LONDON BUSINESS SCHOOL

CA01 B AUT20 - Investment Fundamentals

Final group assignment

Forecasting EUR/USD returns with ARIMA-GARCH model

Active trading to outperform the buy-and-hold strategy

Group 28

Darren Ho, Lexi Su, Lucy Qu, Muslim Dashaev, Olivier Bildstein, Tarek Auf

---

Professor: **M. Roberto Cram**

Summary

Introduction

When asked to think about a hedge fund strategy, our group instantly thought of linking our first two classes from the MFA program: Investment Fundamentals and Data Analytics for Finance. We chose to go with a quantitative approach - using two models - ARIMA[1] and GARCH[2] - to forecast the returns of the most famous foreign exchange pair EUR/USD. Forecasting FX prices is of great interest for investors of course, but also for various other players in the bigger picture such as farmers, policy makers, speculators, and even governments.

But forecasting the future using past data, is this even possible? Doesn't the SEC (Security Exchange Commission) require - via rule 156 - funds to tell investors not to base their expectations of future results on past performances?

After finding evidence in academic literature (cf. 1.1.), we believe forecasting future returns is possible to a certain extent. Through this assignment, we will show the economic rationale of the strategy and then test it using a large dataset processed by an R algorithm. In the end, we will present our findings, explain the limits of our model and how it could be improved.

" Future forecasting is all about testing strategies - it's like a wind tunnel."

- Jamais Cascio

[1] ARIMA stands for Autoregressive Integrated Moving Average
[2] GARCH stands for Generalized Autoregressive Conditional Heteroskedasticity

1. Economic rationale

1.1. Relevant evidence from academic literature

According to Tran Mong Uyen Ngan (2016)[3], there are numbers of theoretical frameworks showing that to a certain extent foreign exchange rates can be predicted. They all use different methods like the neutral networks by Verkooijen (1996)[4], the ARIMA model by Tseng (2001)[5] / Znaczko (2013)[6], the Least Squared model by Hongxing (2007)[7] or the Purchasing Power Parity model and Balassa-Samuelson channel by David (2010)[8].

Out of these researches, we decided to choose the ARIMA model, because it directly enables us to develop an algorithm with our Data Analytics for Finance skills and test our assumptions. Additionally the GARCH function will be incorporated to the ARIMA model to have a more forward looking strategy on predicting volatility.

1.2. Investment strategy

The investment strategy is quite simple: extrapolate a pattern from past data that will enable us to predict future returns. To do so, we will use two functions in R: Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). They provide a statistical analysis model that uses time series

[3] Tran Mong Uyen Ngan (2016), "Forecasting Foreign Exchange Rate by using ARIMA Model: A Case of VND/USD Exchange Rate". In: Research Journal of Finance and Accounting 7.12, pp. 38-44
[4] Verkooijen. (1996). A neutral network approach to long-run exchange rate prediction. Computat.Economic, Vol 9
[5] Tseng, F.-M. (2001). Fuzzy ARIMA model for forecasting the foreign exchange market. Taiwan: Fuzzy Sets and Systems, Vol.118
[6] Znaczko, T. M. (2013). Forecasting Foreign Exchange Rates. Applied Economics Theses, Vol.4
[7] Hongxing, L., Zhaoben, F., & Dongming, Z. (2007). GBP/USD Currency Exchange Rate Time Series Forecasting Using Regularized Least-Squares Regression Model. Proceeding of the World Congress on Engineering, Vol 2
[8] David, H., Lee, J., & Takizawa, H. (2010). In which exchange rate models do forecasters trust? : IMF Working Paper

data and helps to predict future trends. Using our skills learned in the Data Analytics for Finance course, we developed an algorithm in R (cf. code lines) that predicts a next day's return by analyzing data from the last 1000 days.

Depending on the outcome of the algorithm, the investment strategy will be either to go long or to short the FX rate: if the index analyzed is predicted to increase, there will be a buy - if supposed to decrease, a sale.

We will run this algorithm on a 12-year time frame to have a clear view on what our returns can be. We will then benchmark this to the regular EUR/USD one to see if we were able to outperform a typical buy-and-hold strategy.

1.3. Details on the ARIMA function

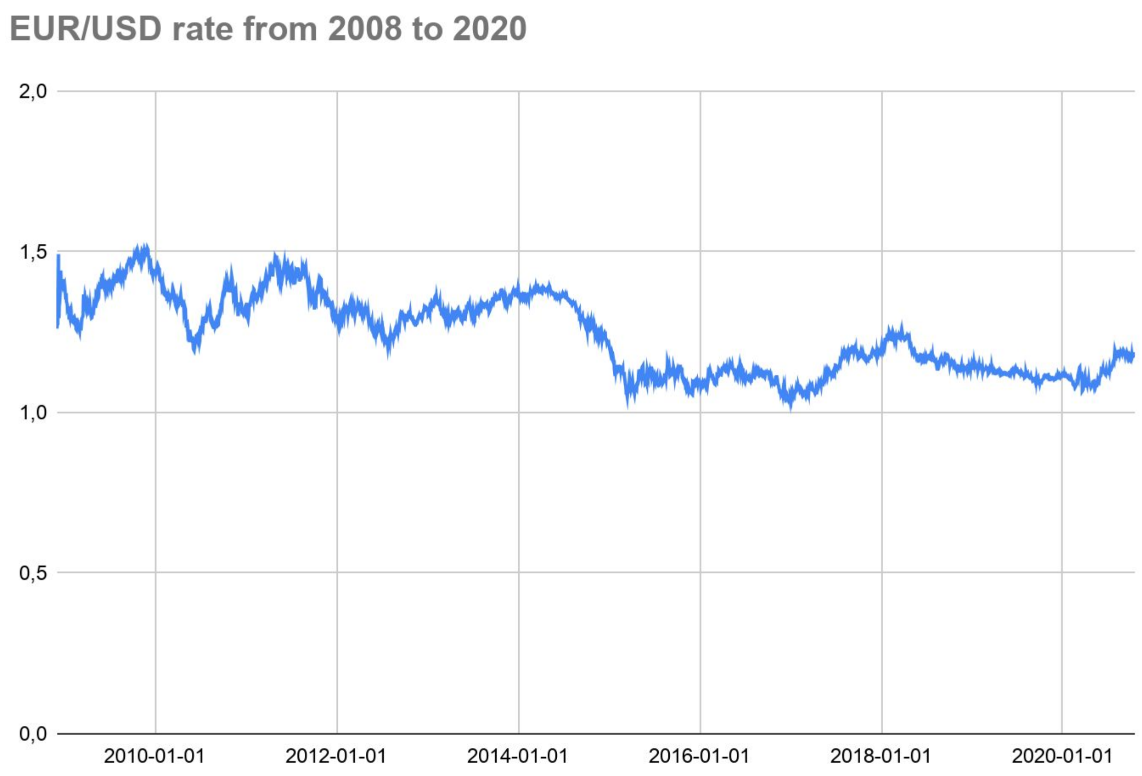
As explained by Meyler (1998)[9], the ARIMA model is a method to forecast time series assuming that past values of the series plus previous error terms contain information for the purpose of forecasting. One of its advantages is that it only requires data on time series in question. The downside is that it is backward looking and therefore poor at predicting turning points (economical cycles, crisis, etc.).

[9] Meyler, A., Kenny, G., & Quinn, T. (1998). Forecasting Irish Inflation Using Arima Model (Vol. Vol.1998, pp. 1-48): Ireland: Central Bank and Financial Services

2. Data section

In order to reduce company and industry specific fluctuations (like the price of oil or new regulations in healthcare), we decided to use a popular foreign exchange rate: the EUR/USD. Indeed, it is one of the most commonly followed rates, gives a wide market overview and tends to be less affected by volatility.

As a timeline, we used twenty years of data from the 10th of December 2008 to the 16th of October 2020. This makes sense because it is spread across various economic cycles and includes a couple of crises (subprime crisis, european debt crisis etc). In fact, through this analysis, we do not simply want to show when our strategy forecasts correctly the markets but also when it is challenged - in order to have exhaustive findings.



Using larger sets of data would have made it challenging for our student computers to process the code. In fact, it already took us about 6 hours to run the code with 12-years worth of information with the 1000 rolling window. So, for each subset of 1000 days that were used to predict the next day's return we calculated 25 ARIMA models and selected the one with the lowest AIC. In total, there were 4361 data subsets with 1000 rolling windows and 109025 ARIMA models were constructed.

To extract the data, we downloaded the EUR/USD closing prices on Yahoo finance. Our code then directly picked the data from the Excel spreadsheet.

3. Trading Approach

3.1 Overview Strategy

The method used here is a rolling window regression for 1000 days. We use the data from Yahoo finance to fit an ARIMA+GARCH model. We forecast the trend of the "next" *n* day differenced and logarithmic return, using the last, *k* (1000) days to fit our model (rolling window). After forecasting the returns, we get a price indication, whether the price will be higher, lower or equal to the last observed price. After that we can either short, if forecast is lower than observed, long if forecast higher than observed or hold/do nothing if forecast equals observed. In this analysis we do not take transaction costs into account, hence when applying this strategy, it could lead to lower returns than here computed.

In the final step, we benchmark our strategy with a very easy, naive and often used model, the buy and hold strategy for the investment horizon. Finally, we visualise both returns, ours and those of the buy-and-hold strategy.

In short, the steps to follow are:

1. For each day, *n*, the previous *k* days of the differenced logarithmic returns of a FX rate are used as a window for fitting an optimal ARIMA and GARCH model.
2. The combined model is used to make a prediction for the next day's return.
3. If the prediction is negative the FX rate is shorted at the previous close, while if it is positive it is longed.
4. If the prediction is the same direction as the previous day, then nothing is changed - if the prediction is different, then the position changes from long to short or short to long depending on the case.

3.2. Model Specification and Fitting

First of all, we run the Augmented Dickey-Fuller test on the daily rates of EUR/USD. We observe that p-value = 0.01 so we do not reject the null hypothesis of non-stationarity. Thus, we take logarithms of each observation and then find the first difference. As a result, we have stationary data series.

In our model we have used *k=1000* and downloaded the data from Yahoo finance. For example, first we consider the period 1-1000 and predict the returns for the day 1001, then we take period 2-1001 and predict for the day 1002, and so on. We also do not take into account transaction costs and slippage (where computer signals to enter/exit, and where the actual entry/exit was). We use these 1000 days of pricing data to forecast the price of the next day. And we do that over and over, while rolling our window forward (hence rolling window forecast) for each forecast we look simultaneously for the best ARIMA model with the lowest AIC (Akaike Information Criterion), our integration value *d* is equal to zero (since we have already taken the difference between logarithms), hence we are actually fitting ARMA and not ARIMA models. After observing the right ARIMA model, we can feed this into the next step, finding the right GARCH model.

Once this model specification step is completed, we can start to fit the ARIMA+GARCH model. R function *ugarchfit()* helps us here a lot as it optimizes the specification. In case the GARCH does not converge, we cannot say with certainty when the prediction is going to happen and need to guess the "long" entry point to the market. But if the model converges, then we have the date and the prediction. So far we have computed our forecasts and entry/exit points. What is now missing, are the returns. We will look at how to compute these in the following lines.

3.3. Calculating the Returns

In order to see what our strategy returns are, we need to calculate the intersection between both datasets, our forecasts and the real data, according to the date (date is the matching key). We finally multiply the real returns with our predicted trend (+/- sign) and obtain our results. In the last and final step we visualize and compare our returns with the naive buy-and-hold model, we can see that our model beats the naive model in the analyzed time frame.

4. Main findings

To better interpret our result, we compared the return of our investment strategy (the orange line) with another benchmark strategy, which is buy and hold the EUR/USD currency pairs (the blue line) over a 12-year horizon. From the chart, we could clearly see that while before 2015 the two strategies oscillated in a similar range without distinctive divergence, our strategy embarked on an outperformance over the naive strategy since 2015 March and is expected to maintain this advantage till the end of the 10-year investment cycle. Numerically speaking, the annual return of our strategic investment strategy peaked at around 25% during 2017 February and is expected to close at around 0% at 2020 end. More interestingly, the return of the investment strategy mirrored that of the naive strategy since the midterm, indicating a negative correlation between the two strategies - whereas the FX trading strategy continuously experienced loss, our investment strategy is expected to harvest a decent return when it draws to the end of the analyzed time frame. Therefore, we believe our model has constituted a worthwhile investment strategy for years ahead.

Conclusion

To summarize, our hedge fund investment principle is quite straightforward - extrapolate a pattern from past data and use them as the main inputs to enable future return prediction. By comparing this strategy with the naive strategy which is to buy and hold the EUR/USD currency pairs, we detected a distinctive arbitrage opportunity using this model which is expected to generate an annualized return as high as 25% during the peak point whereas the downside return is capped at -15%. Therefore, we have high conviction in our model-generated strategy.

Regarding the scope of our research, there still exists some limitations to be further improved. For example, the prediction model is not perfect as we use the empirical price data as the main plugs while don't take into account economic cyclical factors and other maro risk factors. The prediction algorithm could also be further developed with some other tools (like eGARCH) or optimized if by adding some machine learning methodologies. Also, we intentionally restricted parameters of ARMA(p,q) to the maximum level of 5 for each because currently it requires 8 hours of computations. Another suggestion is to consider ARIMA models with higher values of parameters, different sizes of the rolling windows, and larger period of data. We may also further introduce more equity benchmarks for comparison purposes to better visualize the attractiveness of this strategy.

Appendix

Bibliography

Tran Mong Uyen Ngan (2016), “Forecasting Foreign Exchange Rate by using ARIMA Model: A Case of VND/USD Exchange Rate”. In: Research Journal of Finance and Accounting 7.12, pp. 38-44

Verkooijen. (1996). A neutral network approach to long-run exchange rate prediction. Computat.Economic, Vol 9

Tseng, F.-M. (2001). Fuzzy ARIMA model for forecasting the foreign exchange market. Taiwan: Fuzzy Sets and Systems, Vol.118

Znaczko, T. M. (2013). Forecasting Foreign Exchange Rates. Applied Economics Theses, Vol.4

Hongxing, L., Zhaoben, F., & Dongming, Z. (2007). GBP/USD Currency Exchange Rate Time Series Forecasting Using Regularized Least-Squares Regression Model. Proceeding of the World Congress on Engineering, Vol 2

David, H., Lee, J., & Takizawa, H. (2010). In which exchange rate models do forecasters trust? : IMF Working Paper

Meyler, A., Kenny, G., & Quinn, T. (1998). Forecasting Irish Inflation Using Arima Model (Vol. Vol.1998, pp. 1-48): Ireland: Central Bank and Financial Services

Application 1. Code lines

```r
### ARIMA/GARCH trading model
library(quantmod)
library(timeSeries)
library(rugarch)
library(aTSA)
  # get data and initialize objects to hold forecasts
# EURUSD <- read.csv('EURUSD=X', header = T)

getSymbols("^EURUSD=X", from="2003-12-01")
EURUSD <- `EURUSD=X`
EURUSD <- rownames_to_column(as.data.frame(EURUSD), var = "date")
EURUSD[, 1] <- as.Date(as.character(EURUSD[, 1]))

#adf test
aTSA::adf.test(EURUSD[,"EURUSD=X.Close"])

#p-value = 0.01 so we do not reject the null hypothesis of x at 1% sign level

returns <- diff(log(EURUSD[,"EURUSD=X.Close"])) #calc difference in logs of eurusd data
returns <- na.omit(returns)
# forecasts_eurusd <- forecasts

rolling.window <- 1000
forecast_length <- length(returns) - rolling.window
forecasts <- vector(mode="numeric", length=forecast_length)

# loop through every trading day, estimate optimal model parameters from rolling window
# and predict next day's return

```

```
for (i in 0:forecast_length) {
  roll.returns <- returns[(1+i):(rolling.window + i)] # create rolling window
  final.aic <- Inf
  final.order <- c(0,0,0)

  # estimate optimal ARIMA model order

  for (p in 0:5) for (q in 0:5) { # limit possible order to p,q <= 5

    if (p == 0 && q == 0) next # p and q can't both be zero
    fitted_arima <- tryCatch( arima(roll.returns, order = c(p,0,q)),
                              error = function( err ) FALSE,
                              warning = function( err ) FALSE )

    if (!is.logical( fitted_arima)) {
      current.aic <- AIC(fitted_arima)
      if (current.aic < final.aic) { # retain order if AIC is reduced
        final.aic <- current.aic
        final.order <- c(p,0,q)
        final.arima <- arima(roll.returns, order = (final.order))
      }
    }
    else next
  }

  spec = ugarchspec(variance.model = list(garchOrder=c(1,1)),
                    mean.model = list(armaOrder = c(arimaorder(final.arima)[1],
arimaorder(final.arima)[3]),
                                      include.mean = T),
                    distribution.model = "sged")

  fit = tryCatch(ugarchfit(spec, roll.returns, solver = 'hybrid'),
```

```
                  error = function(e) e,
                  warning = function(w) w)

  # calculate next day prediction from fitted mode
  # model does not always converge - assign value of 0 to prediction and p.val
in this case
  if (is(fit, "warning")) {
    forecasts[i+1] <- 0
    print(0)
    # p.val[i+1] <- 0
  }

  else {

    next.day.fore = ugarchforecast(fit, n.ahead = 1)
    x = next.day.fore@forecast$seriesFor
    forecasts[i+1] <- x[1] # actual value of forecast
    print(forecasts[i])

      }
  #understand what % completed
  print(paste(i, round(i/length(forecasts)*100, 2), sep = ","))
}

#in order to save time import the result of the loop
forecasts <- read.csv("~/eur_usd_forecast.csv", dec=",")
forecasts$X <- NULL #delete unnecessary variable
colnames(forecasts) <- c("forecast_returns")

dates <- EURUSD[, 1]
forecasts.ts <- xts(forecasts, dates[(rolling.window):length(returns)])


```

```r
# create lagged series of forecasts and sign of forecast

lag.forecast <- Lag(forecasts.ts, 1)
#if lag of forecast >0 then 1, <0 -1, =0 also 0
sign_forecast <- ifelse(lag.forecast > 0, 1,
                  ifelse(lag.forecast < 0, -1, 0))

# Create the ARIMA/GARCH returns for the directional system
sign_forecast.returns <- sign_forecast *
returns[(rolling.window):length(returns)]
sign_forecast.returns[1] <- 0 # remove NA

# Create the backtests for ARIMA/GARCH and Buy & Hold
cum_returns <- cumsum( sign_forecast.returns)
buy.hold.ts <- xts(returns[(rolling.window):length(returns)],
                   dates[(rolling.window):length(returns)])
buy.hold.curve <- cumsum(buy.hold.ts)
both.curves <- cbind(cum_returns, buy.hold.curve)
both.curves <- rownames_to_column(as.data.frame(both.curves), var = "date")
names(both.curves) <- c("date", "strategy_returns", "buy_and_hold_returns")

both.curves$date <- as.Date(both.curves$date)

ggplot(data = both.curves, aes(x = date)) +
  geom_line(aes(y = strategy_returns, group = 1, color = "stret")) +
  geom_line(aes(y = buy_and_hold_returns, group = 1, color = "bh")) +
  labs(xlab = "Log returns") +
  scale_color_manual(name = "strategy",
                     values = c("stret" ="darkorange",
                                "bh" = "blue"),
                     labels = c("ARIMA + GARCH strategy",
                                "Buy & Hold strategy")) +
  theme_bw() +
```

```
  theme(legend.position = "bottom",
        axis.text=element_text(size=10)) +
  scale_x_date(date_breaks = "12 months",
               date_labels = "%m-%Y") +
  labs(x = "Date",
       y = "Strategy returns",
       title = "ARIMA+GARCH Strategy outperforms Buy&Hold during the 12-year period")
```

Application 2. Forecast output

The output of the loop:

https://www.dropbox.com/s/bv9fsw4l6njl6l8/eur_usd_forecast.csv?dl=0

If you want to rerun the code please run lines 1-25 and then lines 86-132.