Microsoft Stock Time Series Model Project

Khanh Vy Ho

2025-02-02

Introduction

Microsoft Stock Time Series Model Project is my personal project with a goal to answer if weekly Mean Volume and weekly Past Net Price affect weekly Current Net Price of Microsoft stock (from 2015/04/01 to 2021/03/31).

My initial hypothesis is Past volume is positively associated with Current Net Price while Past Net Price is negatively associated with Current Net Price.

Specifically, I aim to apply 2 models using for time-series data including: Autoregressive Model (AR) Autoregressive Distributed Lag Model (ARDL) to select the best fit.

- **Autoregressive Model (AR):** is a type of time-series model where the current value of a variable depends on its past values plus a stochastic (random) error term.
- **Autoregressive Distributed Lag Model (ARDL):** is an extension of the AR model that allows for both short-run and long-run relationships between a dependent variable and one or more independent variables.

Set Up

The raw dataset used for the analysis is available at `./data/Microsoft_Stock.csv`. It was downloaded from Kaggle.

- microsoftdata has 1511 observations of 6 variables. These variables include Date, Open, High, Low, Close, and Volume.
- weekly_microsoft_data is the dataset we will use to analyze. The original dataset provides daily data, which gives non-stationary results. We convert into weekly data for stationary. This dataset has 314 observations of 4 variables. These variables include Week (week), Date (date), Net Price (net_price), and Mean Volume (mean_volume).

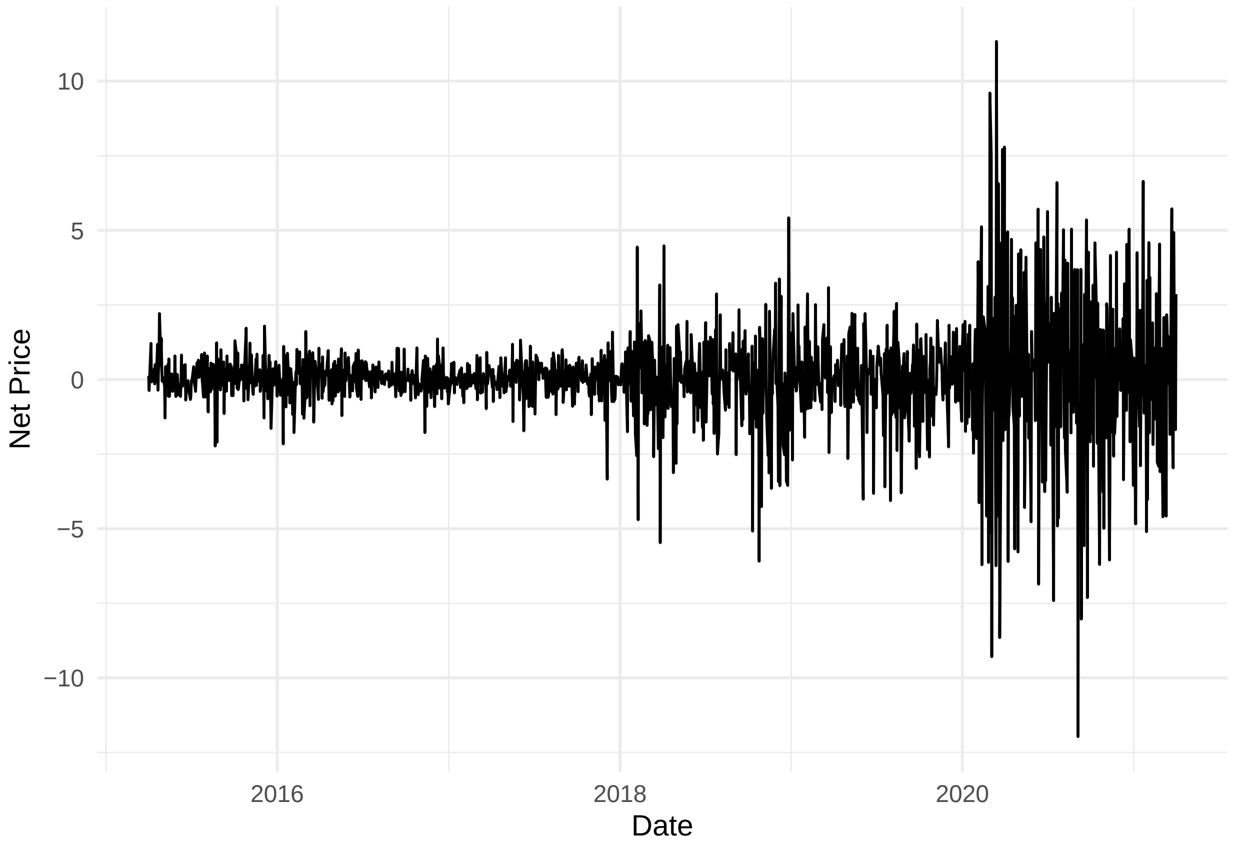
```
library("tidyverse")
microsoftdata <- read.csv("../data/Microsoft_Stock.csv")

#Convert date-time strings into date-time objects
microsoftdata$Date <- as.POSIXct(microsoftdata$Date,
                                  format = "%m/%d/%Y %H:%M:%S")
```

Stationary

Stationary is a key assumption required for external validity of time series regression. Stationary implies that means and variances do not change over time.

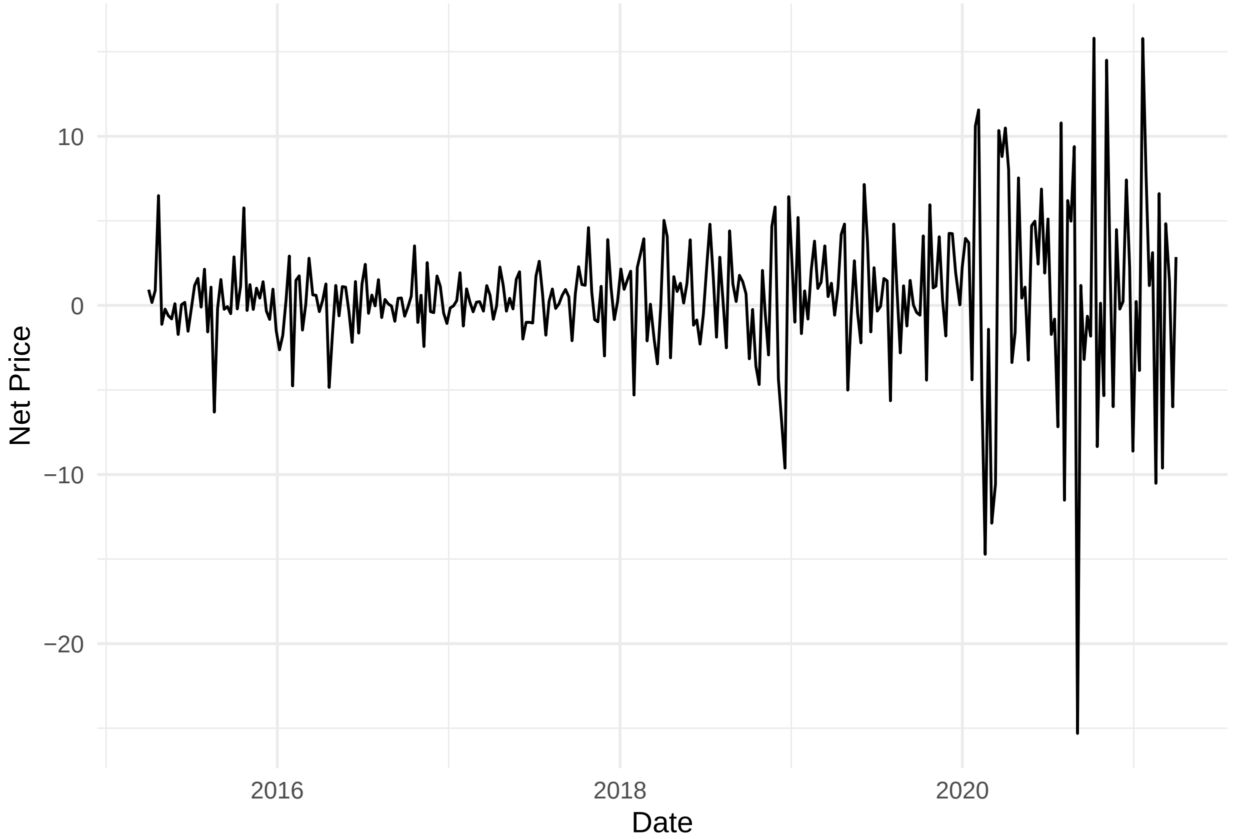
```
microsoftdata |>
  mutate(net_price = Close - Open) |>
  ggplot(mapping = aes(x = Date, y = net_price)) +
  geom_line() +
  theme_minimal() +
  labs(y = "Net Price")
```



Even though the means do not change over time, the variances appear to become larger over time, implying the non-stationary. Therefore, we change the granularity of the data from daily to weekly.

```
# Create column "Net Price" as the difference from Close and Open and column "Week"
first_day <- make_date(2015,04,01)
weekly_microsoft_data <- microsoftdata |>
  mutate(week = floor(difftime(Date, first_day, units = "weeks")))|>
  group_by(week)|>
  summarize(daily_mean_volume = mean(Volume), open_price = first(Open),
            close_price = last(Close),
            date = first(Date), .groups = "drop")|>
  mutate(net_price = close_price - open_price)|>
  select(c(week, date, net_price, daily_mean_volume))
```

```r
weekly_microsoft_data |>
  ggplot(mapping = aes(x = date, y = net_price)) +
  geom_line() +
  theme_minimal() +
  labs(x = "Date", y = "Net Price")
```



After changing to weekly, the means appear to remain constant around 0. The variances still vary, but less noticeable across time. Thus, moving forward, we will use weekly data for our analysis.

Data Summary

```r
#Check for missing observations
sum(is.na(microsoftdata))
```

```
## [1] 0
```

- There is no missing values.

```r
summary(weekly_microsoft_data)
```

```
##      week                date                              net_price
##  Length:314         Min.   :2015-04-01 16:00:00.00   Min.   :-25.3100
##  Class :difftime   1st Qu.:2016-09-30 10:00:00.00   1st Qu.: -0.8375
##  Mode  :numeric    Median :2018-04-01 04:00:00.00   Median :  0.4250
##                    Mean   :2018-03-31 20:18:55.02   Mean   :  0.5271
##                    3rd Qu.:2019-09-30 22:00:00.00   3rd Qu.:  1.9750
##                    Max.   :2021-03-31 16:00:00.00   Max.   : 15.8000
##  daily_mean_volume
##  Min.   :13470253
##  1st Qu.:22333112
##  Median :27709696
##  Mean   :30102903
##  3rd Qu.:34435879
##  Max.   :82774790
```

- The summary appears not have abnormal values. From 04/01/2015 to 03/031/2021, Microsoft stock's highest gain in a week is $15.8 and the highest loss is $25.31. Overall, Microsoft stock has gained $0.5271 every week on average. As for daily trading volume, there is no negative values and the mean is around 30 million shares.

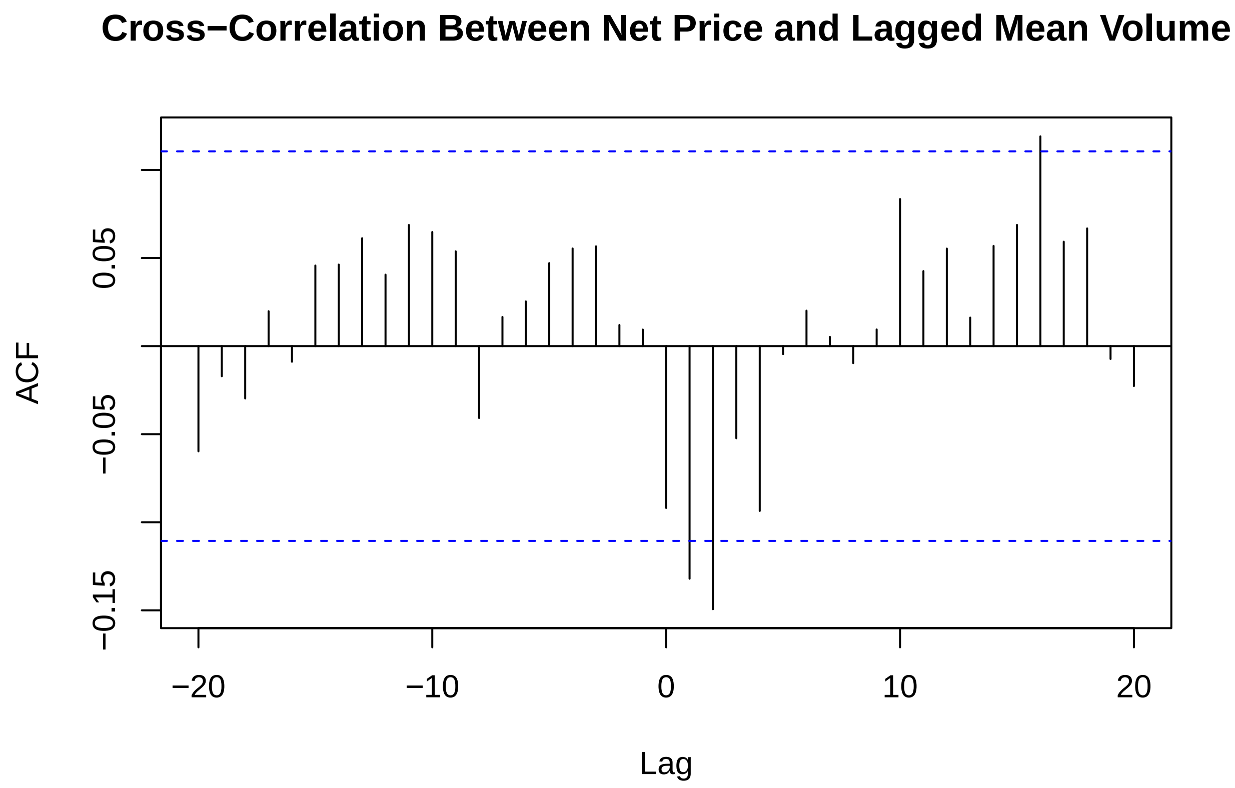
Determine the number of lags using Autocorrelation Function (ACF)

In order to determine the appropriate lag structure, I use cross-correlation function to plot correlation between:

1) Net price (y) and lag of mean volume (x)

```r
# Extract the columns as numeric vectors
net_price <- weekly_microsoft_data$net_price
mean_volume <- weekly_microsoft_data$daily_mean_volume

# Compute the cross-correlation
ccf_result <- ccf(mean_volume, net_price, lag.max = 20, plot = TRUE,
                  main = "Cross-Correlation Between Net Price and Lagged Mean Volume")
```

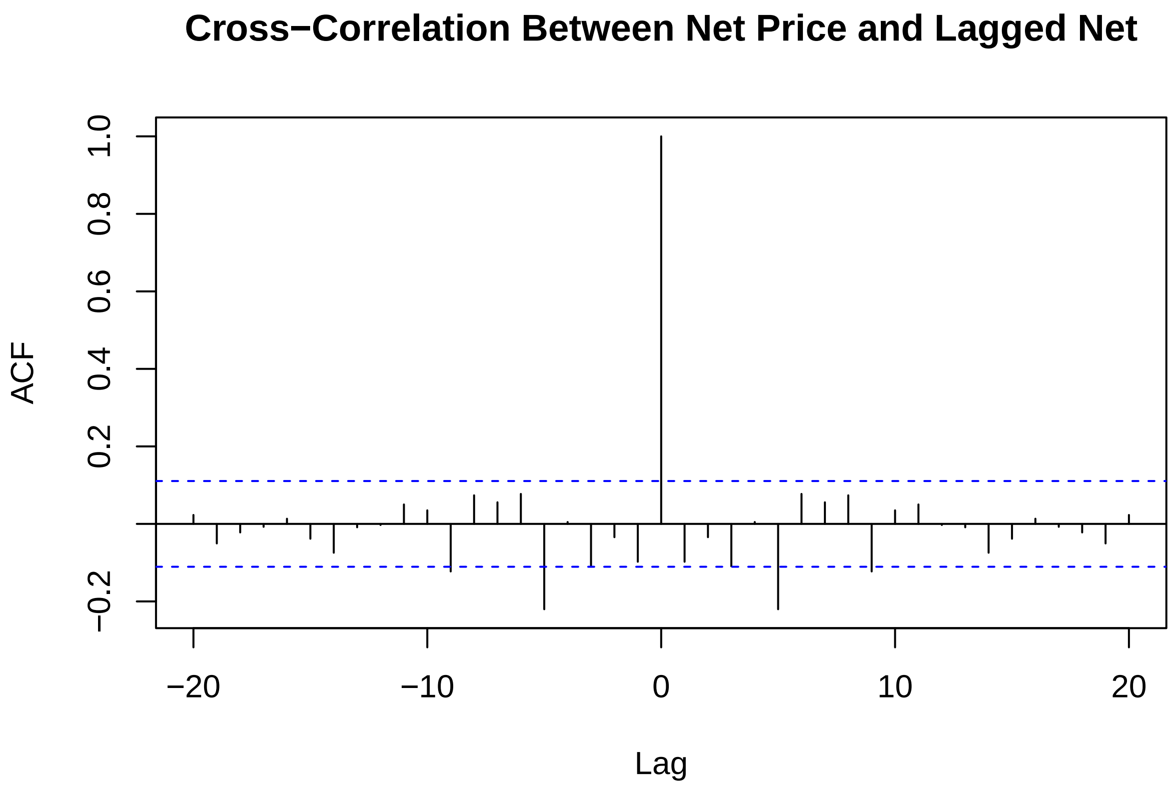


Comment:

- The cross correlation shows how weekly net prices are correlated with the lags of weekly mean trading volume.
- The plot suggests that current net prices (y_t) are significantly correlated with the mean trading volume of thee last 2 weeks (x_{t-1} and x_{t-2}).

2) Net price (y) and lag of itself in the past

```
ccf_result <- ccf(net_price, net_price, lag.max = 20, plot = TRUE,
                  main = "Cross-Correlation Between Net Price and Lagged Net")
```



Comment:

- The cross correlation shows how weekly net prices are correlated with the lags of itself.
- The plot suggests that current net prices (y_t) are significantly correlated with net prices from 3 and 5 weeks earlier (y_{t-3} and y_{t-5}).

Formal Test for Stationary

Since the variances appear non-station, we use Augmented Dickey-Fuller (ADF) test to formally test stationary.

- Null Hypothesis: data is not stationary
- Alternative Hypothesis: data is stationary

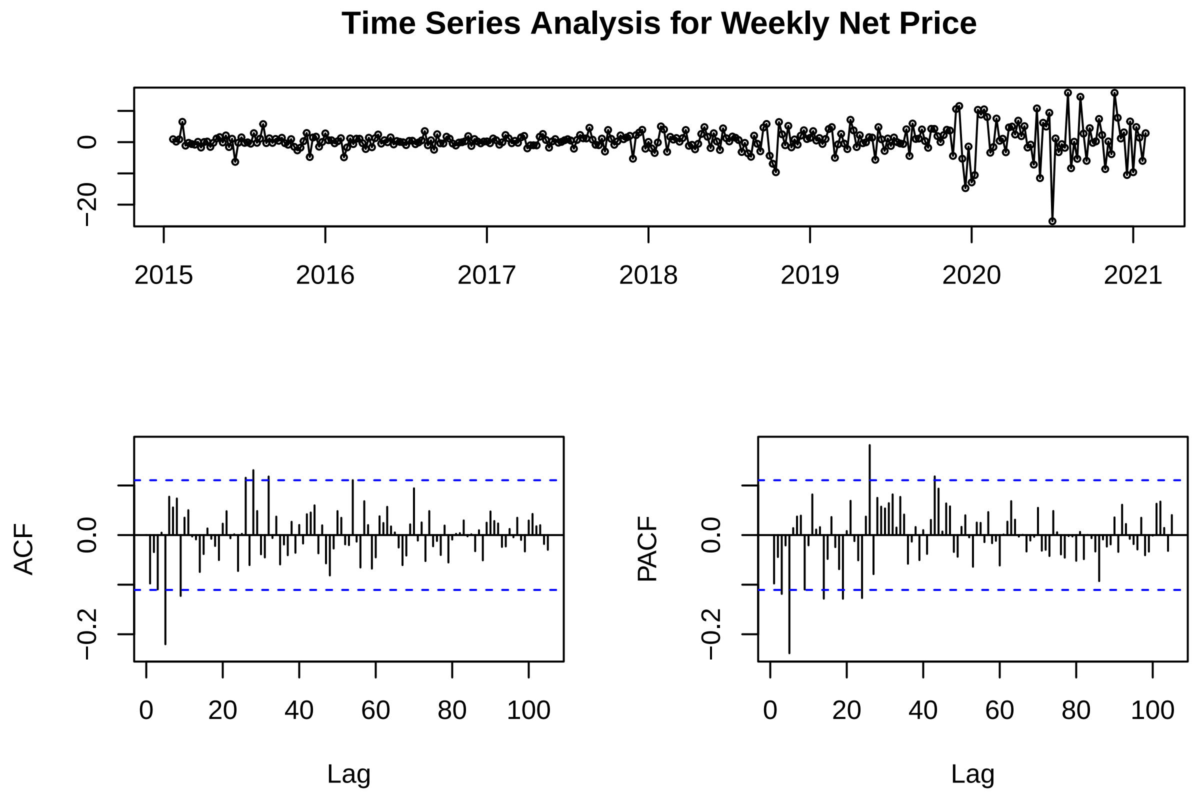
```
library(forecast)
```

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

```r
library(urca)
library(tidyverse)

# Change data to time-series data since it is required for the ADF function
ts_netprice <- ts(weekly_microsoft_data$net_price, start = c(2015,04,01), frequency = 52)
ts_weekly_microsoft_data <- ts(weekly_microsoft_data, start = c(2015,04,01), frequency = 52)

# Plot the time series, ACF, and PACF for the Weekly Net Price to assess stationary visually
tsdisplay(ts_netprice, main = "Time Series Analysis for Weekly Net Price")
```



```r
# Run the Augmented Dickey-Fuller (ADF) test on the time series data
adf_test <- ur.df(ts_netprice, type = "drift", selectlags = "AIC")

# Print the ADF test summary
print(summary(adf_test))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
```

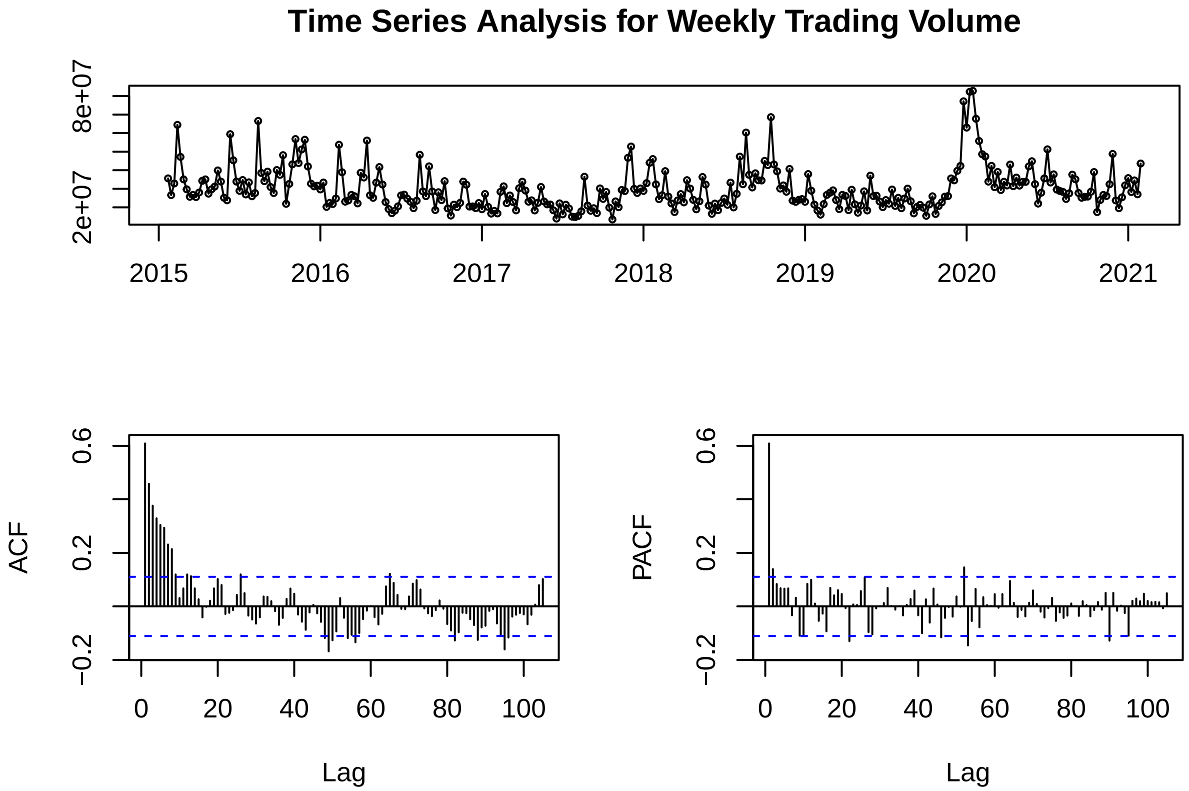
```
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.7355  -1.4054  -0.1276   1.5496  14.9822
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.60383    0.23599   2.559    0.011 *
## z.lag.1     -1.14633    0.08431 -13.597   <2e-16 ***
## z.diff.lag   0.04439    0.05708   0.778    0.437
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.093 on 309 degrees of freedom
## Multiple R-squared:  0.5495, Adjusted R-squared:  0.5466
## F-statistic: 188.4 on 2 and 309 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -13.597 92.4391
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
```

Comment:

- The value of test-statistics is -13.597, which is much lower than all the critical values at the 1%, 5%, and 10% significance levels. Therefore, we reject the null hypothesis that the time-series data is non-stationary.

```
ts_mean_volume <- ts(weekly_microsoft_data$daily_mean_volume, start = c(2015,04,01), frequency = 52)

# Plot the time series, ACF, and PACF for the Weekly Trading Volume to assess stationary visually
tsdisplay(ts_mean_volume, main = "Time Series Analysis for Weekly Trading Volume")
```



```r
# Run the Augmented Dickey-Fuller (ADF) test on the time series data
adf_test <- ur.df(ts_mean_volume, type = "drift", selectlags = "AIC")

# Print the ADF test summary
print(summary(adf_test))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -18760558  -5932452  -1407293   3854160  39246677
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.011e+07  1.580e+06   6.398 5.84e-10 ***
## z.lag.1     -3.348e-01  4.978e-02  -6.725 8.48e-11 ***
## z.diff.lag  -1.364e-01  5.656e-02  -2.412   0.0165 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8973000 on 309 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.2025
## F-statistic: 40.48 on 2 and 309 DF,  p-value: 2.423e-16
##
##
## Value of test-statistic is: -6.7254 22.62
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
```

Comment:

- The value of test-statistics is -6.7254, which is much lower than all the critical values at the 1%, 5%, and 10% significance levels. Therefore, we reject the null hypothesis that the time-series data is non-stationary.

Model Testing

AR Model

According to PACF of weekly net price, the cut off is at 25, which is a hint that this is the order of our AR model. Then, I generate 25 AR models (one for each degree) and use BIC to determine the best fitted model.

```r
library(dynlm)
#Create a matrix to store results
results <- matrix("", nrow = 25, ncol = 1)

#Looping over regressions and choosing the best
for (i in 1:25) {
  #create an AR model with degree i
  mdl <- dynlm(net_price ~ L(net_price, 1:i), data = ts_weekly_microsoft_data)
  N <- nobs(mdl)

  #Schwarz Criterion :
  SC_new <- log(sum(mdl$residuals^2)/N) + (3+i)*log(N)/N

  results[i,1] <- SC_new
}
numeric_results <- matrix(as.numeric(as.matrix(results)), nrow = nrow(results))
round(numeric_results, 3)
```

```
##        [,1]
##  [1,] 2.881
##  [2,] 2.901
##  [3,] 2.909
##  [4,] 2.923
##  [5,] 2.884
##  [6,] 2.906
##  [7,] 2.927
##  [8,] 2.948
##  [9,] 2.958
## [10,] 2.979
## [11,] 2.992
## [12,] 3.014
## [13,] 3.036
## [14,] 3.035
## [15,] 3.056
## [16,] 3.078
## [17,] 3.099
## [18,] 3.120
## [19,] 3.114
## [20,] 3.137
## [21,] 3.147
## [22,] 3.170
## [23,] 3.193
## [24,] 3.186
## [25,] 3.208
```

Comment:

- AR(1) is the best model that has the lowest SC (BIC), but we choose AR(5) to avoid the serial correlation while maintaining the similar SC to AR(1).

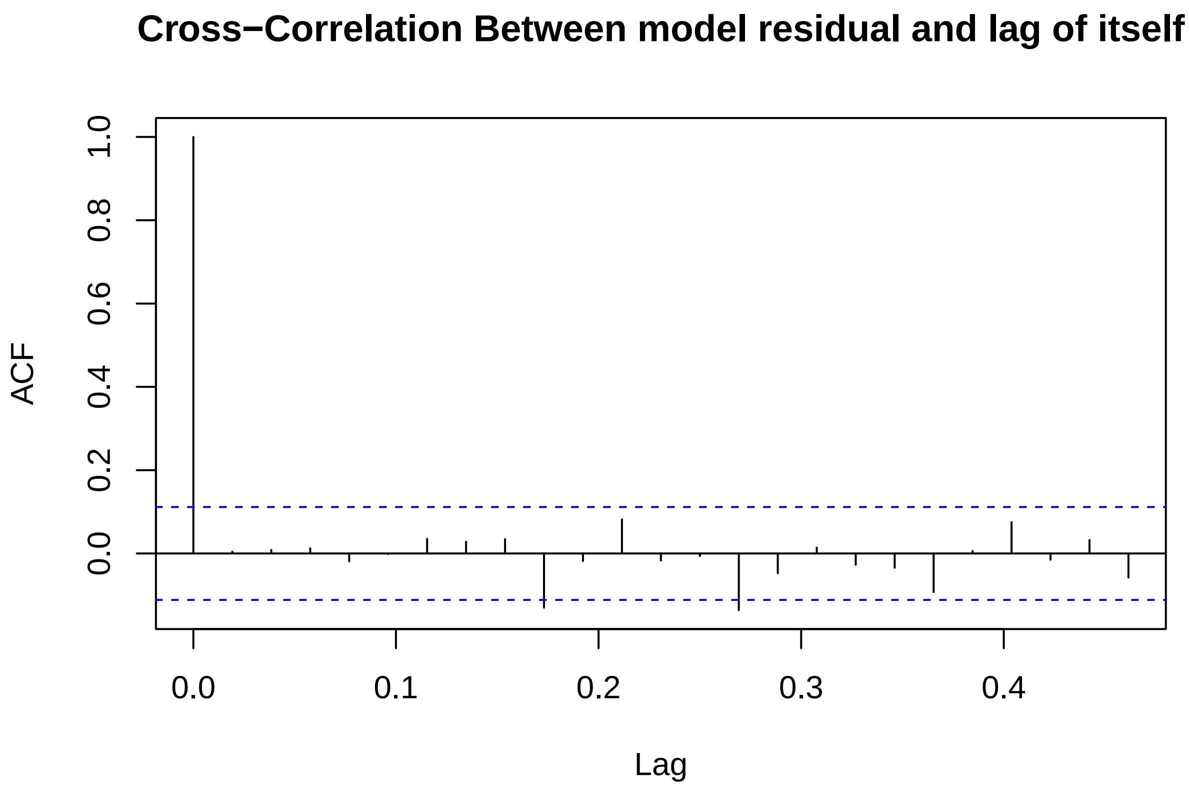
```
mdl_best <- dynlm(net_price ~ L(net_price, 1:5), data = ts_weekly_microsoft_data)
summary(mdl_best)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(9), End = 2021(5)
##
## Call:
## dynlm(formula = net_price ~ L(net_price, 1:5), data = ts_weekly_microsoft_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.6802  -1.5941  -0.1349   1.6170  15.2821
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.84921    0.23851   3.560  0.00043 ***
## L(net_price, 1:5)1 -0.11289    0.05575  -2.025  0.04375 *
## L(net_price, 1:5)2 -0.08979    0.05617  -1.599  0.11097
## L(net_price, 1:5)3 -0.13458    0.05574  -2.414  0.01635 *
## L(net_price, 1:5)4 -0.04470    0.05614  -0.796  0.42651
## L(net_price, 1:5)5 -0.24582    0.05649  -4.351 1.85e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.966 on 303 degrees of freedom
## Multiple R-squared:  0.08322,    Adjusted R-squared:  0.06809
## F-statistic: 5.501 on 5 and 303 DF,  p-value: 7.283e-05
```

Serial Correlation Test for AR Model

```
library(lmtest)
acf(mdl_best$residuals, main = "Cross-Correlation Between model residual and lag of itself")
```



Comment:

- The first 8 lags are not significant, but the 9th lag shows significant different than 0 at the 5% level. We are going to use Breush Godfrey Test to check the existence of the autocorrelation.

```
bgtest(mdl_best, order = 9)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 9
##
## data:  mdl_best
## LM test = 12.163, df = 9, p-value = 0.2043
```

Comment:

- p-value (0.2043) > 0.05 fails to reject the null that errors are uncorrelated. We conclude that there is no significant evidence of serial correlation (autocorrelation).

ARDL Model

According to the cross correlation between outcome net_price and lag of mean_volume above, ACF cuts off at 2, which is a hint that this is the highest suitable value of q for the ARDL model. Besides, based on the result of AR model above, the highest suitable value of p for our ARDL model is 5.

```
#Create a matrix to store results
results <- matrix(0, nrow = 5, ncol = 2)

#Looping over regressions and choosing the best
for (i in 1:5) {
for (j in 1:2) {
mdl <- dynlm(net_price ~ L(net_price, 1:i)+L(daily_mean_volume, 1:j), data = ts_weekly_microsoft_data)
N <- nobs(mdl)

#Schwarz Criterion :
SC_new <- log(sum(mdl$residuals^2)/N) + (3+i+j)*log(N)/N

results[i,j] <- SC_new
}
}
numeric_results <- matrix(as.numeric(as.matrix(results)), nrow = nrow(results))
round(numeric_results, 3)
```

```
##       [,1]  [,2]
## [1,] 2.900 2.921
## [2,] 2.919 2.938
## [3,] 2.926 2.945
## [4,] 2.941 2.959
## [5,] 2.899 2.918
```

Comment:

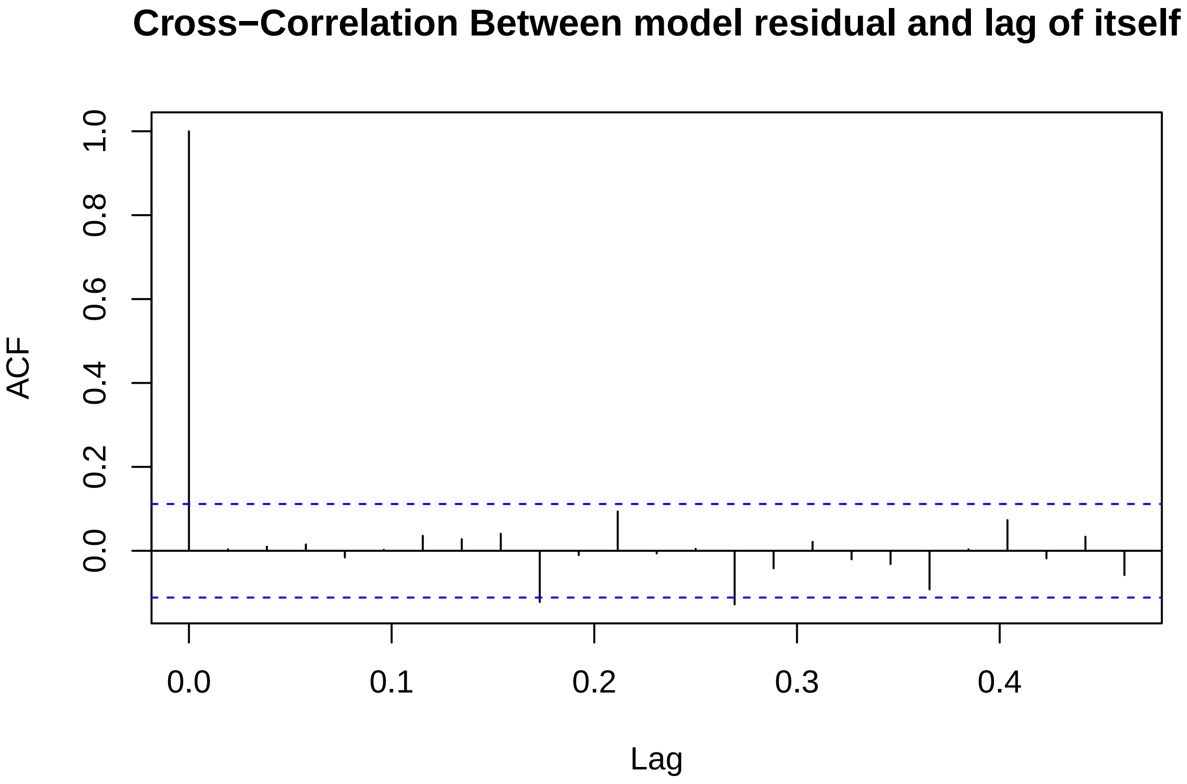
- ARDL(5,2) is the best model that has the lowest SC (BIC) value.

```
ARDL_best <- dynlm(net_price ~ L(net_price, 1:5) + L(daily_mean_volume, 1:2),
                   data = ts_weekly_microsoft_data)
summary(ARDL_best)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(9), End = 2021(5)
##
## Call:
## dynlm(formula = net_price ~ L(net_price, 1:5) + L(daily_mean_volume,
##     1:2), data = ts_weekly_microsoft_data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -21.3523  -1.6515  -0.1034   1.5376  15.1137
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.527e+00  7.571e-01   2.017   0.0446 *
## L(net_price, 1:5)1         -1.220e-01  5.662e-02  -2.154   0.0320 *
## L(net_price, 1:5)2         -1.007e-01  5.723e-02  -1.760   0.0794 .
## L(net_price, 1:5)3         -1.460e-01  5.697e-02  -2.563   0.0109 *
## L(net_price, 1:5)4         -5.104e-02  5.717e-02  -0.893   0.3727
## L(net_price, 1:5)5         -2.534e-01  5.705e-02  -4.441 1.26e-05 ***
## L(daily_mean_volume, 1:2)1 -2.295e-08  2.633e-08  -0.872   0.3840
## L(daily_mean_volume, 1:2)2  1.115e-09  2.567e-08   0.043   0.9654
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.972 on 301 degrees of freedom
## Multiple R-squared:  0.08663,    Adjusted R-squared:  0.06539
## F-statistic: 4.078 on 7 and 301 DF,  p-value: 0.000271
```

```r
acf(ARDL_best$residuals, main = "Cross-Correlation Between model residual and lag of itself")
```



Comment:

- The first 8 lags are not significant, but the 9th lag shows significant different than 0 at the 5% level. I am going to use Breush Godfrey Test to check the existence of the autocorrelation.

```r
bgtest(ARDL_best, order = 9)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 9
##
## data:  ARDL_best
## LM test = 12.018, df = 9, p-value = 0.2123
```

Comment: - p-value (0.2123) > 0.05 fails to reject the null that errors are uncorrelated. I conclude that there is no significant evidence of serial correlation.

AIC Test for AR(5) and ARDL(5,2) to select the best one

```r
# Fit the AR(5) model using dynlm
mdl_best <- dynlm(net_price ~ L(net_price, 1:5), data = ts_weekly_microsoft_data)

# Calculate the AIC value
ar_aic_value <- AIC(mdl_best)
print(ar_aic_value)
```

```
## [1] 1736.309
```

```r
# Fit the ARDL(5,2) model using dynlm
ARDL_best <- dynlm(net_price ~ L(net_price, 1:5) + L(daily_mean_volume, 1:2),
                   data = ts_weekly_microsoft_data)

# Calculate the AIC value
ardl_aic_value <- AIC(ARDL_best)
print(ardl_aic_value)
```

```
## [1] 1739.157
```

```r
summary(mdl_best)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(9), End = 2021(5)
##
## Call:
## dynlm(formula = net_price ~ L(net_price, 1:5), data = ts_weekly_microsoft_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.6802  -1.5941  -0.1349   1.6170  15.2821
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.84921    0.23851   3.560  0.00043 ***
## L(net_price, 1:5)1 -0.11289    0.05575  -2.025  0.04375 *
## L(net_price, 1:5)2 -0.08979    0.05617  -1.599  0.11097
## L(net_price, 1:5)3 -0.13458    0.05574  -2.414  0.01635 *
## L(net_price, 1:5)4 -0.04470    0.05614  -0.796  0.42651
## L(net_price, 1:5)5 -0.24582    0.05649  -4.351 1.85e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.966 on 303 degrees of freedom
## Multiple R-squared:  0.08322,    Adjusted R-squared:  0.06809
## F-statistic: 5.501 on 5 and 303 DF,  p-value: 7.283e-05
```

Conclusion

- AR(5) has lower AIC value (1736.309 < 1739.157), indicating the better fit model.
- Besides, looking that the summary of AR(5) model, the coefficients of lag 1, 3, and 5 are significant. This implies that the weekly net prices of 1, 3, and 5 weeks ago are negatively associated with the current weekly net price of Microsoft stock.
- Coming back to my question and initial hypothesis. Yes, the time-series dataset can help me explore the question "Can weekly mean volume and weekly past net price affect weekly current net price of Microsoft stock?". Regarding my initial hypothesis, ARDL(5,2) shows us past volume is not significantly associated with current net price. Specifically, the coefficient of lag 1 has insignificant negative effect (-2.295e-08) and lag 2 has insignificant positive effect (1.115e-09). However, the hypothesis about past net price is negatively associated with current net price is supported by the results from both AR(5) and ARDL(5,2) models (explained above).

Limitations and Improvements

- Stationarity Constraints (limitation): The data was transformed to weekly observations to address non-stationary concerns, which might have caused loss of information.
- Stationarity Constraints (improvement): To improve robustness, additional stationarity checks like further differencing could ensure the time series is optimal for this analysis.
- Reverse Causality Considerations (issue): The analysis assumes that trading volumes influence weekly price changes. However, there could be reverse causality where weekly price changes affect trading volumes.
- Reverse Causality Considerations (improvement): A Granger causality test could assess if lagged price changes significantly predict trading volume or vice versa.