Predicting Ford stock VaR

Artem Kulik

Stock VaR introduction

Nowadays in fast changing world it’s very important to know how to make money, but most of all it’s important to know how to save them. Stock market is a great opportunity to implement this, but is also a big opportunity to lose all. Specifically, for the reason of this computing VaR for the company stock is really relevant issue.

So, to begin with, what is VaR?

Value at risk (VaR) is a measure of the risk of loss for investments. It estimates how much a set of investments might lose (with a given probability), given normal market conditions, in a set time period such as a day.

In other words, it’s the biggest loss per day with certain probability.

Data set

stock_FORD = get_yahoo_data("F", from = "2016-01-01", to = "2020-01-01")

stock_GAZPROM = get_yahoo_data("OGZPY" , from = "2016-01-01", to = "2020-01-01")

stock_UBS = get_yahoo_data("UBS" , from = "2016-01-01", to = "2020-01-01")

Data was downloaded from Yahoo Finance and contains data from 2016 to 2020 years for 3 different companies: Ford, Gazprom and UBS.

Previous Close	6.26	Market Cap	23.393B
Open	6.06	Beta (5Y Monthly)	1.16
Bid	5.95 x 40700	PE Ratio (TTM)	590.00
Ask	5.95 x 36900	EPS (TTM)	0.01
Day's Range	5.88 - 6.11	Earnings Date	Apr 27, 2020
52 Week Range	5.80 - 10.56	Forward Dividend & Yield	0.60 (9.58%)
Volume	93,087,811	Ex-Dividend Date	Jan 28, 2020
Avg. Volume	64,144,942	1y Target Est	9.36

Data contains 3 datasets with 7 features each:

Date <fctr>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Adj.Close <dbl>	Volume <int>
2016-01-04	18.920	19.140	18.835	19.13	14.92769	1936800
2016-01-05	19.100	19.130	18.920	19.02	14.84185	1966300
2016-01-06	18.740	18.875	18.700	18.78	14.65457	1808000
2016-01-07	18.280	18.470	18.240	18.30	14.28001	2340000
2016-01-08	18.030	18.040	17.350	17.37	13.55431	2856900
2016-01-11	17.700	17.730	17.423	17.58	13.71818	2492100
2016-01-12	17.660	17.760	17.470	17.67	13.78841	2064300
2016-01-13	17.700	17.750	17.150	17.28	13.48408	2929700
2016-01-14	17.210	17.520	17.020	17.42	13.59333	2424700
2016-01-15	16.700	16.940	16.529	16.62	12.96906	2312700

Packages

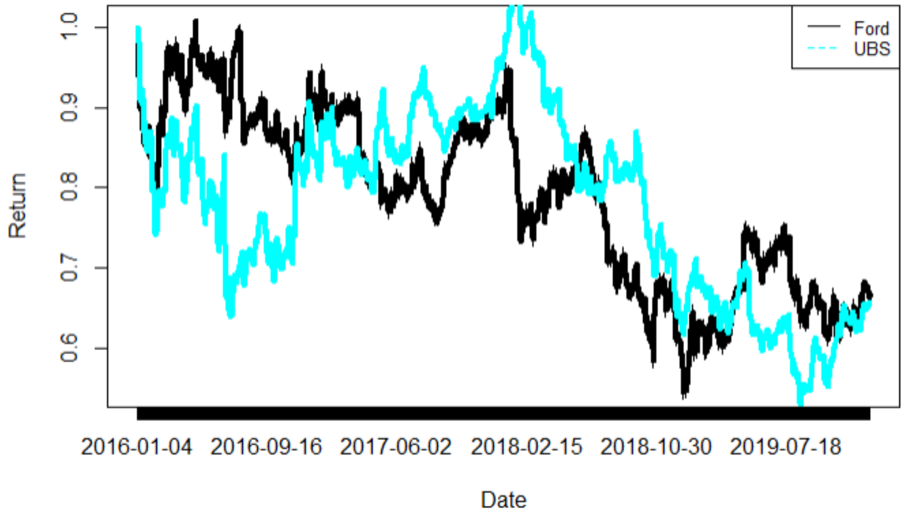
```
library(datasets)
library(evd)
library( QuantTools ) # Market data loading
library( data.table ) # Library for data management
library( ghyp )      # GHP
library( copula )     # Copula
library( fGarch )     # GARCH
library( evd )       # Extreme Value
library(Hmisc)
library(corrplot)
library(pastecs)
library(plotrix)
```

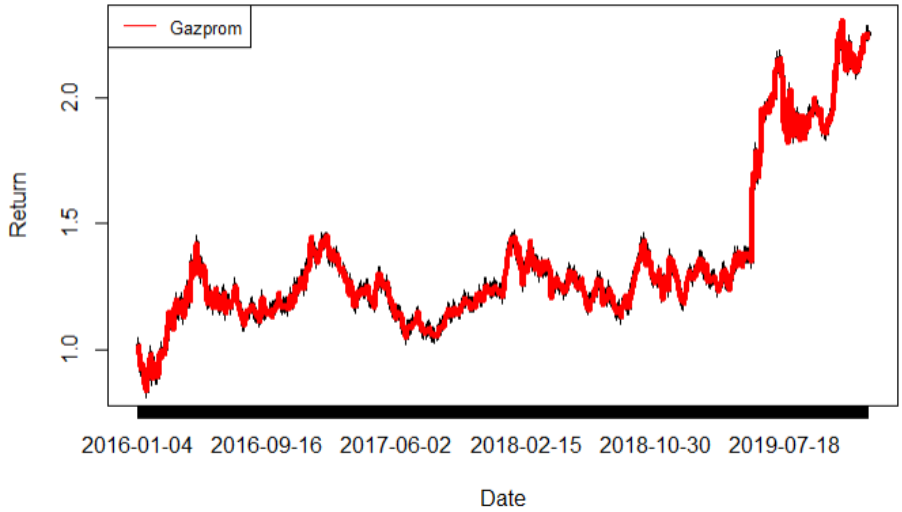
Return computing

```
stock_FORD$return_FORD = stock_FORD$Close / shift(stock_FORD$Close, fill =
stock_FORD$Close[1]) – 1
```

Originally return may be computed as: $\frac{(close_t - close_{t-1})}{close_{t-1}}$

Here are cumulative functions of return for all (3) companies:



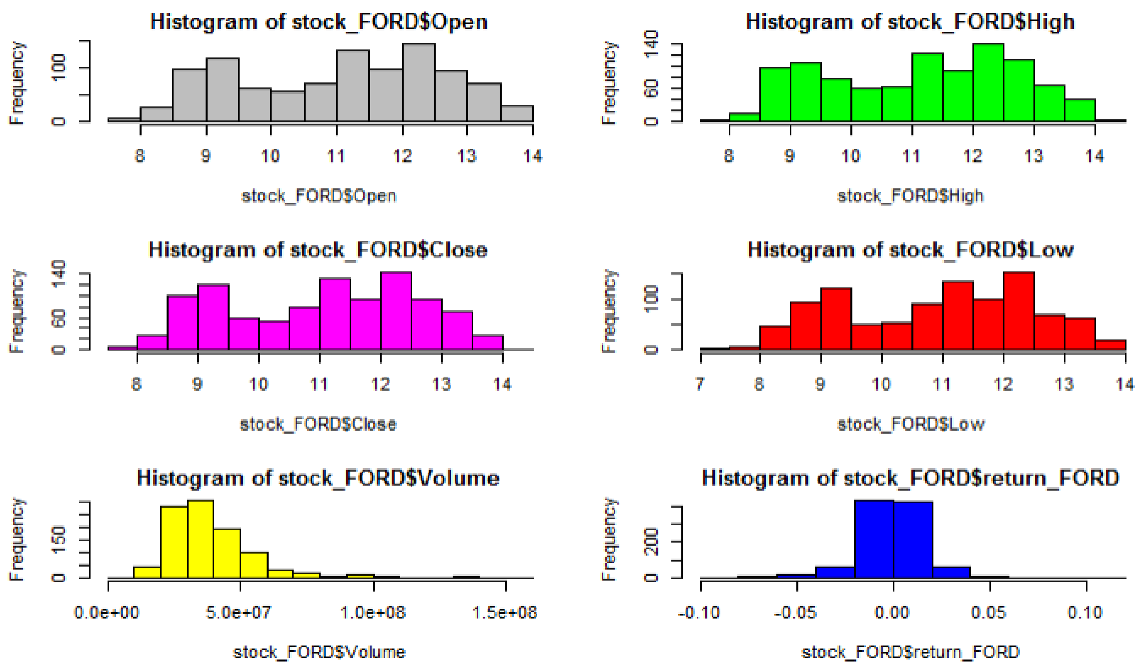


Feature properties

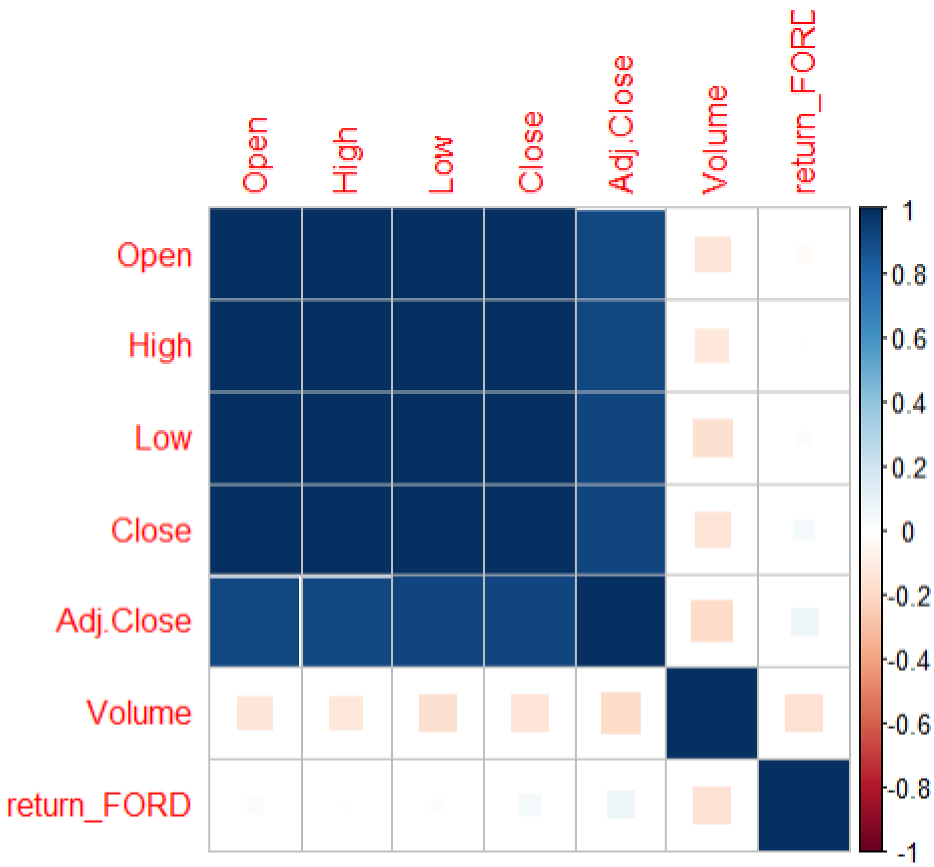
Through the feature distribution histogram it can be clearly seen that return looks similarly like normal distribution but with heavier tails.

Also visual observation gives us the information that first four features are distributed with similar (same) distribution model.

Feature distribution:



After looking through correlation map our hypothesis was proved.



First features are strongly correlated.

VaR computing and modelling

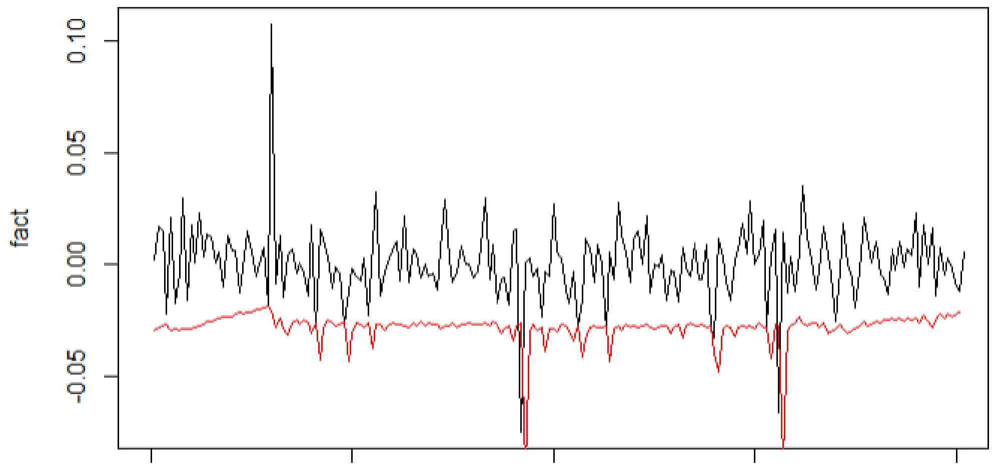
```
return_FORD.gfit <- garchFit( formula=~arma(1,1) + aparch(1,1), data=return_FORD, delta=2, include.delta=FALSE, leverage=TRUE, cond.dist="sged", shape=1.25, include.shape=FALSE, trace=FALSE )

return_FORD.frc <- predict( return_FORD.gfit, n.ahead=5 )

alpha <- 0.05

h <-260  #horizon

VaR <- return_FORD.frc[1, 1]+return_FORD.frc[1, 3] * qsged( alpha, mean=0, sd=1, nu=1.5, xi=return_FORD.gfit@fit$par["skew"] )

sprintf("VaR : %f", VaR)

#plotting VaR curve and checking its quality

VaR <- c()

for ( i in (T_train + 1):T ) {

  h.return_FORD <- return_FORD[(i-h):(i-1)]

  return_FORD.gfit <- garchFit( formula=~arma(1,1) + aparch(1,1), data=h.return_FORD, delta=2, include.delta=FALSE, leverage=TRUE, cond.dist="sged", shape=1.5, include.shape=FALSE, trace=FALSE )

  return_FORD.frc <- predict( return_FORD.gfit, n.ahead=1 )

  VaR[i-T_train] <- return_FORD.frc[1,1] + return_FORD.frc[1,3] * qsged( alpha, mean=0, sd=1, nu=1.5, xi=return_FORD.gfit@fit$par["skew"] )

}

fact <- return_FORD[T_train+1:T]

plot( fact, xlim=c(0, 200), type="l" )

lines( VaR, col="red" )

kupic_test( fact, VaR, T_test, alpha )
```

Data was fragmented on Train and Test (8 / 2) and also horizon was chosen as 260.

It means that for each 260 values of var from train data we get one new value with ARMA-GARCH model and then believe that this new value is train data for next step.

Also some results were computed from other methods like hypothesis that our VaR has GHYP distribution.

ARMA-GARCH



GHYP distribution