Linh tinh

2025-03-12

```r
library(tidyquant)
```

```
## Warning: 程序包'tidyquant'是用R版本4.4.3 来建造的
```

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

```
## Warning: 程序包'PerformanceAnalytics'是用R版本4.4.3 来建造的
```

```
## ── Attaching core tidyquant packages ─────────────────────── tidyquant 1.0.11 ──
## ✔ PerformanceAnalytics 2.0.8      ✔ TTR                  0.24.4
## ✔ quantmod             0.4.26     ✔ xts                  0.14.1
```

```
## ── Conflicts ──────────────────────────────────────────── tidyquant_conflicts() ──
## ✖ zoo::as.Date()                 masks base::as.Date()
## ✖ zoo::as.Date.numeric()         masks base::as.Date.numeric()
## ✖ PerformanceAnalytics::legend() masks graphics::legend()
## ✖ quantmod::summary()            masks base::summary()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

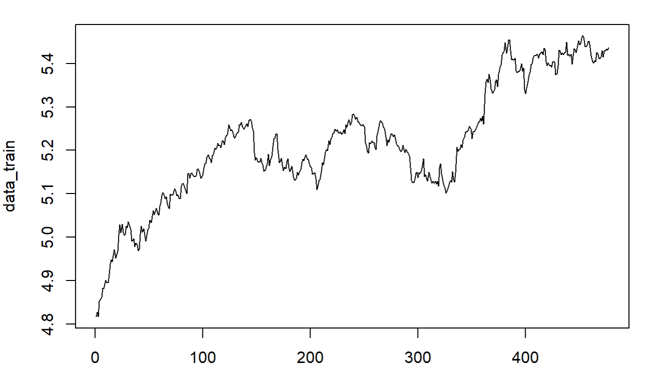
```r
dataset <- tq_get('AAPL',
               from = "2023-01-01",
               to = "2025-01-01",
               get = "stock.prices")

dataset <- dataset[, c("date", "adjusted")]
#dataset$x <- dataset$adjusted
#dataset
dataset$x <- log(dataset$adjusted)
head(dataset)
```

```
## # A tibble: 6 × 3
##   date       adjusted     x
##   <date>        <dbl> <dbl>
## 1 2023-01-03     124.  4.82
## 2 2023-01-04     125.  4.83
## 3 2023-01-05     124.  4.82
## 4 2023-01-06     128.  4.85
## 5 2023-01-09     129.  4.86
## 6 2023-01-10     129.  4.86
```

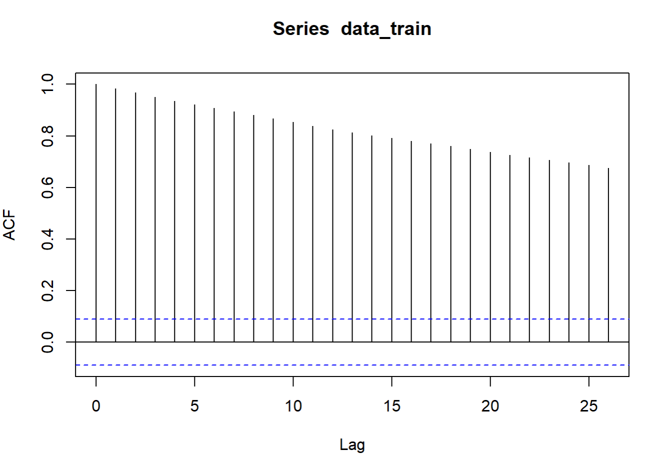
```r
#dataset <- read.csv("C:\\Users\\khuem\\Downloads\\arimadat(1).csv")
#dataset
```

series x is the log transform of the adjusted close price.

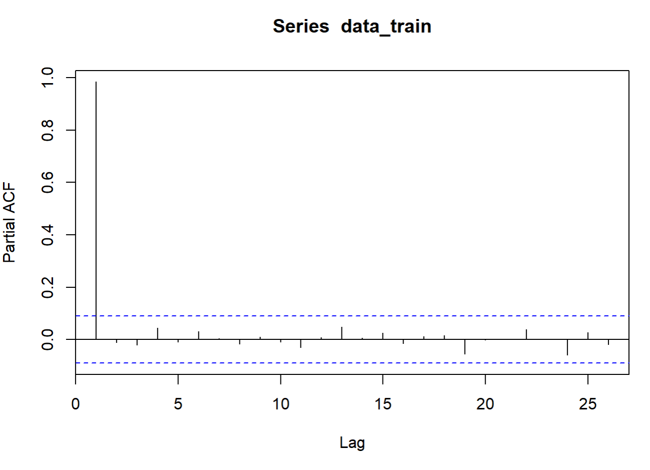
```r
n <- length(dataset$x)
data_train <- dataset$x[1:(n - 25)]
data_test <- dataset$x[(n - 25 + 1):n]
plot(data_train, type = 'l')
```




```r
acf(data_train)
```




```r
pacf(data_train)
```

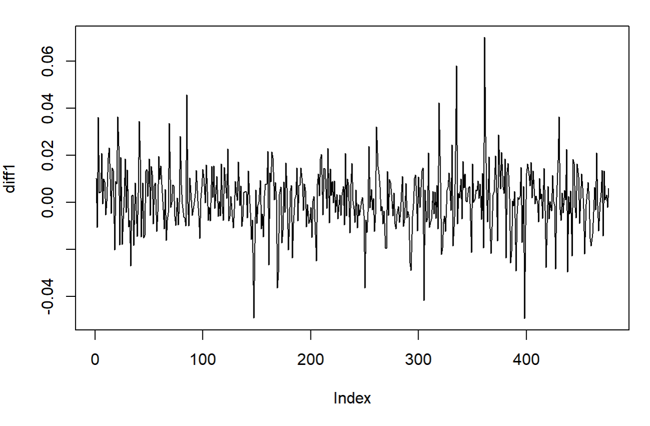


From the Series Plot and the PACF and ACF, the series is highly non-stationary. Therefore, taking difference is needed.

```r
diff1 <- diff(data_train)
diff1[1:10]
```

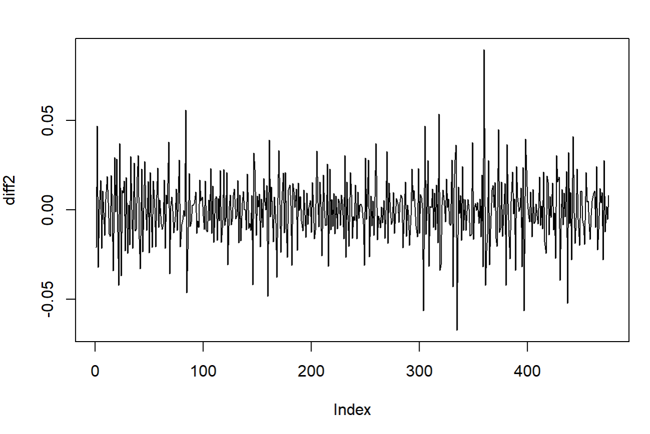
```
##  [1]  0.0102614842 -0.0106612614  0.0361332180  0.0040806730  0.0044463770
##  [6]  0.0208926222 -0.0005995192  0.0100683544  0.0087183437 -0.0053846653
```

```r
plot(diff1, type = 'l')
```

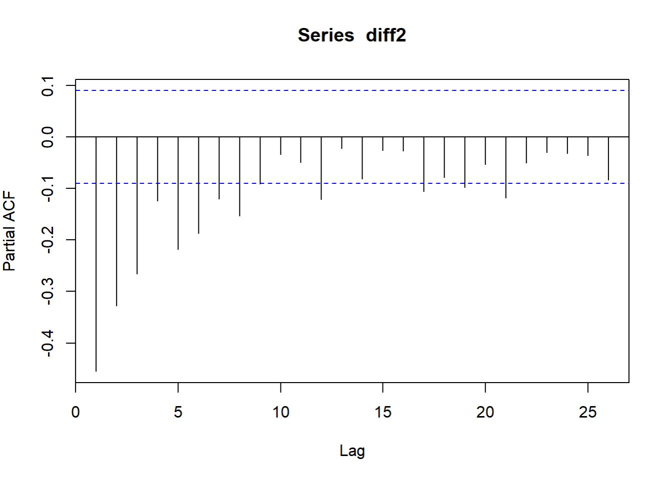


Diff1 series look much more stationary than the original data. However, there is still a visible trend in the data. Therefore, taking the 2nd order difference is needed.

```r
diff2 <- diff(diff1)
plot(diff2, type = 'l')
```




```r
pacf(diff2)
```




```r
ar_demo <- ar(diff2, aic = TRUE, method = 'mle')
ar_demo
```

```
## 
## Call:
## ar(x = diff2, aic = TRUE, method = "mle")
## 
## Coefficients:
##       1        2        3        4        5        6        7        8  
## -0.8565  -0.8074  -0.7513  -0.6489  -0.6669  -0.5604  -0.4397  -0.3642  
##       9       10       11       12  
## -0.2457  -0.1707  -0.1513  -0.1212  
## 
## Order selected 12  sigma^2 estimated as  0.000193
```

The diff2 plot looks very stationary.

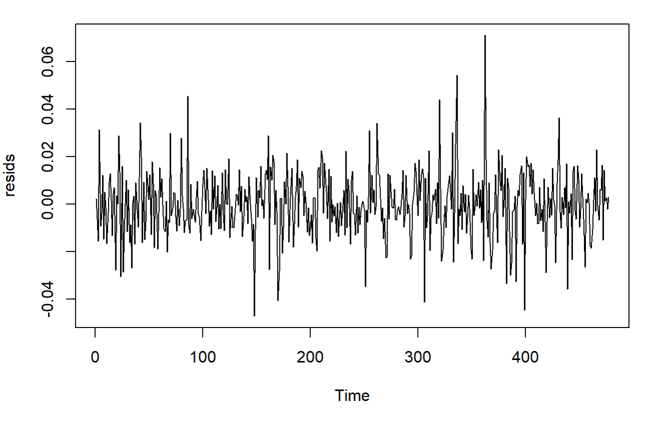
PACF plot of the diff2 series suggests that arima order 11 or 12 should be good.

Fitting ar function model to determine the order for the ARIMA model. The order selected for the arima model by the ar function is 12. Therefore, later a complete ARIMA(12,2,0) will be fitted on the original adjusted close price for forecasting purpose.

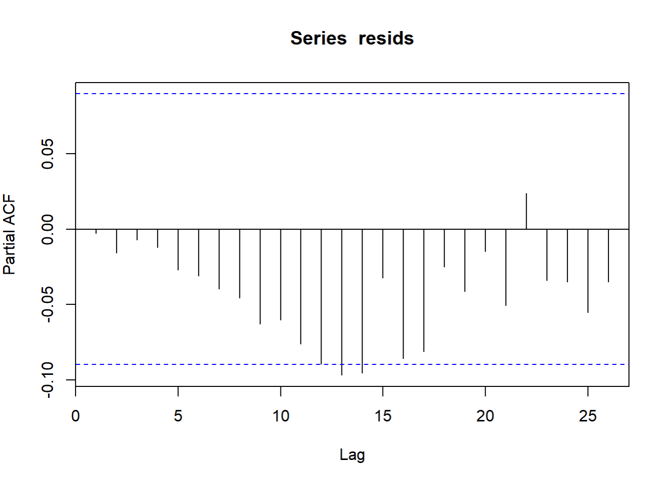
```r
ar_mle <- arima(data_train, order = c(12,2,0))
ar_mle
```

```
## 
## Call:
## arima(x = data_train, order = c(12, 2, 0))
## 
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.8565  -0.8073  -0.7512  -0.6488  -0.6668  -0.5604  -0.4397  -0.3641
## s.e.   0.0456   0.0599   0.0701   0.0773   0.0811   0.0843   0.0842   0.0808
##           ar9     ar10     ar11     ar12
##       -0.2456  -0.1707  -0.1513  -0.1212
## s.e.   0.0770   0.0697   0.0597   0.0458
## 
## sigma^2 estimated as 0.000193:  log likelihood = 1356.44,  aic = -2686.89
```

```r
resids <- ar_mle$resid
plot(resids, type = 'l')
```




```r
pacf(resids)
```




```r
library(tseries)
```

```
## Warning: 程序包'tseries'是用R版本4.4.3 来建造的
```

```r
adf_test <- adf.test(resids, alternative = "stationary")
```

```
## Warning in adf.test(resids, alternative = "stationary"): p-value smaller than
## printed p-value
```

```r
print("ADF Test:")
```

```
## [1] "ADF Test:"
```

```r
print(adf_test)
```

```
## 
##  Augmented Dickey-Fuller Test
## 
## data:  resids
## Dickey-Fuller = -8.6464, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

The model above is the fitted ARIMA(12,2,0) model.

The residuals of the ARIMA model strongly resemblances a white noise process, thus proving that the ARIMA(12,2,0) is enough to capture all the trend of the data series.

The PACF of the residuals also proves the above statement.

To be more confirmative, Augmented Dickey-Fuller Test is conducted. The null hypothesis of non-stationary series is rejected with a p-value = 0.01 (highly significant). The conclusion is stationary.

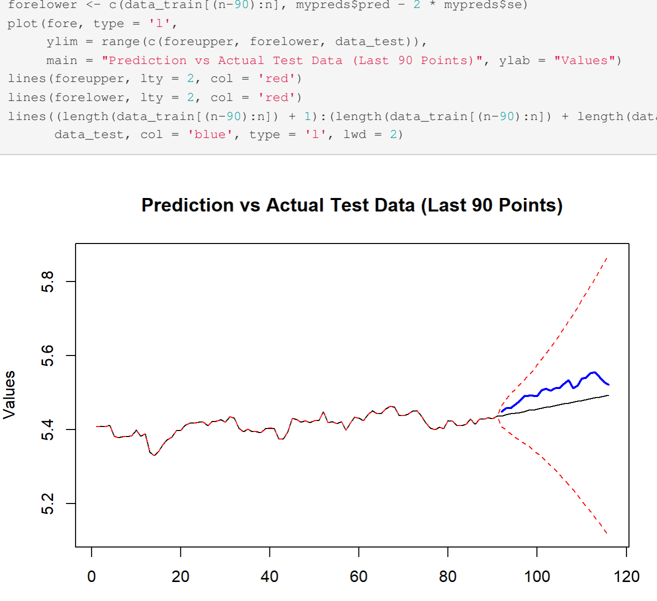
```r
library(forecast);
```

```
## Warning: 程序包'forecast'是用R版本4.4.3 来建造的
```

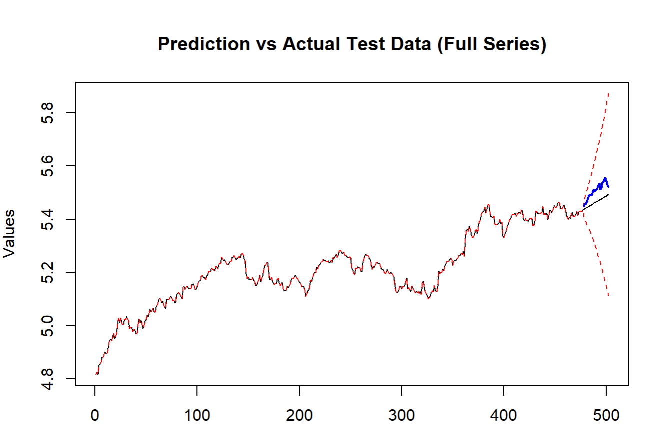
```r
fitted_values <- fitted(ar_mle)
fitted_values[1:10]
```

```
##  [1] 4.815159 4.834015 4.832549 4.821791 4.866554 4.865754 4.870361 4.896607
##  [9] 4.886928 4.902056
```

```r
mypreds <- predict(ar_mle, n.ahead = 25, se.fit = TRUE)
n <- length(data_train)
fore <- c(data_train[(n-90):n], mypreds$pred)
foreupper <- c(data_train[(n-90):n], mypreds$pred + 2 * mypreds$se)
forelower <- c(data_train[(n-90):n], mypreds$pred - 2 * mypreds$se)
plot(fore, type = 'l',
     ylim = range(c(foreupper, forelower, data_test)),
     main = "Prediction vs Actual Test Data (Last 90 Points)", ylab = "Values")
lines(foreupper, lty = 2, col = 'red')
lines(forelower, lty = 2, col = 'red')
lines((length(data_train[(n-90):n]) + 1):(length(data_train[(n-90):n]) + length(data_test)),
      data_test, col = 'blue', type = 'l', lwd = 2)
```




```r
fore <- c(data_train, mypreds$pred)
foreupper <- c(data_train, mypreds$pred + 2 * mypreds$se)
forelower <- c(data_train, mypreds$pred - 2 * mypreds$se)
plot(fore, type = 'l',
     ylim = range(c(foreupper, forelower, data_test)),
     main = "Prediction vs Actual Test Data (Full Series)",
     ylab = "Values", xlab = "Time")
lines(foreupper, lty = 2, col = 'red')
lines(forelower, lty = 2, col = 'red')
lines((n + 1):(n + length(data_test)),
      data_test, col = 'blue', type = 'l', lwd = 2)
```




```r
# performance metrics for diff2 arima model
print('Performance metrics for diff2 arima')
```

```
## [1] "Performance metrics for diff2 arima"
```

```r
mae <- mean(abs(data_test - mypreds$pred))
mse <- mean((data_test - mypreds$pred)^2)
rmse <- sqrt(mse)
mape <- mean(abs((data_test - mypreds$pred) / data_test)) * 100
ss_res <- sum((data_test - mypreds$pred)^2)
ss_tot <- sum((data_test - mean(data_test))^2)
r_squared <- 1 - (ss_res / ss_tot)

cat("MAE:", mae, "\n")
```

```
## MAE: 0.04246977
```

```r
# cat("MSE:", mse, "\n")
cat("RMSE:", rmse, "\n")
```

```
## RMSE: 0.04510931
```

```r
cat("MAPE:", mape, "%\n")
```

```
## MAPE: 0.7697511 %
```

```r
cat("R-squared:", r_squared, "\n")
```

```
## R-squared: -1.398569
```

The error metrics (MAE, MSE, RMSE, MAPE) suggest that the model’s predictions are relatively accurate.