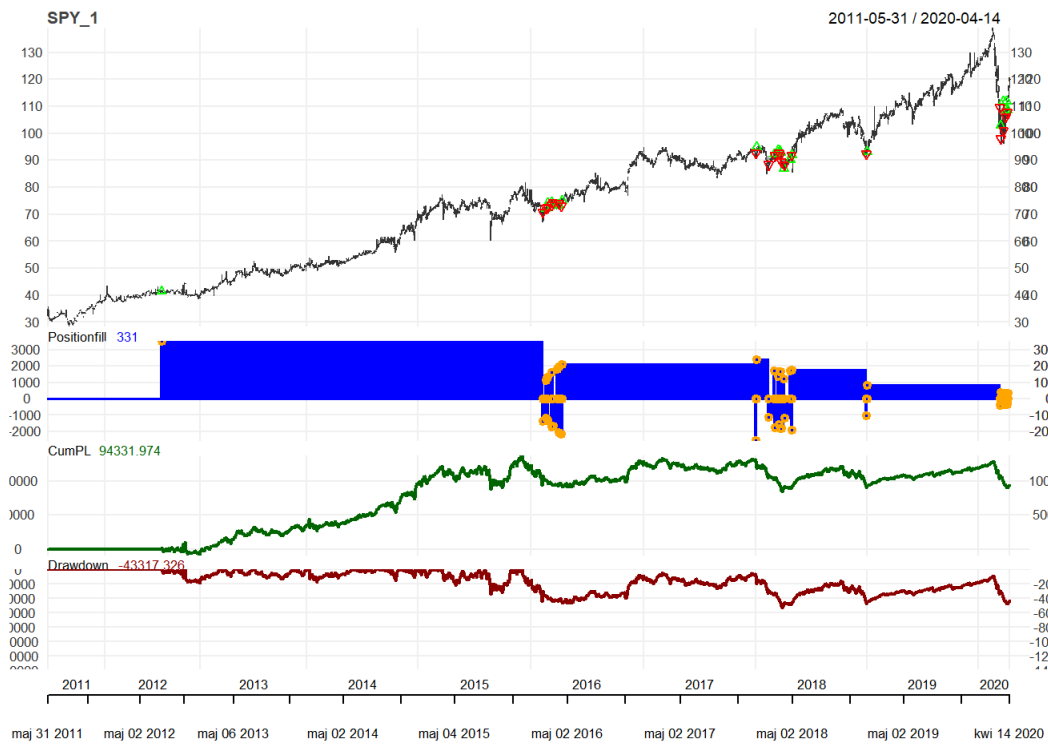


```
tmp <- namedLag(Cl(W20L), k = nLag)
add_TA(tmp$namedLag, col = "blue", on = 1)
```



#Wykres dla ETF\_S&P500

```
chart.Posn(portfolio.st, "SPY_1")
```



```
tmp <- namedLag(Cl(SPY_1), k = nLag)
add_TA(tmp$namedLag, col = "blue", on = 1)
```

