Macroeconometrics, Empirical project

Samuel Hashem Zehi[*] Hochholzer Matthias[†]

xxth June 2021

Contents

[*] student ID 12012285

[†] student ID 11724853

1 Idea

We want to look at the relationship between certain prices and the respective search interest on Google for these prices. Can we find granger causality for this relationship? What are possible issues? For example: modern trading algorithms scrape data from the internet and then buy or sell based on the sentiment. Large spikes in search interest may trigger such algorithms. As media spreads the news of price increases more people will look up prices of goods and commodities, again triggering the algorithms. This is basically a feedback loop.

2 Data

First some notes on the data. The data on the search index of certain prices is taken from Google trends which collects the search queries of people within a specific region (here: United States of America). This data is aggregated on a monthly basis and normalized with a range from zero to 100. Already filtered out are duplicate searches in the sense that the same user made the same search multiple times within a short time-frame. This way we exclude the users which have already invested and constantly checked the prices to look how their investment is doing. Data points are divided by total searches for the month and region to represent the relative popularity, i.e. no over-weighting of regions with more people than others which would, given the same search behavior, lead to differing popularities otherwise.

The data on gold prices comes form the London Bullion Market Association Gold Price and the Federal Reserve Bank of St. Louis. It is measured in USD per troy, daily at 3:00pm. Aggregation is done via prices at the end of each month and it is not seasonally adjusted.

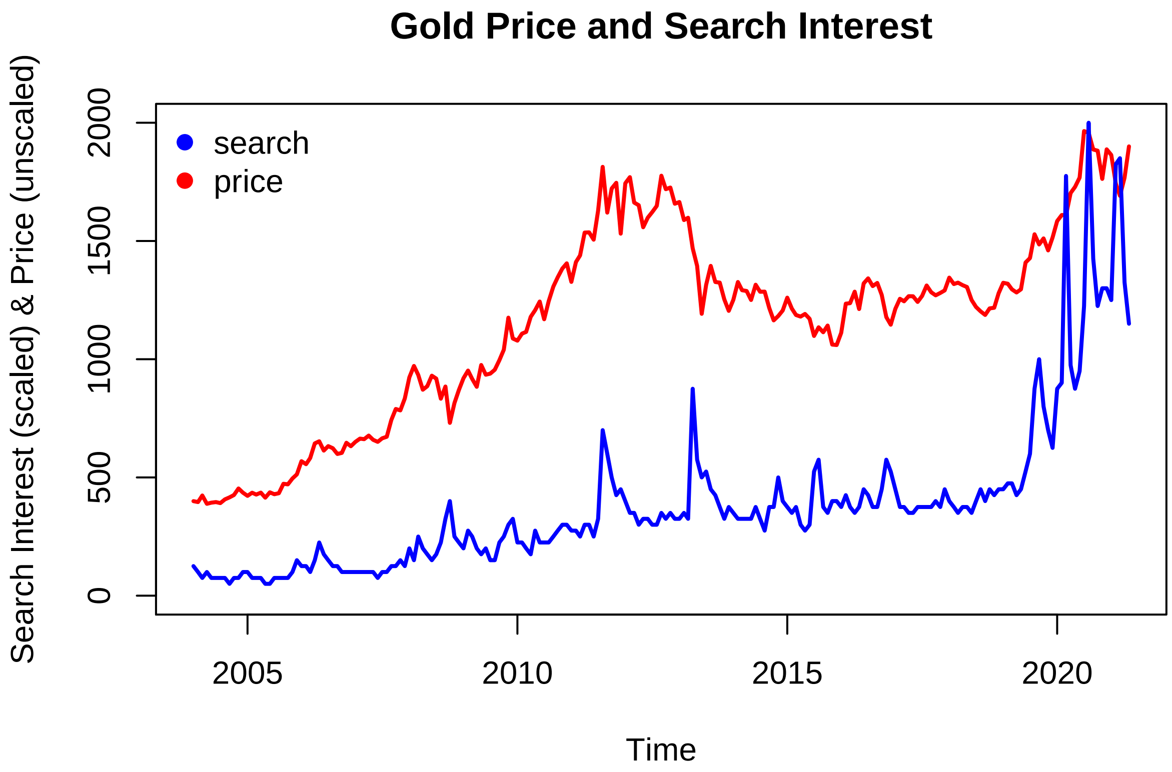
In parts of this project we scale the price and search index to a range from zero to one in order to compare the relative movements more easily.

3 Project Code

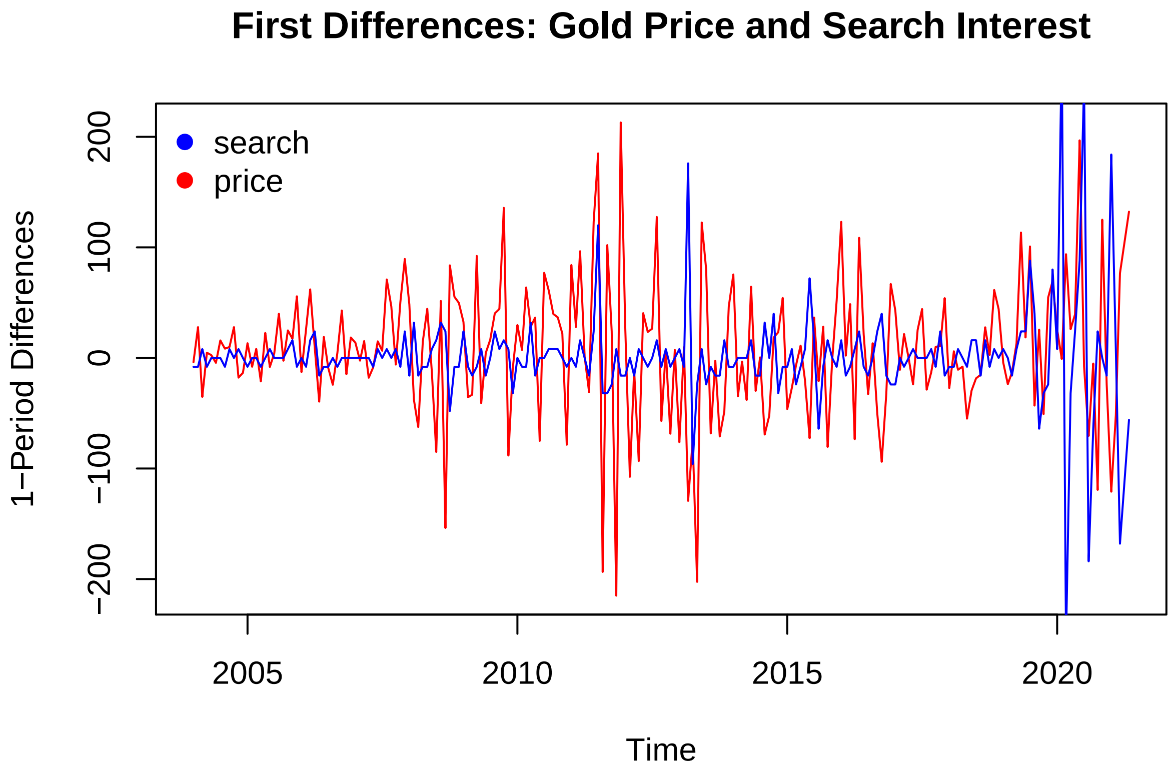
```r
# clear workspace
rm(list=ls())

# load needed libraries
library(readr)
library(vars)
library(tseries)

# set working directory
#setwd("/Users/samue/Downloads/Studium/Economics (Master - Vienna)/2. Semester/Macroeconometrics/Projec

# import search trends
data <-  read_csv("btc-vs-gold-2004.csv", col_types = cols(Month = col_date(format = "%Y-%m")))
# import prices data:
gold_pr <- read_csv("gold-2004.csv", col_types = cols(DATE = col_date(format = "%Y-%m-%d")))
# import high-frequency prices for gold:
gold_HF <- read_csv(’gold-2001-HF.csv’, col_types = cols(DATE = col_date(format = ’%Y-%m-%d’),GOLDPMGBD
```

```r
gold_price <- gold_pr$GOLDPMGBD228NLBM
gold_search <- data$GOLD
```

```
# plot gold price on monthly basis
plot(y=gold_pr$GOLDPMGBD228NLBM,x=gold_pr$DATE,type = ’l’, lwd = 2, col = ’red’,
     ylim = c(0,2000), main = ’Gold Price and Search Interest’,
     xlab = ’Time’, ylab = ’Search Interest (scaled) & Price (unscaled)’)
# add gold search interest scaled up
lines(y=25*data$GOLD,x=gold_pr$DATE, lwd = 2, col = ’blue’)
legend(’topleft’, legend = c(’search’,’price’),
       col = c(’blue’,’red’), bty = "n", pch = c(19,19))
```

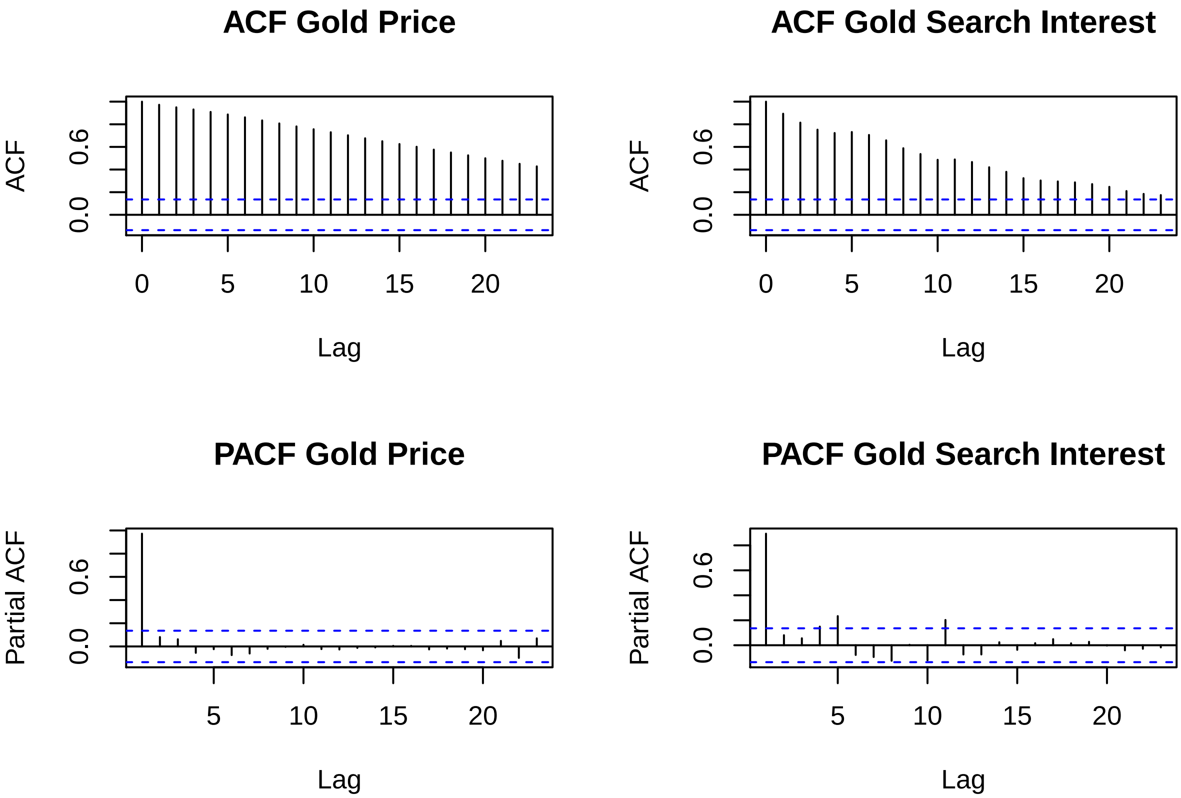


```
# create first differenced prices and search interest
t <- length(gold_pr$DATE)
gold_price_FD <- rep(0,t-1)
for(i in 2:209){gold_price_FD[i-1] <- gold_pr$GOLDPMGBD228NLBM[i]-gold_pr$GOLDPMGBD228NLBM[i-1]}
gold_search_FD <- rep(0,t-1)
for(i in 2:209){gold_search_FD[i-1] <- data$GOLD[i]-data$GOLD[i-1]}
t_1 <- length(gold_HF$DATE)
gold_daily_FD <- rep(0,t_1-1)
for(i in 2:5332){gold_daily_FD[i-1] <- gold_HF$GOLDPMGBD228NLBM[i]-gold_HF$GOLDPMGBD228NLBM[i-1]}

# plot first differenced variables
plot(y=gold_price_FD,x=gold_pr$DATE[1-209], type = ’l’, lwd = 1, col = ’red’,
     xlab = ’Time’, ylab = ’1-Period Differences’,
     main = ’First Differences: Gold Price and Search Interest’)
lines(y=gold_search_FD*8,x=gold_pr$DATE[1-209], lwd = 1, col = ’blue’)
legend(’topleft’, legend = c(’search’,’price’),
       col = c(’blue’,’red’), bty = "n", pch = c(19,19))
```

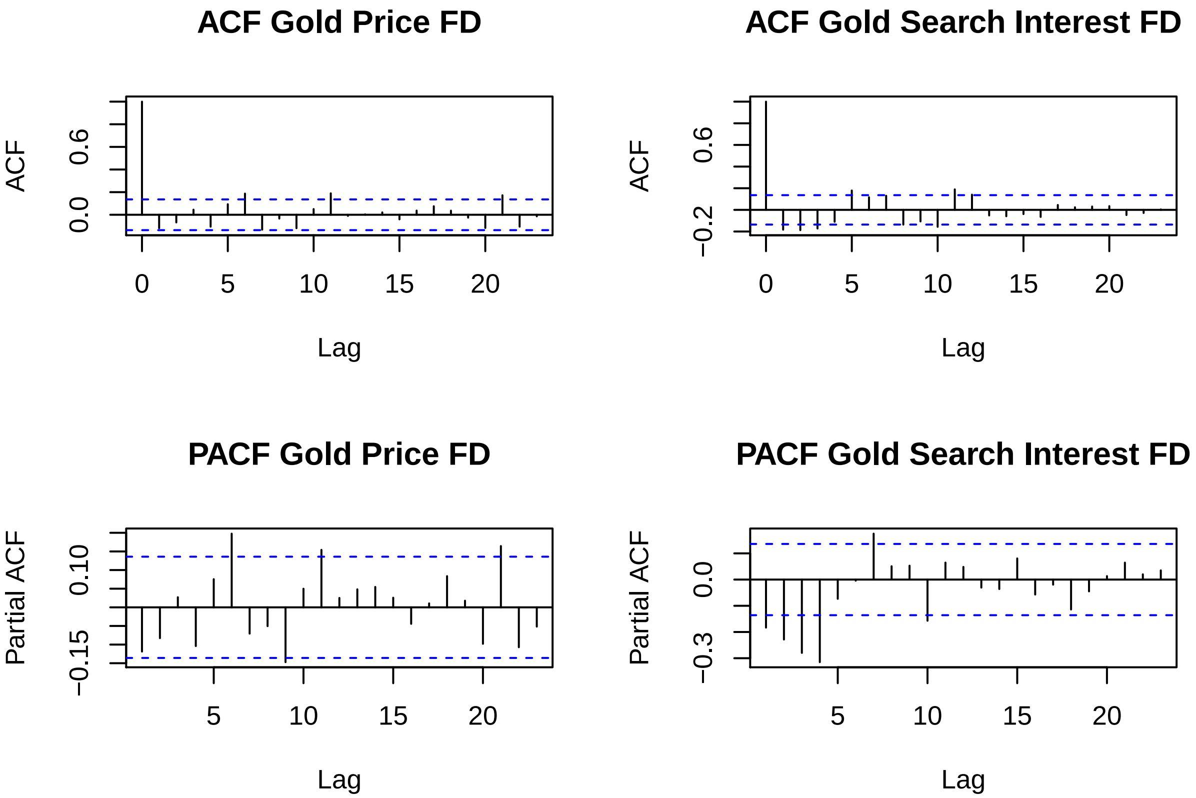


Visually, it appears that the more volatile periods match. An issue seems to be the scaling of the variables.

```
# plot ACF for unmodified variables:
par(mfrow=c(2,2))      # changes the plot layout to more easily compare them
acf(gold_pr$GOLDPMGBD228NLBM, main = 'ACF Gold Price')
acf(data$GOLD, main = 'ACF Gold Search Interest')

# plot PACF for unmodified variables:
pacf(gold_pr$GOLDPMGBD228NLBM, main = 'PACF Gold Price')
pacf(data$GOLD, main = 'PACF Gold Search Interest')
```



```
# plot ACF for differenced variables
acf(gold_price_FD,main = 'ACF Gold Price FD')
acf(gold_search_FD, main = 'ACF Gold Search Interest FD')

# plot PACF for differenced variables
pacf(gold_price_FD,main = 'PACF Gold Price FD')
pacf(gold_search_FD, main = 'PACF Gold Search Interest FD')
```

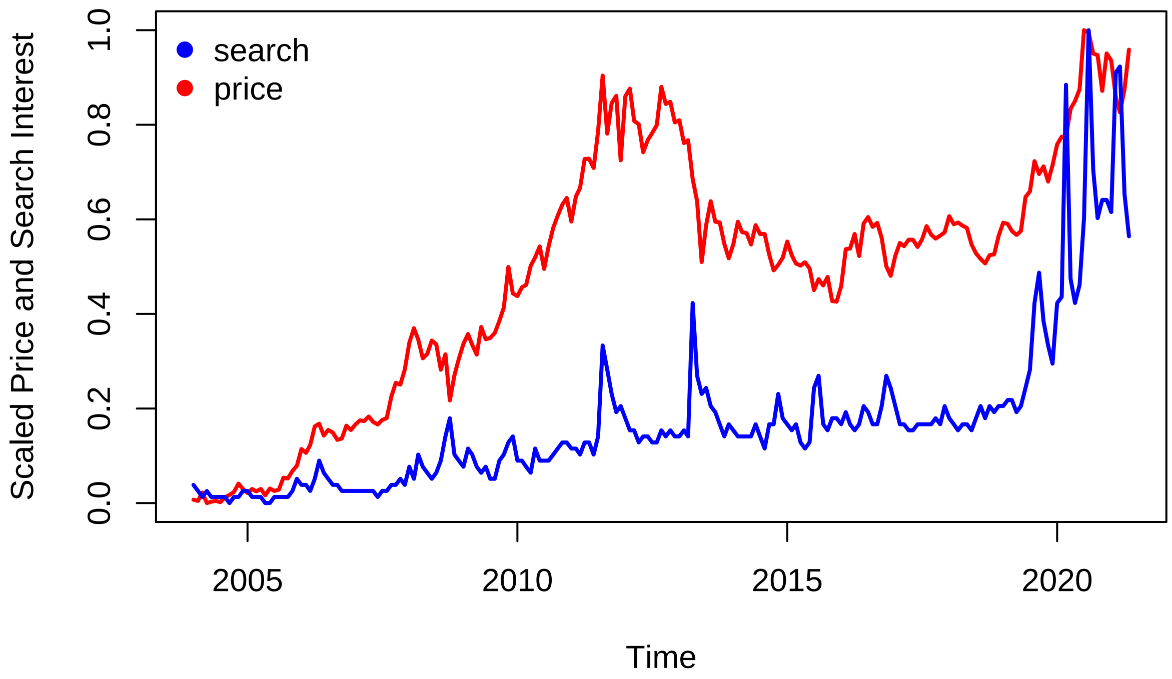


```
par(mfrow = c(1,1))    # revert layout changes
```

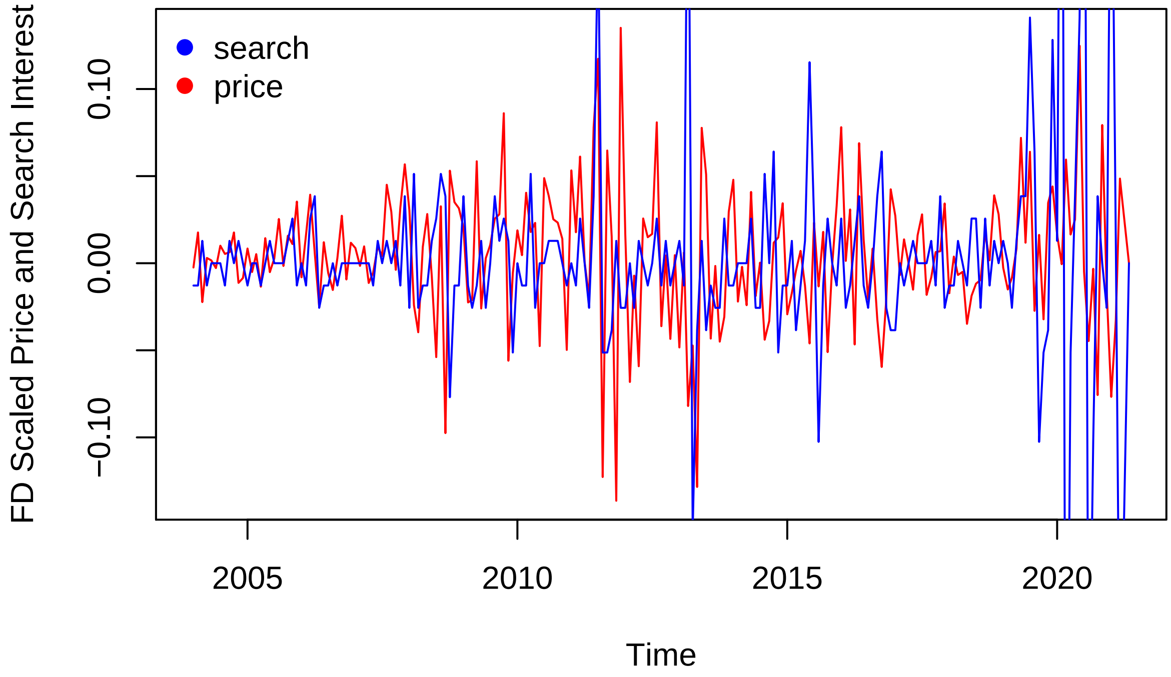
Autocorrelation for the differenced variables seems like no month-on-month relationship between the changes. Kind of like a random walk?

Might help with the interpretation: scale all variables $\mathbf{X}$ such that $X_t \in [0, 1] \forall t \in T$.

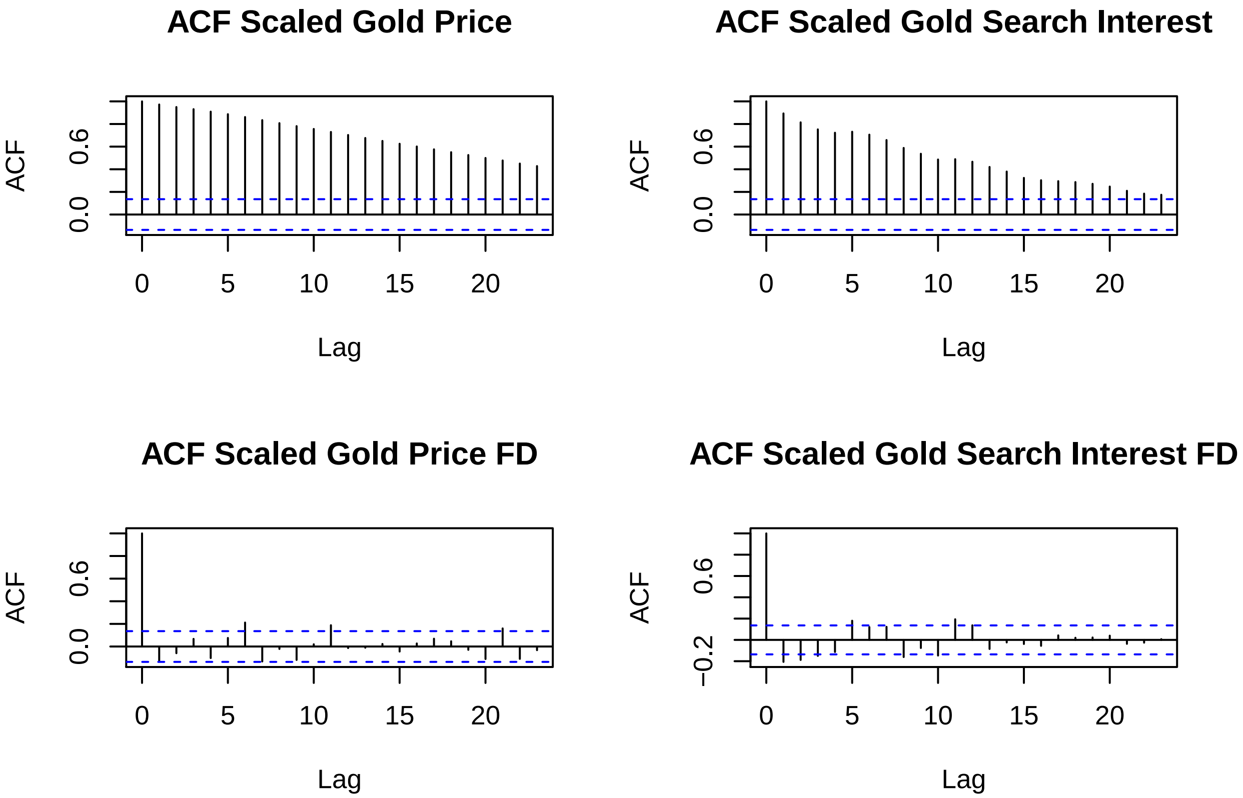
```
range01 <- function(x){(x-min(x))/(max(x)-min(x))}
plot(y=range01(gold_pr$GOLDPMGBD228NLBM),x=gold_pr$DATE, lwd = 2, type = ’l’,
     ylab = ’Scaled Price and Search Interest’,
     xlab = ’Time’, col = ’red’)
lines(y=range01(data$GOLD),x=gold_pr$DATE, lwd = 2, col = ’blue’)
legend(’topleft’, legend = c(’search’,’price’),
       col = c(’blue’,’red’), bty = "n", pch = c(19,19))
```



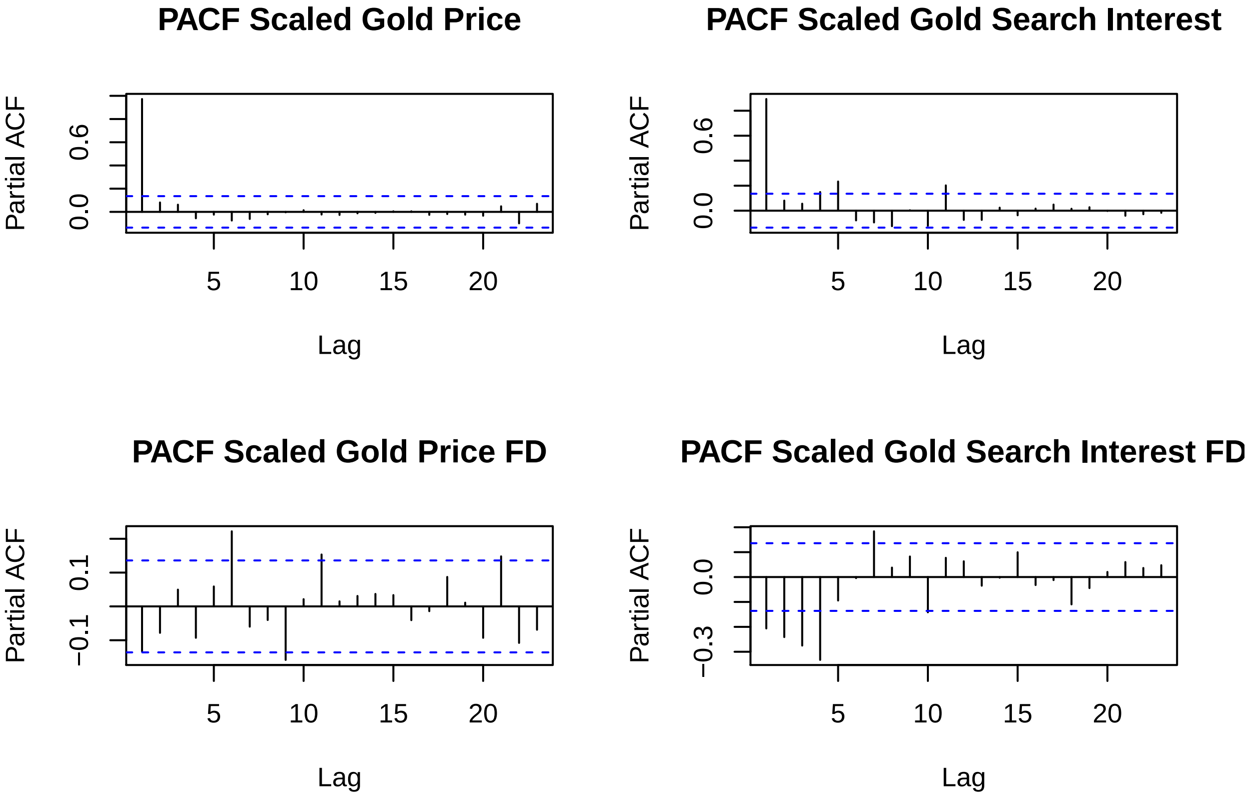
```
# save scaled variables
gold_price_scaled <- range01(gold_pr$GOLDPMGBD228NLBM)
gold_search_scaled <- range01(data$GOLD)

# create first difference on scaled variables:
gold_search_scaled_FD <- rep(0,t-1)
gold_price_scaled_FD <- rep(0,t-1)

for(i in 2:t-1){
  gold_price_scaled_FD[i-1] <- gold_price_scaled[i]-gold_price_scaled[i-1]
}
for(i in 2:t-1){
  gold_search_scaled_FD[i-1] <- gold_search_scaled[i]-gold_search_scaled[i-1]
}

# plot first differenced:
plot(y=gold_price_scaled_FD, x=gold_pr$DATE[1-209], lwd = 1, type = 'l',
     ylab = 'FD Scaled Price and Search Interest',
     xlab = 'Time', col = 'red')
lines(y= gold_search_scaled_FD, x=gold_pr$DATE[1-209], lwd = 1, col = 'blue')
legend('topleft', legend = c('search','price'),
       col = c('blue','red'), bty = "n", pch = c(19,19))
```



```
# plot ACFs
par(mfrow=c(2,2))      # changes the plot layout to more easily compare them
acf(gold_price_scaled, main = 'ACF Scaled Gold Price')
acf(gold_search_scaled, main = 'ACF Scaled Gold Search Interest')
acf(gold_price_scaled_FD,main = 'ACF Scaled Gold Price FD')
acf(gold_search_scaled_FD, main = 'ACF Scaled Gold Search Interest FD')
```



```
par(mfrow = c(1,1))    # revert layout changes


# plot PACFs
par(mfrow=c(2,2))      # changes the plot layout to more easily compare them
pacf(gold_price_scaled, main = ’PACF Scaled Gold Price’)
pacf(gold_search_scaled, main = ’PACF Scaled Gold Search Interest’)
pacf(gold_price_scaled_FD,main = ’PACF Scaled Gold Price FD’)
pacf(gold_search_scaled_FD, main = ’PACF Scaled Gold Search Interest FD’)
```


PACF Scaled Gold Price

PACF Scaled Gold Search Interest

PACF Scaled Gold Price FD

PACF Scaled Gold Search Interest FD

```
par(mfrow = c(1,1))    # revert layout changes
```

Unsurprisingly the rescaling does not matter for the autocorrelation as it is a scaled measure of linear relationships anyway.

ACF Scaled Gold Search Interest FD together with PACF Scaled Gold Search Interest FD gives evidence for an AR(4).

For the Gold Price it’s as you say. Could be a MA(1), AR(1) or an ARMA.

Should we also do DF for gold_search?

```
#####################################################
####### From here on: data saved as time series #####
#####################################################

# "Normal" scaled
# save variable vectors as time series format:
gold_price_scaled <- ts(gold_price_scaled, frequency = 12,
                        start = c(2004, 1), end = c(2021, 5))
gold_search_scaled <- ts(gold_search_scaled, frequency = 12,
                         start = c(2004,1), end = c(2021,5))

# set up data for estimation using ‘VAR()‘
VAR_data_scaled <- window(ts.union(gold_price_scaled, gold_search_scaled),
                   start = c(2004, 1), end = c(2021, 5))

# estimate model coefficients using ‘VAR()‘
VAR_est_scaled <- VAR(y = VAR_data_scaled, p = 5)        # used lag order 5 without testing anything !!
summary(VAR_est_scaled)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_scaled, gold_search_scaled
## Deterministic variables: const
## Sample size: 204
## Log Likelihood: 655.565
## Roots of the characteristic polynomial:
## 1.019 0.9733 0.8642 0.8642 0.7279 0.7279 0.5834 0.5834 0.4733 0.4733
## Call:
## VAR(y = VAR_data_scaled, p = 5)
##
##
## Estimation results for equation gold_price_scaled:
## ==================================================
## gold_price_scaled = gold_price_scaled.l1 + gold_search_scaled.l1 + gold_price_scaled.l2 + gold_searc
##
##                         Estimate Std. Error t value Pr(>|t|)
## gold_price_scaled.l1   0.8680354  0.0727396  11.933   <2e-16 ***
## gold_search_scaled.l1 -0.0308545  0.0425133  -0.726   0.4689
## gold_price_scaled.l2   0.0525540  0.0958267   0.548   0.5840
## gold_search_scaled.l2 -0.0002992  0.0492484  -0.006   0.9952
## gold_price_scaled.l3   0.0990654  0.0952912   1.040   0.2998
## gold_search_scaled.l3  0.0072362  0.0503455   0.144   0.8859
## gold_price_scaled.l4  -0.1320176  0.0958383  -1.378   0.1700
## gold_search_scaled.l4  0.1023473  0.0512323   1.998   0.0472 *
## gold_price_scaled.l5   0.0837110  0.0742230   1.128   0.2608
## gold_search_scaled.l5 -0.0335239  0.0469061  -0.715   0.4757
## const                  0.0125843  0.0061555   2.044   0.0423 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 0.03944 on 193 degrees of freedom
## Multiple R-Squared: 0.977,   Adjusted R-squared: 0.9758
## F-statistic: 819.4 on 10 and 193 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gold_search_scaled:
```

```
## ===================================================
## gold_search_scaled = gold_price_scaled.l1 + gold_search_scaled.l1 + gold_price_scaled.l2 + gold_sear
##
##                         Estimate Std. Error t value Pr(>|t|)
## gold_price_scaled.l1    0.229715   0.116612   1.970   0.0503 .
## gold_search_scaled.l1   0.551036   0.068155   8.085 6.70e-14 ***
## gold_price_scaled.l2    0.008144   0.153624   0.053   0.9578
## gold_search_scaled.l2 -0.021149   0.078952  -0.268   0.7891
## gold_price_scaled.l3  -0.031885   0.152766  -0.209   0.8349
## gold_search_scaled.l3  0.010771   0.080711   0.133   0.8940
## gold_price_scaled.l4  -0.136519   0.153643  -0.889   0.3754
## gold_search_scaled.l4  0.105523   0.082133   1.285   0.2004
## gold_price_scaled.l5  -0.044632   0.118990  -0.375   0.7080
## gold_search_scaled.l5  0.375179   0.075198   4.989 1.35e-06 ***
## const                 -0.009031   0.009868  -0.915   0.3612
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06323 on 193 degrees of freedom
## Multiple R-Squared: 0.8718,  Adjusted R-squared: 0.8651
## F-statistic: 131.2 on 10 and 193 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                    gold_price_scaled gold_search_scaled
## gold_price_scaled          0.0015554         -0.0001557
## gold_search_scaled        -0.0001557          0.0039975
##
## Correlation matrix of residuals:
##                    gold_price_scaled gold_search_scaled
## gold_price_scaled            1.00000           -0.06244
## gold_search_scaled          -0.06244            1.00000
```

```r
# augmented df test on only the gold price
df_test_gold_price <- urca::ur.df(gold_price_scaled, type = c('trend'),
                      selectlags = 'BIC')
summary(df_test_gold_price)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.134978 -0.021219 -0.001006  0.022695  0.130576
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.833e-03  5.949e-03   1.485    0.139
## z.lag.1     -2.821e-02  1.742e-02  -1.620    0.107
## tt           9.382e-05  7.500e-05   1.251    0.212
## z.diff.lag  -1.077e-01  7.042e-02  -1.529    0.128
##
## Residual standard error: 0.0394 on 203 degrees of freedom
## Multiple R-squared:  0.02698,    Adjusted R-squared:  0.0126
## F-statistic: 1.876 on 3 and 203 DF,  p-value: 0.1348
##
##
## Value of test-statistic is: -1.6196 2.0259 1.3129
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

```
# augmented df test on only the gold search
df_test_gold_search <- urca::ur.df(gold_search_scaled, type = c('trend'),
                       selectlags = 'BIC')
summary(df_test_gold_search)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28046 -0.02637 -0.00703  0.00958  0.45812
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0101620  0.0101515  -1.001 0.318001
## z.lag.1     -0.1821778  0.0453131  -4.020 8.19e-05 ***
## tt           0.0004192  0.0001244   3.371 0.000897 ***
## z.diff.lag  -0.0896148  0.0708442  -1.265 0.207337
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07057 on 203 degrees of freedom
## Multiple R-squared:  0.1063, Adjusted R-squared:  0.09308
## F-statistic: 8.047 on 3 and 203 DF,  p-value: 4.307e-05
##
```

```
##
## Value of test-statistic is: -4.0204 5.6202 8.2243
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

```
# the null is always random walk with drift/null
```

For the price we cannot reject the null of a non-stationary process (the random walk with drift+trend is the null), seems to fit conventional wisdom on prices. For the search index we reject the null given the data.

For both, the null cannot be rejected given the data, the null is non-stationarity. Not very unexpected for prices, as they are often thought about as following a random walk and thus being non-stationary. But we can also look at difference-stationarity to check.

```
# First-differences scaled

# save variable vectors as time series format:
gold_price_scaled_FD <- ts(gold_price_scaled_FD, frequency = 12,
                        start = c(2004, 2), end = c(2021, 5))      #took 2004,2, because we are now dea
gold_search_scaled_FD <- ts(gold_search_scaled, frequency = 12,
                         start = c(2004,2), end = c(2021,5))

# set up data for estimation using `VAR()`
VAR_data_scaled_FD <- window(ts.union(gold_price_scaled_FD, gold_search_scaled_FD),
                      start = c(2004, 2), end = c(2021, 5))

# estimate model coefficients using `VAR()`
VAR_est_scaled_FD <- VAR(y = VAR_data_scaled_FD, p = 5)        # used lag order 5 without testing anyth
summary(VAR_est_scaled_FD)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_scaled_FD, gold_search_scaled_FD
## Deterministic variables: const
## Sample size: 203
## Log Likelihood: 655.09
## Roots of the characteristic polynomial:
## 1.023 0.8636 0.8636 0.7413 0.7413 0.613 0.613 0.4371 0.4244 0.4244
## Call:
## VAR(y = VAR_data_scaled_FD, p = 5)
##
##
## Estimation results for equation gold_price_scaled_FD:
## =====================================================
## gold_price_scaled_FD = gold_price_scaled_FD.l1 + gold_search_scaled_FD.l1 + gold_price_scaled_FD.l2 +
##
##                            Estimate Std. Error t value Pr(>|t|)
## gold_price_scaled_FD.l1  -0.134490   0.071705  -1.876   0.0622 .
## gold_search_scaled_FD.l1 -0.018280   0.042583  -0.429   0.6682
## gold_price_scaled_FD.l2  -0.055488   0.072502  -0.765   0.4450  
## gold_search_scaled_FD.l2 -0.020997   0.049795  -0.422   0.6737  
## gold_price_scaled_FD.l3   0.057081   0.072767   0.784   0.4338  
## gold_search_scaled_FD.l3  0.111142   0.050858   2.185   0.0301 *
## gold_price_scaled_FD.l4  -0.077790   0.074279  -1.047   0.2963  
## gold_search_scaled_FD.l4 -0.006915   0.052089  -0.133   0.8945  
## gold_price_scaled_FD.l5   0.028470   0.074084   0.384   0.7012  
## gold_search_scaled_FD.l5 -0.075807   0.046115  -1.644   0.1018  
## const                     0.006617   0.004237   1.562   0.1200  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 0.03903 on 192 degrees of freedom
## Multiple R-Squared: 0.07655, Adjusted R-squared: 0.02846 
## F-statistic: 1.592 on 10 and 192 DF,  p-value: 0.1115
##
##
## Estimation results for equation gold_search_scaled_FD:
## ======================================================
## gold_search_scaled_FD = gold_price_scaled_FD.l1 + gold_search_scaled_FD.l1 + gold_price_scaled_FD.l2
##
##                           Estimate Std. Error t value Pr(>|t|)    
## gold_price_scaled_FD.l1  -0.079646   0.115627  -0.689   0.4918    
## gold_search_scaled_FD.l1  0.533597   0.068666   7.771 4.59e-13 ***
## gold_price_scaled_FD.l2   0.226373   0.116912   1.936   0.0543 .  
## gold_search_scaled_FD.l2  0.009454   0.080296   0.118   0.9064    
## gold_price_scaled_FD.l3   0.216376   0.117339   1.844   0.0667 .  
## gold_search_scaled_FD.l3  0.044653   0.082010   0.544   0.5867    
## gold_price_scaled_FD.l4   0.163597   0.119777   1.366   0.1736    
## gold_search_scaled_FD.l4  0.093875   0.083995   1.118   0.2651    
## gold_price_scaled_FD.l5   0.018623   0.119462   0.156   0.8763    
## gold_search_scaled_FD.l5  0.386010   0.074361   5.191 5.30e-07 ***
## const                    -0.003414   0.006833  -0.500   0.6179    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 0.06293 on 192 degrees of freedom
## Multiple R-Squared: 0.8703,  Adjusted R-squared: 0.8636 
## F-statistic: 128.9 on 10 and 192 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                       gold_price_scaled_FD gold_search_scaled_FD
## gold_price_scaled_FD             1.523e-03             5.586e-06
## gold_search_scaled_FD            5.586e-06             3.960e-03
##
## Correlation matrix of residuals:
##                       gold_price_scaled_FD gold_search_scaled_FD
## gold_price_scaled_FD              1.000000              0.002274
## gold_search_scaled_FD             0.002274              1.000000
```

```
# augmented df test on only the differenced gold price
df_test_gold_price_FD <- urca::ur.df(gold_price_scaled_FD, type = ’none’,
                               selectlags = ’BIC’)
summary(df_test_gold_price_FD)
```

```
## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression none 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.139705 -0.013946  0.004984  0.026891  0.129077 
## 
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)    
## z.lag.1    -1.19119    0.10475 -11.372   <2e-16 ***
## z.diff.lag  0.06367    0.07010   0.908    0.365    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.0393 on 204 degrees of freedom
## Multiple R-squared:  0.562,  Adjusted R-squared:  0.5577 
## F-statistic: 130.9 on 2 and 204 DF,  p-value: < 2.2e-16
## 
## 
## Value of test-statistic is: -11.3722 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
# augmented df test on only the differenced gold search
df_test_gold_search_FD <- urca::ur.df(gold_search_scaled_FD, type = ’none’,
                               selectlags = ’BIC’)
summary(df_test_gold_search_FD)
```

```
## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression none 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30586 -0.01441  0.00017  0.01498  0.45713
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.01322    0.02221  -0.595  0.55242
## z.diff.lag -0.20661    0.07348  -2.812  0.00541 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.0725 on 204 degrees of freedom
## Multiple R-squared:  0.04635,    Adjusted R-squared:  0.037
## F-statistic: 4.958 on 2 and 204 DF,  p-value: 0.0079
##
##
## Value of test-statistic is: -0.5951
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

As the test rejects, given the data we cannot say that the data is not stationary. Which gives evidence for both being I(1).

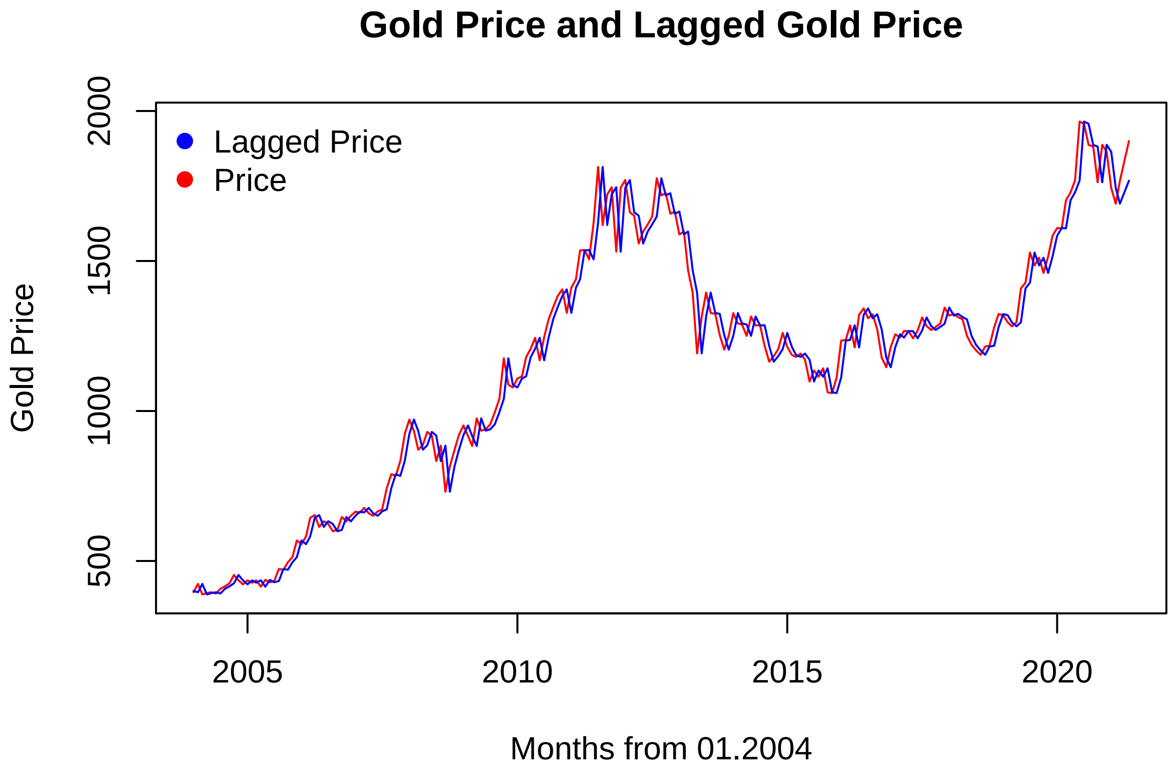
```r
# "Normal" VAR models with unscaled prices

# save variable vectors as time series format:
gold_price <- ts(gold_price, frequency = 12,
                        start = c(2004, 1), end = c(2021, 5))
gold_search <- ts(gold_search, frequency = 12,
                         start = c(2004,1), end = c(2021,5))

# set up data for estimation using ‘VAR()‘
VAR_data <- window(ts.union(gold_price, gold_search),
                   start = c(2004, 1), end = c(2021, 5))

# estimate model coefficients using ‘VAR()‘
VAR_est <- VAR(y = VAR_data, p = 5, type = ’both’)     # used lag order 5 without testing anything !!!!!
summary(VAR_est)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price, gold_search
## Deterministic variables: both
## Sample size: 204
## Log Likelihood: -1734.597
## Roots of the characteristic polynomial:
## 1.004 0.9764 0.8623 0.8623 0.7272 0.7272 0.5785 0.5785 0.4665 0.4665
## Call:
## VAR(y = VAR_data, p = 5, type = "both")
```

```
##
##
## Estimation results for equation gold_price:
## ===========================================
## gold_price = gold_price.l1 + gold_search.l1 + gold_price.l2 + gold_search.l2 + gold_price.l3 + gold_
##
##                Estimate Std. Error t value Pr(>|t|)
## gold_price.l1   0.86638    0.07300  11.868   <2e-16 ***
## gold_search.l1 -0.65623    0.86459  -0.759   0.4488
## gold_price.l2   0.05228    0.09603   0.544   0.5868
## gold_search.l2 -0.02053    0.99806  -0.021   0.9836
## gold_price.l3   0.09919    0.09550   1.039   0.3003
## gold_search.l3  0.12412    1.02105   0.122   0.9034
## gold_price.l4  -0.13158    0.09605  -1.370   0.1723
## gold_search.l4  2.04160    1.03963   1.964   0.0510 .
## gold_price.l5   0.08117    0.07463   1.088   0.2781
## gold_search.l5 -0.70911    0.95302  -0.744   0.4577
## const          29.38675   14.55194   2.019   0.0448 *
## trend           0.05721    0.13694   0.418   0.6766
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 62.3 on 192 degrees of freedom
## Multiple R-Squared: 0.977,   Adjusted R-squared: 0.9757
## F-statistic: 741.7 on 11 and 192 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gold_search:
## ============================================
## gold_search = gold_price.l1 + gold_search.l1 + gold_price.l2 + gold_search.l2 + gold_price.l3 + gold_
##
##                  Estimate Std. Error t value Pr(>|t|)
## gold_price.l1   0.0110561  0.0057789   1.913   0.0572 .
## gold_search.l1  0.5449368  0.0684403   7.962 1.45e-13 ***
## gold_price.l2   0.0003516  0.0076020   0.046   0.9632
## gold_search.l2 -0.0238542  0.0790058  -0.302   0.7630
## gold_price.l3  -0.0015551  0.0075594  -0.206   0.8372
## gold_search.l3  0.0066386  0.0808259   0.082   0.9346
## gold_price.l4  -0.0066729  0.0076032  -0.878   0.3812
## gold_search.l4  0.1005054  0.0822964   1.221   0.2235
## gold_price.l5  -0.0026836  0.0059077  -0.454   0.6502
## gold_search.l5  0.3692793  0.0754407   4.895 2.08e-06 ***
## const          -1.1808366  1.1519232  -1.025   0.3066
## trend           0.0106875  0.0108403   0.986   0.3254
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 4.932 on 192 degrees of freedom
## Multiple R-Squared: 0.8724,  Adjusted R-squared: 0.8651
## F-statistic: 119.4 on 11 and 192 DF,  p-value: < 2.2e-16
##
##
```

```
##
## Covariance matrix of residuals:
##             gold_price gold_search
## gold_price      3881.9      -19.90
## gold_search      -19.9       24.32
##
## Correlation matrix of residuals:
##             gold_price gold_search
## gold_price     1.00000    -0.06477
## gold_search   -0.06477     1.00000
```

```
#####################################################
####### Sollten wir hier beim AR(1) nicht die First differences verwenden.
# Weil wir ja einen I(1) prozess haben. Und sollten wir nicht einfach mit
# dem besteren ARMA modell arbeiten und nicht AR(1) ? #####
#####################################################

# compare the VAR to the AR(1) model for the prices
T <-length(gold_price)
gold_price_2 <- as.numeric(gold_price[-1])
gold_price_lagged <- as.numeric(gold_price[-T])

plot(y=gold_price_2,x=gold_pr$DATE[1-209], type = ’l’, lwd = 1, col = ’red’,
     main = ’Gold Price and Lagged Gold Price’,
     ylab = ’Gold Price’, xlab = ’Months from 01.2004’)
lines(y=gold_price_lagged,x=gold_pr$DATE[1-209], lwd = 1, col = ’blue’)
legend(’topleft’, legend = c(’Lagged Price’,’Price’),
       col = c(’blue’,’red’), bty = "n", pch = c(19,19))
```



```
# estimate model
gold_price_AR1 <- lm(gold_price_2 ~ gold_price_lagged)
# estimate robust standard errors
coeftest(gold_price_AR1, vcov. = vcovHC, type = "HC1")
```

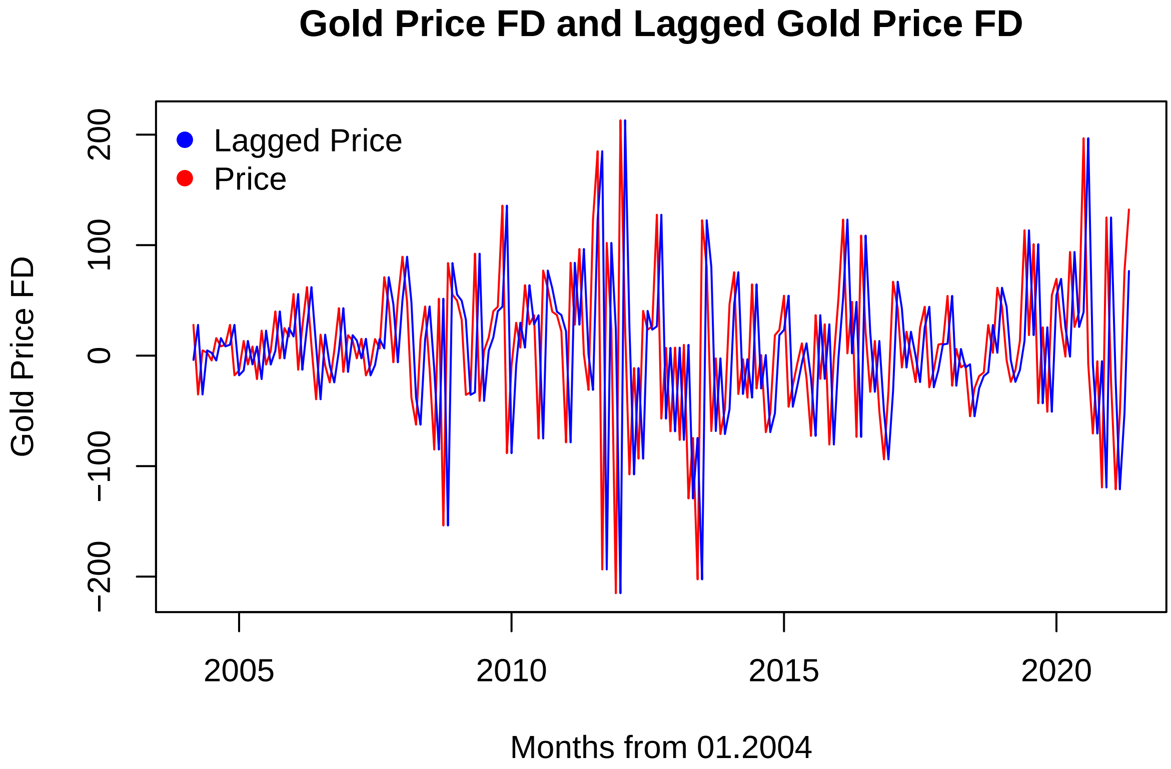
```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       20.49216   10.59726  1.9337  0.05452 .
## gold_price_lagged  0.98841    0.01114 88.7232  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
# "First-difference" VAR models with unscaled prices

# save variable vectors as time series format:
gold_price_FD <- ts(gold_price_FD, frequency = 12,
                        start = c(2004, 2), end = c(2021, 5))       #took 2004,2, because we are now de
gold_search_FD <- ts(gold_search_FD, frequency = 12,
                         start = c(2004,2), end = c(2021,5))

# set up data for estimation using ‘VAR()‘
VAR_data_FD <- window(ts.union(gold_price_FD, gold_search_FD),
                   start = c(2004, 2), end = c(2021, 5))

# estimate model coefficients using ‘VAR()‘
VAR_est_FD <- VAR(y = VAR_data_FD, p = 5, type = 'both')    # used lag order 5 without testing anything
summary(VAR_est_FD)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_FD, gold_search_FD
## Deterministic variables: both
## Sample size: 203
## Log Likelihood: -1725.2
## Roots of the characteristic polynomial:
## 0.8753 0.8753 0.701 0.701 0.6903 0.6903 0.6008 0.6008 0.5209 0.5175
## Call:
## VAR(y = VAR_data_FD, p = 5, type = "both")
##
##
## Estimation results for equation gold_price_FD:
## ==============================================
## gold_price_FD = gold_price_FD.l1 + gold_search_FD.l1 + gold_price_FD.l2 + gold_search_FD.l2 + gold_p
##
##                    Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1  -0.12029    0.07303  -1.647    0.101
## gold_search_FD.l1 -0.27550    0.92218  -0.299    0.765
## gold_price_FD.l2  -0.06214    0.07445  -0.835    0.405
## gold_search_FD.l2 -0.27834    0.98212  -0.283    0.777
## gold_price_FD.l3   0.03511    0.07461   0.471    0.639
## gold_search_FD.l3 -0.30539    1.00703  -0.303    0.762
## gold_price_FD.l4  -0.09339    0.07599  -1.229    0.221
## gold_search_FD.l4  1.53168    1.04556   1.465    0.145
## gold_price_FD.l5   0.04520    0.07540   0.600    0.550
## gold_search_FD.l5  1.40669    0.98117   1.434    0.153
## const             10.15844    9.36052   1.085    0.279
## trend             -0.01704    0.07733  -0.220    0.826
##
##
## Residual standard error: 62.58 on 191 degrees of freedom
## Multiple R-Squared: 0.06793, Adjusted R-squared: 0.01425
## F-statistic: 1.265 on 11 and 191 DF,  p-value: 0.2472
##
##
## Estimation results for equation gold_search_FD:
## ===============================================
## gold_search_FD = gold_price_FD.l1 + gold_search_FD.l1 + gold_price_FD.l2 + gold_search_FD.l2 + gold_
##
##                     Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1   0.011787   0.005705   2.066   0.0402 *
## gold_search_FD.l1 -0.497041   0.072032  -6.900 7.43e-11 ***
## gold_price_FD.l2   0.009622   0.005816   1.655   0.0997 .
## gold_search_FD.l2 -0.533478   0.076714  -6.954 5.48e-11 ***
## gold_price_FD.l3   0.010226   0.005828   1.755   0.0809 .
## gold_search_FD.l3 -0.532586   0.078660  -6.771 1.54e-10 ***
## gold_price_FD.l4   0.004996   0.005935   0.842   0.4010
## gold_search_FD.l4 -0.451789   0.081670  -5.532 1.03e-07 ***
## gold_price_FD.l5   0.007956   0.005889   1.351   0.1783
## gold_search_FD.l5 -0.138608   0.076640  -1.809   0.0721 .
## const             -1.045051   0.731155  -1.429   0.1545
## trend              0.014756   0.006041   2.443   0.0155 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 4.888 on 191 degrees of freedom
## Multiple R-Squared: 0.3364,  Adjusted R-squared: 0.2982
## F-statistic: 8.802 on 11 and 191 DF,  p-value: 1.412e-12
##
##
##
## Covariance matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        3916.56         -18.13
## gold_search_FD        -18.13          23.90
##
## Correlation matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        1.00000       -0.05925
## gold_search_FD      -0.05925        1.00000
```

```
# compare the VAR to the AR(1) model for the prices first-differences
T <-length(gold_price_FD)
gold_price_FD_2 <- as.numeric(gold_price_FD[-1])
```

```
gold_price_FD_lagged <- as.numeric(gold_price_FD[-T])

plot(y=gold_price_FD_2,x=gold_pr$DATE[3:209], type = 'l', lwd = 1, col = 'red',
     main = 'Gold Price FD and Lagged Gold Price FD',
     ylab = 'Gold Price FD', xlab = 'Months from 01.2004')
lines(y=gold_price_FD_lagged,x=gold_pr$DATE[3:209], lwd = 1, col = 'blue')
legend('topleft', legend = c('Lagged Price','Price'),
       col = c('blue','red'), bty = "n", pch = c(19,19))
```



```
# estimate model
gold_price_FD_AR1 <- lm(gold_price_FD_2 ~ gold_price_FD_lagged)
# estimate robust standard errors
coeftest(gold_price_FD_AR1, vcov. = vcovHC, type = "HC1")
```

```
##
## t test of coefficients:
##
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           8.066698   4.377330  1.8428   0.0668 .
## gold_price_FD_lagged -0.121139   0.096469 -1.2557   0.2106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The values on the intercept seem to differ, but the estimated coefficient on the lag seems to fit.

```
# verify the 'by-hand' results with built-in function for ARIMA
ar.ols(gold_price, order.max = 5, intercept = T)
```

```
##
## Call:
## ar.ols(x = gold_price, order.max = 5, intercept = T)
##
## Coefficients:
##      1
## 0.9884
##
## Intercept: 7.171 (4.301)
##
## Order selected 1  sigma^2 estimated as  3848
```

```
forecast::auto.arima(gold_price, ic = 'aic')
```

```
## Series: gold_price
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.1411  7.1279
## s.e.   0.0740  3.6766
##
## sigma^2 estimated as 3842:  log likelihood=-1152.52
## AIC=2311.05   AICc=2311.16   BIC=2321.06
```

```
#for FD
ar.ols(gold_price_FD, order.max = 5, intercept = T) #just verifies the above ARIMA(0,1,1)
```

```
##
## Call:
## ar.ols(x = gold_price_FD, order.max = 5, intercept = T)
##
## Coefficients:
##       1
## -0.1211
##
## Intercept: -0.01952 (4.303)
##
## Order selected 1  sigma^2 estimated as  3832
```

```
forecast::auto.arima(gold_price_FD, ic = 'aic')
```

```
## Series: gold_price_FD
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##           ma1    mean
##       -0.1411  7.1279
## s.e.   0.0740  3.6766
##
## sigma^2 estimated as 3842:  log likelihood=-1152.52
## AIC=2311.05   AICc=2311.16   BIC=2321.06
```

The model is automated to difference such that the data is stationary, then the function finds the best forecasting model via the AIC. Here this would be an ARMA(0,1) model:

$$\Delta\widehat{\text{gold price}}_t = \underset{3.6766}{(7.1279)} + \epsilon_t + \underset{(0.0740)}{(-0.1411)}\epsilon_{t-1}$$

```r
# estimate model coefficients using AIC

VAR_lag <- VAR(y = VAR_data, type = 'both', ic = 'AIC', lag.max = 15)
summary(VAR_lag)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price, gold_search
## Deterministic variables: both
## Sample size: 198
## Log Likelihood: -1654.705
## Roots of the characteristic polynomial:
## 0.9968 0.9968 0.9901 0.9901 0.9211 0.9211 0.9195 0.9195 0.9102 0.9102 0.8787 0.8416 0.8416 0.8333 0.8
## Call:
## VAR(y = VAR_data, type = "both", lag.max = 15, ic = "AIC")
##
##
## Estimation results for equation gold_price:
## ===========================================
## gold_price = gold_price.l1 + gold_search.l1 + gold_price.l2 + gold_search.l2 + gold_price.l3 + gold_s
##
##                  Estimate Std. Error t value Pr(>|t|)
## gold_price.l1     0.87441    0.07633  11.455  < 2e-16 ***
## gold_search.l1   -0.09896    0.97900  -0.101  0.91960
## gold_price.l2     0.08168    0.09980   0.818  0.41422
## gold_search.l2    0.43364    1.07400   0.404  0.68689
## gold_price.l3     0.09959    0.10068   0.989  0.32394
## gold_search.l3   -0.67663    1.08498  -0.624  0.53369
## gold_price.l4    -0.15542    0.10093  -1.540  0.12541
## gold_search.l4    1.87254    1.10489   1.695  0.09191 .
## gold_price.l5     0.10802    0.09939   1.087  0.27862
## gold_search.l5   -0.79868    1.17455  -0.680  0.49741
## gold_price.l6     0.16395    0.09995   1.640  0.10275
## gold_search.l6    0.19169    1.21396   0.158  0.87471
## gold_price.l7    -0.29844    0.10264  -2.908  0.00412 **
## gold_search.l7   -1.33615    1.13699  -1.175  0.24153
## gold_price.l8     0.05090    0.10696   0.476  0.63472
## gold_search.l8   -0.32111    1.15305  -0.278  0.78097
## gold_price.l9    -0.10628    0.10652  -0.998  0.31980
## gold_search.l9    1.60014    1.16680   1.371  0.17202
## gold_price.l10    0.23204    0.10700   2.169  0.03147 *
## gold_search.l10   0.65500    1.36858   0.479  0.63283
## gold_price.l11   -0.08451    0.08375  -1.009  0.31435
## gold_search.l11  -0.63300    1.25273  -0.505  0.61399
## const            29.87096   15.55263   1.921  0.05641 .
## trend             0.04859    0.15009   0.324  0.74651
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 61.6 on 174 degrees of freedom
## Multiple R-Squared: 0.9772,  Adjusted R-squared: 0.9741
## F-statistic: 323.7 on 23 and 174 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gold_search:
## ============================================
## gold_search = gold_price.l1 + gold_search.l1 + gold_price.l2 + gold_search.l2 + gold_price.l3 + gold_
##
##                  Estimate Std. Error t value Pr(>|t|)
## gold_price.l1    0.008780   0.005712   1.537 0.126067
## gold_search.l1   0.494927   0.073254   6.756 2.05e-10 ***
## gold_price.l2   -0.005736   0.007468  -0.768 0.443507
## gold_search.l2  -0.067768   0.080363  -0.843 0.400230
## gold_price.l3    0.005739   0.007533   0.762 0.447230
## gold_search.l3   0.060180   0.081184   0.741 0.459524
## gold_price.l4   -0.004481   0.007552  -0.593 0.553680
## gold_search.l4   0.082621   0.082673   0.999 0.319003
## gold_price.l5    0.005676   0.007437   0.763 0.446357
## gold_search.l5   0.373837   0.087886   4.254 3.43e-05 ***
## gold_price.l6   -0.011852   0.007479  -1.585 0.114828
## gold_search.l6  -0.039448   0.090834  -0.434 0.664616
## gold_price.l7    0.022381   0.007680   2.914 0.004037 **
## gold_search.l7   0.137440   0.085075   1.616 0.108013
## gold_price.l8   -0.011973   0.008003  -1.496 0.136450
## gold_search.l8  -0.080319   0.086277  -0.931 0.353176
## gold_price.l9    0.001943   0.007970   0.244 0.807708
## gold_search.l9  -0.044945   0.087306  -0.515 0.607348
## gold_price.l10  -0.004495   0.008006  -0.561 0.575249
## gold_search.l10 -0.241279   0.102404  -2.356 0.019579 *
## gold_price.l11  -0.006814   0.006266  -1.087 0.278368
## gold_search.l11  0.342986   0.093736   3.659 0.000335 ***
## const           -0.991835   1.163729  -0.852 0.395225
## trend            0.018826   0.011231   1.676 0.095485 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.609 on 174 degrees of freedom
## Multiple R-Squared: 0.8962,  Adjusted R-squared: 0.8824
## F-statistic: 65.29 on 23 and 174 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             gold_price gold_search
## gold_price    3793.969       4.728
## gold_search      4.728      21.242
##
## Correlation matrix of residuals:
##             gold_price gold_search
## gold_price     1.00000     0.01665
## gold_search    0.01665     1.00000
```

```
VAR_lag_select <- VARselect(y = VAR_data, type = 'both', lag.max = 15)
VAR_lag_select
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     11      5      1     11
##
## $criteria
##                   1            2            3            4            5
## AIC(n)      11.82853     11.82522     11.82092     11.75218     11.67071
## HQ(n)       11.88309     11.90707     11.93005     11.88860     11.83441
## SC(n)       11.96328     12.02736     12.09043     12.08907     12.07498
## FPE(n) 137110.04209 136661.36323 136081.91073 127053.33408 117127.87221
##                   6            7            8            9           10
## AIC(n)      11.67736     11.66562     11.61794     11.61773     11.61210
## HQ(n)       11.86835     11.88389     11.86349     11.89056     11.91222
## SC(n)       12.14901     12.20465     12.22435     12.29152     12.35327
## FPE(n) 117932.21775 116584.51726 111191.70935 111212.74329 110642.66475
##                  11           12           13           14           15
## AIC(n)      11.56747     11.56862     11.58512     11.60526     11.59713
## HQ(n)       11.89487     11.92331     11.96709     12.01451     12.03366
## SC(n)       12.37601     12.44454     12.52842     12.61594     12.67518
## FPE(n) 105875.58850 106071.11783 107923.72935 110223.26619 109448.48022
```

```
#for FD
VAR_lag_FD <- VAR(y = VAR_data_FD, type = 'both', ic = 'AIC', lag.max = 15)
summary(VAR_lag_FD)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_FD, gold_search_FD
## Deterministic variables: both
## Sample size: 198
## Log Likelihood: -1656.605
## Roots of the characteristic polynomial:
## 0.9889 0.9889 0.9207 0.9207 0.9202 0.9202 0.9084 0.9084 0.8755 0.8379 0.8379 0.8311 0.8311 0.8191 0.8
## Call:
## VAR(y = VAR_data_FD, type = "both", lag.max = 15, ic = "AIC")
##
##
## Estimation results for equation gold_price_FD:
## ==============================================
## gold_price_FD = gold_price_FD.l1 + gold_search_FD.l1 + gold_price_FD.l2 + gold_search_FD.l2 + gold_pr
##
##                   Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1  -0.11056    0.07605  -1.454   0.1478
## gold_search_FD.l1 -0.08358    0.98071  -0.085   0.9322
## gold_price_FD.l2  -0.02939    0.07718  -0.381   0.7038
## gold_search_FD.l2  0.40212    1.10592   0.364   0.7166
## gold_price_FD.l3    0.06943    0.07658   0.907   0.3658  
## gold_search_FD.l3  -0.28147    1.22923  -0.229   0.8191  
## gold_price_FD.l4   -0.08251    0.07761  -1.063   0.2892  
## gold_search_FD.l4   1.52539    1.30402   1.170   0.2437  
## gold_price_FD.l5    0.02844    0.07693   0.370   0.7121  
## gold_search_FD.l5   0.75620    1.38864   0.545   0.5867  
## gold_price_FD.l6    0.19098    0.07760   2.461   0.0148 *
## gold_search_FD.l6   0.96453    1.34815   0.715   0.4753  
## gold_price_FD.l7   -0.10576    0.07960  -1.329   0.1857  
## gold_search_FD.l7  -0.52307    1.30032  -0.402   0.6880  
## gold_price_FD.l8   -0.05299    0.08216  -0.645   0.5198  
## gold_search_FD.l8  -0.95842    1.22935  -0.780   0.4367  
## gold_price_FD.l9   -0.15845    0.08239  -1.923   0.0561 .
## gold_search_FD.l9   0.62239    1.21170   0.514   0.6081  
## gold_price_FD.l10   0.07567    0.08355   0.906   0.3663  
## gold_search_FD.l10  1.00005    1.16074   0.862   0.3901  
## const               9.60691   10.13961   0.947   0.3447  
## trend              -0.01649    0.08512  -0.194   0.8466  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## 
## Residual standard error: 61.78 on 176 degrees of freedom
## Multiple R-Squared: 0.1622,  Adjusted R-squared: 0.06226 
## F-statistic: 1.623 on 21 and 176 DF,  p-value: 0.04857 
## 
## 
## Estimation results for equation gold_search_FD: 
## =============================================== 
## gold_search_FD = gold_price_FD.l1 + gold_search_FD.l1 + gold_price_FD.l2 + gold_search_FD.l2 + gold_
## 
##                      Estimate Std. Error t value Pr(>|t|)    
## gold_price_FD.l1    0.009149   0.005646   1.620 0.106924    
## gold_search_FD.l1  -0.504513   0.072807  -6.929 7.67e-11 ***
## gold_price_FD.l2    0.003424   0.005730   0.598 0.550891    
## gold_search_FD.l2  -0.571249   0.082102  -6.958 6.55e-11 ***
## gold_price_FD.l3    0.009154   0.005685   1.610 0.109122    
## gold_search_FD.l3  -0.511228   0.091257  -5.602 8.03e-08 ***
## gold_price_FD.l4    0.004738   0.005762   0.822 0.412051    
## gold_search_FD.l4  -0.429967   0.096809  -4.441 1.58e-05 ***
## gold_price_FD.l5    0.010462   0.005711   1.832 0.068667 .  
## gold_search_FD.l5  -0.055661   0.103092  -0.540 0.589939    
## gold_price_FD.l6   -0.001426   0.005761  -0.247 0.804819    
## gold_search_FD.l6  -0.095101   0.100085  -0.950 0.343312    
## gold_price_FD.l7    0.020980   0.005909   3.550 0.000493 ***
## gold_search_FD.l7   0.039133   0.096535   0.405 0.685689    
## gold_price_FD.l8    0.009038   0.006100   1.482 0.140196    
## gold_search_FD.l8  -0.043579   0.091266  -0.477 0.633599    
## gold_price_FD.l9    0.011006   0.006117   1.799 0.073681 .  
## gold_search_FD.l9  -0.088944   0.089955  -0.989 0.324139    
## gold_price_FD.l10   0.006574   0.006202   1.060 0.290611    
## gold_search_FD.l10 -0.335833   0.086173  -3.897 0.000138 ***
## const              -1.470108   0.752756  -1.953 0.052409 .  
## trend               0.016808   0.006319   2.660 0.008540 ** 
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 4.587 on 176 degrees of freedom
## Multiple R-Squared: 0.4615,  Adjusted R-squared: 0.3972
## F-statistic: 7.181 on 21 and 176 DF,  p-value: 7.4e-15
##
##
##
## Covariance matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD       3817.324          6.271
## gold_search_FD         6.271         21.039
##
## Correlation matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        1.00000        0.02213
## gold_search_FD       0.02213        1.00000
```

```
VAR_lag_FD_select <- VARselect(y = VAR_data_FD, type = ’both’, lag.max = 15)
VAR_lag_FD_select
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      4      4     10
##
## $criteria
##                     1            2            3            4            5
## AIC(n)       11.89902     11.85386     11.75817     11.66242     11.66707
## HQ(n)        11.95379     11.93601     11.86771     11.79934     11.83138
## SC(n)        12.03426     12.05672     12.02865     12.00052     12.07280
## FPE(n) 147124.09896 140631.40080 127805.39947 116146.35986 116703.74827
##                     6            7            8            9           10
## AIC(n)       11.66085     11.60882     11.61500     11.61547     11.56011
## HQ(n)        11.85254     11.82789     11.86146     11.88931     11.86133
## SC(n)        12.13419     12.14978     12.22359     12.29167     12.30393
## FPE(n) 116001.84875 110147.95111 110867.37695 110963.87937 105040.07209
##                    11           12           13           14           15
## AIC(n)       11.56529     11.58768     11.60751     11.60284     11.62814
## HQ(n)        11.89390     11.94367     11.99088     12.01361     12.06629
## SC(n)        12.37674     12.46674     12.55419     12.61715     12.71007
## FPE(n) 105649.00712 108117.44187 110373.90910 109965.81854 112907.13064
```

```
#Problem! All roots are inside the unit circle --> unstable. This is why we try it without a trend -->
```

```
VAR_lag <- VAR(y = VAR_data, type = ’const’, ic = ’AIC’, lag.max = 15)
summary(VAR_lag)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price, gold_search
## Deterministic variables: const
## Sample size: 198
## Log Likelihood: -1656.34
## Roots of the characteristic polynomial:
##  1.03 0.9916 0.9916 0.9802 0.9221 0.9221 0.9201 0.9201 0.9151 0.9151 0.878 0.8434 0.8434 0.8344 0.834
## Call:
## VAR(y = VAR_data, type = "const", lag.max = 15, ic = "AIC")
##
##
## Estimation results for equation gold_price:
## ===========================================
## gold_price = gold_price.l1 + gold_search.l1 + gold_price.l2 + gold_search.l2 + gold_price.l3 + gold_s
##
##                  Estimate Std. Error t value Pr(>|t|)
## gold_price.l1    0.87513    0.07610  11.499  < 2e-16 ***
## gold_search.l1  -0.05621    0.96757  -0.058  0.95374
## gold_price.l2    0.08278    0.09949   0.832  0.40654
## gold_search.l2   0.43843    1.07115   0.409  0.68281
## gold_price.l3    0.10007    0.10041   0.997  0.32032
## gold_search.l3  -0.66179    1.08123  -0.612  0.54128
## gold_price.l4   -0.15626    0.10064  -1.553  0.12229
## gold_search.l4   1.88889    1.10091   1.716  0.08798 .
## gold_price.l5    0.10651    0.09902   1.076  0.28357
## gold_search.l5  -0.79764    1.17153  -0.681  0.49687
## gold_price.l6    0.16350    0.09969   1.640  0.10278
## gold_search.l6   0.16789    1.20862   0.139  0.88968
## gold_price.l7   -0.29871    0.10238  -2.918  0.00399 **
## gold_search.l7  -1.32694    1.13372  -1.170  0.24342
## gold_price.l8    0.04955    0.10660   0.465  0.64263
## gold_search.l8  -0.31966    1.15009  -0.278  0.78139
## gold_price.l9   -0.10611    0.10625  -0.999  0.31932
## gold_search.l9   1.59749    1.16378   1.373  0.17161
## gold_price.l10   0.23276    0.10670   2.181  0.03049 *
## gold_search.l10  0.68828    1.36122   0.506  0.61375
## gold_price.l11  -0.08065    0.08268  -0.975  0.33071
## gold_search.l11 -0.56506    1.23187  -0.459  0.64701
## const           30.00235   15.50751   1.935  0.05464 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 61.44 on 175 degrees of freedom
## Multiple R-Squared: 0.9771,  Adjusted R-squared: 0.9743
## F-statistic: 340.1 on 22 and 175 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gold_search:
## ============================================
## gold_search = gold_price.l1 + gold_search.l1 + gold_price.l2 + gold_search.l2 + gold_price.l3 + gold_
##
##                   Estimate Std. Error t value Pr(>|t|)
## gold_price.l1     0.009061   0.005739   1.579 0.116152
## gold_search.l1    0.511486   0.072959   7.011 4.95e-11 ***
## gold_price.l2   -0.005312   0.007502  -0.708 0.479821
## gold_search.l2  -0.065911   0.080769  -0.816 0.415590
## gold_price.l3    0.005925   0.007571   0.783 0.434954
## gold_search.l3   0.065926   0.081529   0.809 0.419832
## gold_price.l4   -0.004810   0.007589  -0.634 0.527047
## gold_search.l4   0.088957   0.083013   1.072 0.285376
## gold_price.l5    0.005094   0.007467   0.682 0.496030
## gold_search.l5   0.374240   0.088339   4.236 3.67e-05 ***
## gold_price.l6   -0.012029   0.007517  -1.600 0.111336
## gold_search.l6  -0.048670   0.091135  -0.534 0.593991
## gold_price.l7    0.022277   0.007720   2.886 0.004397 **
## gold_search.l7   0.141009   0.085487   1.649 0.100845
## gold_price.l8   -0.012497   0.008038  -1.555 0.121829
## gold_search.l8  -0.079754   0.086721  -0.920 0.359020
## gold_price.l9    0.002008   0.008011   0.251 0.802349
## gold_search.l9  -0.045973   0.087754  -0.524 0.601024
## gold_price.l10  -0.004215   0.008046  -0.524 0.601041
## gold_search.l10 -0.228383   0.102642  -2.225 0.027355 *
## gold_price.l11  -0.005319   0.006235  -0.853 0.394709
## gold_search.l11  0.369307   0.092888   3.976 0.000102 ***
## const           -0.940933   1.169333  -0.805 0.422099
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.633 on 175 degrees of freedom
## Multiple R-Squared: 0.8945,   Adjusted R-squared: 0.8812
## F-statistic: 67.43 on 22 and 175 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             gold_price gold_search
## gold_price    3774.561       5.581
## gold_search      5.581      21.461
##
## Correlation matrix of residuals:
##             gold_price gold_search
## gold_price     1.00000     0.01961
## gold_search    0.01961     1.00000
```

```
VAR_lag_select <- VARselect(y = VAR_data, type = 'const', lag.max = 15)
VAR_lag_select
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     11      5      1     11
##
## $criteria
##                   1            2            3            4            5
## AIC(n)     11.83777     11.83070     11.81954     11.74277     11.65647
## HQ(n)      11.87869     11.89891     11.91504     11.86554     11.80653
## SC(n)      11.93884     11.99914     12.05537     12.04597     12.02705
## FPE(n) 138382.06056 137409.34210 135890.80282 125856.97520 115463.95383
##                 6            7            8            9           10
## AIC(n)     11.66376     11.65285     11.61132     11.61462     11.61814
## HQ(n)      11.84110     11.85747     11.84323     11.87381     11.90461
## SC(n)      12.10172     12.15819     12.18404     12.25472     12.32561
## FPE(n) 116326.78161 115089.66397 110439.76870 110844.30016 111283.75901
##                11           12           13           14           15
## AIC(n)     11.56397     11.56907     11.58852     11.60400     11.60497
## HQ(n)      11.87773     11.91012     11.95685     11.99961     12.02786
## SC(n)      12.33882     12.41130     12.49813     12.58099     12.64933
## FPE(n) 105472.55908 106080.53909 108245.81312 110030.76954 110248.54472
```

```
#for FD
VAR_lag_FD <- VAR(y = VAR_data_FD, type = 'const', ic = 'AIC', lag.max = 15)
summary(VAR_lag_FD)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_FD, gold_search_FD
## Deterministic variables: const
## Sample size: 198
## Log Likelihood: -1660.541
## Roots of the characteristic polynomial:
## 0.9857 0.9857 0.9186 0.9186 0.9167 0.9167 0.9003 0.9003 0.8751 0.8348 0.8348 0.8216 0.8216 0.8193 0.8
## Call:
## VAR(y = VAR_data_FD, type = "const", lag.max = 15, ic = "AIC")
##
##
## Estimation results for equation gold_price_FD:
## ==============================================
## gold_price_FD = gold_price_FD.l1 + gold_search_FD.l1 + gold_price_FD.l2 + gold_search_FD.l2 + gold_p
##
##                    Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1   -0.11001    0.07579  -1.451   0.1484
## gold_search_FD.l1  -0.11483    0.96472  -0.119   0.9054
## gold_price_FD.l2   -0.02844    0.07681  -0.370   0.7116
## gold_search_FD.l2   0.35291    1.07343   0.329   0.7427
## gold_price_FD.l3    0.07057    0.07614   0.927   0.3552
## gold_search_FD.l3  -0.34611    1.17986  -0.293   0.7696
## gold_price_FD.l4   -0.08162    0.07726  -1.056   0.2922
## gold_search_FD.l4   1.45542    1.24960   1.165   0.2457
## gold_price_FD.l5    0.02949    0.07653   0.385   0.7005
## gold_search_FD.l5   0.66982    1.31153   0.511   0.6102
## gold_price_FD.l6    0.19237    0.07705   2.497   0.0135 *
## gold_search_FD.l6   0.87667    1.26613   0.692   0.4896
## gold_price_FD.l7   -0.10507    0.07930  -1.325   0.1869
## gold_search_FD.l7  -0.59765    1.23864  -0.483   0.6300
## gold_price_FD.l8   -0.05174    0.08169  -0.633   0.5273
## gold_search_FD.l8  -1.01551    1.19026  -0.853   0.3947
## gold_price_FD.l9   -0.15695    0.08181  -1.919   0.0566 .
## gold_search_FD.l9   0.57031    1.17828   0.484   0.6290
## gold_price_FD.l10   0.07740    0.08284   0.934   0.3514
## gold_search_FD.l10  0.97309    1.14923   0.847   0.3983
## const               7.88628    4.87849   1.617   0.1078
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 61.62 on 177 degrees of freedom
## Multiple R-Squared: 0.162,   Adjusted R-squared: 0.06736
## F-statistic: 1.711 on 20 and 177 DF,  p-value: 0.03509
##
##
## Estimation results for equation gold_search_FD:
## ==================================================
## gold_search_FD = gold_price_FD.l1 + gold_search_FD.l1 + gold_price_FD.l2 + gold_search_FD.l2 + gold_
##
##                    Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1    0.008587   0.005738   1.496 0.136312
## gold_search_FD.l1  -0.472657   0.073038  -6.471 9.21e-10 ***
## gold_price_FD.l2    0.002451   0.005815   0.421 0.673902
## gold_search_FD.l2  -0.521095   0.081268  -6.412 1.26e-09 ***
## gold_price_FD.l3    0.007988   0.005764   1.386 0.167578
## gold_search_FD.l3  -0.445345   0.089325  -4.986 1.46e-06 ***
## gold_price_FD.l4    0.003826   0.005850   0.654 0.513886
## gold_search_FD.l4  -0.358655   0.094605  -3.791 0.000206 ***
## gold_price_FD.l5    0.009390   0.005794   1.621 0.106844
## gold_search_FD.l5   0.032388   0.099294   0.326 0.744673
## gold_price_FD.l6   -0.002851   0.005834  -0.489 0.625674
## gold_search_FD.l6  -0.005552   0.095857  -0.058 0.953877
## gold_price_FD.l7    0.020280   0.006004   3.378 0.000898 ***
## gold_search_FD.l7   0.115154   0.093775   1.228 0.221086
## gold_price_FD.l8    0.007763   0.006184   1.255 0.211047
## gold_search_FD.l8   0.014607   0.090113   0.162 0.871413
## gold_price_FD.l9    0.009474   0.006193   1.530 0.127860
## gold_search_FD.l9  -0.035861   0.089206  -0.402 0.688171
## gold_price_FD.l10   0.004807   0.006272   0.767 0.444377
## gold_search_FD.l10 -0.308345   0.087006  -3.544 0.000504 ***
## const               0.283681   0.369343   0.768 0.443470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.665 on 177 degrees of freedom
## Multiple R-Squared: 0.4398,  Adjusted R-squared: 0.3765
## F-statistic: 6.948 on 20 and 177 DF,  p-value: 5.647e-14
##
##
##
## Covariance matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        3796.57           5.41
## gold_search_FD          5.41          21.76
##
## Correlation matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        1.00000        0.01882
## gold_search_FD       0.01882        1.00000
```

```
VAR_lag_FD_select <- VARselect(y = VAR_data_FD, type = ’const’, lag.max = 15)
VAR_lag_FD_select
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n) 
##     10      4      4     10 
## 
## $criteria
##                   1            2            3            4            5
## AIC(n)     11.87983     11.83746     11.74978     11.66585     11.67760
## HQ(n)      11.92090     11.90592     11.84563     11.78908     11.82821
## SC(n)      11.98126     12.00651     11.98645     11.97014     12.04951
## FPE(n) 144326.02180 138341.25377 126733.82294 116539.10567 117930.03872
##                   6            7            8            9           10
## AIC(n)     11.67244     11.61517     11.61999     11.62050     11.58150
## HQ(n)      11.85044     11.82055     11.85276     11.88065     11.86903
## SC(n)      12.11197     12.12233     12.19477     12.26290     12.29152
## FPE(n) 117341.99966 110835.17776 111403.09662 111500.44501 107283.44156
##                  11           12           13           14           15
## AIC(n)     11.58501     11.59947     11.61361     11.60789     11.63345
## HQ(n)      11.89993     11.94177     11.98330     12.00496     12.05791
## SC(n)      12.36265     12.44473     12.52649     12.58839     12.68157
## FPE(n) 107719.62262 109359.83845 111002.49194 110467.49347 113444.11328
```

```
#Force it to 1 lag

# estimate model coefficients using AIC

VAR_lag <- VAR(y = VAR_data, type = ’both’, p=1)
summary(VAR_lag)
```

```
## 
## VAR Estimation Results:
## ========================= 
## Endogenous variables: gold_price, gold_search 
## Deterministic variables: both 
## Sample size: 208 
## Log Likelihood: -1799.017 
## Roots of the characteristic polynomial:
## 0.9694 0.7759
## Call:
## VAR(y = VAR_data, p = 1, type = "both")
## 
## 
## Estimation results for equation gold_price: 
## =========================================== 
## gold_price = gold_price.l1 + gold_search.l1 + const + trend 
## 
##                 Estimate Std. Error t value Pr(>|t|)    
## gold_price.l1    0.96788    0.01798  53.837   <2e-16 ***
## gold_search.l1   0.10532    0.50011   0.211   0.8334    
## const           25.95499   13.78646   1.883   0.0612 .  
## trend           0.15617    0.13017   1.200   0.2317
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 62.32 on 204 degrees of freedom
## Multiple R-Squared: 0.9772,  Adjusted R-squared: 0.9769
## F-statistic:  2913 on 3 and 204 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gold_search:
## ============================================
## gold_search = gold_price.l1 + gold_search.l1 + const + trend
##
##                 Estimate Std. Error t value Pr(>|t|)
## gold_price.l1   0.002848   0.001578   1.805   0.0725 .
## gold_search.l1  0.777456   0.043888  17.715   <2e-16 ***
## const          -2.158855   1.209834  -1.784   0.0758 .
## trend           0.023485   0.011423   2.056   0.0411 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 5.469 on 204 degrees of freedom
## Multiple R-Squared: 0.8359,  Adjusted R-squared: 0.8335
## F-statistic: 346.5 on 3 and 204 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             gold_price gold_search
## gold_price      3883.7      -12.50
## gold_search      -12.5       29.91
##
## Correlation matrix of residuals:
##             gold_price gold_search
## gold_price     1.00000    -0.03669
## gold_search   -0.03669     1.00000
```

```
#for FD
VAR_lag_FD <- VAR(y = VAR_data_FD, type = 'both', p=1)
summary(VAR_lag_FD)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_FD, gold_search_FD
## Deterministic variables: both
## Sample size: 207
## Log Likelihood: -1796.533
## Roots of the characteristic polynomial:
## 0.1819 0.1819
## Call:
## VAR(y = VAR_data_FD, p = 1, type = "both")
```

```
##
##
## Estimation results for equation gold_price_FD:
## ==============================================
## gold_price_FD = gold_price_FD.l1 + gold_search_FD.l1 + const + trend
##
##                    Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1  -0.123089   0.070274  -1.752   0.0814 .
## gold_search_FD.l1 -0.662835   0.755885  -0.877   0.3816
## const              8.352675   8.788579   0.950   0.3430
## trend             -0.001137   0.072641  -0.016   0.9875
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 62.39 on 203 degrees of freedom
## Multiple R-Squared: 0.01812, Adjusted R-squared: 0.003612
## F-statistic: 1.249 on 3 and 203 DF,  p-value: 0.2931
##
##
## Estimation results for equation gold_search_FD:
## ===============================================
## gold_search_FD = gold_price_FD.l1 + gold_search_FD.l1 + const + trend
##
##                    Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1   0.016363   0.006338   2.582  0.01054 *
## gold_search_FD.l1 -0.180752   0.068176  -2.651  0.00865 **
## const             -0.250411   0.792679  -0.316  0.75240
## trend              0.003687   0.006552   0.563  0.57427
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 5.627 on 203 degrees of freedom
## Multiple R-Squared: 0.06579, Adjusted R-squared: 0.05199
## F-statistic: 4.766 on 3 and 203 DF,  p-value: 0.003117
##
##
##
## Covariance matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD         3892.9         -11.90
## gold_search_FD         -11.9          31.67
##
## Correlation matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD         1.0000        -0.0339
## gold_search_FD       -0.0339         1.0000
```

```r
VAR_lag <- VAR(y = VAR_data, type = ’const’, p=1)
summary(VAR_lag)
```

```
##
## VAR Estimation Results:
```

```
## =========================
## Endogenous variables: gold_price, gold_search
## Deterministic variables: const
## Sample size: 208
## Log Likelihood: -1801.96
## Roots of the characteristic polynomial:
## 0.9901 0.8059
## Call:
## VAR(y = VAR_data, p = 1, type = "const")
##
##
## Estimation results for equation gold_price:
## ===========================================
## gold_price = gold_price.l1 + gold_search.l1 + const
##
##                Estimate Std. Error t value Pr(>|t|)
## gold_price.l1   0.98036    0.01467  66.810   <2e-16 ***
## gold_search.l1  0.35938    0.45355   0.792   0.4291
## const          24.25247   13.72791   1.767   0.0788 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 62.39 on 205 degrees of freedom
## Multiple R-Squared: 0.977,   Adjusted R-squared: 0.9768
## F-statistic:  4359 on 2 and 205 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gold_search:
## ============================================
## gold_search = gold_price.l1 + gold_search.l1 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## gold_price.l1   0.004726   0.001296   3.646 0.000338 ***
## gold_search.l1  0.815662   0.040070  20.356  < 2e-16 ***
## const          -2.414886   1.212840  -1.991 0.047798 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 5.512 on 205 degrees of freedom
## Multiple R-Squared: 0.8325,  Adjusted R-squared: 0.8309
## F-statistic: 509.6 on 2 and 205 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             gold_price gold_search
## gold_price    3892.000      -8.343
## gold_search     -8.343      30.379
##
## Correlation matrix of residuals:
##             gold_price gold_search
## gold_price     1.00000    -0.02426
## gold_search   -0.02426     1.00000
```

```
#for FD
VAR_lag_FD <- VAR(y = VAR_data_FD, type = ’const’, p=1)
summary(VAR_lag_FD)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gold_price_FD, gold_search_FD
## Deterministic variables: const
## Sample size: 207
## Log Likelihood: -1796.695
## Roots of the characteristic polynomial:
## 0.1813 0.1813
## Call:
## VAR(y = VAR_data_FD, p = 1, type = "const")
##
##
## Estimation results for equation gold_price_FD:
## ==============================================
## gold_price_FD = gold_price_FD.l1 + gold_search_FD.l1 + const
##
##                   Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1  -0.12307    0.07009  -1.756   0.0806 .
## gold_search_FD.l1 -0.66331    0.75343  -0.880   0.3797
## const              8.23328    4.35479   1.891   0.0601 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 62.24 on 204 degrees of freedom
## Multiple R-Squared: 0.01812, Adjusted R-squared: 0.008495
## F-statistic: 1.882 on 2 and 204 DF,  p-value: 0.1549
##
##
## Estimation results for equation gold_search_FD:
## ===============================================
## gold_search_FD = gold_price_FD.l1 + gold_search_FD.l1 + const
##
##                    Estimate Std. Error t value Pr(>|t|)
## gold_price_FD.l1   0.016307   0.006327   2.577  0.01066 *
## gold_search_FD.l1 -0.179223   0.068008  -2.635  0.00905 **
## const              0.136697   0.393083   0.348  0.72838
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##
## Residual standard error: 5.618 on 204 degrees of freedom
## Multiple R-Squared: 0.06434, Adjusted R-squared: 0.05516
## F-statistic: 7.014 on 2 and 204 DF,  p-value: 0.001133
##
##
##
```

```
## Covariance matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        3873.81         -11.86
## gold_search_FD        -11.86          31.56
##
## Correlation matrix of residuals:
##                gold_price_FD gold_search_FD
## gold_price_FD        1.00000       -0.03391
## gold_search_FD      -0.03391        1.00000
```

4 Phillips-Ouliaris Cointegration Test

```
po.test(VAR_data, demean = TRUE, lshort = TRUE)
```

```
##
##  Phillips-Ouliaris Cointegration Test
##
## data:  VAR_data
## Phillips-Ouliaris demeaned = -19.431, Truncation lag parameter = 2,
## p-value = 0.06318
```

```
# for FD
po.test(VAR_data_FD, demean = TRUE, lshort = TRUE)
```

```
## Warning in po.test(VAR_data_FD, demean = TRUE, lshort = TRUE): p-value smaller
## than printed p-value
```

```
##
##  Phillips-Ouliaris Cointegration Test
##
## data:  VAR_data_FD
## Phillips-Ouliaris demeaned = -224.2, Truncation lag parameter = 2,
## p-value = 0.01
```

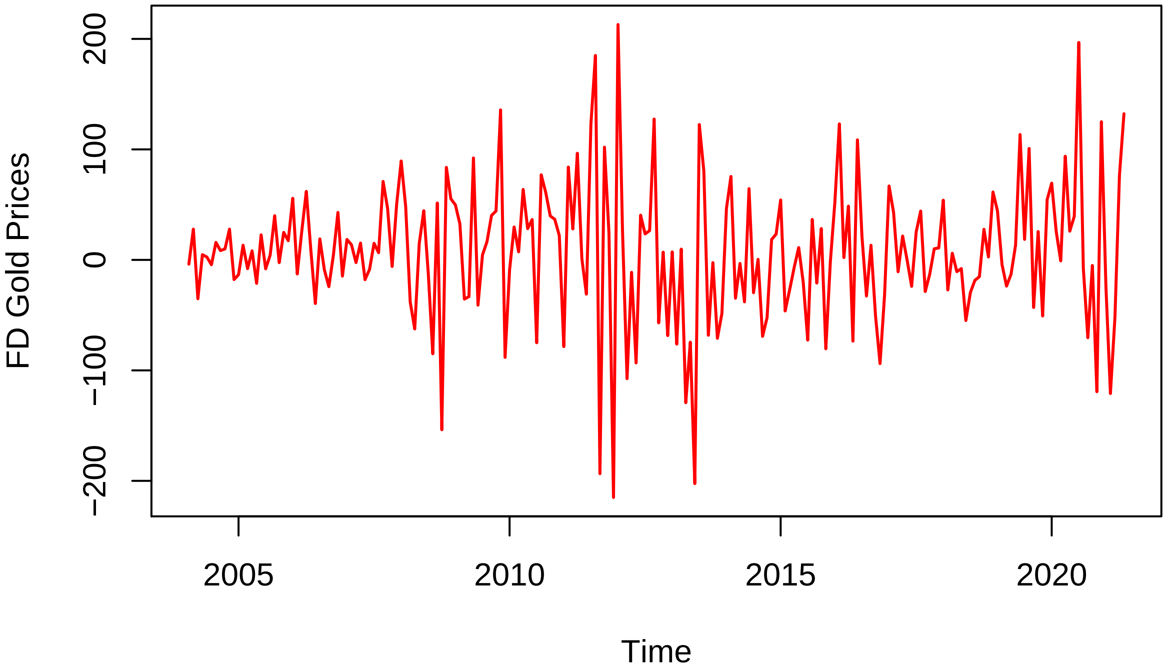
We cannot reject the null of the residuals being I(1). Thus we cannot rule out the case of a spurious or unbalanced regression. Note: as we only have one I(1) process and one non I(1) process this test makes no real sense.

5 Ideas for the first part: Univariate Time Series

Sollten wir hier nicht mit MA(1) weiterarbeiten? Ich habs mal weiter unten mit MA(1) gemacht. Den AR(1) part aber noch nicht gelöscht ####LÖSCHEN####

6 Differenced AR(1) and ARCH Model for Gold Prices ####LÖSCHEN

```
# Differenced AR(1) and ARCH Model for Gold Prices
gold_price_FD <- ts(gold_price_FD, frequency = 12,
                    start = c(2004, 2), end = c(2021, 5))
plot(gold_price_FD, ylab = 'FD Gold Prices', col = 'red', lwd = 1.5)
```

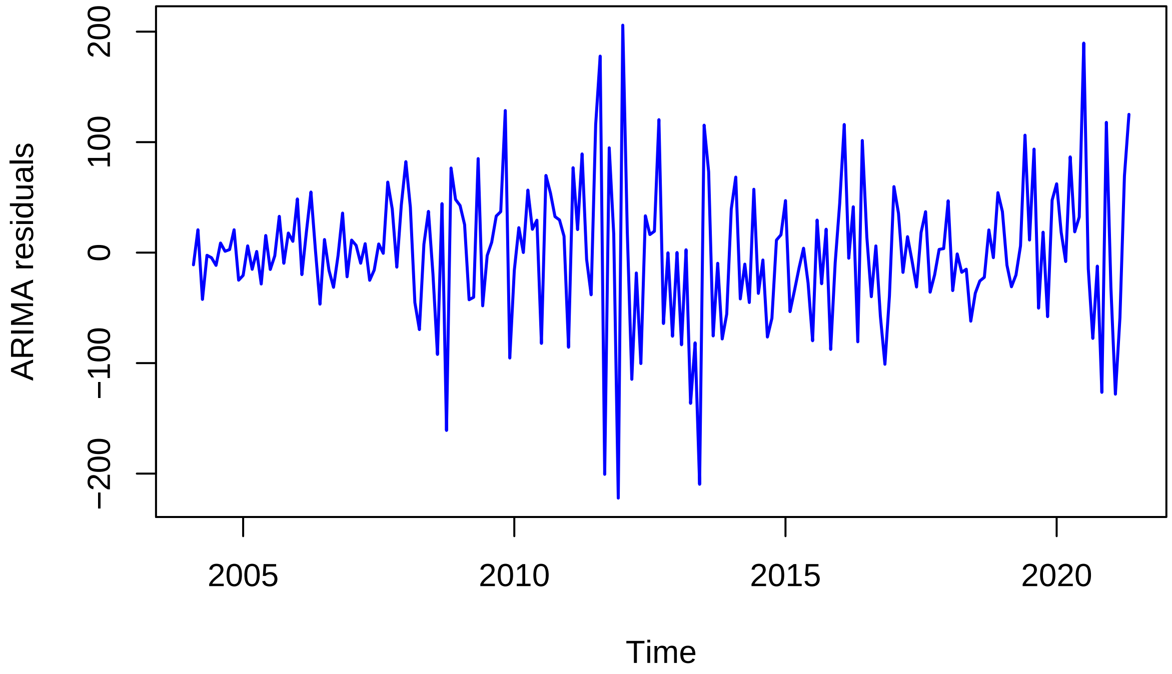


```
ar1mod_FD <- arima(gold_price_FD, order = c(1,0,0))
ar1mod_FD
```

```
##
## Call:
## arima(x = gold_price_FD, order = c(1, 0, 0))
##
## Coefficients:
##           ar1  intercept
##       -0.1205     7.1540
## s.e.   0.0694     3.8237
##
## sigma^2 estimated as 3814:  log likelihood = -1152.78,  aic = 2311.57
```

```
plot(forecast::arima.errors(ar1mod_FD),type = 'l', lwd = 1.5, col = 'blue',
     ylab = 'ARIMA residuals')
```

```
## Deprecated, use residuals.Arima(object, type='regression') instead
```



```
mean(forecast::arima.errors(ar1mod_FD))
```

```
## Deprecated, use residuals.Arima(object, type='regression') instead

## [1] 0.05853392
```

Going by the plot, it does not appear that the variance of the residuals is constant over time but rather has times of higher and lower volatility.

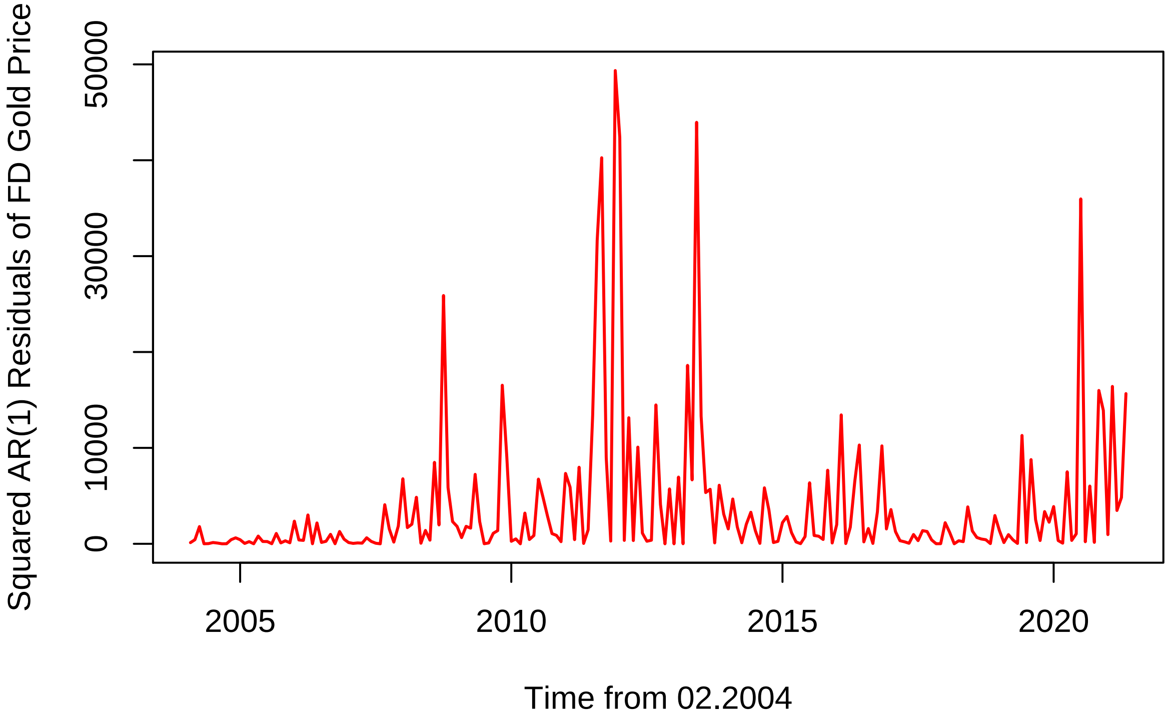
```
resi_ar1_FD_2 <- (forecast::arima.errors(ar1mod_FD))^2
```

```
## Deprecated, use residuals.Arima(object, type='regression') instead
```

```
resi_arch1_FD_2_model <- arima(resi_ar1_FD_2, order = c(1,0,0))
resi_arch1_FD_2_model
```

```
##
## Call:
## arima(x = resi_ar1_FD_2, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.3095  3887.0806
## s.e.  0.0662   727.6653
##
## sigma^2 estimated as 52737140:  log likelihood = -2144.4,  aic = 4294.79
```

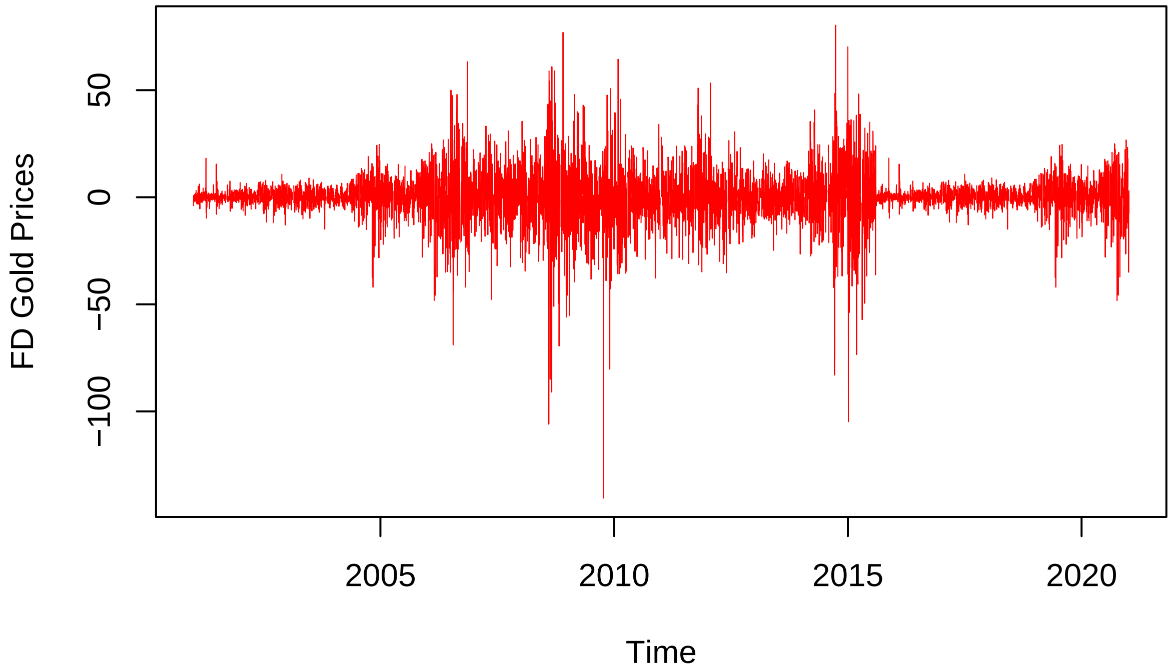
```
# plot the squared residuals:
plot(resi_ar1_FD_2, ylab = 'Squared AR(1) Residuals of FD Gold Price',
     xlab = 'Time from 02.2004', col = 'red', lwd = 1.5)
```



Mit MA(1), ändert nicht viel

7 Differenced MA(1) and ARCH Model for Gold Prices

```
# Differenced MA(1) and ARCH Model for Gold Prices

gold_price_FD <- ts(gold_daily_FD, frequency = 365,
                    start = c(2001, 1, 1), end = c(2021, 6, 10))
plot(gold_price_FD, ylab = 'FD Gold Prices', col = 'red', lwd = 0.5)
```

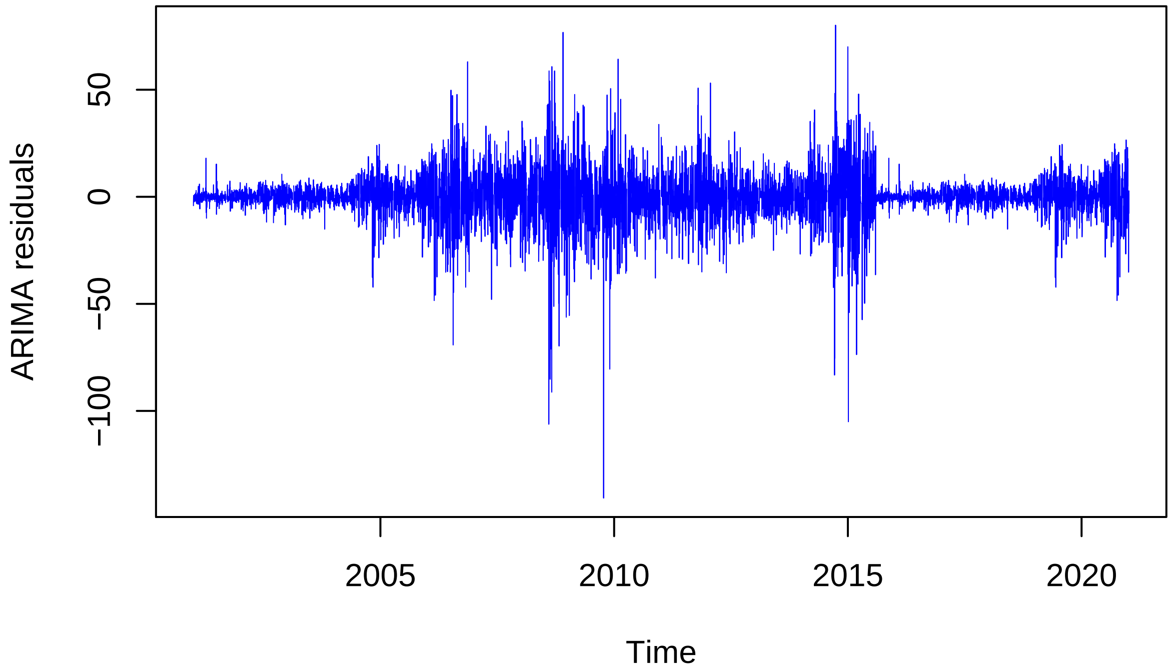


```
model_HF_FD <- forecast::auto.arima(gold_price_FD, ic = 'aic')
model_HF_FD
```

```
## Series: gold_price_FD
## ARIMA(4,0,2) with non-zero mean
##
## Coefficients:
##           ar1     ar2      ar3      ar4     ma1      ma2    mean
##       -0.0270  0.6155  -0.0033  -0.0384  0.0278  -0.6164  0.2022
## s.e.   0.2394  0.2277   0.0162   0.0162  0.2398   0.2285  0.1213
##
## sigma^2 estimated as 120.9:  log likelihood=-26138.47
## AIC=52292.95   AICc=52292.97   BIC=52348.12
```

```
plot(forecast::arima.errors(model_HF_FD),type = 'l', lwd = 0.5, col = 'blue',
     ylab = 'ARIMA residuals')
```

```
## Deprecated, use residuals.Arima(object, type='regression') instead
```



Going by the plot, it does not appear that the variance of the residuals is constant over time but rather has times of higher and lower volatility.

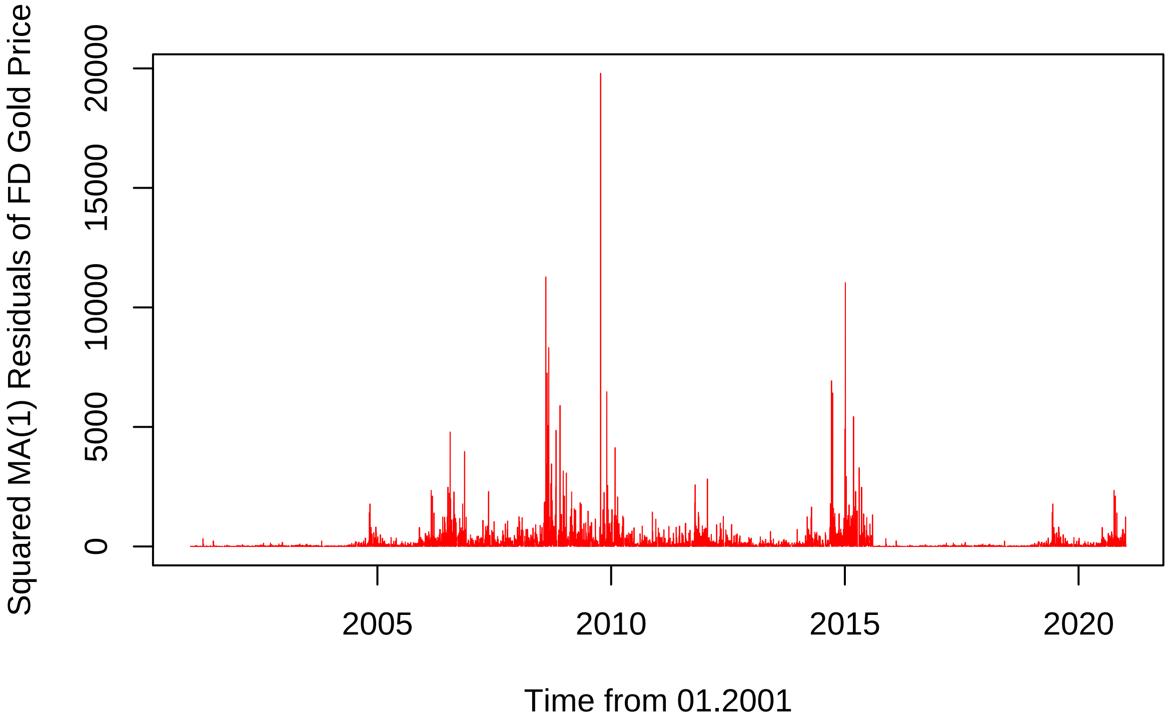
```
resi_ma1_FD_2 <- (forecast::arima.errors(model_HF_FD))^2
```

```
## Deprecated, use residuals.Arima(object, type='regression') instead
```

```
resi_arch1_FD_2_model <- arima(resi_ma1_FD_2, order = c(1,0,0))
resi_arch1_FD_2_model
```

```
##
## Call:
## arima(x = resi_ma1_FD_2, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##        0.112   121.0271
## s.e.   0.012     6.4950
##
## sigma^2 estimated as 229290:  log likelihood = -51994.71,  aic = 103995.4
```

```
# plot the squared residuals:
plot(resi_ma1_FD_2, ylab = 'Squared MA(1) Residuals of FD Gold Price',
     xlab = 'Time from 01.2001', col = 'red', lwd = 0.5)
```



8 GARCH

```
#Try GARCH(1,1)

gold_price_FD_clean <- na.remove(gold_price_FD) #always got an error for NA

garch_gold_price_FD <- garch(x=gold_price_FD_clean,order=c(1,1))
```

```
##
##  ***** ESTIMATION WITH ANALYTICAL GRADIENT *****
##
##
##      I     INITIAL X(I)        D(I)
##
##      1     1.089412e+02     1.000e+00
##      2     5.000000e-02     1.000e+00
##      3     5.000000e-02     1.000e+00
##
##     IT   NF      F         RELDF    PRELDF    RELDX   STPPAR   D*STEP   NPRELDF
##      0    1  1.968e+04
##      1    3  1.959e+04  4.67e-03  4.66e-02  2.2e-03  4.0e+03  4.8e-01  9.42e+01
##      2    5  1.958e+04  5.08e-04  9.24e-04  1.8e-04  4.7e+00  4.8e-02  2.39e+00
##      3    6  1.957e+04  3.06e-04  3.87e-04  1.6e-04  2.0e+00  4.8e-02  1.01e+00
##      4    7  1.956e+04  8.15e-04  1.04e-03  4.1e-04  2.0e+00  9.5e-02  7.67e-01
##      5    9  1.955e+04  1.50e-04  5.87e-04  2.7e-04  2.0e+00  7.4e-02  1.79e-01
##      6   10  1.955e+04  1.72e-04  4.98e-04  2.6e-04  2.0e+00  7.4e-02  2.80e-02
##      7   11  1.955e+04  9.75e-05  1.01e-04  3.2e-04  2.0e+00  7.4e-02  1.95e-03
##      8    14  1.955e+04  3.77e-06  1.19e-05  3.6e-05  4.1e+00  9.7e-03  2.19e-03
##      9    15  1.955e+04  4.93e-06  9.15e-06  3.4e-05  2.0e+00  9.7e-03  1.38e-03
##     10    16  1.955e+04  5.23e-06  5.24e-06  4.3e-05  2.0e+00  9.7e-03  1.26e-03
##     11    17  1.955e+04  5.04e-06  5.92e-06  8.7e-05  2.0e+00  1.9e-02  1.26e-03
##     12    22  1.950e+04  2.58e-03  1.26e-03  3.4e-02  0.0e+00  7.1e+00  1.26e-03
##     13    24  1.883e+04  3.42e-02  1.80e-02  8.4e-01  0.0e+00  9.3e+01  1.80e-02
##     14    25  1.875e+04  4.28e-03  6.49e-02  5.8e-01  0.0e+00  2.4e+01  6.49e-02
##     15    27  1.843e+04  1.69e-02  1.28e-02  2.2e-01  8.1e-01  1.2e+01  5.94e-02
##     16    29  1.836e+04  4.14e-03  4.01e-03  6.1e-02  2.0e+00  2.4e+00  1.57e+01
##     17    31  1.820e+04  8.37e-03  8.41e-03  1.5e-01  2.0e+00  4.7e+00  9.41e+02
##     18    32  1.805e+04  8.30e-03  1.61e-02  5.5e-01  2.0e+00  9.4e+00  2.70e+01
##     19    34  1.753e+04  2.89e-02  2.80e-02  1.4e-01  2.0e+00  9.4e-01  3.99e-01
##     20    36  1.744e+04  5.34e-03  5.28e-03  1.9e-01  2.0e+00  9.4e-01  2.04e+00
##     21    37  1.733e+04  6.40e-03  1.07e-02  8.6e-01  2.0e+00  1.9e+00  5.43e+00
##     22    40  1.731e+04  7.08e-04  1.72e-03  1.2e-02  8.6e+00  1.9e-02  5.40e-02
##     23    41  1.730e+04  6.70e-04  6.58e-04  1.0e-02  2.0e+00  1.9e-02  1.39e-02
##     24    42  1.729e+04  8.99e-04  8.83e-04  1.7e-02  1.9e+00  3.8e-02  1.38e-02
##     25    44  1.722e+04  3.80e-03  3.55e-03  4.1e-02  5.7e-01  7.5e-02  1.33e-02
##     26    45  1.718e+04  2.46e-03  2.84e-03  8.3e-02  1.9e+00  1.5e-01  6.74e-02
##     27    48  1.718e+04  9.66e-05  7.50e-04  4.8e-03  2.9e+00  1.0e-02  3.28e-02
##     28    49  1.717e+04  4.73e-04  4.72e-04  4.7e-03  2.0e+00  1.0e-02  3.15e-02
##     29    50  1.716e+04  3.83e-04  4.63e-04  9.9e-03  2.0e+00  2.0e-02  3.27e-02
##     30    51  1.715e+04  8.00e-04  1.02e-03  1.9e-02  1.9e+00  4.1e-02  2.14e-02
##     31    52  1.714e+04  4.32e-04  6.32e-04  1.9e-02  1.8e+00  4.1e-02  7.81e-03
##     32    53  1.714e+04  1.39e-04  6.04e-04  1.6e-02  8.9e-01  4.1e-02  1.04e-03
##     33    54  1.714e+04  1.60e-05  1.27e-04  2.2e-02  1.2e+00  4.1e-02  3.14e-04
##     34    55  1.714e+04  9.36e-05  1.51e-04  2.0e-02  6.9e-01  4.1e-02  1.89e-04
##     35    56  1.714e+04  2.15e-06  1.97e-05  4.8e-03  0.0e+00  8.9e-03  1.97e-05
##     36    57  1.714e+04  9.16e-06  9.29e-06  2.8e-03  0.0e+00  5.1e-03  9.29e-06
##     37    58  1.714e+04  2.65e-08  2.66e-08  1.0e-04  0.0e+00  2.7e-04  2.66e-08
##     38    59  1.714e+04  2.22e-10  8.62e-11  1.0e-05  0.0e+00  1.9e-05  8.62e-11
##     39    60  1.714e+04  2.92e-11  7.86e-13  2.2e-06  0.0e+00  4.1e-06  7.86e-13
##     40    61  1.714e+04  2.14e-12  4.95e-15  1.7e-07  0.0e+00  3.1e-07  4.95e-15
##     41    62  1.714e+04  7.43e-15  1.22e-18  4.7e-10  0.0e+00  1.1e-09  1.22e-18
##     42    63  1.714e+04 -1.70e-15  3.34e-22  1.9e-11  0.0e+00  3.6e-11  3.34e-22
##
##  ***** X- AND RELATIVE FUNCTION CONVERGENCE *****
##
##  FUNCTION     1.713619e+04   RELDX        1.946e-11
##  FUNC. EVALS      63         GRAD. EVALS      42
##  PRELDF       3.337e-22      NPRELDF      3.337e-22
##
##      I      FINAL X(I)        D(I)          G(I)
##
##      1    1.679893e-01     1.000e+00     3.931e-09
##      2    8.431307e-02     1.000e+00    -1.874e-06
##      3    9.200609e-01     1.000e+00    -2.105e-06
```

```
## Warning in sqrt(pred$e): NaNs produced
```

```
summary(garch_gold_price_FD)
```

```
##
```

```
## Call:
## garch(x = gold_price_FD_clean, order = c(1, 1))
##
## Model:
## GARCH(1,1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -10.19636  -0.49038   0.04738   0.61628   7.36207
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0  0.167989    0.011448    14.67   <2e-16 ***
## a1  0.084313    0.002692    31.32   <2e-16 ***
## b1  0.920061    0.002636   349.06   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 5586.5, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.00050691, df = 1, p-value = 0.982
```

```
AIC_GARCH_1<-AIC(garch_gold_price_FD)
AIC_GARCH_1
```

```
## [1] 46816.38
```

```
# Check, if the above GARCH(1,1) works with rugarch

#load rugarch library
library(rugarch)
```

```
## Loading required package: parallel
```

```
##
## Attaching package: ’rugarch’
```

```
## The following object is masked from ’package:stats’:
##
##     sigma
```

```
#fit the rugarch sGarch model
spec = ugarchspec(variance.model=list(model="sGARCH", garchOrder=c(1,1)), mean.model=list(armaOrder=c(0
test_garch_gold_price_FD<- ugarchfit(spec=spec, data=gold_price_FD_clean)
test_garch_gold_price_FD
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.192548    0.057573   3.3444 0.000825
## omega   0.191148    0.029141   6.5594 0.000000
## alpha1  0.075676    0.006018  12.5746 0.000000
## beta1   0.923324    0.005853 157.7537 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.192548    0.059760   3.2220 0.001273
## omega   0.191148    0.072008   2.6545 0.007942
## alpha1  0.075676    0.017852   4.2390 0.000022
## beta1   0.923324    0.016907  54.6125 0.000000
##
## LogLikelihood : -23432.46
##
## Information Criteria
## ------------------------------------
##
## Akaike       6.8428
## Bayes        6.8468
## Shibata      6.8428
## Hannan-Quinn 6.8441
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      4.426 0.03539
## Lag[2*(p+q)+(p+q)-1][2]     5.153 0.03701
## Lag[4*(p+q)+(p+q)-1][5]     6.695 0.06146
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1029  0.7484
## Lag[2*(p+q)+(p+q)-1][5]    2.6455  0.4756
## Lag[4*(p+q)+(p+q)-1][9]    5.6242  0.3437
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.00802 0.500 2.000 0.92864
## ARCH Lag[5]   5.06256 1.440 1.667 0.09940
## ARCH Lag[7]   6.72811 2.315 1.543 0.09931
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.0948
## Individual Statistics:
## mu     0.01941
## omega  0.02079
## alpha1 1.15660
## beta1  1.07772
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.07 1.24 1.6
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           1.6935 0.09041   *
## Negative Sign Bias  0.2509 0.80192
## Positive Sign Bias  1.0492 0.29410
## Joint Effect        8.0823 0.04434  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     288.1    5.465e-50
## 2    30     317.4    2.829e-50
## 3    40     352.1    4.965e-52
## 4    50     369.0    1.204e-50
##
##
## Elapsed time : 0.672426
```

```
#Summarizes coeff.
coef(test_garch_gold_price_FD)
```

```
##         mu      omega     alpha1      beta1
## 0.19254758 0.19114774 0.07567601 0.92332398
```

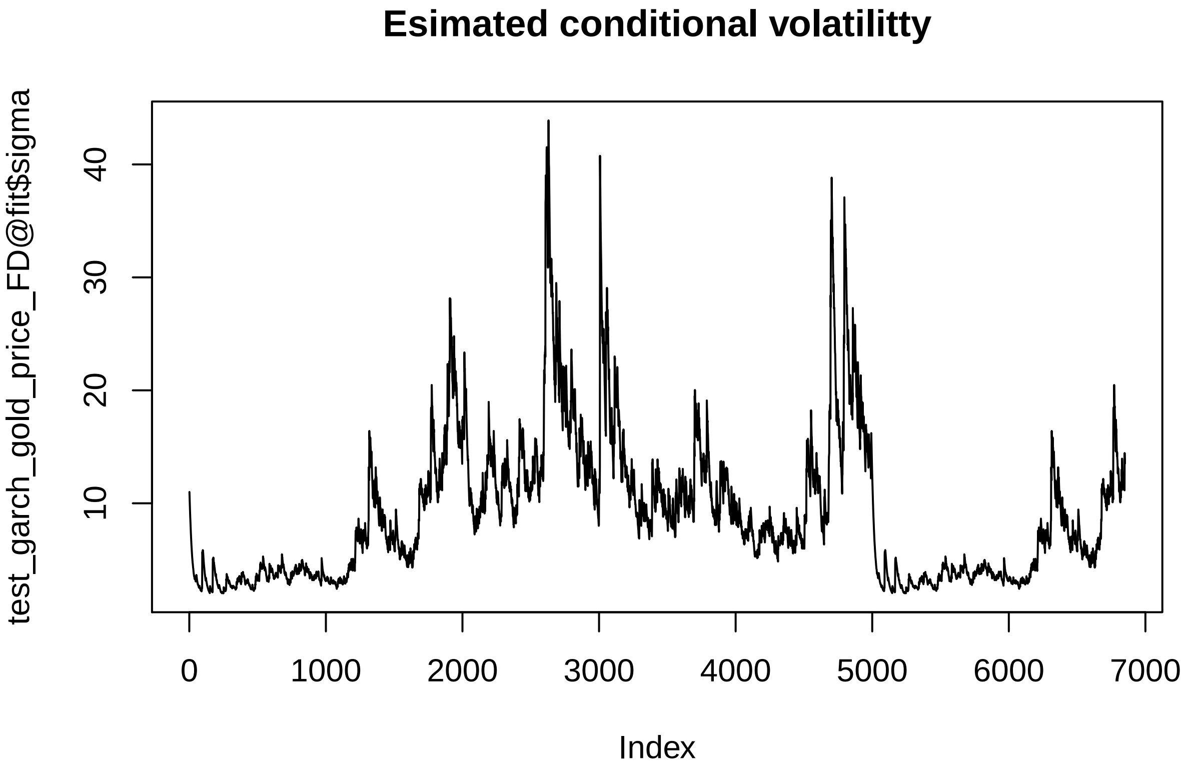
```
garch_gold_price_FD$coef  # for comparison with the package tseries and garch from above, Checked onlin
```

```
##         a0         a1         b1
## 0.16798931 0.08431307 0.92006094
```

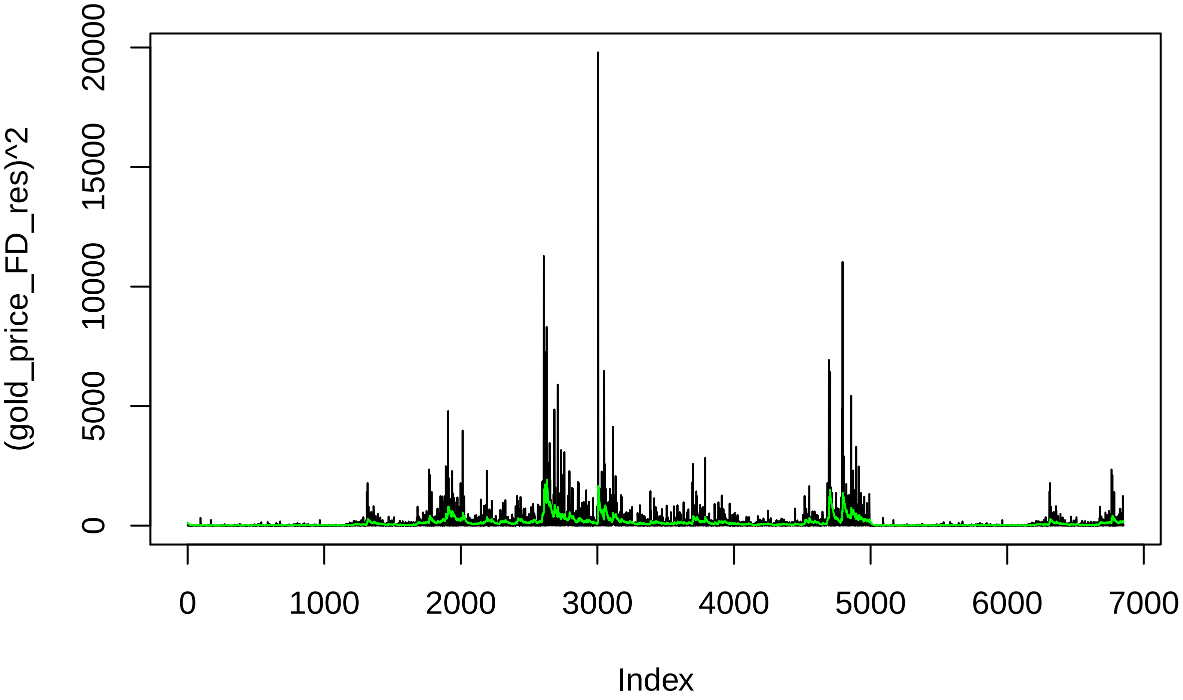
```
#calculating AIC:
AIC_GARCH_2 <- 6.8428*length(gold_price_FD_clean)
AIC_GARCH_2
```

```
## [1] 46873.18
```

```
# Plotting the estimated conditional volatility
plot(test_garch_gold_price_FD@fit$sigma, type = "l", main = "Esimated conditional volatilitty")
```



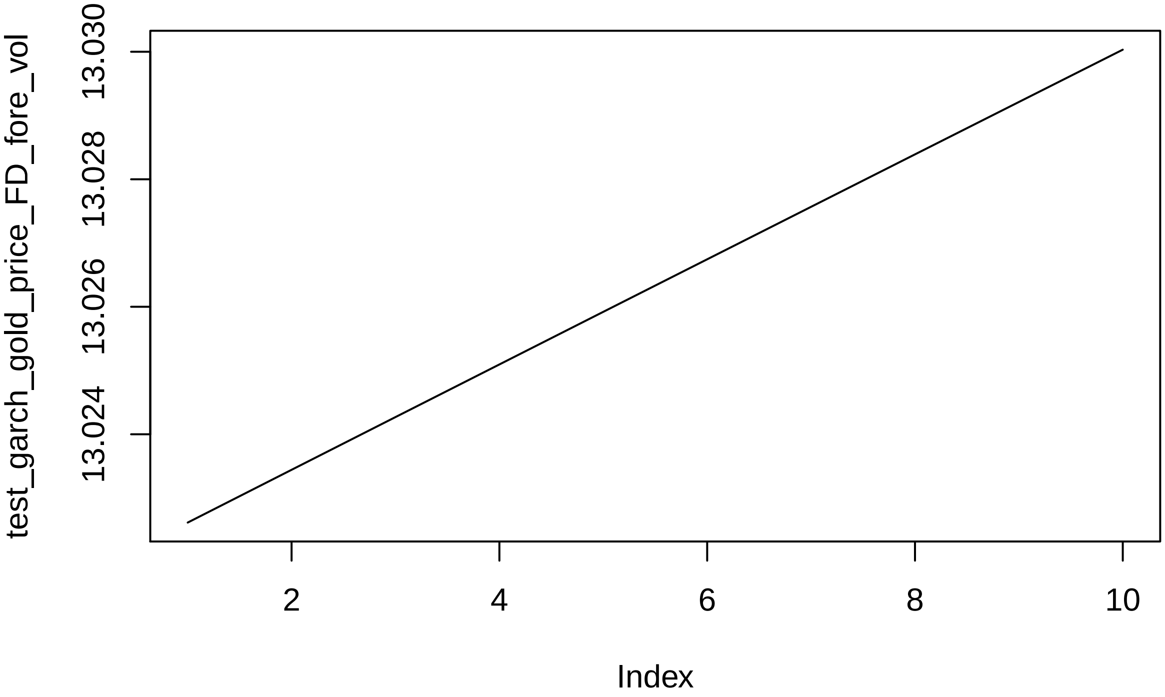
```
#Plot of squared residuals and est. cond. variance
gold_price_FD_res<-test_garch_gold_price_FD@fit$residuals
gold_price_FD_var<-test_garch_gold_price_FD@fit$var
plot((gold_price_FD_res)^2, type = "l")
lines(gold_price_FD_var, col="green")
```



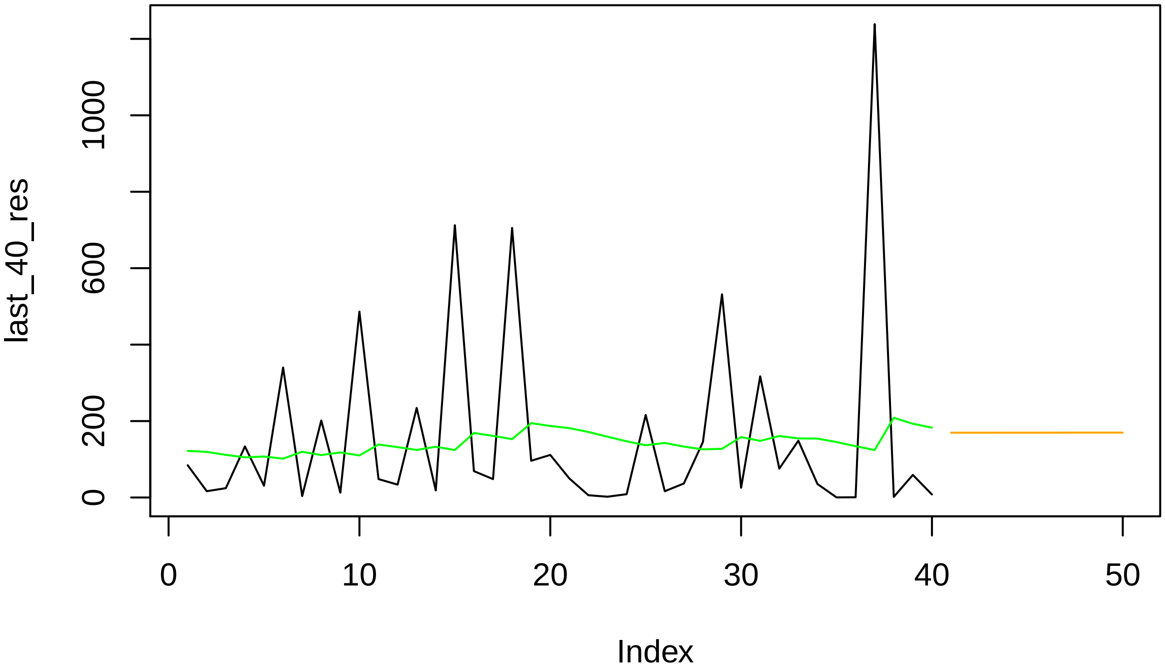
```
#Model forecasting
test_garch_gold_price_FD_fore <- ugarchforecast (test_garch_gold_price_FD, n.ahaed = 10)
test_garch_gold_price_FD_fore
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 10
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1988-10-03 01:00:00]:
##      Series Sigma
## T+1  0.1925 13.02
## T+2  0.1925 13.02
## T+3  0.1925 13.02
## T+4  0.1925 13.03
## T+5  0.1925 13.03
## T+6  0.1925 13.03
## T+7  0.1925 13.03
## T+8  0.1925 13.03
## T+9  0.1925 13.03
## T+10 0.1925 13.03
```

```
#Plot of the conditional volatitily forecst:
test_garch_gold_price_FD_fore_vol <- test_garch_gold_price_FD_fore@forecast$sigmaFor
plot(test_garch_gold_price_FD_fore_vol, type= "l")
```



```
last_40_var <- c(tail(gold_price_FD_var,40),rep(NA,10))
last_40_res <- c(tail((gold_price_FD_res)^2,40),rep(NA,10))
test_garch_gold_price_FD_fore_vol <- c(rep(NA,40),(test_garch_gold_price_FD_fore_vol)^2)

plot(last_40_res, type= "l")
lines(test_garch_gold_price_FD_fore_vol, col="orange")
lines(last_40_var, col="green")
```



Since rugarch uses a different version of AIC one needs to mutiply it with the length: 6.8428*6850=46873.18

```
#Try eGARCH

spec = ugarchspec(variance.model=list(model="eGARCH", garchOrder=c(1,1)), mean.model=list(armaOrder=c(0
egarch_gold_price_FD<- ugarchfit(spec=spec, data=gold_price_FD_clean, solver = 'hybrid')
egarch_gold_price_FD
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : eGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.300007    0.040505 7.4066e+00        0
## omega   0.023223    0.001112 2.0885e+01        0
## alpha1  0.045680    0.005196 8.7916e+00        0
## beta1   0.996760    0.000003 3.2281e+05        0
## gamma1  0.145725    0.001178 1.2367e+02        0
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.300007    0.028986    10.3500 0.000000
## omega   0.023223    0.002227    10.4273 0.000000
## alpha1  0.045680    0.011905     3.8371 0.000124
## beta1   0.996760    0.000011 91180.3976 0.000000
## gamma1  0.145725    0.002737    53.2488 0.000000
##
## LogLikelihood : -23391.54
##
## Information Criteria
## ------------------------------------
##
## Akaike       6.8311
## Bayes        6.8361
## Shibata      6.8311
## Hannan-Quinn 6.8328
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      2.695 0.10065
## Lag[2*(p+q)+(p+q)-1][2]     3.777 0.08655
## Lag[4*(p+q)+(p+q)-1][5]     5.570 0.11335
## d.o.f=0
## H0 : No serial correlation
##
```

```
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      3.359 0.06682
## Lag[2*(p+q)+(p+q)-1][5]     4.542 0.19378
## Lag[4*(p+q)+(p+q)-1][9]     6.390 0.25585
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.03502 0.500 2.000  0.8515
## ARCH Lag[5]   2.28941 1.440 1.667  0.4107
## ARCH Lag[7]   3.74288 2.315 1.543  0.3852
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  0.9715
## Individual Statistics:
## mu     0.08420
## omega  0.15354
## alpha1 0.32759
## beta1  0.22569
## gamma1 0.04047
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias           1.6743 0.094123   *
## Negative Sign Bias  0.9843 0.324992
## Positive Sign Bias  1.2341 0.217220
## Joint Effect       13.8424 0.003128 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     284.9    2.443e-49
## 2    30     315.1    8.284e-50
## 3    40     327.9    2.373e-47
## 4    50     343.2    9.024e-46
##
##
## Elapsed time : 4.352819
```

```
AIC_eGARCH <- 6.8311*length(gold_price_FD_clean)
AIC_eGARCH
```

```
## [1] 46793.04
```

```
#Try iGARCH

spec = ugarchspec(variance.model=list(model="iGARCH", garchOrder=c(1,1)), mean.model=list(armaOrder=c(0
igarch_gold_price_FD<- ugarchfit(spec=spec, data=gold_price_FD_clean)
igarch_gold_price_FD
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : iGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu       0.19294    0.057572   3.3513 0.000804
## omega    0.18470    0.025701   7.1863 0.000000
## alpha1   0.07675    0.005293  14.5015 0.000000
## beta1    0.92325          NA       NA       NA
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu       0.19294    0.061023   3.1618 0.001568
## omega    0.18470    0.064045   2.8838 0.003929
## alpha1   0.07675    0.011744   6.5355 0.000000
## beta1    0.92325          NA       NA       NA
##
## LogLikelihood : -23430.77
##
## Information Criteria
## ------------------------------------
##
## Akaike       6.842
## Bayes        6.845
## Shibata      6.842
## Hannan-Quinn 6.843
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      4.397 0.03601
## Lag[2*(p+q)+(p+q)-1][2]     5.137 0.03738
## Lag[4*(p+q)+(p+q)-1][5]     6.700 0.06127
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                    0.09725  0.7552
## Lag[2*(p+q)+(p+q)-1][5]   2.60608  0.4837
## Lag[4*(p+q)+(p+q)-1][9]   5.53124  0.3557
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]  0.007351 0.500 2.000  0.9317
## ARCH Lag[5]  4.974918 1.440 1.667  0.1041
## ARCH Lag[7]  6.585417 2.315 1.543  0.1065
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  0.4255
## Individual Statistics:
## mu     0.01943
## omega  0.01849
## alpha1 0.24788
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          0.846 1.01 1.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           1.6990 0.08937   *
## Negative Sign Bias  0.3058 0.75973
## Positive Sign Bias  1.0883 0.27650
## Joint Effect        8.2011 0.04203  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     285.4    1.972e-49
## 2    30     326.5    4.527e-52
## 3    40     348.7    2.213e-51
## 4    50     378.9    1.584e-52
##
##
## Elapsed time : 0.296921
```

```
AIC_iGARCH <- 6.842*length(gold_price_FD_clean)
AIC_iGARCH
```

```
## [1] 46867.7
```

```
# Summarizing all coeff:

garch_gold_price_FD$coef  #GARCH t-series
```

```
##         a0         a1         b1 
## 0.16798931 0.08431307 0.92006094
```

```
coef(test_garch_gold_price_FD) #sGARCH
```

```
##         mu      omega     alpha1      beta1 
## 0.19254758 0.19114774 0.07567601 0.92332398
```

```
coef(egarch_gold_price_FD) #eGARCH
```

```
##         mu      omega     alpha1      beta1     gamma1 
## 0.30000707 0.02322293 0.04568017 0.99676025 0.14572522
```

```
coef(igarch_gold_price_FD)  #iGARCH
```

```
##         mu      omega     alpha1      beta1 
## 0.19294107 0.18469719 0.07674977 0.92325023
```

```
# Summarizing AICs:
AIC_GARCH_1 #GARCH t-series
```

```
## [1] 46816.38
```

```
AIC_GARCH_2 #sGARCH
```

```
## [1] 46873.18
```

```
AIC_eGARCH #eGARCH
```

```
## [1] 46793.04
```

```
AIC_iGARCH #iGARCH
```

```
## [1] 46867.7
```

```
# According to AIC it seems like eGARCH performs the best, but isn't better than MA(1)
```