

No Pot of Gold at the End of the Rainbow

Can we forecast gold prices using classical time series models?

June 5, 2016

- Introduction
- Methodology
- Exploratory Analysis
- ARIMA-GARCH
- Reversing the Transformations
- Exponential Smoothing Forecast
- Conclusions
- References

This work project was developed by Dário Soares, Francisco Lourenço, Hugo Ricardo and Ricardo Costa Mendes.

Introduction

The objective of this project is to achieve an appropriate model to fit and forecast the gold ETF (GLD) daily prices in USD, an exchange traded fund listed in the NYSE, that replicates the physical price of gold.

We have picked the methods learnt within the scope of the forecasting methods course, that seemed to be more suitable to recover the time series structure.

The data was obtained from Quandl, for the period between 2004-11-18 and 2016-05-04, and it contains 2845 observations. The time series was loaded without reference to any time stamps, holidays or weekends.

```
# Load data
gold <- read.csv("project_gold_gld_daily.txt",header=F, sep="," , as.is=TRUE, skip=
54, col.names = c("Date", "Value"))

# Set as time-series
gold.ts0 <- ts(gold$Value)

# Remove last 6 observations
gold.ts <- head(gold.ts0,-6)
length(gold.ts0)
```

```
[1] 2845
```

```
length(gold.ts)
```

```
[1] 2839
```

Methodology

We have extracted a daily time series of gold ETF prices from Quandl and took off the last six observations for demonstrative purposes.

Then we have attempted to model and forecast a time-series of daily prices of the Gold ETF (GLD) using the following forecasting methods: ARIMA, ARCH-GARCH and Exponential Smoothing.

In order to build an ARIMA model we have analysed the time series' stationarity, made some scale transformations, analysed the ACF and PACF plots and finally carried out the Ljung-Box test.

After concluding that a simple ARIMA model could not be applied, we modelled the volatility with ARCH and GARCH methods, analysing the ACF and PACF plots of the (squared) residuals.

We simulated multiple GARCH(r,s) models, and selected the best using both the AIC and the parsimony principle.

We then proceeded to forecast six days forward and compare with the real data that was initially left out.

Finally we applied a Exponential Smoothing forecast for trended series. For each of the following methods we made a simulation: Holt's Linear Trend method, Exponential Trend, Damped Trend and Damped Exponential Trend with optimal smoothing parameters estimation.

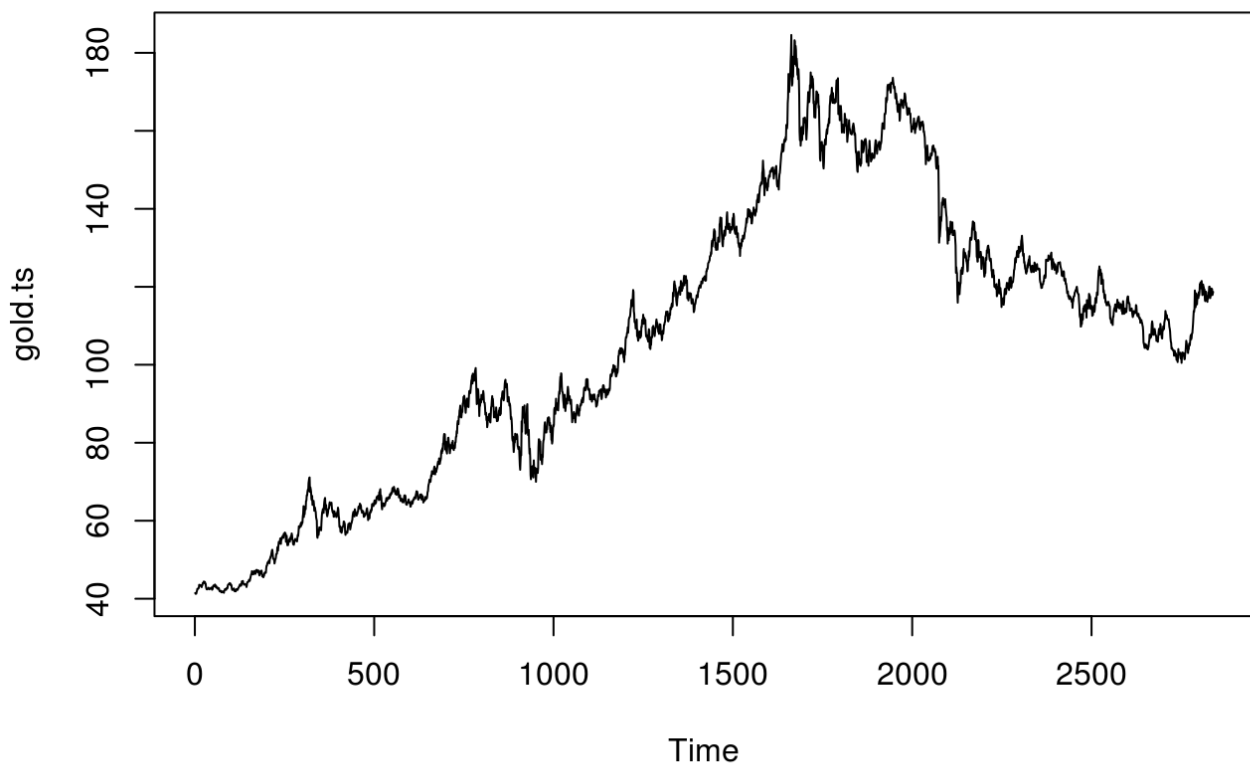
We conclude with a brief overview of the results.

Exploratory Analysis

We started by plotting the original time series of Gold ETF prices in USD.

```
plot.ts(gold.ts, main="Gold ETF Prices (Daily)")
```

Gold ETF Prices (Daily)

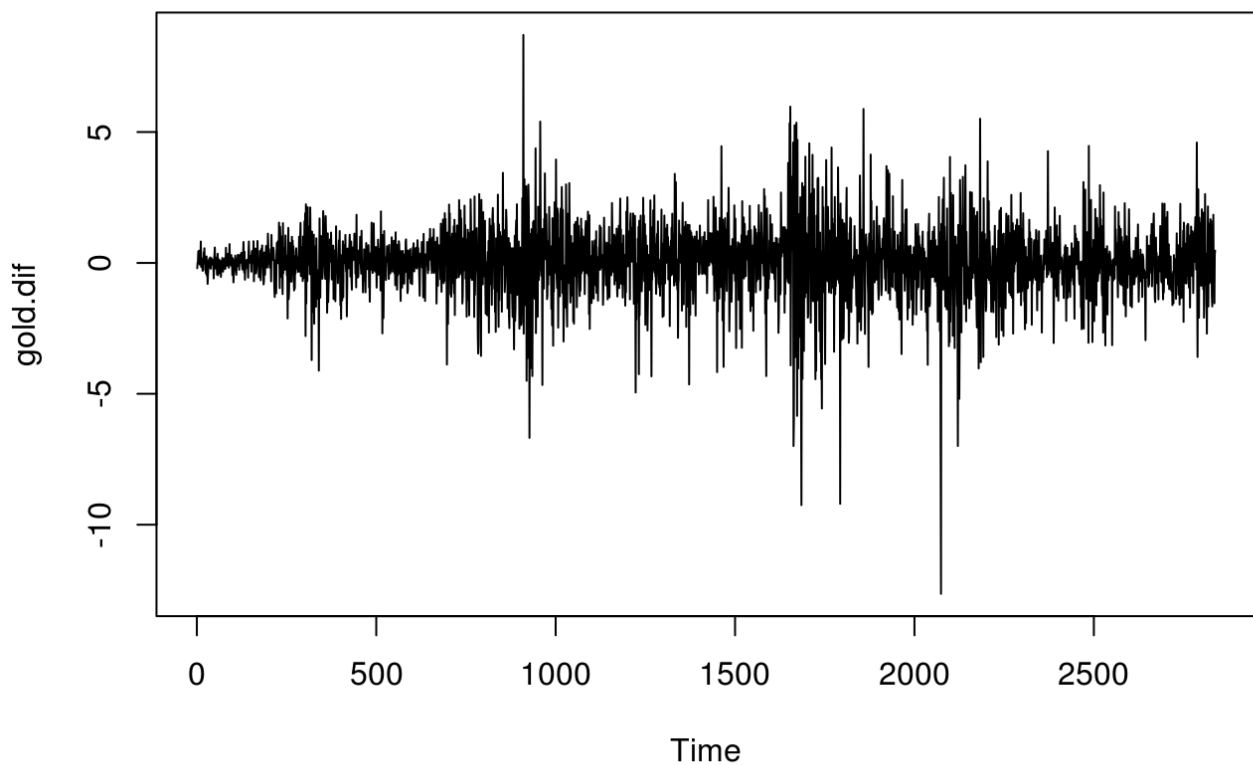


Theoretically seasonality should not be present in a financial time series as it would mean an arbitrage edge.

The time series does not seem to be stationary due to the presence of a trend. Because of this we cannot apply directly the Box-Jenkins methodology, so we will evaluate if first differences are sufficient to turn the time series stationary.

```
gold.dif <- diff(gold.ts)
plot.ts(gold.dif, main="Gold ETF Prices (Daily Differences)")
```

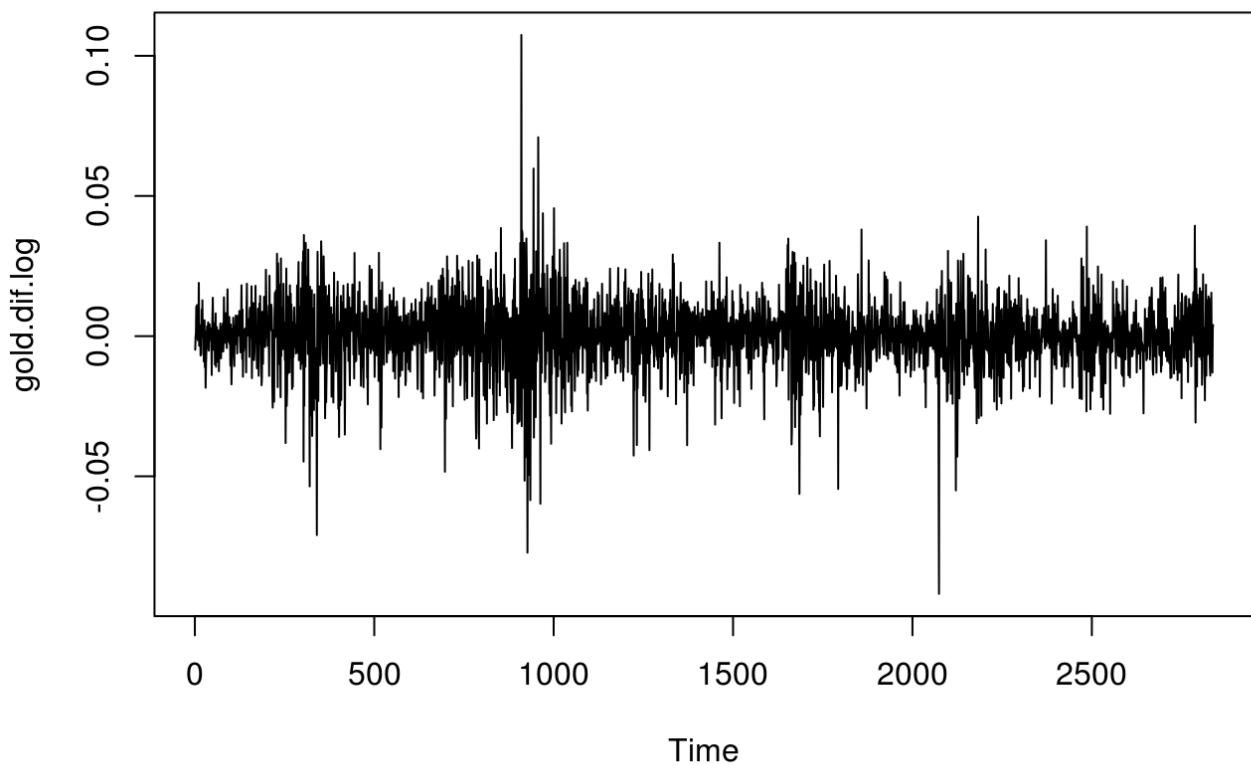
Gold ETF Prices (Daily Differences)



The variance seems to change over time. Therefore, we will apply a logarithmic transformation and a subsequent first difference.

```
gold.dif.log <- diff(log(gold.ts))  
plot.ts(gold.dif.log, main="Gold ETF Prices (Daily Differences of Log(Prices))")
```

Gold ETF Prices (Daily Differences of Log(Prices))

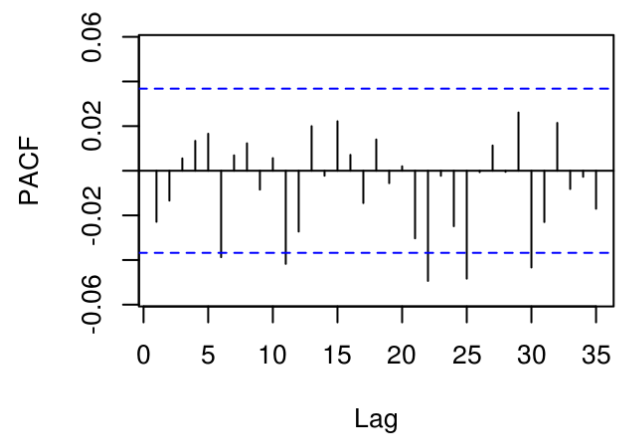
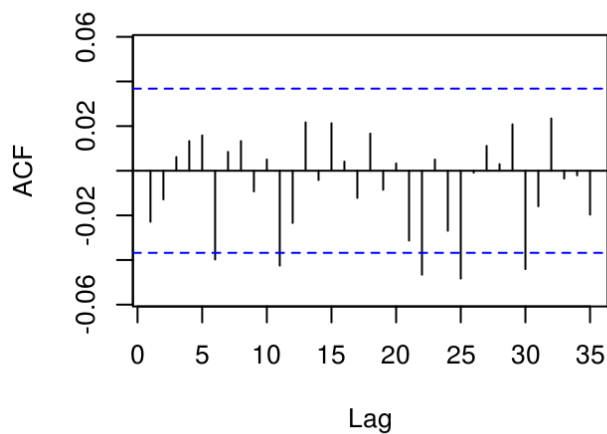
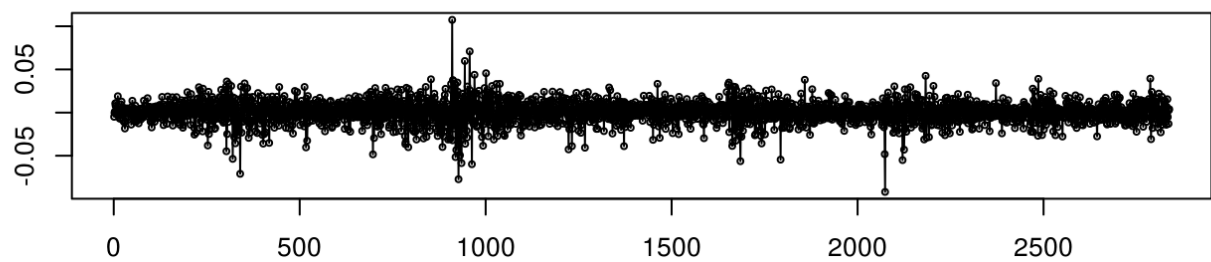


Unfortunately the differences of the logarithmic transformation did not overcome the variance issues. This suggests that an ARIMA-GARCH method is a better approach.

ARIMA-GARCH

```
# ACF and PACF of the transformed series  
tsdisplay(gold.dif.log)
```

gold.dif.log



```
Acf(gold.dif.log, plot = FALSE)
```

Autocorrelations of series 'gold.dif.log', by lag

0	1	2	3	4	5	6	7	8	9
1.000	-0.023	-0.013	0.006	0.013	0.016	-0.040	0.008	0.013	-0.009
10	11	12	13	14	15	16	17	18	19
0.005	-0.042	-0.023	0.022	-0.004	0.021	0.004	-0.012	0.017	-0.009
20	21	22	23	24	25	26	27	28	29
0.003	-0.031	-0.047	0.005	-0.027	-0.048	-0.001	0.011	0.003	0.021
30	31	32	33	34					
-0.044	-0.016	0.023	-0.003	-0.002					

```
Pacf(gold.dif.log, plot = FALSE)
```

Partial autocorrelations of series 'x', by lag

0	1	2	3	4	5	6	7	8	9
-0.023	-0.013	0.006	0.013	0.017	-0.039	0.007	0.012	-0.008	0.006
10	11	12	13	14	15	16	17	18	19
-0.042	-0.027	0.020	-0.002	0.022	0.007	-0.014	0.014	-0.006	0.002
20	21	22	23	24	25	26	27	28	29
-0.030	-0.049	-0.002	-0.025	-0.048	-0.001	0.011	0.000	0.026	-0.043
30	31	32	33						
-0.023	0.021	-0.008	-0.003						

```
# AutoArima
gold.log <- log(gold.ts)
auto.arima(gold.log)
```

Series: gold.log
ARIMA(0,1,0) with drift

Coefficients:
drift
4e-04
s.e. 2e-04

sigma^2 estimated as 0.0001574: log likelihood=8399.02
AIC=-16794.04 AICc=-16794.04 BIC=-16782.14

```
# Ljung-Box test
Box.test(gold.dif.log, lag = 20, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: gold.dif.log
## X-squared = 19.676, df = 20, p-value = 0.4783
```

There is no well defined temporal structure in the transformed data (gold.dif.log).

The Auto Arima points to a random walk.

The Ljung-Box test applied to the transformed data does not reject the null hypothesis (inexistence of autocorrelations), with a p-value of 0.4783.

Given these assertions we will fit a linear model to the transformed data with only the intercept in order to impose a zero mean series.

Then we will analyse the ACF and PACF of the linear model residuals and decide for a GARCH case.

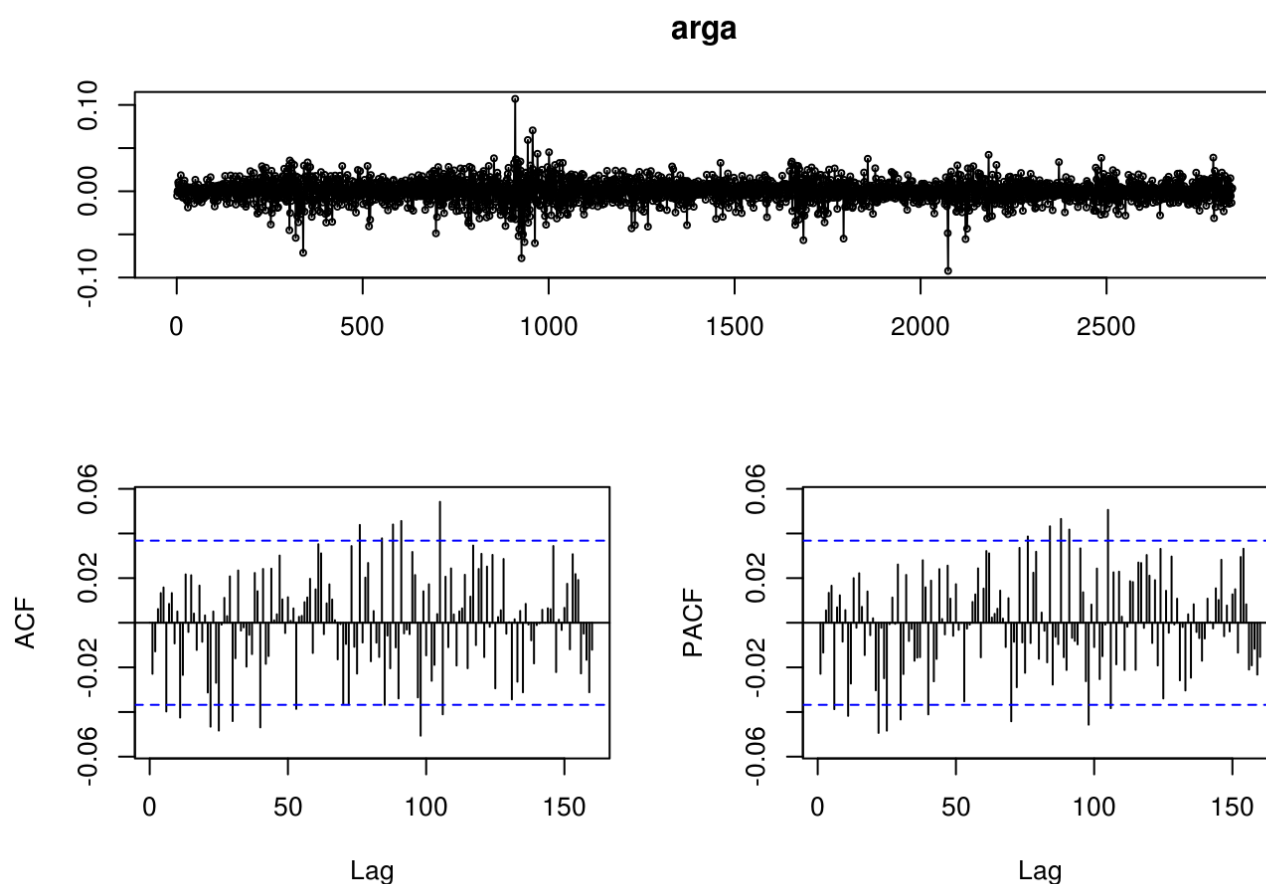
```
# fit of the linear model with just an intercept
fitlin <- Arima(gold.dif.log, order = c(0,0,0))
summary(fitlin)
```

```
## Series: gold.dif.log
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##      intercept
##           4e-04
## s.e.       2e-04
##
## sigma^2 estimated as 0.0001574:  log likelihood=8399.02
## AIC=-16794.04   AICc=-16794.04   BIC=-16782.14
##
## Training set error measures:
##              ME      RMSE      MAE  MPE MAPE      MASE
## Training set -2.301834e-15 0.01254511 0.008865689 -Inf  Inf 0.6658542
##              ACF1
## Training set -0.0228644
```

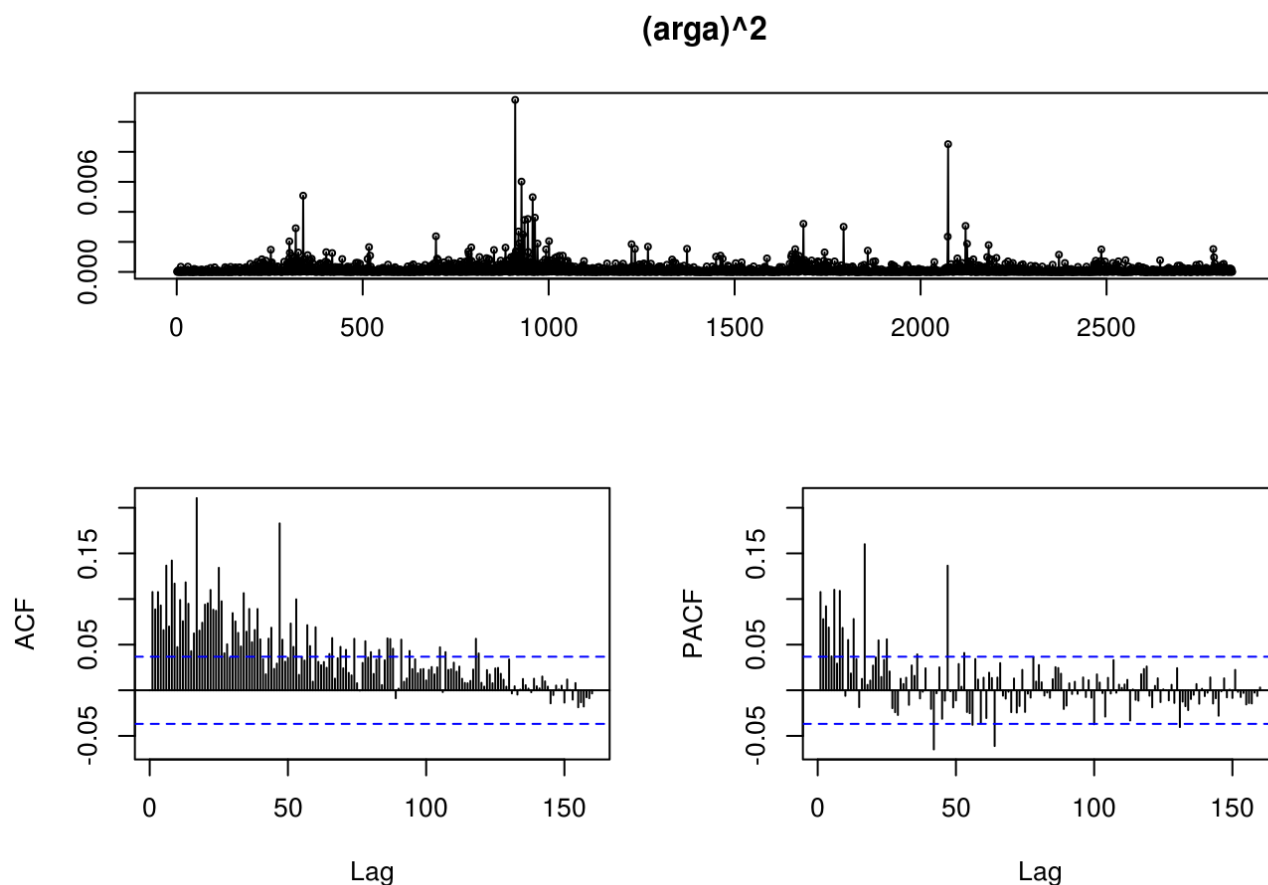
The intercept is statistically significant for a 5% error Type I.

We will try to fit an ARCH-GARCH as it follows:

```
# analysis of the residuals
arga <- fitlin$residuals
tsdisplay(arga, lag.max = 160)
```



```
tsdisplay((arga)**2, lag.max = 160)
```

From the analysis of the ACF and PACF of the squares of the linear model residuals (arga) we conclude that there is a case for a changing variance over time. As the pattern is conducive to an ARMA model we will do a simulation with multiple GARCH models to assist us on the fit decision.

Model	AIC
GARCH(1,1)	-6.096633
GARCH(1,0)	-5.944167
GARCH(2,0)	-5.971479
GARCH(3,0)	-5.996618
GARCH(4,0)	-6.014606
GARCH(5,0)	-6.025685
GARCH(6,0)	-6.050757
GARCH(2,1)	-6.095755
GARCH(3,1)	-6.094847
GARCH(4,1)	-6.093946
GARCH(5,1)	-6.093174
GARCH(6,1)	-6.092338
GARCH(1,2)	-6.095783
GARCH(1,3)	-6.094890

Model	AIC
GARCH(1,4)	-6.094029
GARCH(1,5)	-6.093382
GARCH(1,6)	-6.092544
GARCH(2,2)	-6.096449
GARCH(2,3)	-6.095632
GARCH(2,4)	-6.094740
GARCH(2,5)	-6.094125
GARCH(2,6)	-6.093348
GARCH(3,2)	-6.095585
GARCH(3,3)	-6.094197
GARCH(3,4)	-6.093292
GARCH(3,5)	-6.093531
GARCH(3,6)	-6.092720
GARCH(4,2)	-6.094991
GARCH(4,3)	-6.093292
GARCH(4,4)	-6.093847
GARCH(4,5)	-6.093599
GARCH(4,6)	-6.092924
GARCH(5,2)	-6.094174
GARCH(5,3)	-6.092448
GARCH(5,4)	-6.093027
GARCH(5,5)	-6.093050
GARCH(5,6)	-6.092220
GARCH(6,2)	-6.093563
GARCH(6,3)	-6.093281
GARCH(6,4)	-6.093205
GARCH(6,5)	-6.093205
GARCH(6,6)	-6.093019

From this simulation we conclude the best model is a GARCH(1,1).

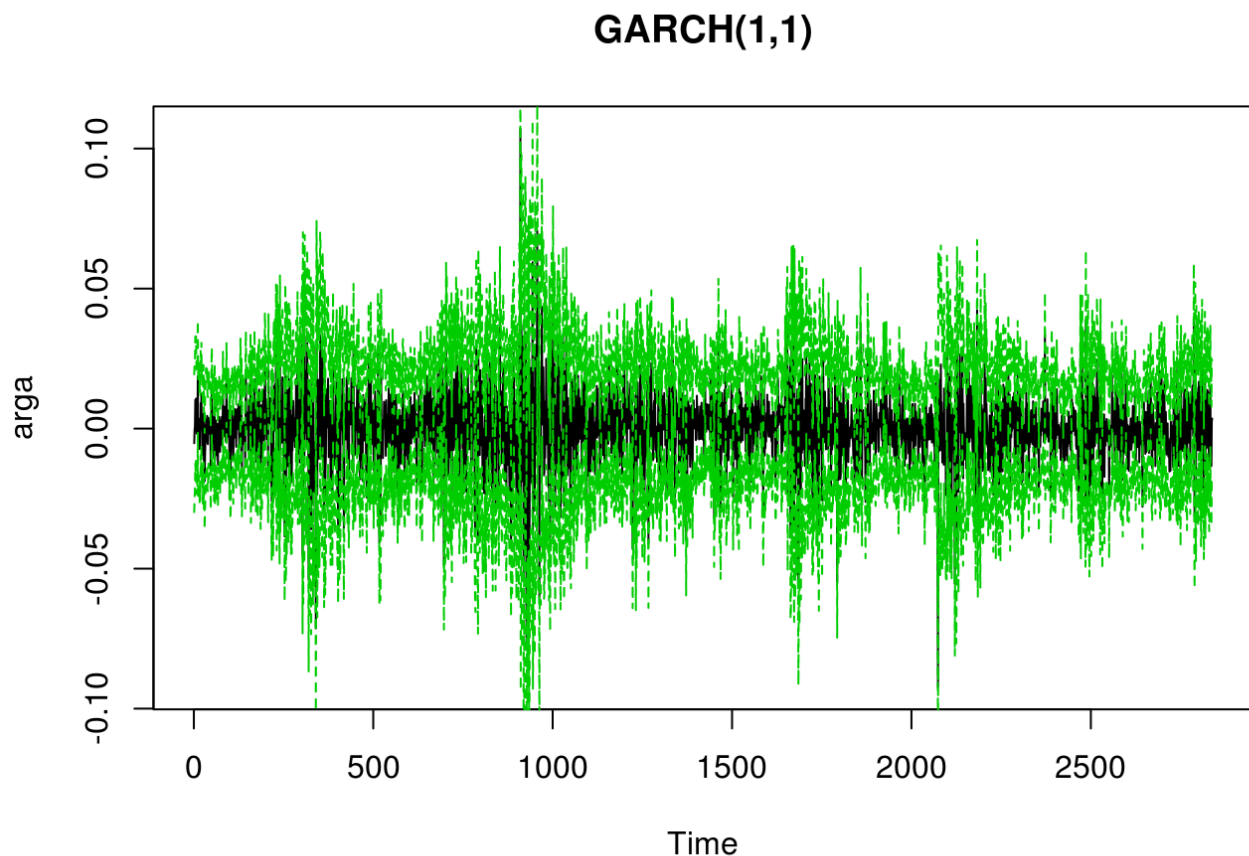
```
gfit <- garchFit(~garch(1,1), arga, trace=F)
summary(gfit)
```

```
##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~garch(1, 1), data = arga, trace = F)
##
## Mean and Variance Equation:
## data ~ garch(1, 1)
## <environment: 0x49866a0>
## [data = arga]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##          mu          omega        alpha1        beta1
## -1.8078e-15  1.9279e-06  5.7395e-02  9.3103e-01
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      -1.808e-15  1.976e-04   0.000      1
## omega    1.928e-06  4.891e-07   3.942 8.08e-05 ***
## alpha1   5.739e-02  7.288e-03   7.875 3.33e-15 ***
## beta1    9.310e-01  8.606e-03  108.183 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 8655.123    normalized:  3.049726
##
## Description:
## Thu Jun  9 22:23:58 2016 by user: dario
##
##
## Standardised Residuals Tests:
##
##              Statistic p-Value
## Jarque-Bera Test  R    Chi^2  748.1844  0
## Shapiro-Wilk Test  R    W      0.9771714  0
## Ljung-Box Test     R    Q(10)  5.21801  0.8761469
## Ljung-Box Test     R    Q(15)  11.53687  0.7136916
## Ljung-Box Test     R    Q(20)  12.36139  0.9030757
## Ljung-Box Test     R^2  Q(10)  11.668   0.3078906
## Ljung-Box Test     R^2  Q(15)  16.75008  0.3340229
## Ljung-Box Test     R^2  Q(20)  20.89322  0.4034416
## LM Arch Test       R    TR^2   13.70953  0.3196408
##
## Information Criterion Statistics:
##          AIC          BIC          SIC          HQIC
## -6.096633 -6.088246 -6.096637 -6.093608
```

The tests results were good except for normality. The fit wasn't able to recover and capture the extreme values idiosyncrasy.

We will go to forecasting and look at the final results.

```
u <- gfit@sigma.t # conditional sd
argapplusu <- arga + 1.96 * u
argaminusu <- arga - 1.96 * u
plot(arga, main="GARCH(1,1)")
lines(argapplusu, lty= 2, col = 3)
lines(argaminusu, lty= 2, col = 3)
```



As we can see above, the GARCH(1,1) allowed to capture a significant part of the changing variance. The arga plot is indeed quite good.

```
# Forecast for 6 periods
x <- predict(gfit, n.ahead = 6)
x
```

```
##      meanForecast  meanError standardDeviation
## 1 -1.807752e-15  0.01048150          0.01048150
## 2 -1.807752e-15  0.01051277          0.01051277
## 3 -1.807752e-15  0.01054360          0.01054360
## 4 -1.807752e-15  0.01057397          0.01057397
## 5 -1.807752e-15  0.01060391          0.01060391
## 6 -1.807752e-15  0.01063343          0.01063343
```

Reversing the Transformations

Now we are going to reverse the transformations that we made on the original time series
(`(diff(log(gold.ts)))`).

```
# Back transformation
# Intercept of fitlin is fitlin$coef

argaforecast <- x$meanForecast
argaforecastplusu <- argaforecast + 1.96*(x$meanError)
argaforecastminusu <- argaforecast - 1.96*(x$meanError)

argaI <- arga + fitlin$coef
argapplusuI <- argapplusu + fitlin$coef
argaminusuI <- argaminusu + fitlin$coef

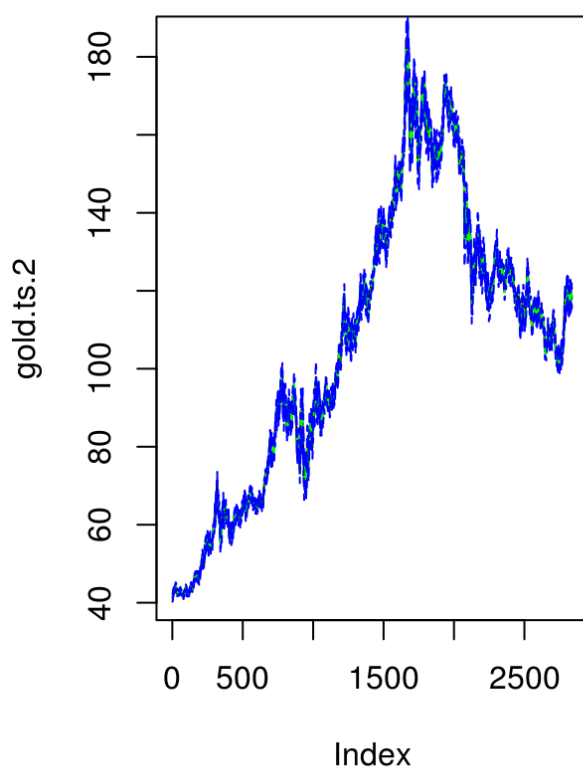
argaforecastI <- argaforecast + fitlin$coef
argaforecastplusuI <- argaforecastplusu + fitlin$coef
argaforecastminusuI <- argaforecastminusu + fitlin$coef

gold.ts.1 <- head(gold.ts, -1)
gold.ts.2 <- tail(gold.ts, (length(gold.ts) - 2))
gold.ts.3 <- tail(gold.ts, 6)

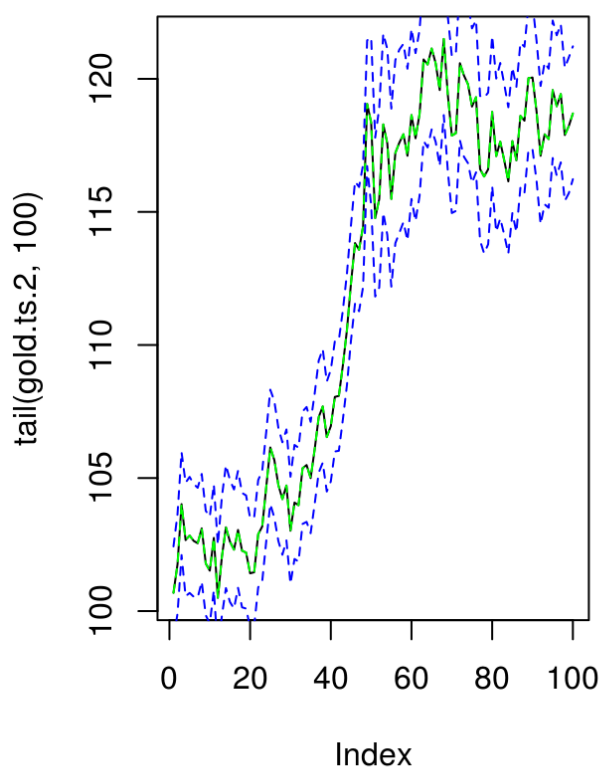
gold.ts.arga <- (exp(argaI))*gold.ts.1
gold.ts.argapplusu <- (exp(argapplusuI))*gold.ts.1
gold.ts.argaminusu <- (exp(argaminusuI))*gold.ts.1

par(mfrow=c(1,2))
plot(gold.ts.2, type="l", main="Series (Since Inception)")
lines(gold.ts.arga, lty= 2, col = "green")
lines(gold.ts.argapplusu, lty= 2, col = "blue")
lines(gold.ts.argaminusu, lty= 2, col = "blue")
plot(tail(gold.ts.2,100), type="l", main="Series (Last 100 Obs)")
lines(tail(gold.ts.arga,100), lty= 2,col = "green")
lines(tail(gold.ts.argapplusu,100), lty= 2, col = "blue")
lines(tail(gold.ts.argaminusu,100), lty= 2, col = "blue")
```

Series (Since Inception)



Series (Last 100 Obs)



The point forecast method is the naive case, given that there is no significant temporal structure in the time series. It gathers all characteristics of a random walk except for the homoskedasticity.

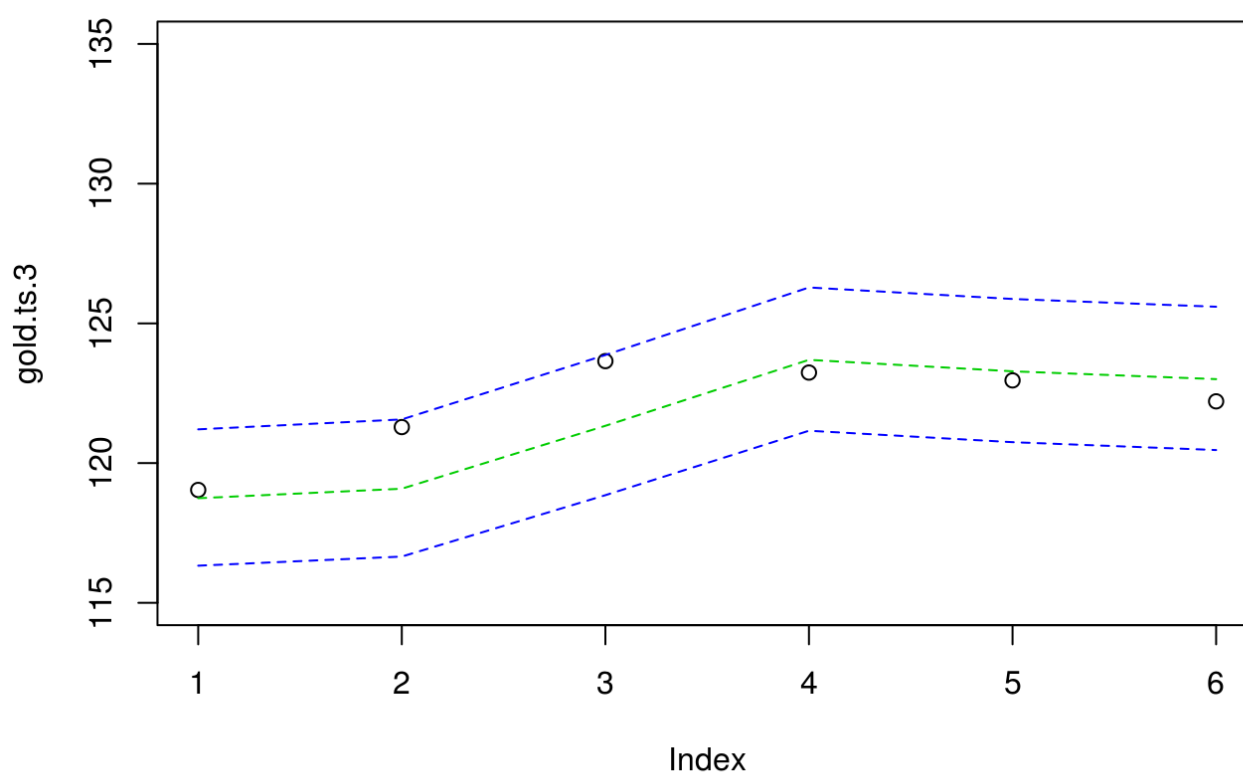
The width of the confidence/forecast intervals vary over time due to the GARCH fit. These intervals are not symmetrical because of the back logarithmic transformation.

```
f.argaforecast <- function(n, seriesx, series0, series1){
  x.arga <- c()
  for(i in 1:n){
    x.arga[i] <- (exp(seriesx[i])) * series0[length(series1)+i-1]
  }
  x.arga
}

gold.ts.argaforecast <- ts(f.argaforecast(6, argaforecastI, gold.ts0, gold.ts))
gold.ts.argaforecastplusu <- ts(f.argaforecast(6, argaforecastplusuI, gold.ts0, gold.ts))
gold.ts.argaforecastminusu <- ts(f.argaforecast(6, argaforecastminusuI, gold.ts0, gold.ts))

plot(gold.ts.3, ylim=c(115,135), main="Forecast (6 days)")
lines(gold.ts.argaforecast, lty= 2,col = 3)
lines(gold.ts.argaforecastplusu, lty= 2,col = 4)
lines(gold.ts.argaforecastminusu, lty= 2,col = 4)
```

Forecast (6 days)



As can be seen above, the last six observations that were left out on purpose fall within the forecast interval of our model.

Exponential Smoothing Forecast

Now let's try to model and forecast the Gold ETF by applying exponential smoothing methods.

We will skip the Simple Exponential Smoothing method, because our time series has a trend. Since there is no seasonality, the most suitable methods are the **Holt's Linear Trend**, the **Exponential Trend**, the **Damped Linear Trend** and the **Damped Exponential Trend**.

If we use the *optimal* setting to automatically estimate the exponential smoothing parameters that minimize the sum-squared residuals, the results are as follows:

```
# Fit the models
# h = number of periods for forecasting
fit1<-holt(gold.ts, h=6)
fit2<-holt(gold.ts, initial='optimal', exponential=TRUE, h=6)
fit3<-holt(gold.ts, initial='optimal', damped=TRUE, h=6)
fit4<-holt(gold.ts, initial='optimal', exponential=TRUE, damped=TRUE, h=6)

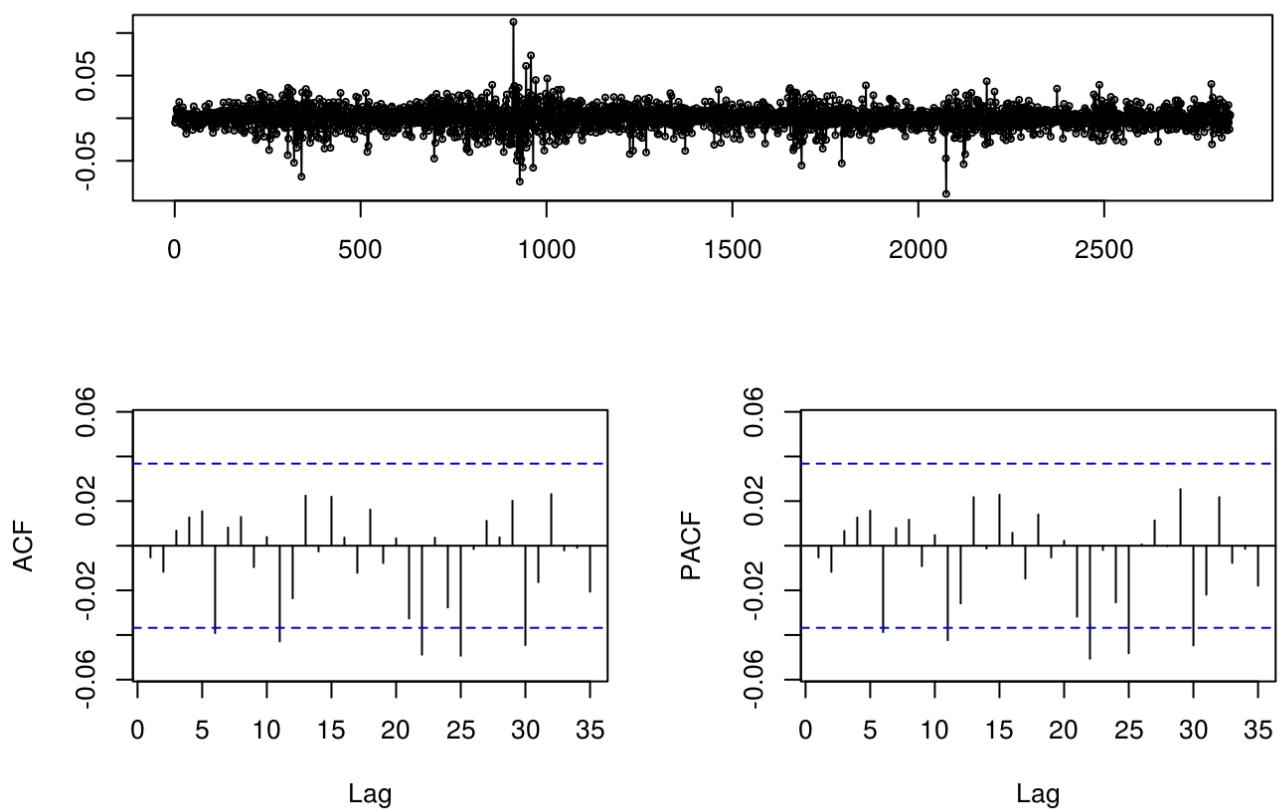
# Show AIC results
m.holt <- AIC(fit1$model)
m.holt.exponential <- AIC(fit2$model)
m.holt.damped <- AIC(fit3$model)
m.holt.exp.damped <- AIC(fit4$model)
cbind(m.holt, m.holt.exponential, m.holt.damped, m.holt.exp.damped)
```

```
##           m.holt m.holt.exponential m.holt.damped m.holt.exp.damped
## [1,] 24299.55      23814.21      24302.38      23817.22
```

The Exponential Trend method has the lowest AIC, but the methods don't differ far from each other.

```
res2 <- fit2$residuals
tsdisplay(res2, main = "Holt's Exponential Residuals")
```

Holt's Exponential Residuals



```
Box.test(res2, lag = 20, type = "Ljung-Box")
```

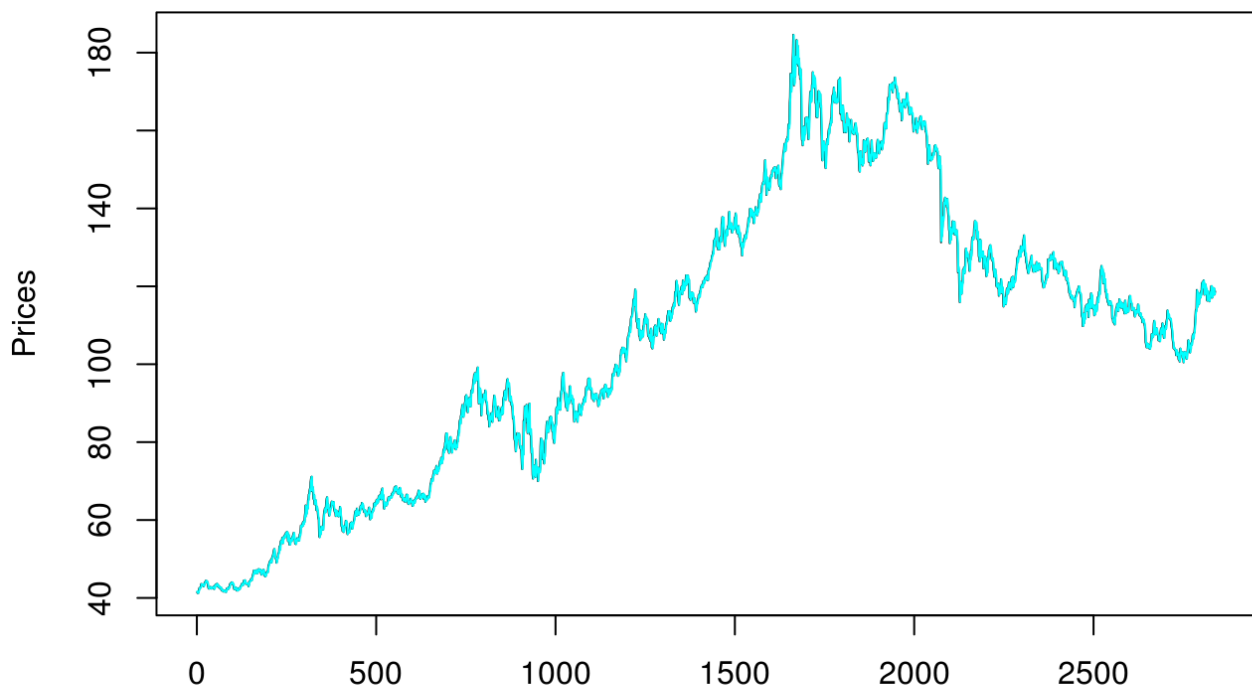


```
##
## Box-Ljung test
##
## data: res2
## X-squared = 18.117, df = 20, p-value = 0.5797
```

The residuals do not show any visible temporal structure.

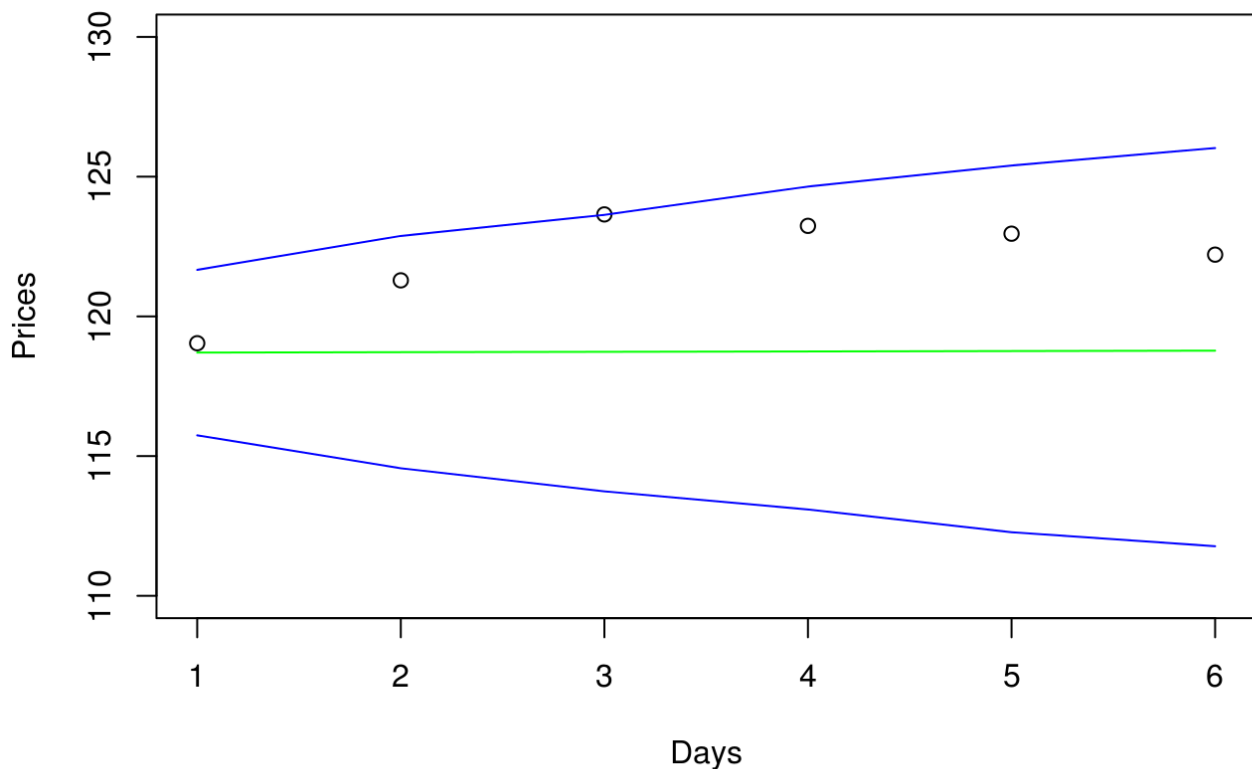
```
# plot forecast values
plot(gold.ts, main= paste("Holt's Exponential Trend"," AIC=", round(AIC(fit2$model
),1)),
     xlab="", ylab = "Prices", col="black", type="l", cex.main=0.9)
lines(fitted(fit2), col="cyan")
```

Holt's Exponential Trend AIC= 23814.2



```
# Plot forecasts for the next 6 days
t <- 1:6
plot(t, tail(gold.ts,6), col="black", xlab="Days", ylab="Prices", main="Holt's Ex
ponential (6 days forecast)", ylim=c(110,130))
lines(t, fit2$mean, col="green")
lines(t, fit2$lower[,2], col="blue")
lines(t, fit2$upper[,2], col="blue")
```

Holt's Exponential (6 days forecast)



As can be seen above, the last six observations that were left out on purpose also fall within the forecast interval of the model.

Conclusions

The ARIMA-GARCH model main drawback is the non-normality of the GARCH residuals. The Holt's Exponential model did not show any particular problem.

Both models should be tested in real time market in order to ascertain the return-risk profile of the following trading strategies:

1. Sell when the price reaches the upper limit and buy when it reaches the lower limit.
2. Buy when the upper limit is broken out and sell when the lower limit is broken out.

References

- Hyndman, R.J. and Athanasopoulos, G. Forecasting: Principles and Practice, OTexts, 2012
- Makridakis, S., Wheelwright, S.C., Hyndman, R.J. Forecasting: Methods and Applications, 3rd edition, John Wiley & Sons, 1998.
- Shumway, R.H. and Stoffer, D.S. Time Series Analysis and its Applications with Examples in R, 3rd edition,

Springer, 2011.

Tsay, R.S. Analysis of Financial Time Series, 2nd edition, Wiley, 2005