Exercise 2 - AR, ARMA, ARIMA Financial Time Series Modelling (in R)

September 16, 2017

Auto-Regressive *- Lags of the variable itself*
Integrated *- Differencing steps required to make stationary*
Moving Average *- Lags of previous information shocks*

ARIMA(p,d,q)
AR Model *- If only AR terms are there, i.e. ARIMA(1,0,0) = AR (1)*
MA Model *- If only error terms are there, i.e. ARIMA(0,0,1) = MA (1)*
ARMA *- If both are there, i.e. ARIMA(1,0,1) = ARMA(1,1)*
ARIMA *- If differencing term is also included, i.e. ARIMA(1,1,1) = ARMA(1,1) with first differencing*
ARIMAX *- If some exogenous variables are also included.*

AR Modelling of Financial Times Series (BTC/USD ~16 days)

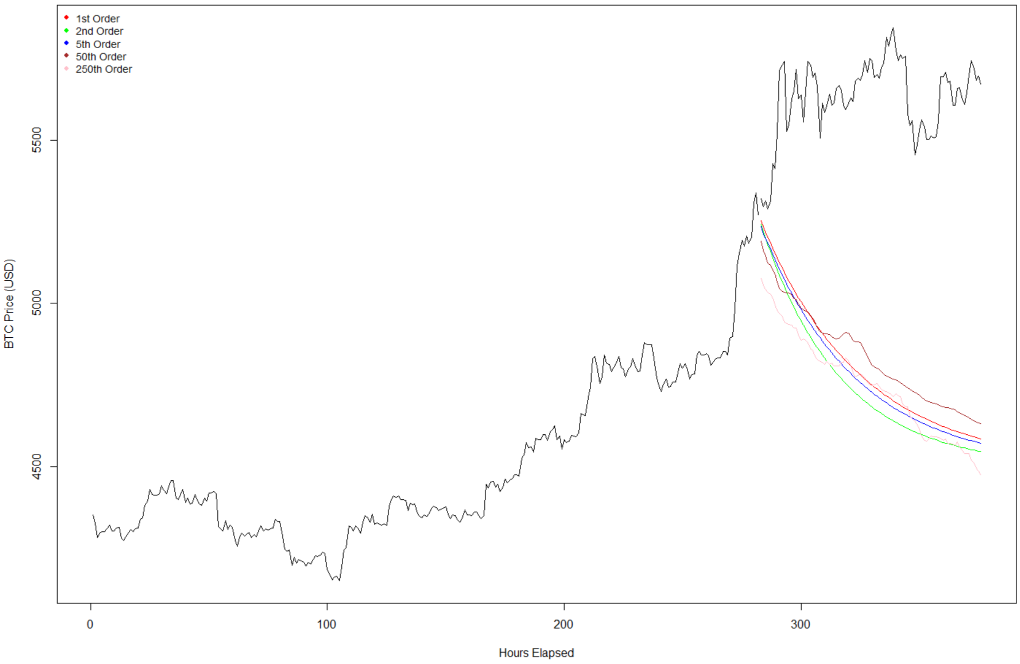

Figure 1.1 - Bitcoin price last 400h (16days) with AR Yule Prediction

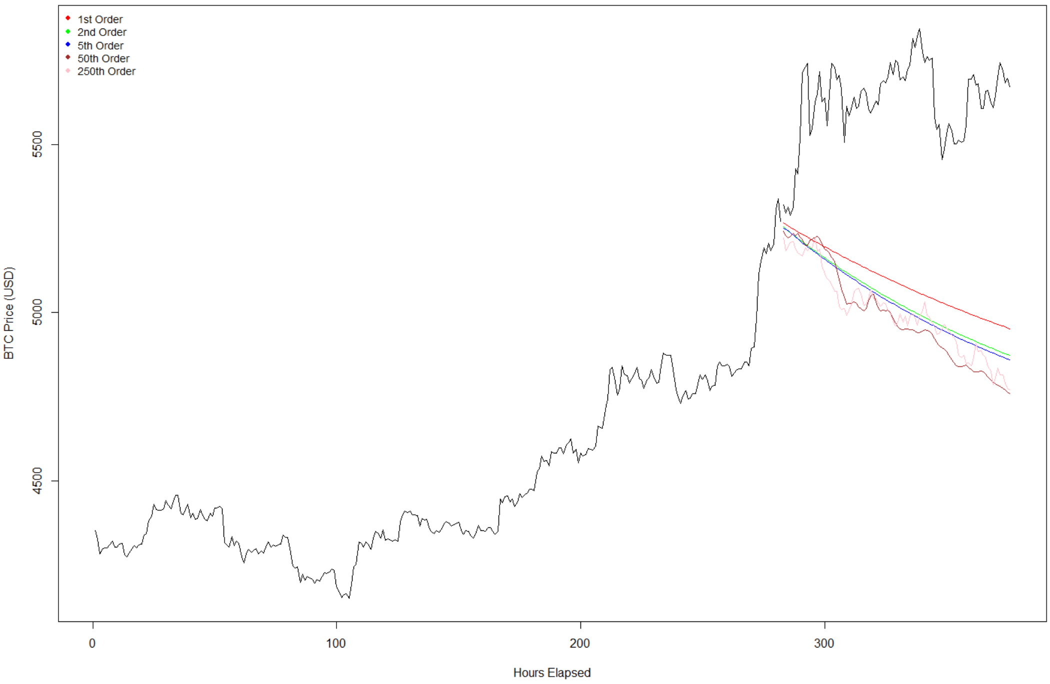

Figure 1.2 - Bitcoin price last 400h (16days) with AR Burg Prediction

ARMA Modelling of Financial Times Series (BTC/USD ~16 days)

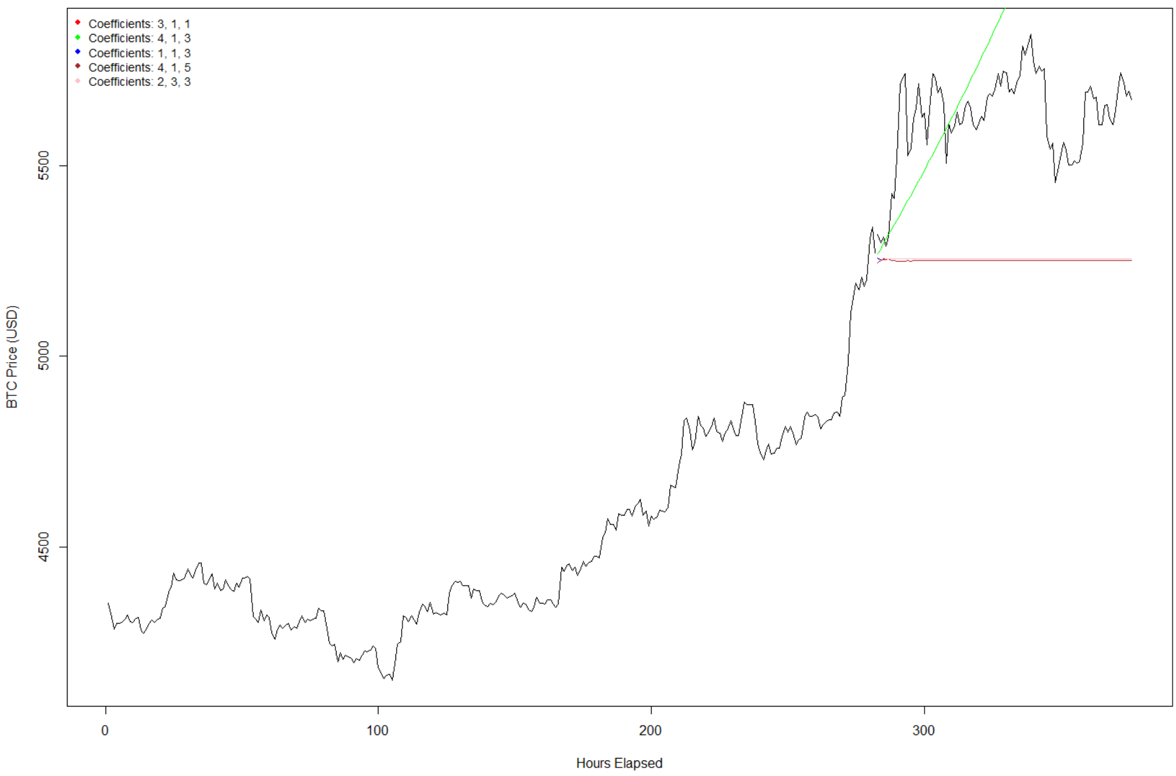

Figure 1.3 - Bitcoin price last 400h (16days) with ARMA Prediction

A common approach in statistics to quantify the goodness of fit test is the AIC (for Akaike Information Criteria) statistic.

ARIMA Modelling of Financial Time Series (BTC/USD ~16 days)

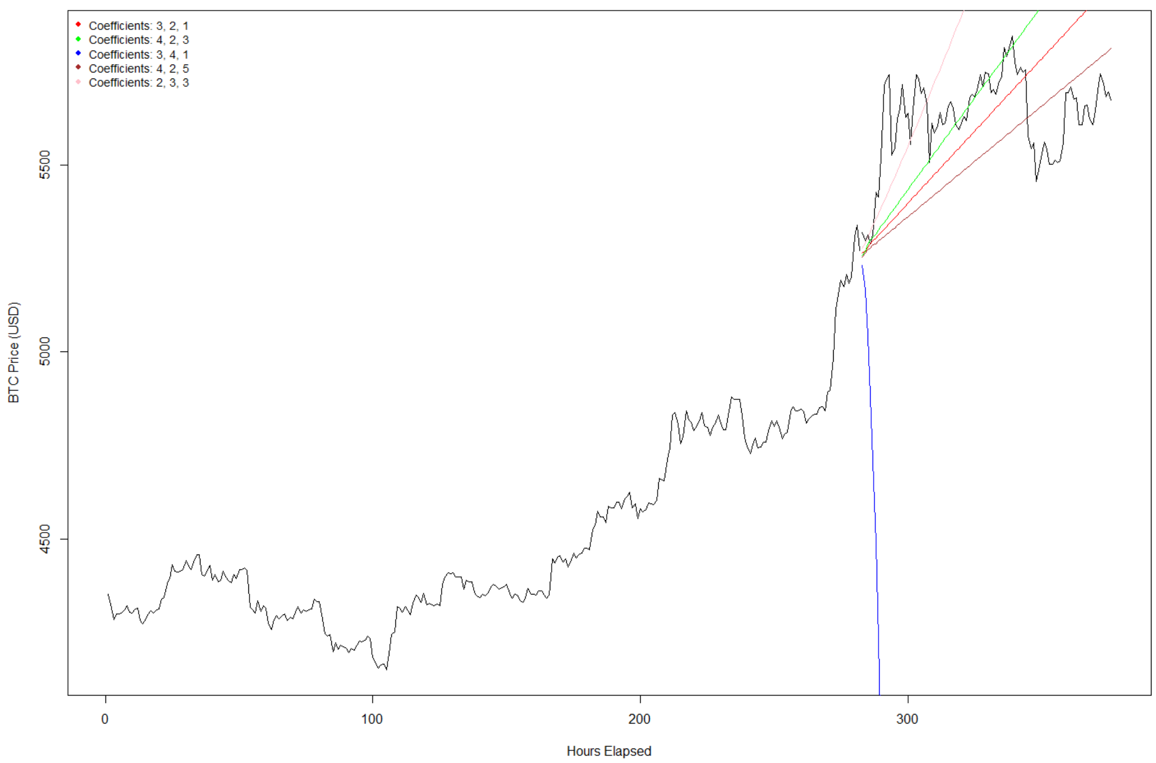

Figure 1.4 - Bitcoin price last 400h (16days) with ARIMA Prediction

Fitting an ARIMA model requires the series to be stationary. A series is said to be stationary when its mean, variance, and autocovariance are time invariant

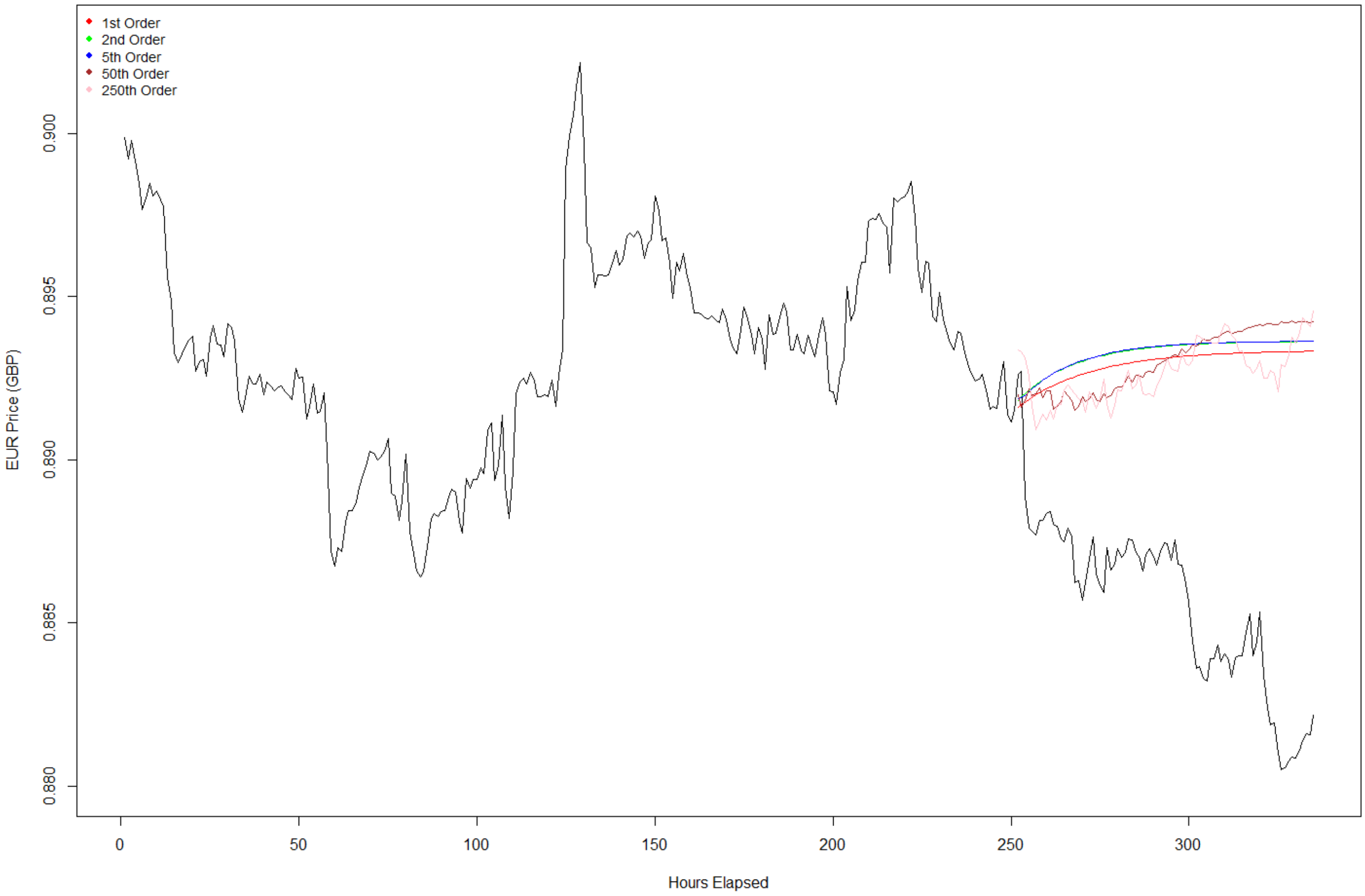

Figure 2.1 - Euro price last 400h (16days) with AR Yule Prediction

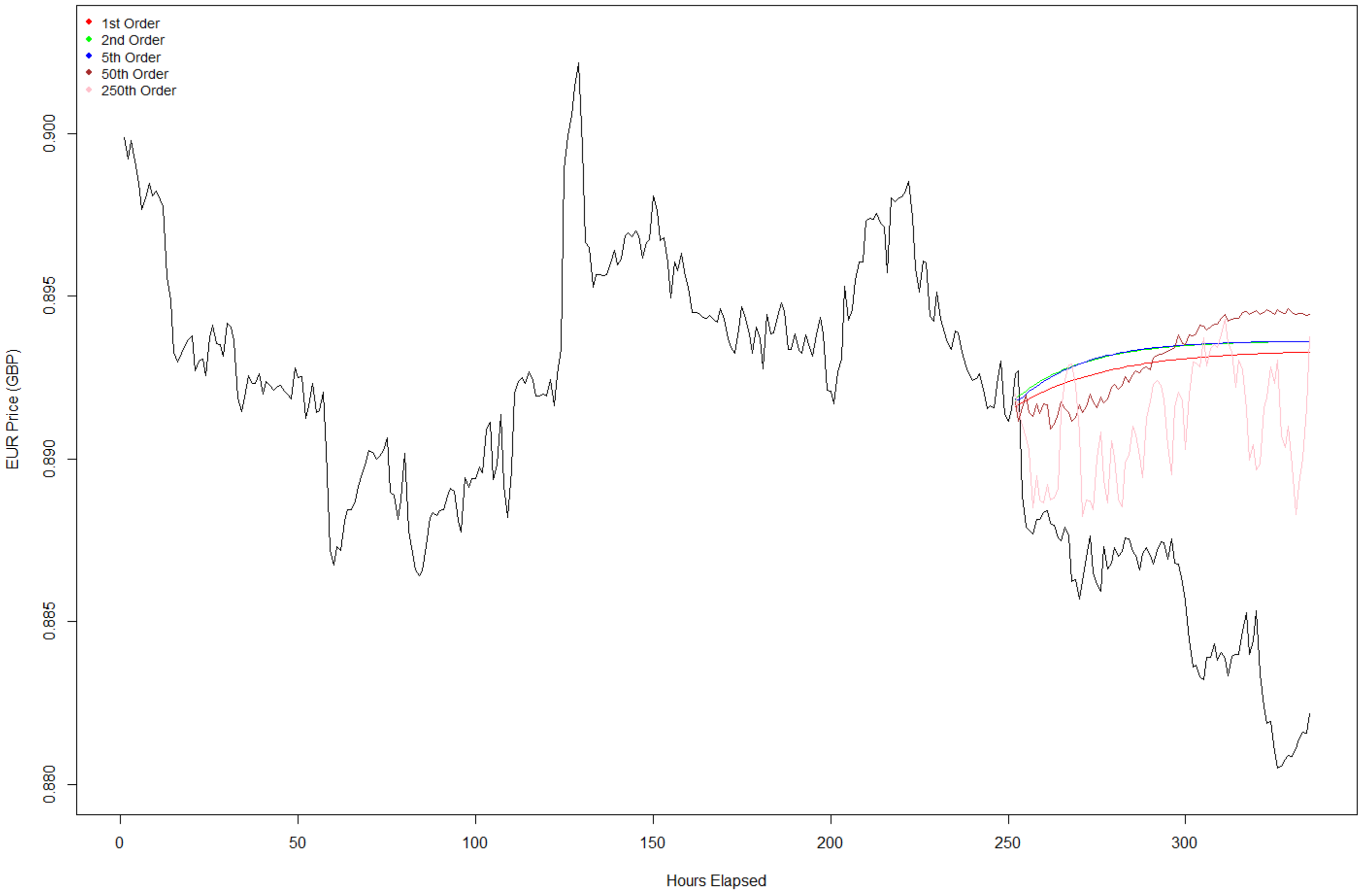

Figure 2.2 - Euro price last 400h (16days) with AR Burg Prediction

ARMA Modelling of Financial Times Series (EUR/GBP ~16 days)

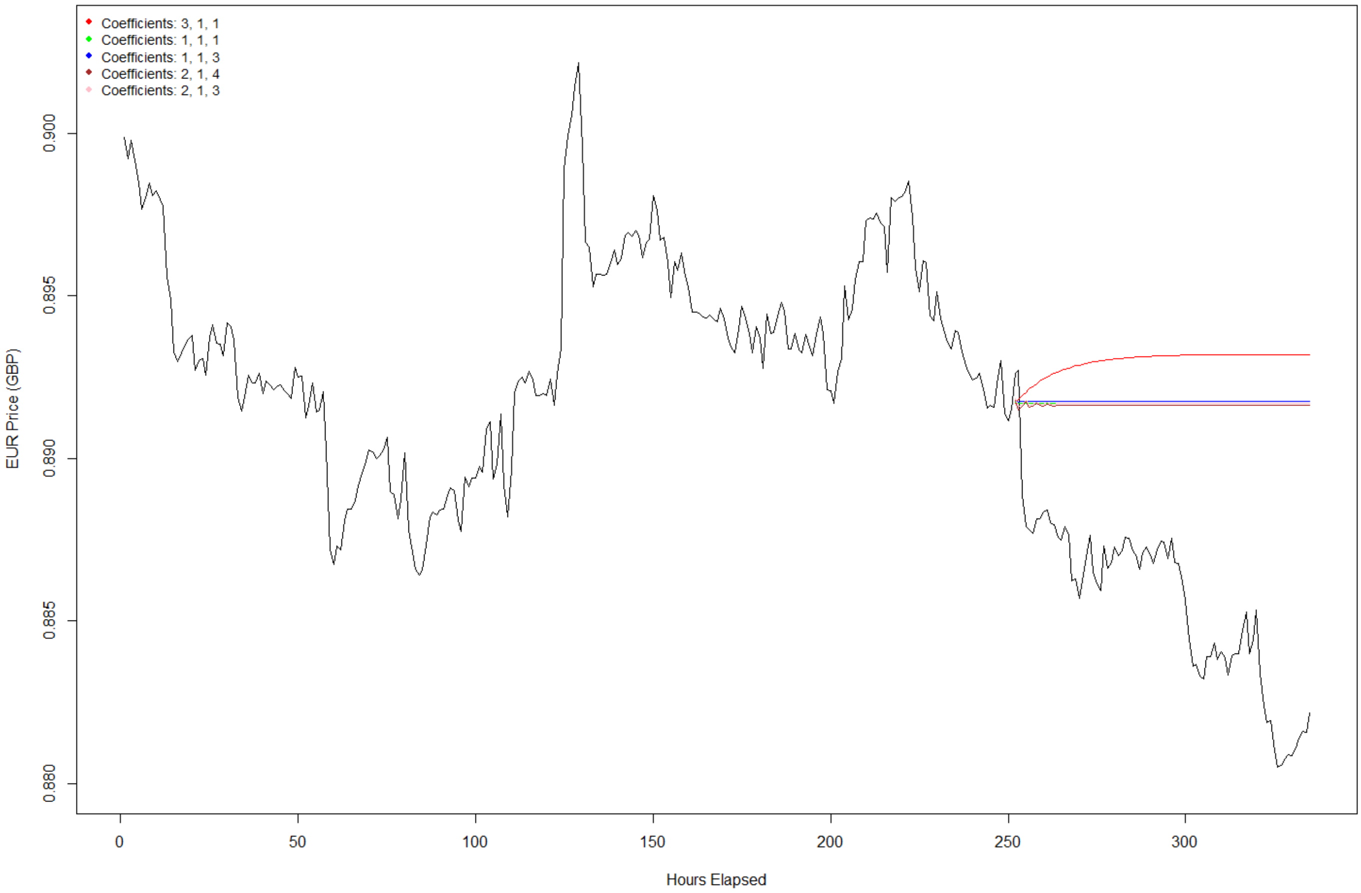

Figure 2.3 - Euro price last 400h (16days) with ARMA

ARIMA Modelling of Financial Time Series (EUR/GBP ~16 days)

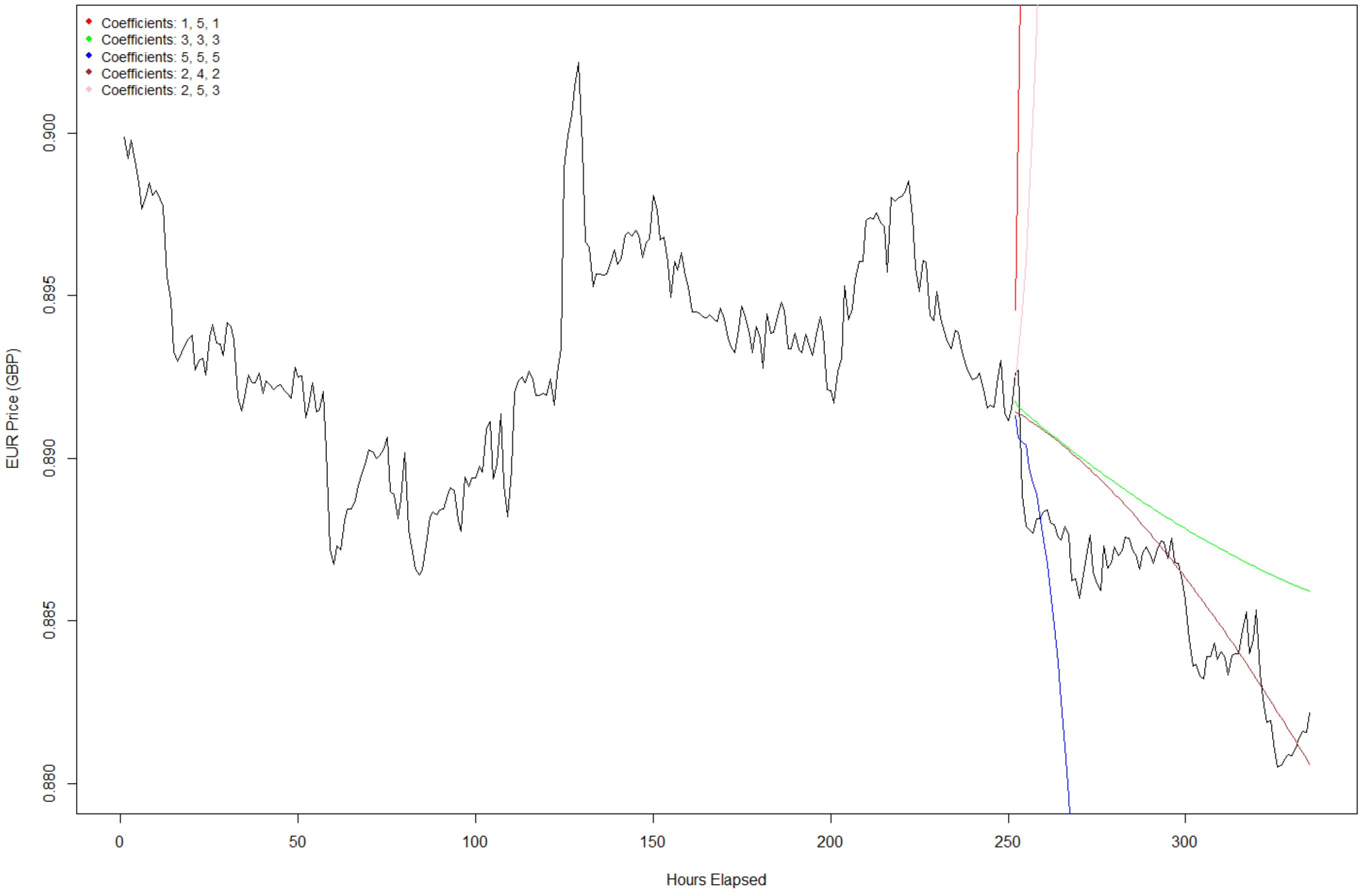

Figure 3.1 - Euro price last 400h (16days) with ARIMA Prediction

<u>R Script used:</u>

```
data <- read.csv("R/BTC400h.csv")

data_train <- data[1:282,2]
data_test <- data[283:376,2]

btc_ar <- arima(data_train, order = c(3, 1, 1), method = "CSS")
btc_pred <- predict(object = btc_ar, n.ahead = 94)

x1 <- seq(283,376,1)
plot(data_train, ylim=range(data[,2]), xlim=range(1,376), xlab="Hours Elapsed", ylab="BTC Price
(USD)", type = "l", col = "black")
lines(x = x1, data_test, ylim=range(data[,2]), xlim=range(1,376), axes = FALSE, xlab = "", ylab
= "", type = "l", col = "black")
lines(x = x1, btc_pred$pred, ylim=range(data[,2]), xlim=range(1,376), axes = FALSE, xlab = "",
ylab = "", type = "l", col = "red")

btc_ar <- arima(data_train, order = c(4,1,3), method = "CSS")
btc_pred <- predict(object = btc_ar, n.ahead = 94)
lines(x = x1, btc_pred$pred, ylim=range(data[,2]), xlim=range(1,376), axes = FALSE, xlab = "",
ylab = "", type = "l", col = "green")

btc_ar <- arima(data_train, order = c(1,1,3), method = "CSS")
btc_pred <- predict(object = btc_ar, n.ahead = 94)
lines(x = x1, btc_pred$pred, ylim=range(data[,2]), xlim=range(1,376), axes = FALSE, xlab = "",
ylab = "", type = "l", col = "blue")

btc_ar <- arima(data_train, order = c(4,1,5), method = "CSS")
btc_pred <- predict(object = btc_ar, n.ahead = 94)
lines(x = x1, btc_pred$pred, ylim=range(data[,2]), xlim=range(1,376), axes = FALSE, xlab = "",
ylab = "", type = "l", col = "brown")

btc_ar <- arima(data_train, order = c(2,1,3), method = "CSS")
btc_pred <- predict(object = btc_ar, n.ahead = 94)
lines(x = x1, btc_pred$pred, ylim=range(data[,2]), xlim=range(1,376), axes = FALSE, xlab = "",
ylab = "", type = "l", col = "pink")

legend("topleft", legend=c("Coefficients: 3, 1, 1", "Coefficients: 4, 1, 3", "Coefficients: 1,
1, 3", "Coefficients: 4, 1, 5", "Coefficients: 2, 3, 3"), col=c("red", "green", "blue",
"brown", "pink"), bty = "n", cex = 0.9, pch = 16, text.col = "black")
```