HuYuDataInsight LLC Apr 26-29, 2024

Zhaowei Cai

2024-04-29

1-5

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 4.2.3

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.2.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 4.2.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(urca)
```

```
## Warning: package 'urca' was built under R version 4.2.3
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

```
library(fGarch)
```

```
## Warning: package 'fGarch' was built under R version 4.2.3

## NOTE: Packages 'fBasics', 'timeDate', and 'timeSeries' are no longer
## attached to the search() path when 'fGarch' is attached.
##
## If needed attach them yourself in your R script by e.g.,
##         require("timeSeries")

##
## Attaching package: 'fGarch'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```
library(zoo)
library(tseries)
library(rugarch)
```

```
## Warning: package 'rugarch' was built under R version 4.2.3

## Loading required package: parallel

##
## Attaching package: 'rugarch'

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

```
library(ARDL)
```

```
## Warning: package 'ARDL' was built under R version 4.2.3

## To cite the ARDL package in publications:
##
## Use this reference to refer to the validity of the ARDL package.
##
##   Natsiopoulos, Kleanthis, and Tzeremes, Nickolaos G. (2022). ARDL
##   bounds test for cointegration: Replicating the Pesaran et al. (2001)
##   results for the UK earnings equation using R. Journal of Applied
##   Econometrics, 37(5), 1079-1090. https://doi.org/10.1002/jae.2919
##
```

```
## Use this reference to cite this specific version of the ARDL package.
##
##   Kleanthis Natsiopoulos and Nickolaos Tzeremes (2023). ARDL: ARDL, ECM
##   and Bounds-Test for Cointegration. R package version 0.2.4.
##   https://CRAN.R-project.org/package=ARDL
```

```
library(vars)
```

```
## Warning: package 'vars' was built under R version 4.2.3

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.2.3

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.2.3

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.2.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.2.3
```

```
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
data = na.omit(read.csv('AAPL2.csv'))
qqq = read.csv('QQQ2.csv')
```

1-5

```
closing = na.omit(data$Close) # closing price
log_closing = na.omit(log(data$Close)) # log closing price
log_return = na.omit(diff(log(data$Close))) # log return
time = as.Date(data$Date, format = '%m/%d/%y')

##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
summary(ur.df(log_return, type='trend', lags=20, selectlags="BIC"))
```

```
##
## ###############################################
## # 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
## -0.044441 -0.009042 -0.000271  0.009709  0.066383
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.871e-04  2.318e-03   0.340    0.734
## z.lag.1     -9.690e-01  9.028e-02 -10.733   <2e-16 ***
## tt           2.475e-06  1.541e-05   0.161    0.873
## z.diff.lag   6.294e-02  6.679e-02   0.942    0.347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01542 on 225 degrees of freedom
## Multiple R-squared:  0.4561, Adjusted R-squared:  0.4488
## F-statistic: 62.88 on 3 and 225 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -10.7329 38.4099 57.6049
##
## 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
```

```
# No drift or time trend
adf.test(log_return)
```

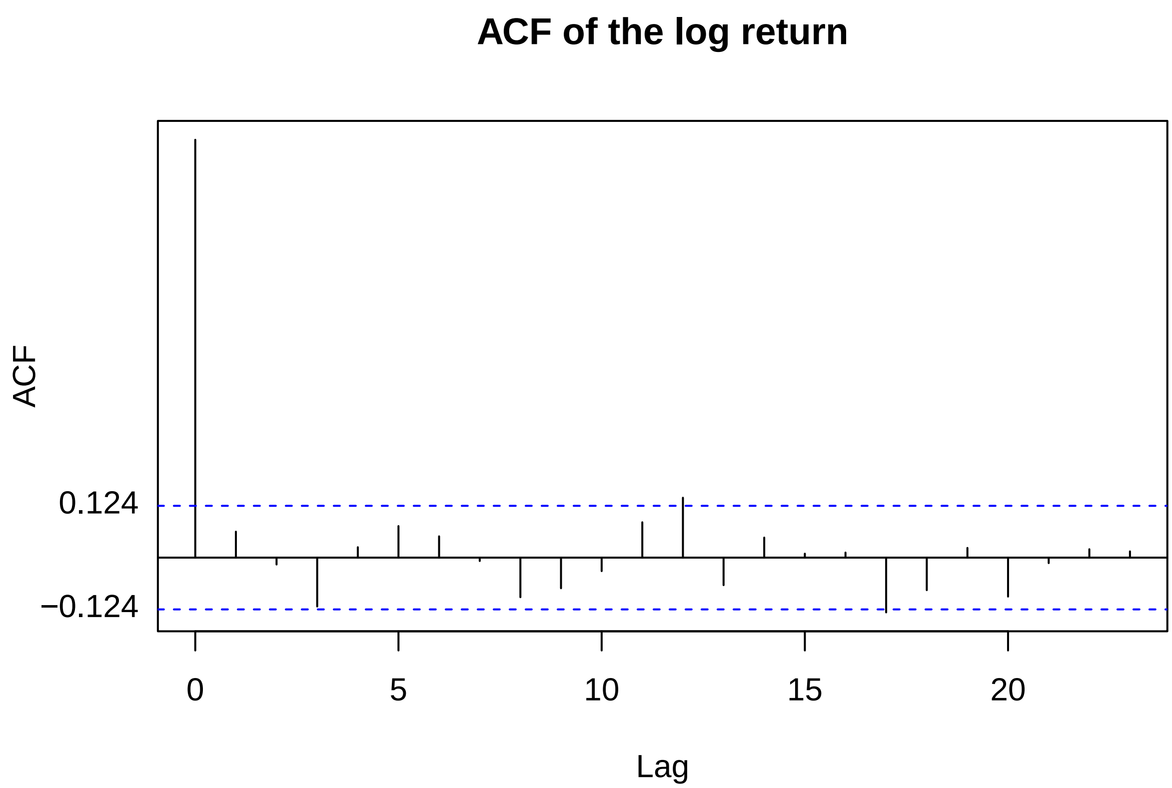
```
## Warning in adf.test(log_return): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log_return
## Dickey-Fuller = -5.4017, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

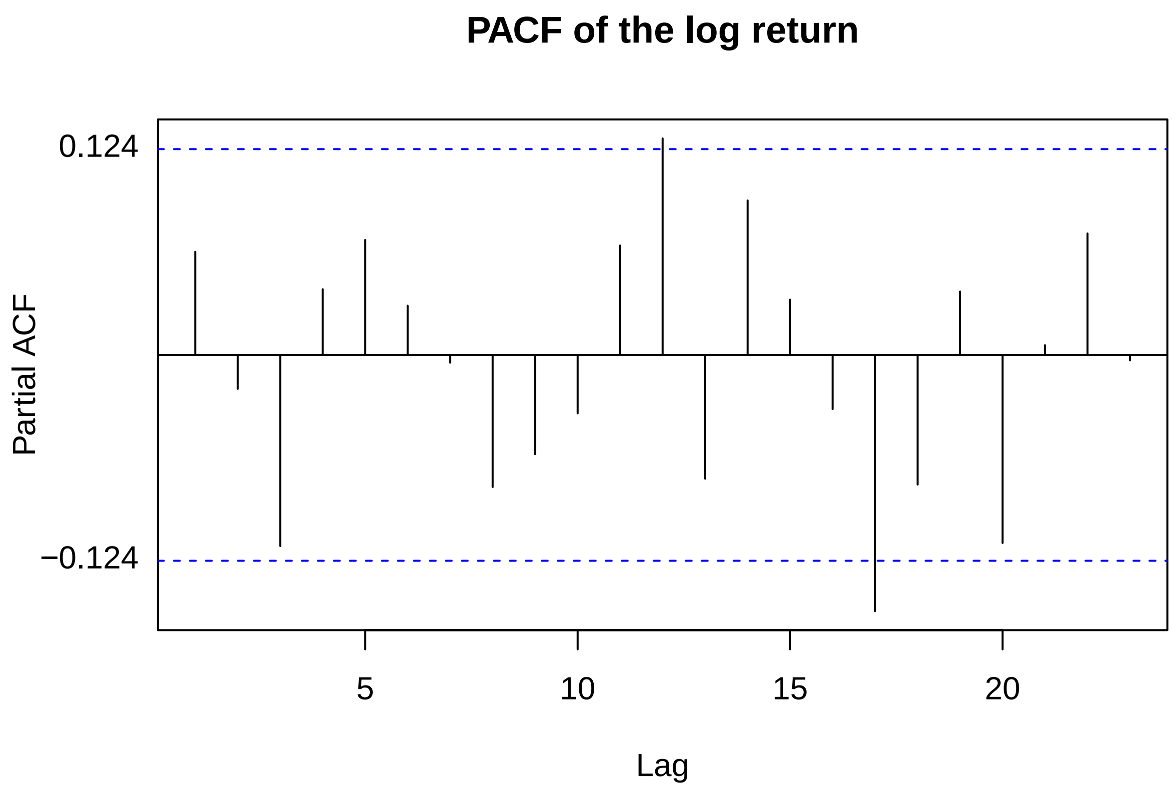
```
# The data is stationary. Difference is not needed.

##Check for the seasonality
n = length(log_return)
```

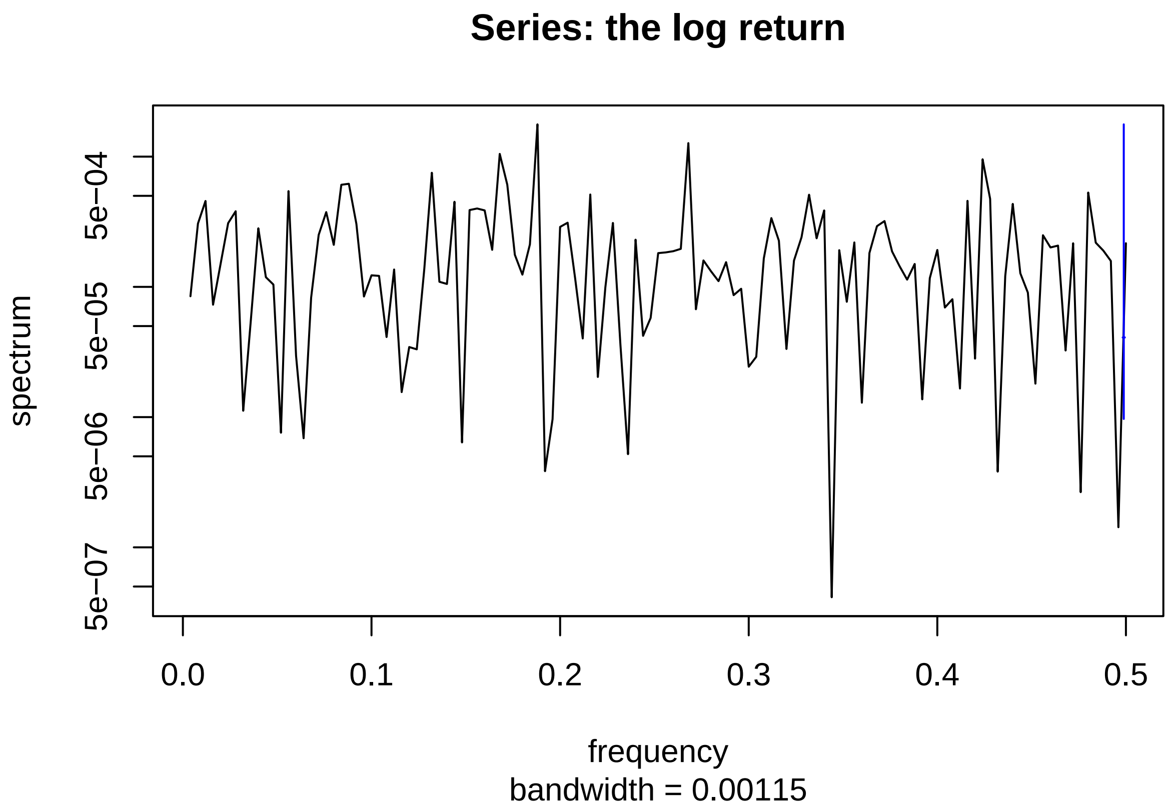
```
acf(log_return,main="ACF of the log return",yaxt="n")
ci=qnorm(c(0.025, 0.975))/sqrt(n)
text(y=ci,par("usr")[1],labels=round(ci,4),pos=2,xpd=TRUE)
```




```
pacf(log_return,main="PACF of the log return",yaxt="n")
text(y=ci,par("usr")[1],labels=round(ci,4),pos=2,xpd=TRUE)
```




```
spec.pgram(log_return,main="Series: the log return")
```

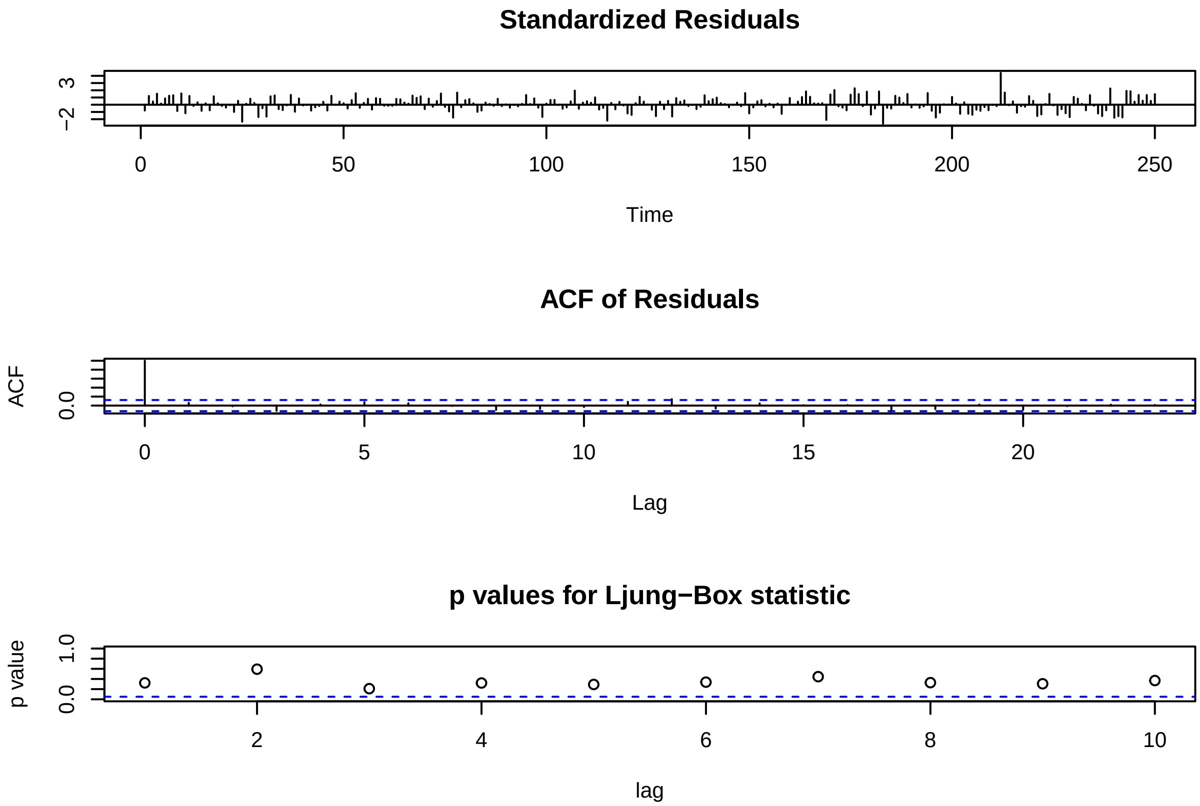



```r
# we cannot find any evidence for seasonality.

# ARIMA
fit = auto.arima(log_return, max.p=25, max.q=25, ic="bic",
                        seasonal=F, lambda=NULL,
                        stepwise=FALSE, approximation=FALSE
                        )
summary(fit)
```

```
## Series: log_return
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 0.0002344:  log likelihood = 690.1
## AIC=-1378.2   AICc=-1378.18   BIC=-1374.68
##
## Training set error measures:
##                       ME       RMSE        MAE MPE MAPE      MASE       ACF1
## Training set 0.001441796 0.01530871 0.01193403 100  100 0.7343912 0.06213279
```

```r
# ARIMA(0,0,0)
# AIC=-1378.2   AICc=-1378.18   BIC=-1374.68
tsdiag(fit)
```




```
shapiro.test(fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.98867, p-value = 0.04692
```

```
# The null-hypothesis of this test is that the population is normally distributed.
# The null hypothesis is rejected and there is evidence that the residuals tested are not normally dist

# ARIMA-Garch
arma_model <- fit
garch_spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                         mean.model = list(armaOrder = c(0,0)))
garch_fit <- ugarchfit(spec = garch_spec, data = arma_model$residuals)
garch_fit
```

```
##
## *---------------------------------*
## *          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      0.001543    0.000918  1.68050 0.092860
## omega   0.000002    0.000003  0.67482 0.499791
## alpha1  0.028370    0.015937  1.78008 0.075063
## beta1   0.967161    0.019342 50.00390 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001543    0.000916  1.68405 0.092173
## omega   0.000002    0.000007  0.24506 0.806409
## alpha1  0.028370    0.061755  0.45939 0.645953
## beta1   0.967161    0.070149 13.78725 0.000000
##
## LogLikelihood : 694.839
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.5267
## Bayes        -5.4704
## Shibata      -5.5272
## Hannan-Quinn -5.5040
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.5299  0.4667
## Lag[2*(p+q)+(p+q)-1][2]    0.5510  0.6701
## Lag[4*(p+q)+(p+q)-1][5]    2.8812  0.4292
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.05026  0.8226
## Lag[2*(p+q)+(p+q)-1][5]   1.03722  0.8512
## Lag[4*(p+q)+(p+q)-1][9]   2.66768  0.8123
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     1.149 0.500 2.000  0.2837
## ARCH Lag[5]     1.233 1.440 1.667  0.6652
## ARCH Lag[7]     2.333 2.315 1.543  0.6470
##
## Nyblom stability test
```

```
## ------------------------------------
## Joint Statistic:  58.8476
## Individual Statistics:
## mu     0.03939
## omega  3.60133
## alpha1 0.31726
## beta1  0.32200
##
## 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          0.18511 0.8533
## Negative Sign Bias 0.01524 0.9879
## Positive Sign Bias 0.16633 0.8680
## Joint Effect       0.04623 0.9974
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     11.12       0.9197
## 2    30     30.08       0.4100
## 3    40     38.64       0.4861
## 4    50     43.60       0.6911
##
##
## Elapsed time : 0.08401489
```

```r
# infocriteria(garch_fit)
# ARIMA-GARCH would be more apt for modeling time series data with volatility clustering, which is a ch

arma_model <- auto.arima(log_closing)
arma_model # difference --> return
```

```
## Series: log_closing
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       0.0014
## s.e.  0.0010
##
## sigma^2 = 0.0002333:  log likelihood = 691.21
## AIC=-1378.42   AICc=-1378.38   BIC=-1371.38
```

```r
garch_spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                         mean.model = list(armaOrder = c(0,0)))
garch_fit <- ugarchfit(spec = garch_spec, data = arma_model$residuals)
garch_fit
```

```
##
## *---------------------------------*
## *          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      0.000119    0.000915  0.13011 0.896477
## omega   0.000002    0.000002  0.74409 0.456824
## alpha1  0.028018    0.013999  2.00147 0.045342
## beta1   0.968322    0.016707 57.95811 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000119    0.000912  0.13064  0.89606
## omega   0.000002    0.000005  0.30768  0.75833
## alpha1  0.028018    0.051076  0.54855  0.58331
## beta1   0.968322    0.056970 16.99699  0.00000
##
## LogLikelihood : 698.1017
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.5307
## Bayes        -5.4745
## Shibata      -5.5312
## Hannan-Quinn -5.5081
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.4889  0.4844
## Lag[2*(p+q)+(p+q)-1][2]    0.5036  0.6919
## Lag[4*(p+q)+(p+q)-1][5]    2.8332  0.4384
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.05025  0.8226
## Lag[2*(p+q)+(p+q)-1][5]   1.02183  0.8547
## Lag[4*(p+q)+(p+q)-1][9]   2.66387  0.8129
## d.o.f=2
##
## Weighted ARCH LM Tests
```

```
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     1.127 0.500 2.000  0.2885
## ARCH Lag[5]     1.208 1.440 1.667  0.6723
## ARCH Lag[7]     2.330 2.315 1.543  0.6478
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  62.2614
## Individual Statistics:
## mu     0.0410
## omega  4.0988
## alpha1 0.3090
## beta1  0.3138
##
## 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          0.18133 0.8563
## Negative Sign Bias 0.01285 0.9898
## Positive Sign Bias 0.17080 0.8645
## Joint Effect       0.04578 0.9974
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     10.59       0.9368
## 2    30     29.28       0.4506
## 3    40     36.65       0.5776
## 4    50     47.61       0.5298
##
##
## Elapsed time : 0.1007771
```

```
forecasted_returns <- ugarchforecast(garch_fit, n.ahead = 1)
last_close_price <- closing[length(closing)]
(price_forecast <-as.numeric(last_close_price*exp(forecasted_returns@forecast$seriesFor)))
```

```
## [1] 174.0907
```

```
(lower_interval <- as.numeric(price_forecast*exp(qnorm(0.025)*forecasted_returns@forecast$sigmaFor)))
```

```
## [1] 167.6181
```

```
(upper_interval <- as.numeric(price_forecast*exp(qnorm(0.975)*forecasted_returns@forecast$sigmaFor)))
```

```
## [1] 180.8133
```

```r
# Print the forecasted closing price and prediction interval
cat("1-day ahead closing price forecast:", price_forecast, "\n")
```

```
## 1-day ahead closing price forecast: 174.0907
```

```r
cat("95% Prediction Interval: (", lower_interval, ", ", upper_interval, ")\n")
```

```
## 95% Prediction Interval: ( 167.6181 ,  180.8133 )
```

```r
# The true value 174.72 is inside the 95% CI.
```

6-10

```r
lc_AAPL = log_closing
lc_QQQ = na.omit(log(qqq$Close))
ardl_data = data.frame(cbind(lc_AAPL, lc_QQQ))
ardl_model <- ardl(lc_AAPL~lc_QQQ, data = ardl_data, order = c(1,1))
summary(ardl_model)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 251
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.028231 -0.005032 -0.000332  0.004966  0.042486
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.098822   0.054645  -1.808   0.0718 .
## L(lc_AAPL, 1)   0.998128   0.006966 143.293   <2e-16 ***
## lc_QQQ          0.955097   0.045068  21.192   <2e-16 ***
## L(lc_QQQ, 1)   -0.936557   0.045937 -20.388   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0091 on 246 degrees of freedom
## Multiple R-squared:  0.993,  Adjusted R-squared:  0.9929
## F-statistic: 1.156e+04 on 3 and 246 DF,  p-value: < 2.2e-16
```

```r
# UECM (Unrestricted Error Correction Model)
uecm_model <- uecm(ardl_model)
summary(uecm_model)
```

```
##
## Time series regression with "ts" data:
```

```
## Start = 2, End = 251
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.028231 -0.005032 -0.000332  0.004966  0.042486
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.098822   0.054645  -1.808   0.0718 .
## L(lc_AAPL, 1) -0.001872   0.006966  -0.269   0.7883
## L(lc_QQQ, 1)   0.018539   0.011802   1.571   0.1175
## d(lc_QQQ)      0.955097   0.045068  21.192   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0091 on 246 degrees of freedom
## Multiple R-squared:  0.6492, Adjusted R-squared:  0.6449
## F-statistic: 151.7 on 3 and 246 DF,  p-value: < 2.2e-16
```

```r
# RECM (Restricted Error Correction Model)
recm_model <- recm(ardl_model, case = 2)
summary(recm_model)
```

```
##
## Time series regression with "zooreg" data:
## Start = 2, End = 251
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.028231 -0.005032 -0.000332  0.004966  0.042486
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## d(lc_QQQ)  0.9550966  0.0443248  21.548   <2e-16 ***
## ect       -0.0018724  0.0007572  -2.473   0.0141 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009063 on 248 degrees of freedom
##   (0 observations deleted due to missingness)
## Multiple R-squared:  0.6523, Adjusted R-squared:  0.6495
## F-statistic: 232.6 on 2 and 248 DF,  p-value: < 2.2e-16
```

```r
# VAR
VARselect(ardl_data, lag.max = 4, type = 'const')
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
##
## $criteria
##                    1             2             3             4
## AIC(n) -1.807892e+01 -1.805673e+01 -1.802690e+01 -1.802858e+01
## HQ(n)  -1.804460e+01 -1.799952e+01 -1.794682e+01 -1.792562e+01
## SC(n)  -1.799367e+01 -1.791465e+01 -1.782799e+01 -1.777284e+01
## FPE(n)  1.407426e-08  1.439023e-08  1.482617e-08  1.480175e-08
```

```
# estimation
vare_diff = VAR(ardl_data, p = 1, type = 'const')
summary(vare_diff)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: lc_AAPL, lc_QQQ
## Deterministic variables: const
## Sample size: 250
## Log Likelihood: 1557.564
## Roots of the characteristic polynomial:
## 0.9853 0.9853
## Call:
## VAR(y = ardl_data, p = 1, type = "const")
##
##
## Estimation results for equation lc_AAPL:
## ========================================
## lc_AAPL = lc_AAPL.l1 + lc_QQQ.l1 + const
##
##            Estimate Std. Error t value Pr(>|t|)
## lc_AAPL.l1 0.985328   0.011641  84.641   <2e-16 ***
## lc_QQQ.l1  0.004387   0.019767   0.222    0.825
## const      0.049024   0.090920   0.539    0.590
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01527 on 247 degrees of freedom
## Multiple R-Squared: 0.9801,  Adjusted R-squared: 0.9799
## F-statistic:  6080 on 2 and 247 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation lc_QQQ:
## =======================================
## lc_QQQ = lc_AAPL.l1 + lc_QQQ.l1 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## lc_AAPL.l1 -0.013401   0.009797  -1.368   0.1726
## lc_QQQ.l1   0.985182   0.016636  59.221   <2e-16 ***
## const       0.154797   0.076518   2.023   0.0441 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01285 on 247 degrees of freedom
## Multiple R-Squared: 0.9589,  Adjusted R-squared: 0.9586
## F-statistic:  2880 on 2 and 247 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##           lc_AAPL    lc_QQQ
## lc_AAPL 0.0002331 0.0001577
## lc_QQQ  0.0001577 0.0001651
##
## Correlation matrix of residuals:
##         lc_AAPL lc_QQQ
## lc_AAPL  1.0000 0.8038
## lc_QQQ   0.8038 1.0000
```

```
# residuals test
serial.test(vare_diff)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object vare_diff
## Chi-squared = 46.079, df = 60, p-value = 0.9071
```

```
# forecast of differenced data
varf_diff = predict(vare_diff, n.ahead = 1, ci = 0.95)
exp(varf_diff$fcst$lc_AAPL)
```

```
##                  fcst    lower    upper       CI
## lc_AAPL.fcst 173.9203 168.7935 179.2028 1.030373
```

```
exp(varf_diff$fcst$lc_QQQ)
```

```
##                 fcst    lower    upper       CI
## lc_QQQ.fcst 359.0921 350.1625 368.2493 1.025501
```

```
# The true values 174.72 and 359.35 are inside the 95% CIs.

# 5
upper_interval-lower_interval # range
```

```
## [1] 13.19513
```

```
abs(price_forecast-174.72) # error
```

```
## [1] 0.6292605
```

```r
# 10
exp(varf_diff$fcst$lc_AAPL)[3]-exp(varf_diff$fcst$lc_AAPL)[2] # range
```

```
## [1] 10.40934
```

```r
abs(exp(varf_diff$fcst$lc_AAPL)[1]-174.72) # error
```

```
## [1] 0.7996923
```

```r
# The 95 CI of VAR(1) is narrower, but the prediction error of ARIMA-garch is lower
# Both have their own advantages
```