HuYuDataInsight LLC

Zhaowei Cai

2024-05-06

(a)

```
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
```

```r
library(tseries)
```

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

```r
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
```

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

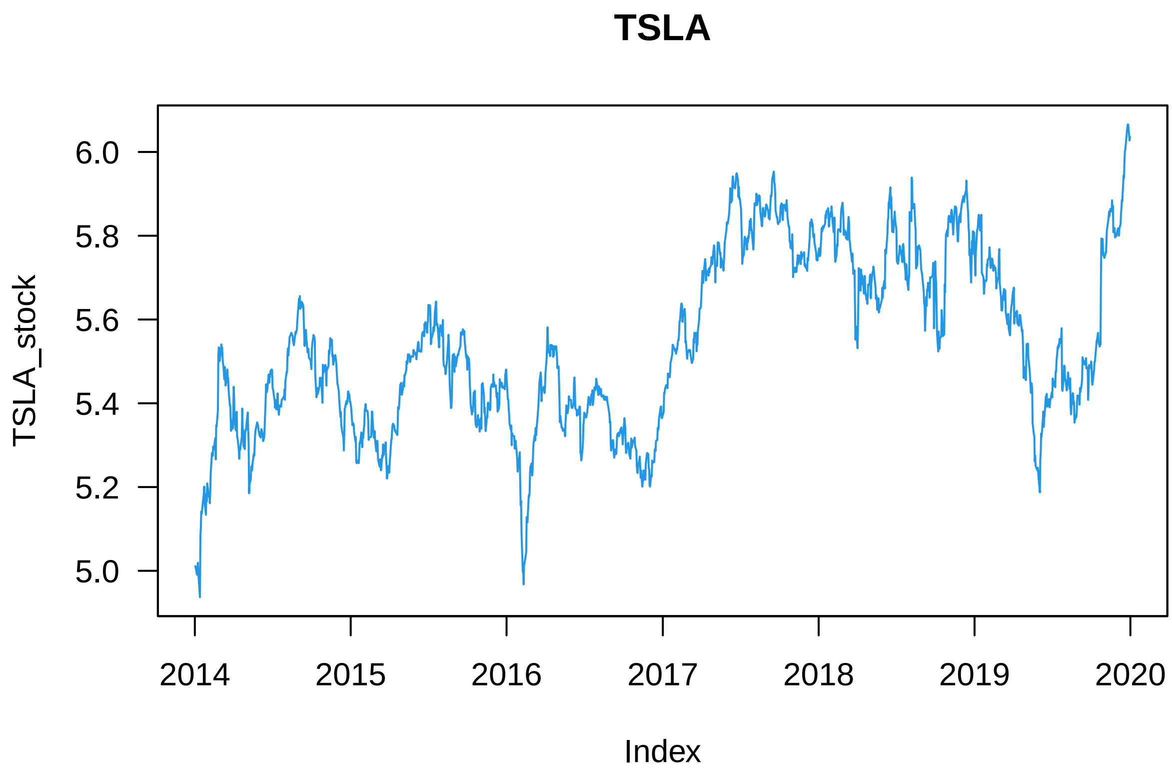
```
## Warning: package 'rugarch' was built under R version 4.2.3
```

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

```
##
## Attaching package: 'rugarch'
```

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

```r
data = read.csv('TSLA1.csv')
closing = data$Close # closing price
log_closing = log(data$Close) # log closing price
log_return = na.omit(diff(log(data$Close))) # log return

# Visualize the data
time = as.Date(data$Date, format = '%m/%d/%y')
df = data.frame(datefield = time, TSLA = log_closing)
TSLA_stock = with(df, zoo(TSLA, order.by = time))
plot.zoo(TSLA_stock, col=4, las=1, main="TSLA")
```

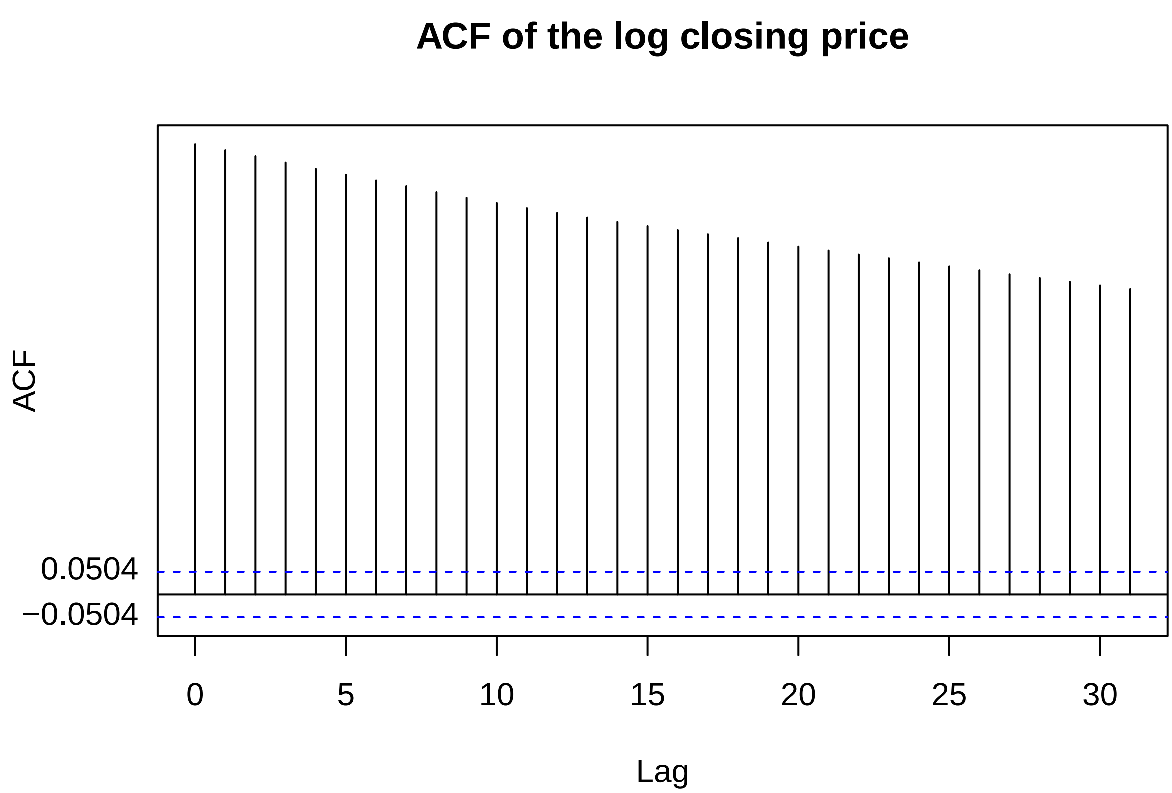



```
##Check for the trend (the Augmented Dickey-Fuller (ADF) test)
summary(ur.df(log_closing, type='trend', lags=20, selectlags="BIC"))
```

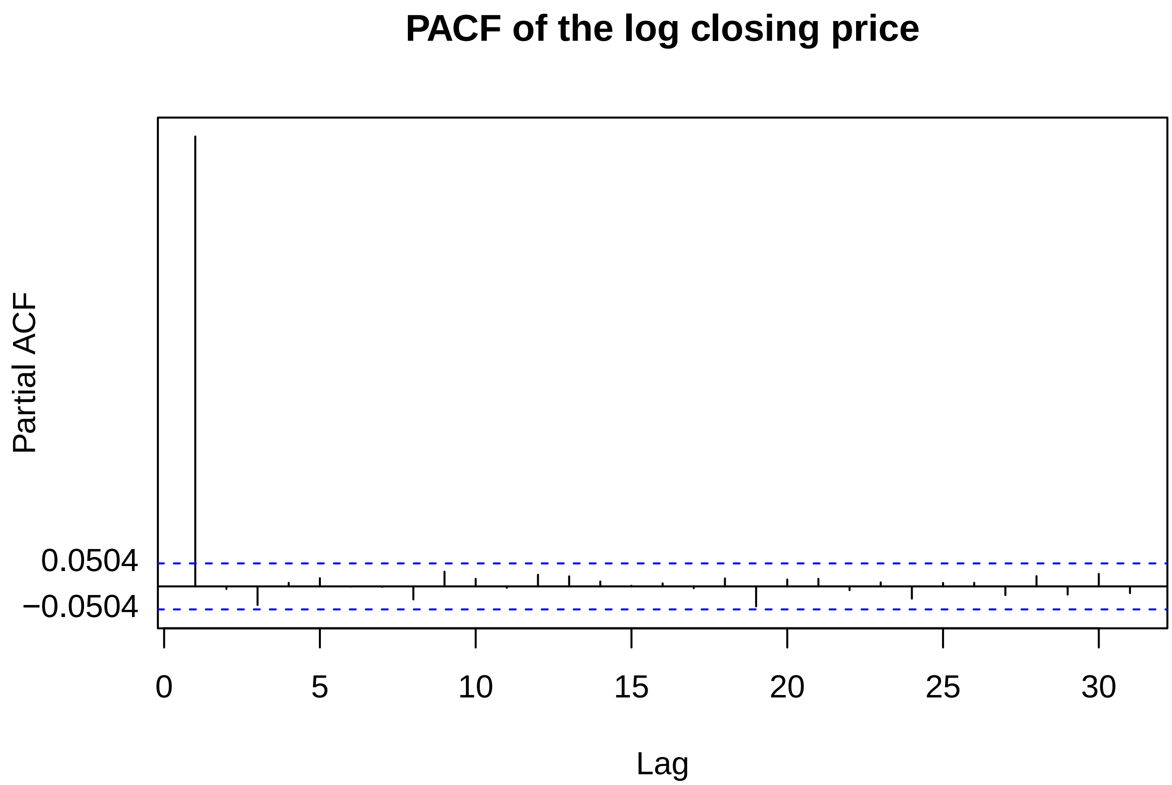
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.149411 -0.014050 -0.000271  0.015169  0.159921
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.117e-02  2.313e-02   3.077  0.00213 **
## z.lag.1     -1.325e-02  4.313e-03  -3.072  0.00217 **
## tt           3.796e-06  2.037e-06   1.864  0.06259 .
## z.diff.lag   1.171e-02  2.597e-02   0.451  0.65220
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02827 on 1485 degrees of freedom
## Multiple R-squared:  0.006368,   Adjusted R-squared:  0.004361
## F-statistic: 3.172 on 3 and 1485 DF,  p-value: 0.02344
##
##
## Value of test-statistic is: -3.0718 3.3546 4.7416
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

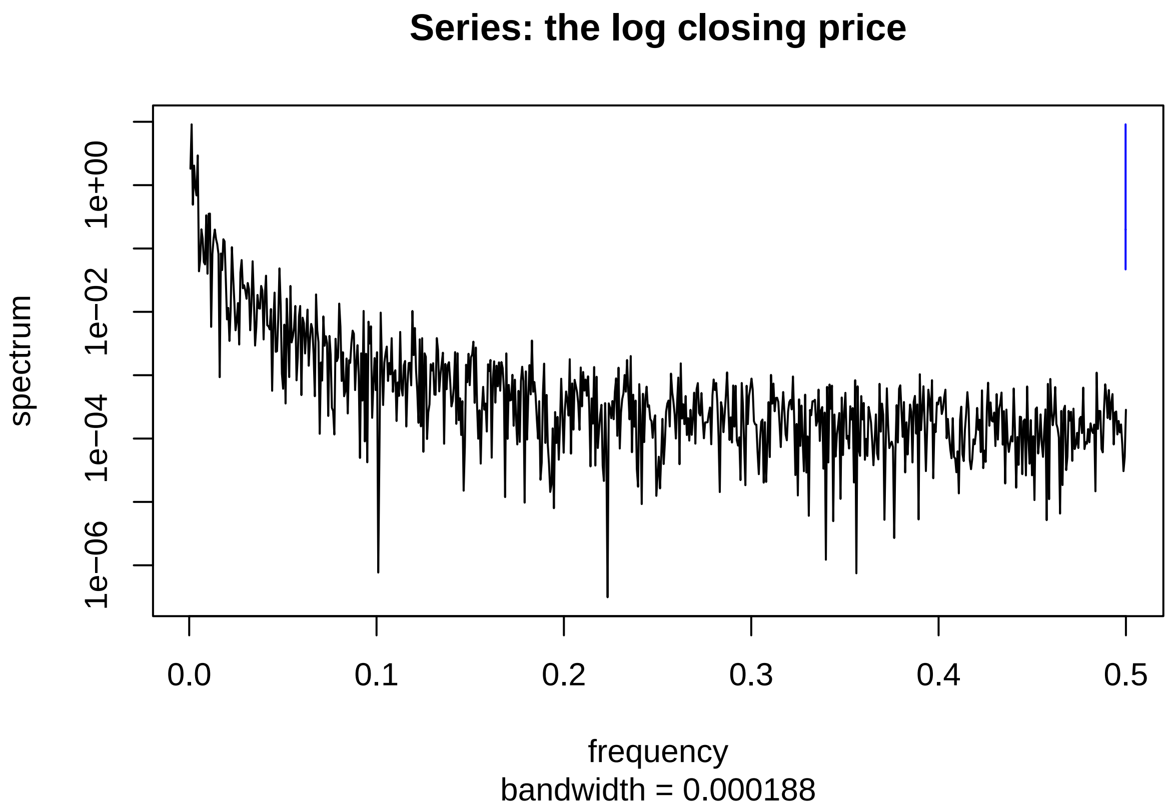
```
# From the result, we can see that the intercept is significantly different from 0. It means that the m
# Also, there is no linear trend for this time series because the coefficient for tt is not significant


##Check for the seasonality
n = length(log_closing)
acf(log_closing,main="ACF of the log closing price",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_closing,main="PACF of the log closing price",yaxt="n")
text(y=ci,par("usr")[1],labels=round(ci,4),pos=2,xpd=TRUE)
```




```
spec.pgram(log_closing,main="Series: the log closing price")
```




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

# also
adf.test(log_closing)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log_closing
## Dickey-Fuller = -3.1586, Lag order = 11, p-value = 0.09505
## alternative hypothesis: stationary
```

```r
# accept the null hypothesis of non-stationary
# difference is needed.
# log_return = diff(log_closing)
```

(b)

```r
# Remove the drift
# Demean or Difference
adf.test(log_closing)
```

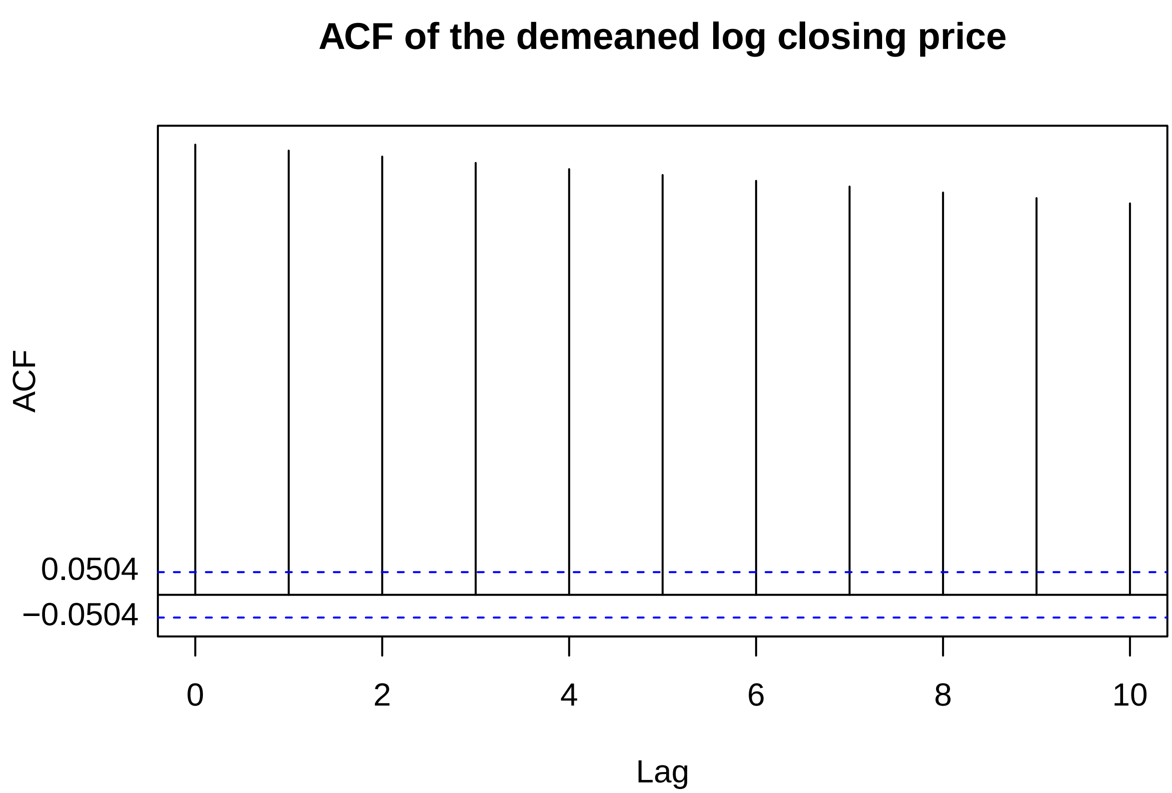
```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  log_closing
## Dickey-Fuller = -3.1586, Lag order = 11, p-value = 0.09505
## alternative hypothesis: stationary
```

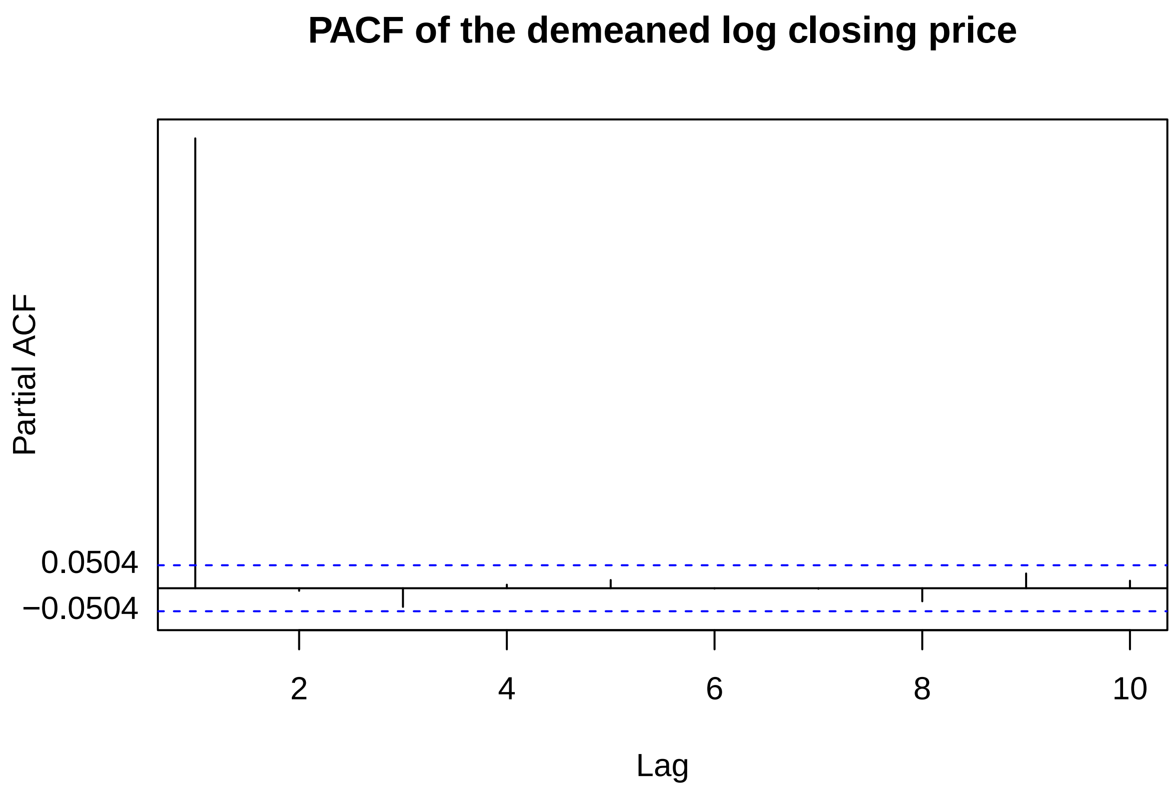
```r
# We know that difference is needed

# 1) demean:
mean(log_closing)
```

```
## [1] 5.543693
```

```r
log_closing1=log_closing-mean(log_closing)

acf(log_closing1,lag=10,main="ACF of the demeaned log closing price",yaxt="n")
text(y=ci,par("usr")[1],labels=round(ci,4),pos=2,xpd=TRUE)
```



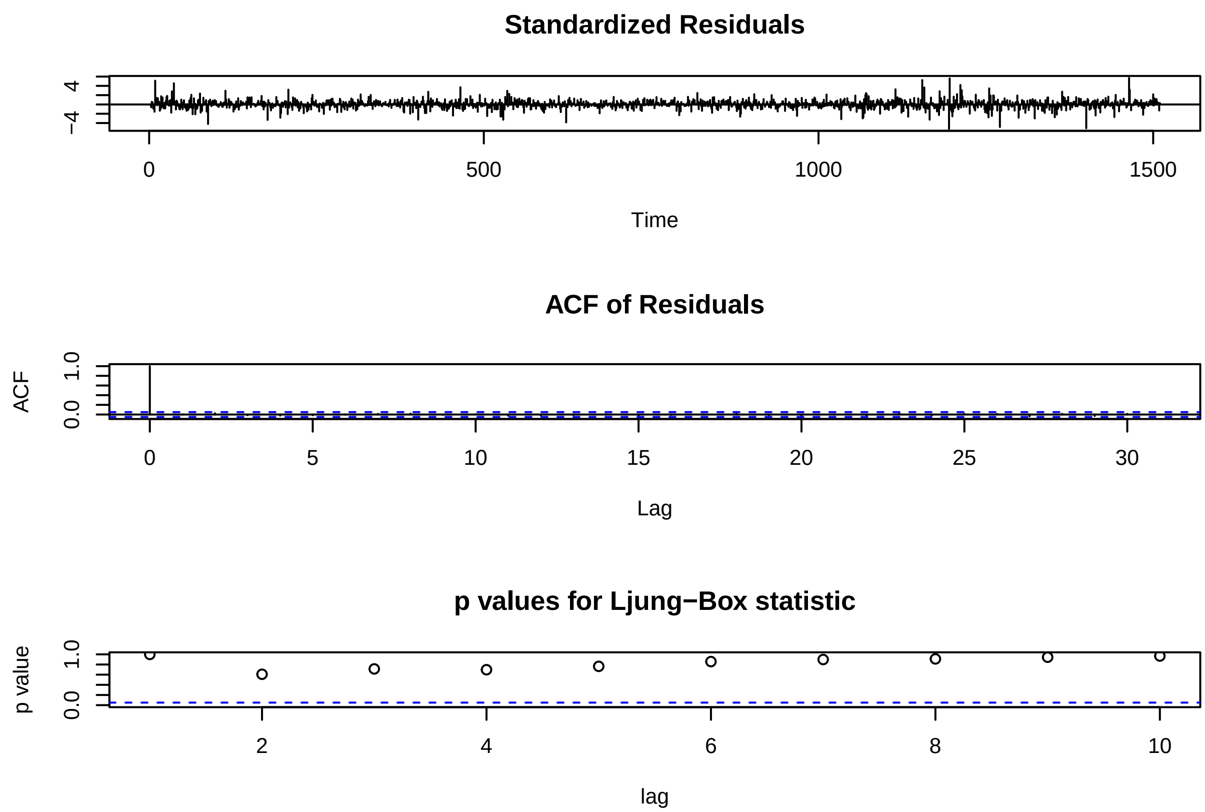

```r
pacf(log_closing1,lag=10,main="PACF of the demeaned log closing price",yaxt="n")
text(y=ci,par("usr")[1],labels=round(ci,4),pos=2,xpd=TRUE)
```




```
fit1 = auto.arima(log_closing1, max.p=25, max.q=25, ic="bic",
                        seasonal=F, lambda=NULL,
                        stepwise=FALSE, approximation=FALSE
                        )
summary(fit1)
```

```
## Series: log_closing1
## ARIMA(0,1,0)
##
## sigma^2 = 0.0008158:  log likelihood = 3224.32
## AIC=-6446.64   AICc=-6446.64   BIC=-6441.33
##
## Training set error measures:
##                        ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.0006784347 0.02855288 0.02009258 -16.66241 48.88883 0.9993553
##                       ACF1
## Training set -5.077176e-05
```

```
# ARIMA(0,1,0)
# also shows that difference is needed
tsdiag(fit1)
```




```r
shapiro.test(fit1$residuals)
```

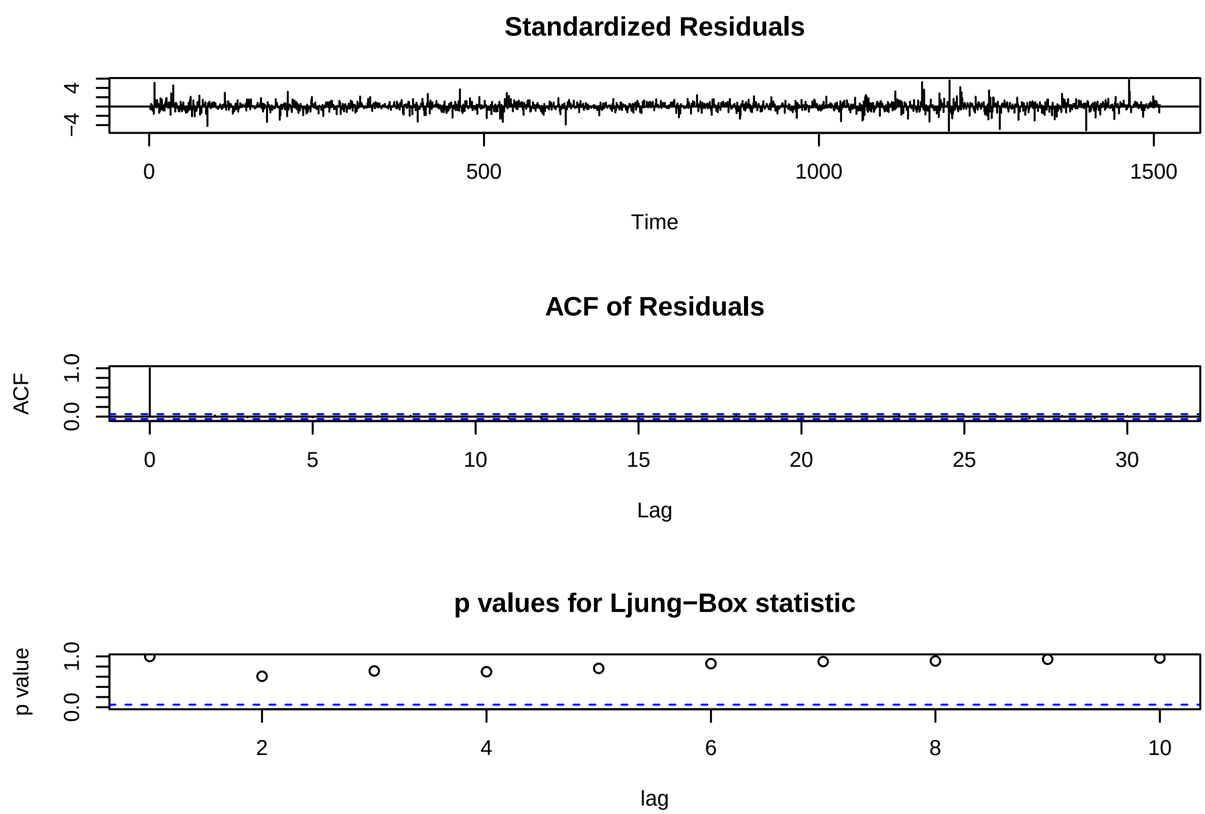
```
##
##  Shapiro-Wilk normality test
##
## data:  fit1$residuals
## W = 0.9454, p-value < 2.2e-16
```

```r
# 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

# 2) difference
# log return = diff(log closing)
# we can difference the data first and fit the log return
fit2 = auto.arima(log_return, max.p=25, max.q=25, ic="bic",
                       seasonal=F, lambda=NULL,
                       stepwise=FALSE, approximation=FALSE
                       )
summary(fit2) # ARMA(0,0)
```

```
## Series: log_return
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 0.0008158:  log likelihood = 3224.32
## AIC=-6446.64   AICc=-6446.64   BIC=-6441.33
```

```
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE
## Training set 0.0006792371 0.02856234 0.02010554 100  100 0.6782608
##                       ACF1
## Training set -5.498476e-05
```

```r
tsdiag(fit2)
```




```r
# Check the normality
shapiro.test(fit2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit2$residuals
## W = 0.94548, p-value < 2.2e-16
```

(c)

```r
prediction <- forecast(fit1, h=1, level=0.95)
(lower_interval <- as.numeric(exp(prediction$lower+mean(log_closing))))
```

```
## [1] 395.5548
```

```r
(price_forecast <- as.numeric(exp(prediction$mean+mean(log_closing))))
```

```
## [1] 418.33
```

```r
(upper_interval <- as.numeric(exp(prediction$upper+mean(log_closing))))
```

```
## [1] 442.4165
```

```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: 418.33
```

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

```
## 95% Prediction Interval: ( 395.5548 ,  442.4165 )
```

(d)

```r
# using log return
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
## -0.150974 -0.014073 -0.000256  0.015433  0.161771
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.372e-04  1.501e-03   0.225    0.822
## z.lag.1     -9.729e-01  3.661e-02 -26.572   <2e-16 ***
## tt           2.942e-07  1.712e-06   0.172    0.864
## z.diff.lag  -2.265e-02  2.596e-02  -0.873    0.383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02836 on 1484 degrees of freedom
## Multiple R-squared:  0.4981, Adjusted R-squared:  0.4971
## F-statistic: 490.9 on 3 and 1484 DF,  p-value: < 2.2e-16
##
```

```
##
## Value of test-statistic is: -26.5723 235.3642 353.0462
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

```
# No drift, no trend
# Stationary

# 1) default mean model of ARMA(1,1)
garch_spec <- ugarchspec()
garch_fit1 <- ugarchfit(spec = garch_spec, data = log_return)
garch_fit1
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000750    0.000717    1.04679 0.295198
## ar1     0.275770    0.544296    0.50665 0.612397
## ma1    -0.255409    0.547007   -0.46692 0.640556
## omega   0.000004    0.000003    1.25627 0.209018
## alpha1  0.014946    0.003187    4.69012 0.000003
## beta1   0.979437    0.000958 1022.33579 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000750    0.000708    1.06048  0.28892
## ar1     0.275770    0.225387    1.22354  0.22112
## ma1    -0.255409    0.229451   -1.11313  0.26565
## omega   0.000004    0.000017    0.25652  0.79755
## alpha1  0.014946    0.014901    1.00299  0.31587
## beta1   0.979437    0.000910 1076.79977  0.00000
##
## LogLikelihood : 3259.295
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.3119
## Bayes        -4.2907
```

```
## Shibata      -4.3119
## Hannan-Quinn -4.3040
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                      0.1607  0.6885
## Lag[2*(p+q)+(p+q)-1][5]     1.0799  1.0000
## Lag[4*(p+q)+(p+q)-1][9]     1.7562  0.9925
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                       5.801 0.01601
## Lag[2*(p+q)+(p+q)-1][5]      8.065 0.02839
## Lag[4*(p+q)+(p+q)-1][9]      9.602 0.06095
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     2.562 0.500 2.000  0.1094
## ARCH Lag[5]     3.075 1.440 1.667  0.2791
## ARCH Lag[7]     4.047 2.315 1.543  0.3399
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  3.0783
## Individual Statistics:
## mu     0.0439
## ar1    0.1005
## ma1    0.1013
## omega  0.6558
## alpha1 0.2302
## beta1  0.2492
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.49 1.68 2.12
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias            1.457 0.1452217
## Negative Sign Bias   3.496 0.0004858 ***
## Positive Sign Bias   1.135 0.2565069
## Joint Effect        14.891 0.0019121 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
```

```
## 1    20     105.4    5.619e-14
## 2    30     108.1    4.724e-11
## 3    40     130.3    9.124e-12
## 4    50     135.7    4.511e-10
##
##
## Elapsed time : 0.303895
```

```r
# 2) Fit the mean model first
arma_model <- auto.arima(log_return)
arma_model
```

```
## Series: log_return
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 0.0008158:  log likelihood = 3224.32
## AIC=-6446.64   AICc=-6446.64   BIC=-6441.33
```

```r
garch_spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                         mean.model = list(armaOrder = c(0,0)))
garch_fit2 <- ugarchfit(spec = garch_spec, data = arma_model$residuals)
garch_fit2
```

```
##
## *---------------------------------*
## *          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      0.000799    0.000698    1.1449  0.25226
## omega   0.000004    0.000003    1.3713  0.17029
## alpha1  0.014608    0.002816    5.1869  0.00000
## beta1   0.979899    0.000918 1067.9325  0.00000
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.000799    0.000704    1.13484  0.25644
## omega   0.000004    0.000014    0.30178  0.76282
## alpha1  0.014608    0.012086    1.20874  0.22676
## beta1   0.979899    0.000778 1260.20382  0.00000
##
## LogLikelihood : 3258.961
##
## Information Criteria
## ------------------------------------
```

```
##
## Akaike       -4.3141
## Bayes        -4.3000
## Shibata      -4.3141
## Hannan-Quinn -4.3088
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1390  0.7092
## Lag[2*(p+q)+(p+q)-1][2]    0.4474  0.7188
## Lag[4*(p+q)+(p+q)-1][5]    1.2400  0.8034
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      6.243 0.01247
## Lag[2*(p+q)+(p+q)-1][5]     8.596 0.02092
## Lag[4*(p+q)+(p+q)-1][9]    10.107 0.04759
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     2.705 0.500 2.000  0.1000
## ARCH Lag[5]     3.144 1.440 1.667  0.2695
## ARCH Lag[7]     4.089 2.315 1.543  0.3340
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  3.4246
## Individual Statistics:
## mu     0.05688
## omega  0.73809
## alpha1 0.24092
## beta1  0.26804
##
## 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            1.328 0.1844894
## Negative Sign Bias   3.406 0.0006775 ***
## Positive Sign Bias   1.037 0.2998853
## Joint Effect        14.186 0.0026630 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
```

```
##   group statistic p-value(g-1)
## 1    20     110.4    6.831e-15
## 2    30     123.9    1.085e-13
## 3    40     137.8    5.975e-13
## 4    50     143.5    3.284e-11
##
##
## Elapsed time : 0.124557
```

```
# 3) If difference is needed (here no need)
arma_model <- auto.arima(diff(log_return))
arma_model
```

```
## Series: diff(log_return)
## ARIMA(5,0,0) with zero mean
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5
##       -0.8236  -0.6236  -0.4739  -0.3332  -0.1744
## s.e.   0.0254   0.0319   0.0336   0.0319   0.0254
##
## sigma^2 = 0.0009615:  log likelihood = 3100.27
## AIC=-6188.55   AICc=-6188.49   BIC=-6156.64
```

```
garch_spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
                         mean.model = list(armaOrder = c(5,0)))
garch_spec <- ugarchspec()
garch_fit3 <- ugarchfit(spec = garch_spec, data = arma_model$residuals)
garch_fit3
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error     t value Pr(>|t|)
## mu      0.000013    0.000007  1.8511e+00 0.064150
## ar1     0.769639    0.016431  4.6841e+01 0.000000
## ma1    -0.998579    0.000036 -2.7406e+04 0.000000
## omega   0.000005    0.000005  9.8996e-01 0.322195
## alpha1  0.016088    0.004199  3.8310e+00 0.000128
## beta1   0.977968    0.001087  8.9943e+02 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error     t value Pr(>|t|)
```

```
## mu      0.000013    0.000008     1.60791  0.10785
## ar1     0.769639    0.051922    14.82299  0.00000
## ma1    -0.998579    0.000573 -1742.08978  0.00000
## omega   0.000005    0.000026     0.18593  0.85250
## alpha1  0.016088    0.023025     0.69870  0.48474
## beta1   0.977968    0.002178   448.99692  0.00000
##
## LogLikelihood : 3228.07
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.2733
## Bayes        -4.2521
## Shibata      -4.2733
## Hannan-Quinn -4.2654
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic   p-value
## Lag[1]                      9.418 2.148e-03
## Lag[2*(p+q)+(p+q)-1][5]    15.163 0.000e+00
## Lag[4*(p+q)+(p+q)-1][9]    31.946 5.773e-15
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic  p-value
## Lag[1]                      6.754 0.009356
## Lag[2*(p+q)+(p+q)-1][5]     9.156 0.015115
## Lag[4*(p+q)+(p+q)-1][9]     9.872 0.053424
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     2.628 0.500 2.000  0.1050
## ARCH Lag[5]     2.670 1.440 1.667  0.3414
## ARCH Lag[7]     2.755 2.315 1.543  0.5610
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.3965
## Individual Statistics:
## mu     0.07879
## ar1    0.07333
## ma1    0.15970
## omega  0.43944
## alpha1 0.21470
## beta1  0.24712
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.49 1.68 2.12
```

```
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias          0.42987 0.667350
## Negative Sign Bias 2.80928 0.005029 ***
## Positive Sign Bias 0.03424 0.972692
## Joint Effect       9.94908 0.019004  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     110.0    7.847e-15
## 2    30     133.0    2.925e-15
## 3    40     139.2    3.597e-13
## 4    50     157.8    2.335e-13
##
##
## Elapsed time : 0.3071699
```

(e)

```r
# Use the garch_fit2 from d)
forecasted_returns <- ugarchforecast(garch_fit2, n.ahead = 1)

# Assuming the last observed closing price is on December 31, 2019
# You may need to replace this with the actual closing price date
last_close_price <- closing[1510]

# Forecast one day ahead (January 2, 2020)
(price_forecast <- as.numeric(last_close_price*exp(forecasted_returns@forecast$seriesFor)))
```

```
## [1] 418.6645
```

```r
# Calculate the 95% prediction interval
(lower_interval <- as.numeric(price_forecast * exp(qnorm(0.025) * forecasted_returns@forecast$sigmaFor))
```

```
## [1] 395.6544
```

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

```
## [1] 443.0126
```

```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: 418.6645
```

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

```
## 95% Prediction Interval: ( 395.6544 ,  443.0126 )
```

```r
# wider than the interval in c)
```