R Notebook

Code

Name : Ayush Singoria
Roll number: 6A
Assignment 2 RTSM

Hide

```
# # Required Packages
packages = c('quantmod','car','forecast','tseries','FinTS', 'rugarch','utf8','ggplot2')
#
# # Install all Packages with Dependencies
#
#install.packages(packages, dependencies = TRUE)
#
# # Load all Packages
lapply(packages, require, character.only = TRUE)
```

```
[[1]]
[1] TRUE

[[2]]
[1] TRUE

[[3]]
[1] TRUE

[[4]]
[1] TRUE

[[5]]
[1] TRUE

[[6]]
[1] TRUE

[[7]]
[1] TRUE

[[8]]
[1] TRUE
```

Downloading the HUL Stock data

Hide

```
getSymbols(Symbols = 'HINDUNILVR.NS',
           src = 'yahoo',
           from = as.Date('2018-01-01'),
           to = as.Date('2023-12-31'),
           periodicity = 'daily')
```

```
[1] "HINDUNILVR.NS"
```

Hide

```
# xts (Time-Series) Object
```

Sanitization of the data

Hide

```
# Extract Adjusted Closing Price and remove missing values
stock_price = na.omit(HINDUNILVR.NS$HINDUNILVR.NS.Adjusted)
class(stock_price)
```

```
[1] "xts" "zoo"
```

Hide

```
View(stock_price)
```

Examine the structure of the stock price object.

Result:

- The output confirms stock is an xts object containing daily adjusted closing prices for "HINDUNILVR.NS" stock from January 1st, 2018, to December 29th, 2023 (1481 observations).

Analysis:

- The data is in a time series format with dates as the index.
- Knowing the data source (Yahoo) and update time (March 24th, 2024) provides context.

Hide

```
#Checking for the structure of the stock_price
str(stock_price)
```

```
An xts object on 2018-01-01 / 2023-12-29 containing:
  Data:    double [1481, 1]
  Columns: HINDUNILVR.NS.Adjusted
  Index:   Date [1481] (TZ: "UTC")
  xts Attributes:
    $ src    : chr "yahoo"
    $ updated: POSIXct[1:1], format: "2024-03-25 11:10:36"
```

Confirming if there are any null values in the series
output: No there is no Null value

Hide

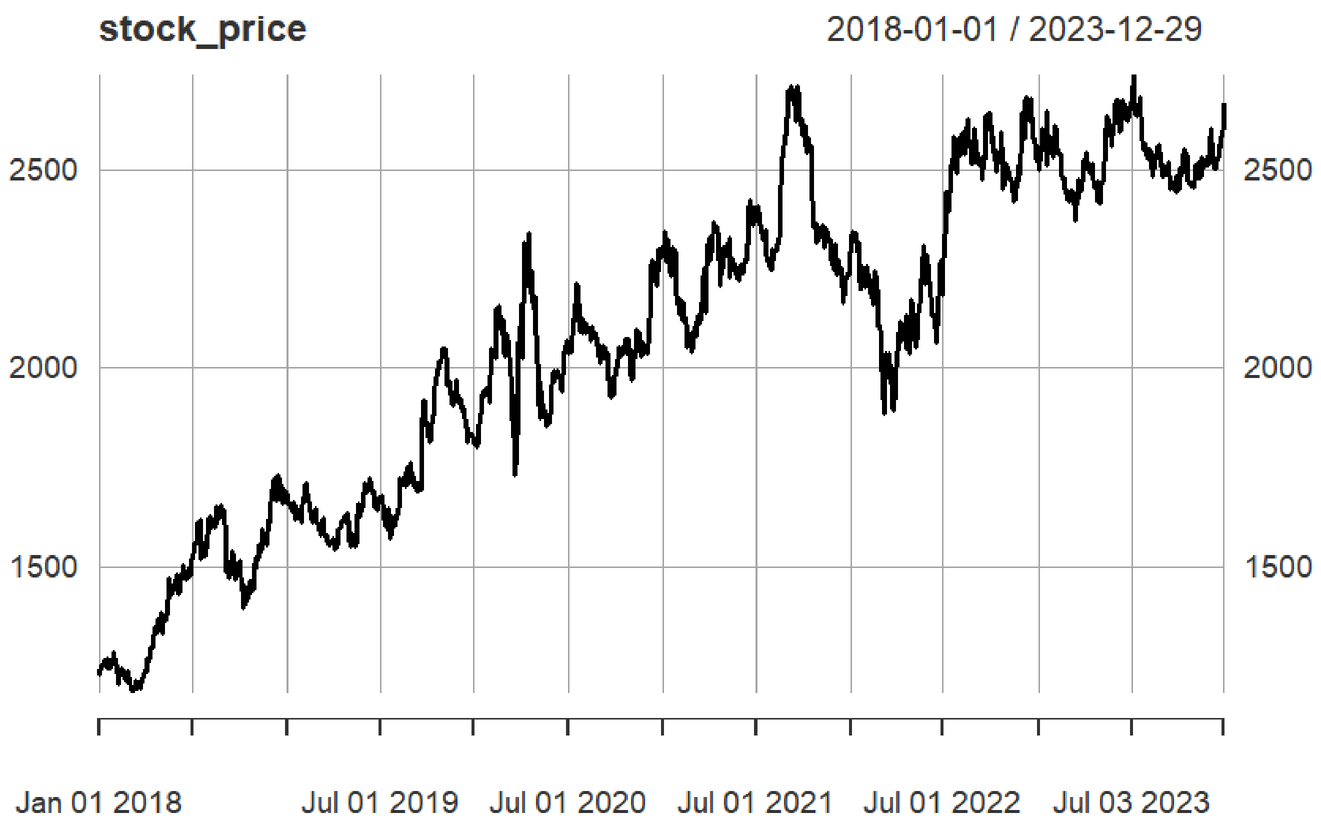
```
# Confirming if there are any null values in the series
any_null = any(is.null(stock_price))
any_null
```

```
[1] FALSE
```

Visualising the time series data

Hide

```
#Visualising the time series data
plot(stock_price)
```



Forecasting using Simple Moving Average (SMA)

Objective: To demonstrate the computation and visualization of the Simple Moving Average (SMA) and the Simple Moving Average Random Walk Forecast with Drift for a given stock price dataset.

Analysis:

- The SMA with a window size of 4 provides a smoothed representation of the original stock price data, allowing for easier identification of trends.
- The plot illustrates how the SMA line follows the general trend of the stock price while smoothing out short-term fluctuations.

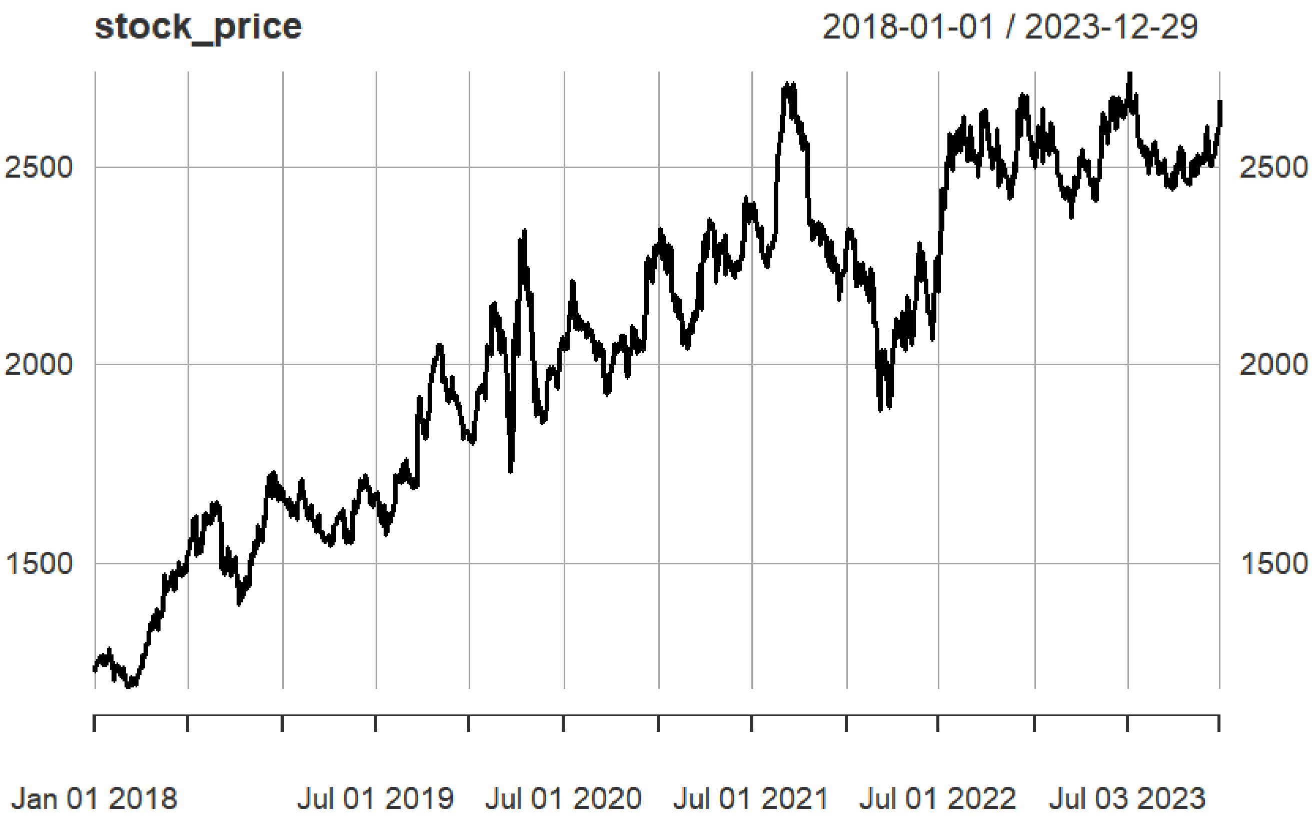
Hide

```
# Simple Moving Average [SMA]

stock_price_ma4 = ma(stock_price, order = 4)

plot(stock_price, lwd = 2)

lines(stock_price_ma4, col = 'blue', lwd = 20)
```



Hide

```
NA
NA
```

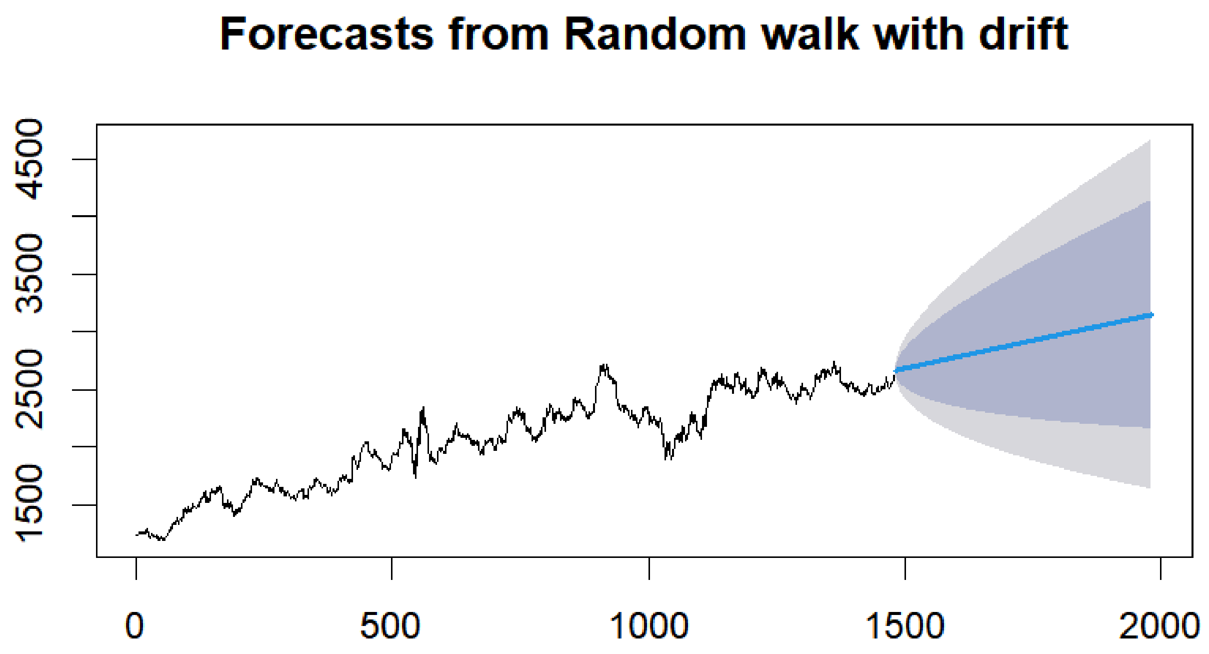
Hide

```
# Simple Moving Average : Random Walk (with Drift) Forecast
stock_price_ma8 = rwf(stock_price, h = 500, drift = TRUE)
accuracy(stock_price_ma8)
```

```
                       ME     RMSE      MAE          MPE     MAPE
Training set 8.849443e-14 29.95028 20.72031 -0.007367167 1.008978
                 MASE        ACF1
Training set 1.000849 -0.04875345
```

Hide

```
plot(stock_price_ma8)
```



Forecasting using Exponentially Weighted Moving Average (EWMA)

Objective: Forecast the future stock prices for the next 500 periods using Simple Exponential Smoothing (SES).

Results:

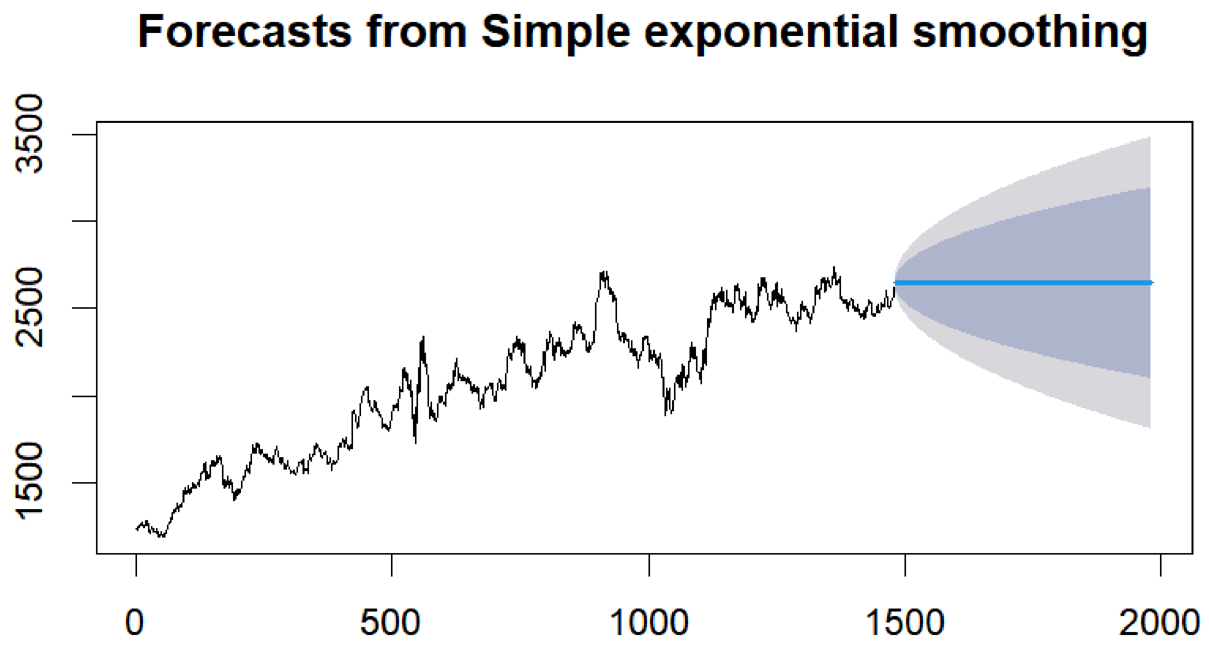
- The output table displays several accuracy measures:
  - ME (Mean Error): 1.176946 (indicates a slight bias towards underestimating future prices)
  - RMSE (Root Mean Squared Error): 21.98572 (measures the overall magnitude of the forecast errors)
  - MAE (Mean Absolute Error): 13.13919 (represents the average absolute difference between forecasts and actual prices)
  - MPE (Mean Percentage Error): 0.1053299 (expresses the average error as a percentage of the actual values)
  - MAPE (Mean Absolute Percentage Error): 1.129559 (similar to MPE but uses absolute values)
  - MASE (Mean Absolute Scaled Error): 1.081948 (scales the errors by the average absolute value of the series)
  - ACF1 (Autocorrelation Function at lag 1): 0.3454723 (measures the linear dependence between forecasts and actual prices one period apart)

````
```r
 #Forecasting using Exponentially Weighted Moving Average (EWMA)
stock_price_es = ses(stock_price, h = 500, alpha = 0.6)
accuracy(stock_price_es)
```
````

```
```
                    ME   RMSE   MAE   MPE   MAPE   MASE
Training set 1.591267 31.85333 22.43343 0.06943971 1.092961 1.083598
                   ACF1
Training set 0.3426256
```
```

Hide

```
plot(stock_price_es)
```



Forecasting with Time-Series Data (Univariate) : ARIMA | GARCH

Augmented Dickey-Fuller (ADF) test

Objective: to assess the stationarity of a time series data on HUL daily stocjk prise using the Augmented Dickey-Fuller (ADF) test.

Results:

The ADF test output indicates:

- Dickey-Fuller Statistic: -3.956
- Lag Order: 11
- p-value: 0.01119

Analysis

- A negative Dickey-Fuller statistic suggests rejection of the null hypothesis (H0) of non-stationarity.

- The p-value (0.01119) is less than the commonly used significance level of 0.05. This further strengthens the evidence against non-stationarity.
- Therefore, based on the ADF test results, we can conclude that the stock price data is likely stationary.

Hide

```
adf_test_stock_price = adf.test(stock_price);
adf_test_stock_price
```

```

	Augmented Dickey-Fuller Test

data:  stock_price
Dickey-Fuller = -3.956, Lag order = 11, p-value = 0.01119
alternative hypothesis: stationary
```

Hide

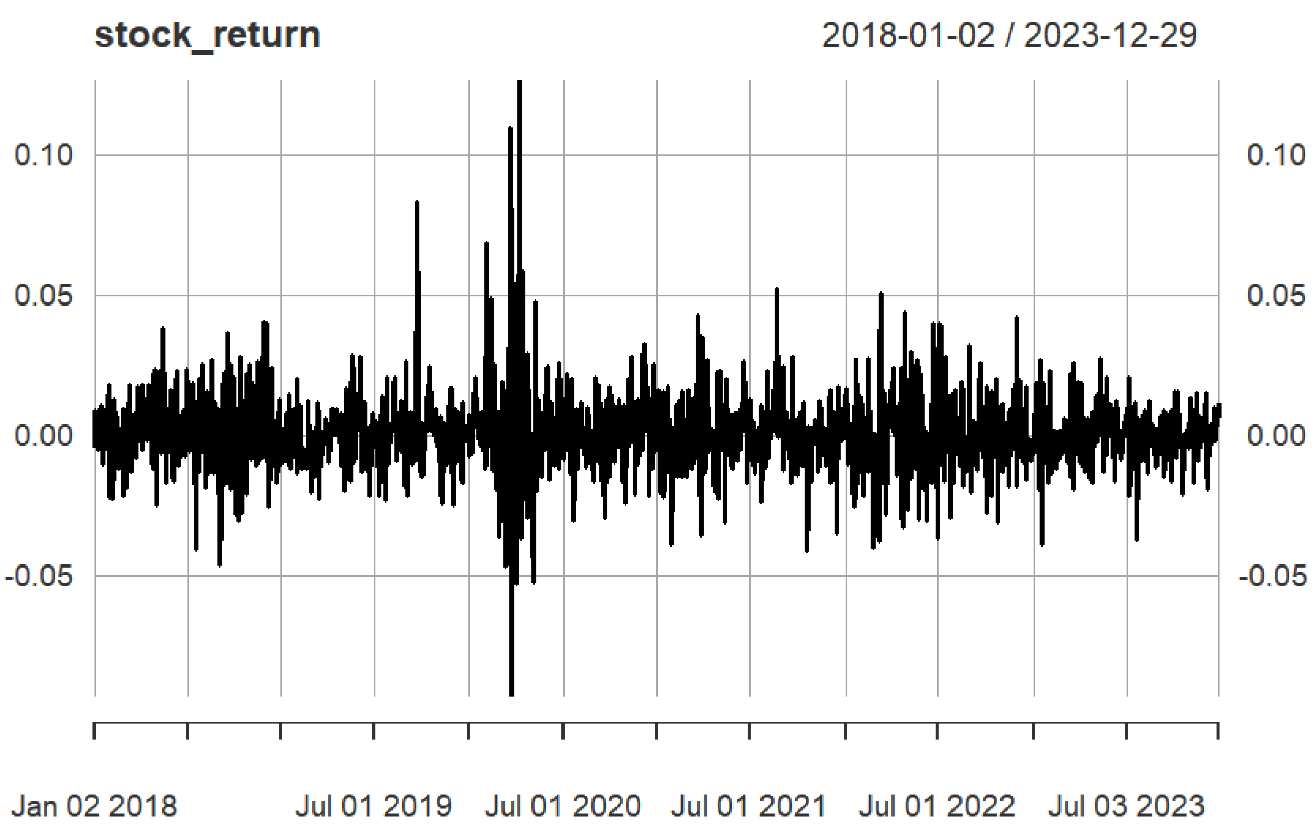
```
# H0 - Not Stationary (>0.05)
# H1 - Stationary (<0.05)
```

Stock return data

To get Stock return the stock price data is transformed by taking the log difference of stock price.

Hide

```
stock_return = na.omit(diff(log(stock_price)));
plot(stock_return)
```



Augmented Dickey-Fuller (ADF) test for stationarity on the daily returns

Result:
The Augmented Dickey-Fuller test for stationarity on HUL daily price and returns yields the following results:
- Dickey-Fuller statistic: -11.087
- Lag order: 11
- p-value: 0.01
- Alternative hypothesis: Stationary

Managerial Implication:
The ADF test suggests that both daily stock returns and daily price of HUL stock is stationary.
We can go ahead with out prediction by using any data, stock price or stock return. This report focus on prediction using Stock return data.

Hide

```
#ADF test for Stationery on stock return

adf_test1_stock_return = adf.test(stock_return); adf_test1_stock_return
```

```
Warning: p-value smaller than printed p-value
```

```


	Augmented Dickey-Fuller Test

data:  stock_return
Dickey-Fuller = -11.087, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```

Ljung-Box test for autocorrelation on the daily returns

Result:

X-squared statistic: 5.5273

Degrees of freedom:1

p-value: 0.01872

Implication:
The Ljung-Box test indicates significant autocorrelation in the HUL stock daily returns. The smaller p-value (smaller than 0.05) suggests evidence against the null hypothesis of no autocorrelation.

Action:
Given the presence of autocorrelation, it may be advisable to consider an autoARIMA model for time series forecasting. AutoARIMA can help in automatically selecting an appropriate ARIMA model with differencing to account for the observed autocorrelation.

Hide

```
#Ljung-Box test for autocorrelation on the daily returns of HUL stock.
lb_test_stock_return = Box.test(stock_return);
lb_test_stock_return
```

```
    Box-Pierce test

data:  stock_return
X-squared = 5.5274, df = 1, p-value = 0.01872
```

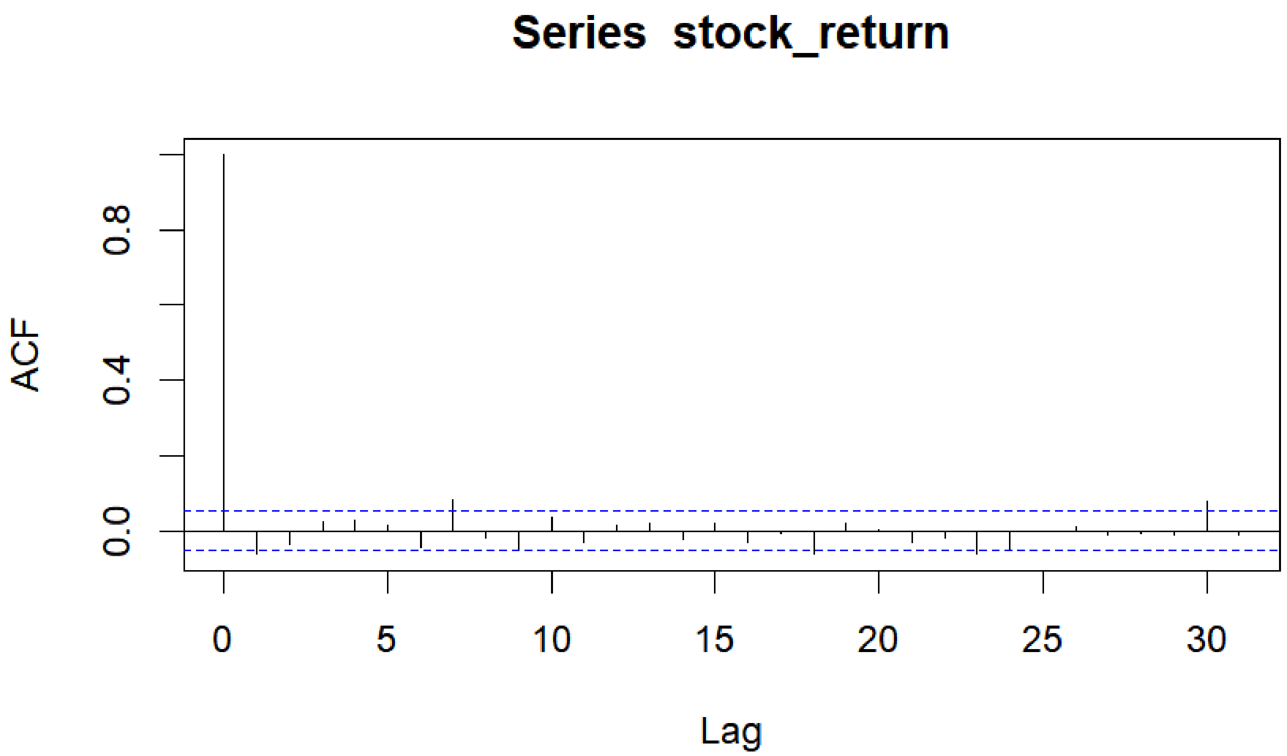
Hide

```
# H0 - No Auto-correlation (>0.05)
# H1 - Auto-correlation exists (<0.05)
```

ACF PACF TEST To get the Order of p and q in Arima

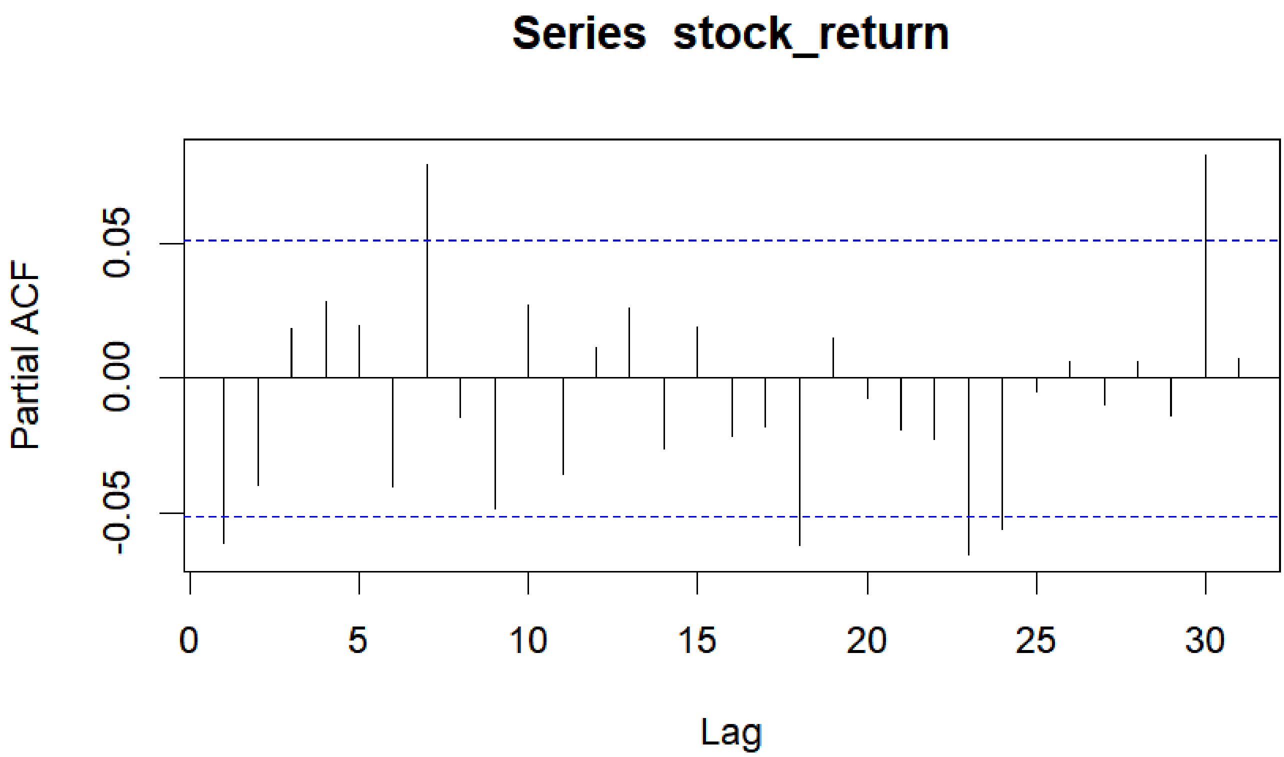
Hide

```
acf(stock_return)
```



Hide

```
pacf(stock_return)
```



Hide

```
# ACF of Stock returns (Stationary) Series
# PACF of Stock returns and price (Stationary) Series
# p(ACF) and q(PACF) values for ARIMA
```

AutoARIMA modeling on the daily returns

The `auto.arima` function to automatically identify and fit an ARIMA model for the `stock_return` data.

Result: The output indicates:

- Model: ARIMA(0,0,1) - This means there are no autoregressive (AR) terms, one moving average (MA) term of order 1, and a non-zero mean.
- Coefficients:
  - ma1: -0.0658 (coefficient of the MA term)
  - mean: 5e-04 (estimated mean of the series)
- Statistics:
  - sigma^2: 0.0002107 (variance of the error term)
  - Log likelihood: 4165.09 (measure of model fit)
- Information Criteria:
  - AIC: -8324.19
  - AICc: -8324.17
  - BIC: -8308.29 (lower values indicate better model fit)

Implication: The autoARIMA models provide a statistical framework to capture the underlying patterns in daily returns of HUL stock. These models can be used for forecasting future values, and the AIC, AICc, and BIC values help in model comparison.

Hide

```
#Auto Arima
arma_pq_stock_return = auto.arima(stock_return);
arma_pq_stock_return
```

```
Series: stock_return
ARIMA(0,0,1) with non-zero mean

Coefficients:
          ma1   mean
      -0.0658  5e-04
s.e.   0.0268  4e-04

sigma^2 = 0.0002107:  log likelihood = 4165.09
AIC=-8324.19   AICc=-8324.17   BIC=-8308.29
```

Determining Residuals of Auto ARIMA model

The provided residuals represent the differences between the observed stock returns and the values predicted by the Auto ARIMA model. These residuals are essential indicators of model accuracy. Positive residuals suggest underestimation of returns, while negative residuals indicate overestimation. Close-to-zero residuals imply a good fit between observed and predicted values, whereas larger residuals may signal areas where the model could be improved. Managers should closely monitor residuals to identify opportunities for refining the model and enhancing forecasting accuracy.

Hide

```
Residuals_returns = arma_pq_stock_return$residuals
head(Residuals_returns)
```

```
Time Series:
Start = 1
End = 6
Frequency = 1
[1] -0.004908563  0.007896463  0.001404388  0.002746156  0.008356903
[6] -0.005173256
```

Hide

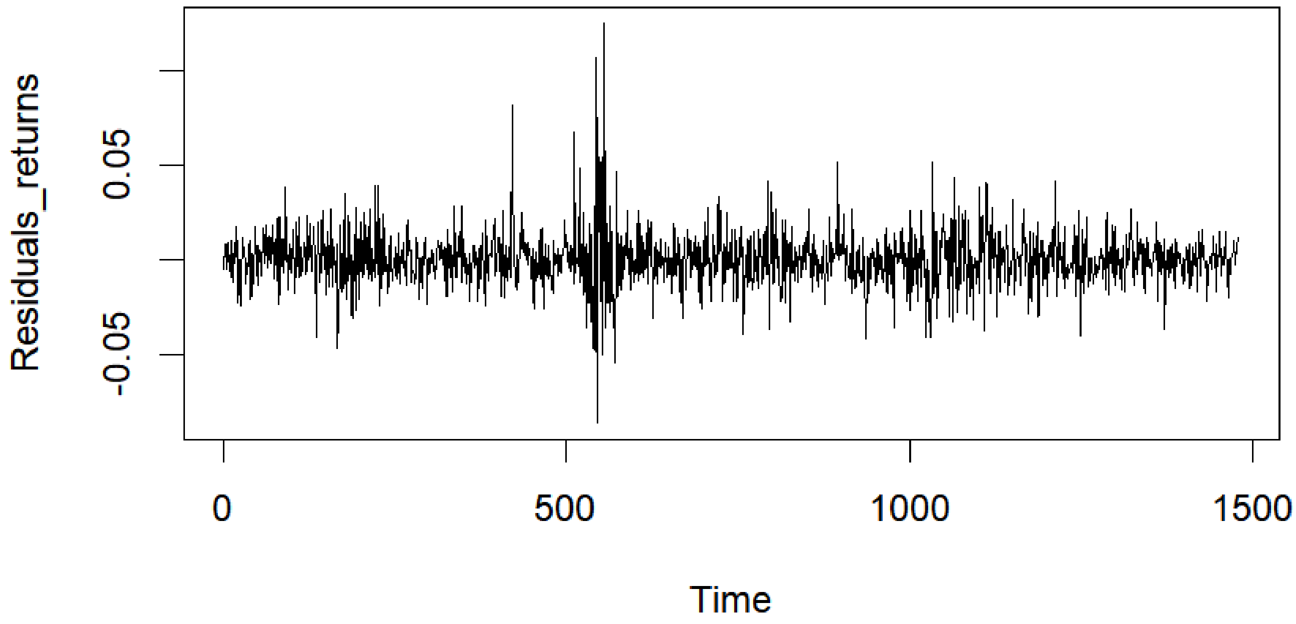
```
#length of residuals of stock return data
length(Residuals_returns)
```

```
[1] 1480
```

Visualization of residual of stock return

Hide

```
plot(Residuals_returns)
```



Ljung-Box test for autocorrelation on the residuals of the ARIMA model.

Results:
Ljung-Box Test for Autocorrelation on Residuals:
- X-squared statistic: 0.0066014
- Degrees of freedom: 1
- p-value: 0.9352

Implication:
The Ljung-Box test indicates significant autocorrelation in the residuals of the ARIMA(0, 0, 1) model. The High p-value (0.9352) suggests that there is no evidence against the null hypothesis of no autocorrelation.

Action: Since the Residuals have no auto-correlation the ARIMA model is appropriate and We do forecasting using ARIMA model

Hide

```
lb_test_arma_pq_stock_return = Box.test(Residuals_returns);
lb_test_arma_pq_stock_return
```

```


	Box-Pierce test

data:  Residuals_returns
X-squared = 0.0066015, df = 1, p-value = 0.9352
```

Hide

```
# H0 - No Auto-correlation { Favourable }
# H1 - Auto-correlationn Exists
#After this no autocorrelation exists
```

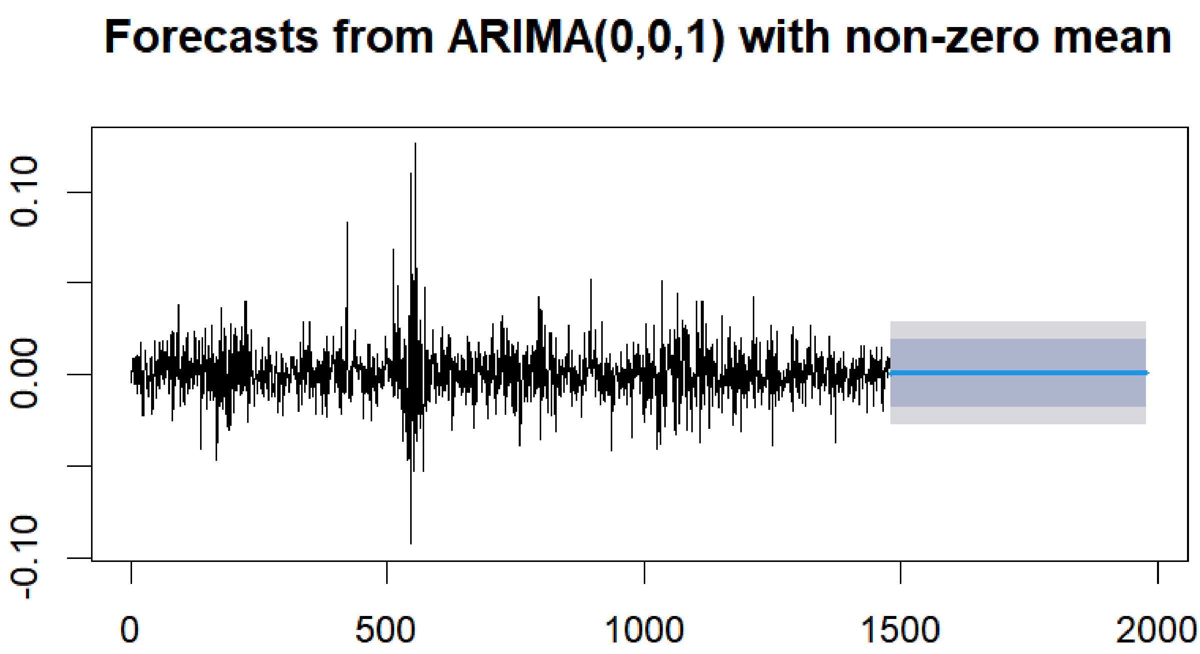
Forecasting using ARIMA model
Generated forecasts for the next 500 time points using the fitted ARIMA model.

Plot:
The plot displays the original time series of daily returns along with the forecasted values.

Implication: The ARIMA(0, 0, 1) model is fitted to the historical daily returns of HUL stock, providing insights into the underlying patterns. The generated forecast can be used for future predictions, and the plot visually represents the model's performance.

Hide

```
stock_return_fpq_20 = forecast(arma_pq_stock_return, h = 500)
plot(stock_return_fpq_20)
```



Test for Volatility Clustering and heteroscedasticity : Box Test and Arch test

Results:

1. Box Test for Volatility Clustering:
    - X-squared statistic: 113.93
    - Degrees of freedom: 1
    - p-value: < 2.2e-16 Inference: The Box test indicates significant evidence against the null hypothesis, suggesting that the return variance series exhibits volatility clustering.
2. ARCH Test for heteroscedasticity:
    - Chi-squared statistic: 240.15
    - Degrees of freedom: 10
    - p-value: <2.2e-16
      Inference: The ARCH test also provides strong evidence against the null hypothesis, supporting the presence of ARCH effects in the return series. This implies that the returns have heteroscedasticity.

Implication:
The results from both tests suggest that the residuals of the ARIMA(0, 0, 1) model exhibit volatility clustering or heteroskedasticity. This suggest we should go for GARCH Modeling for further analysis and forecasting.

Hide

```
# Test for Volatility Clustering: Box Test
lb_test_arma_pq_stock_return_square = Box.test(Residuals_returns^2);
lb_test_arma_pq_stock_return_square
```

```

	Box-Pierce test

data:  Residuals_returns^2
X-squared = 113.93, df = 1, p-value < 2.2e-16
```

Hide

```
# Test for heteroscedasticity: ARCH Test
Residual_square_arch_test_returns = ArchTest(Residuals_returns^2, lags = 10)
Residual_square_arch_test_returns
```

```

	ARCH LM-test; Null hypothesis: no ARCH effects

data:  Residuals_returns^2
Chi-squared = 240.15, df = 10, p-value < 2.2e-16
```

GARCH Modelling

Result

- **Log-likelihood:** 9357.344
- **Information Criteria:** Akaike (-12.640), Bayes (-12.625), Shibata (-12.640), Hannan-Quinn (1-12.634) - Lower values indicate better fit.
- **Tests:**
  - Ljung-Box tests on standardized residuals suggest rejection of no serial correlation (meaning there might be serial correlation in the residuals).
  - Ljung-Box tests on standardized squared residuals don’t reject no serial correlation.
  - ARCH LM tests fail to reject ARCH effects

Analysis:
The GARCH(1,1) model seems to capture some aspects of the volatility dynamics in the squared residuals.

Hide

```
#Garch model return
garch_model1_return = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1,
1)), mean.model = list(armaOrder = c(0,0), include.mean = TRUE))

Residuals_square_garch1_return = ugarchfit(garch_model1_return, data = Residuals_returns^2);
Residuals_square_garch1_return
```

```
*---------------------------------*
*          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      0.000140    0.000012  11.298286  0.00000
omega   0.000000    0.000000   0.025438  0.97971
alpha1  0.057576    0.007732   7.446696  0.00000
beta1   0.918199    0.004711 194.912387  0.00000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000140    0.025904 0.005391  0.99570
omega   0.000000    0.002026 0.000003  1.00000
alpha1  0.057576   24.746484 0.002327  0.99814
beta1   0.918199    5.314435 0.172775  0.86283

LogLikelihood : 9357.344

Information Criteria
------------------------------------

Akaike       -12.640
Bayes        -12.625
Shibata      -12.640
Hannan-Quinn -12.634

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic   p-value
Lag[1]                      11.09 0.0008683
Lag[2*(p+q)+(p+q)-1][2]     12.52 0.0004121
Lag[4*(p+q)+(p+q)-1][5]     16.49 0.0001761
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                   0.001247  0.9718
Lag[2*(p+q)+(p+q)-1][5]  0.026112  0.9999
Lag[4*(p+q)+(p+q)-1][9]  0.046012  1.0000
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
```

```
            Statistic Shape Scale P-Value
ARCH Lag[3]   0.02532 0.500 2.000  0.8736
ARCH Lag[5]   0.03752 1.440 1.667  0.9967
ARCH Lag[7]   0.04032 2.315 1.543  0.9999

Nyblom stability test
------------------------------------
Joint Statistic:  130.7261
Individual Statistics:
mu      1.2584
omega  60.8801
alpha1  0.2378
beta1   0.2714

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 <dbl>	prob <dbl>	sig <chr>
Sign Bias	0.9128415	0.3614751	
Negative Sign Bias	1.3226642	0.1861521	
Positive Sign Bias	0.4157736	0.6776362	
Joint Effect	2.1326659	0.5453326	

4 rows

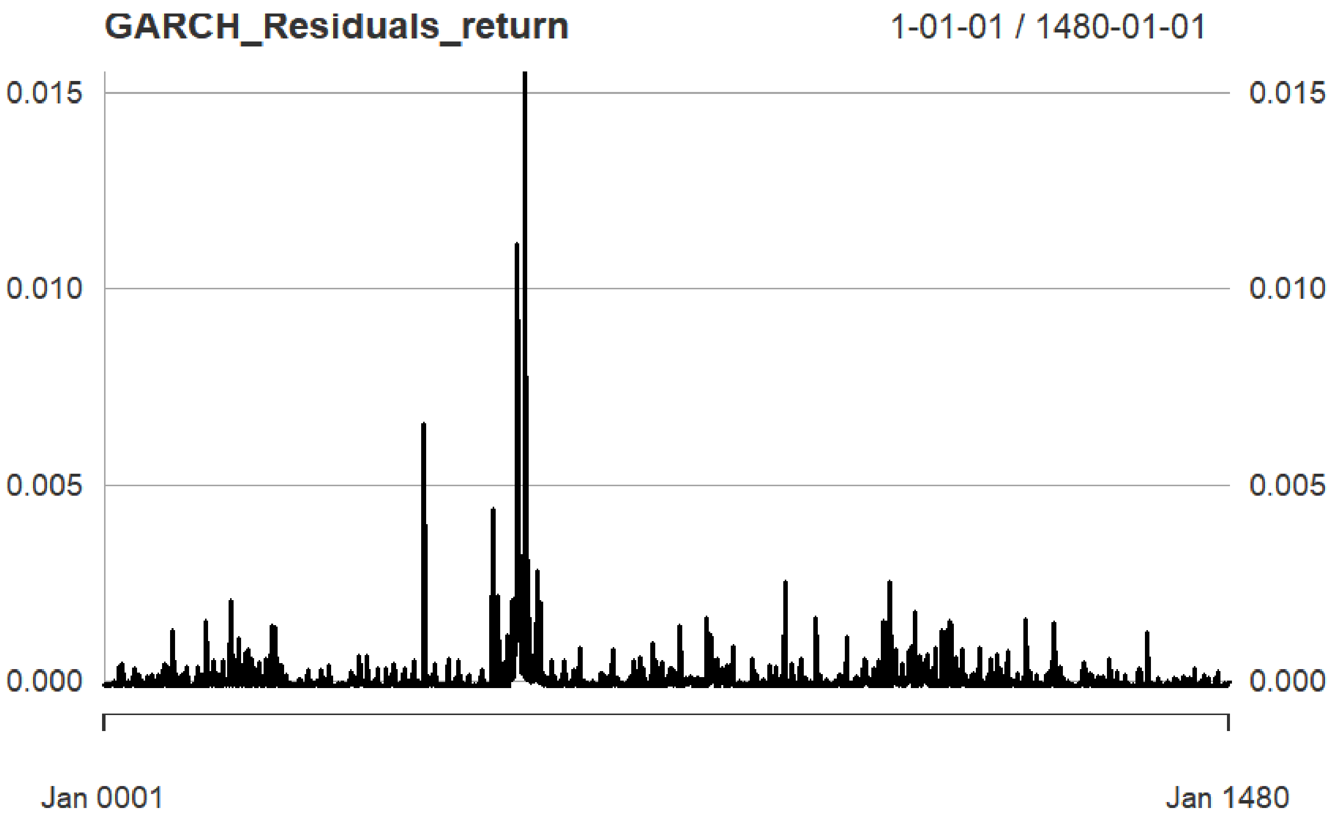
```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20      2848            0
2    30      2896            0
3    40      2930            0
4    50      2986            0


Elapsed time : 0.267956
```

GARCH Residual

Hide

```
#GARCH Residual
GARCH_Residuals_return = residuals(Residuals_square_garch1_return)
plot(GARCH_Residuals_return)
```



Tests for the presence of ARCH effects (Autoregressive Conditional Heteroscedasticity) in the squared residuals

The output shows the results of an ARCH LM-test:

- Chi-squared statistic: 288.64
- Degrees of freedom (df): 20
- p-value: less than 2.2e-16 (essentially 0)

  Analysis: A very high Chi-squared statistic and a p-value close to zero provide strong evidence against the null hypothesis of no ARCH effects. This suggests that the squared residuals exhibit heteroscedasticity, meaning the variance of the residuals is not constant over time.

Note: Since the test is performed on squared residuals of a GARCH model, this doesn't necessarily imply heteroscedasticity in the original data. However, it highlights potential issues with the GARCH model's ability to fully capture the volatility dynamics.

Managerial Implication

- The presence of ARCH effects indicates that the GARCH model might not be the most suitable choice for capturing the volatility dynamics in the data.
- It might be necessary to explore alternative GARCH models with higher order terms or explore other volatility modeling techniques.

Hide

```
#test of heteroscedasticity
GARCH_Residuals_square_arch_test_return = ArchTest(GARCH_Residuals_return, lags = 20)
GARCH_Residuals_square_arch_test_return
```

```
    ARCH LM-test; Null hypothesis: no ARCH effects

data:  GARCH_Residuals_return
Chi-squared = 288.64, df = 20, p-value < 2.2e-16
```

If there would have been no ARCH effect the we would have further fitted the GARCH model in original Stock return data using the following code

Hide

```
garch_model_stock_return = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c
(1,1)), mean.model = list(armaOrder = c(0,1), include.mean = TRUE))

stock_return_Garch_model_1 = ugarchfit(garch_model_stock_return, data = stock_return);
stock_return_Garch_model_1
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(0,0,1)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000507    0.000320  1.58193  0.11367
ma1    -0.010431    0.029080 -0.35872  0.71981
omega   0.000013    0.000000 46.75605  0.00000
alpha1  0.081942    0.005768 14.20538  0.00000
beta1   0.850111    0.009967 85.29395  0.00000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000507    0.000345  1.46940  0.14172
ma1    -0.010431    0.031404 -0.33217  0.73976
omega   0.000013    0.000001 23.46885  0.00000
alpha1  0.081942    0.008981  9.12367  0.00000
beta1   0.850111    0.017071 49.79933  0.00000

LogLikelihood : 4314.742

Information Criteria
------------------------------------

Akaike       -5.8240
Bayes        -5.8061
Shibata      -5.8240
Hannan-Quinn -5.8173

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.04333  0.8351
Lag[2*(p+q)+(p+q)-1][2]   0.37107  0.9874
Lag[4*(p+q)+(p+q)-1][5]   0.82909  0.9691
d.o.f=1
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.4854  0.4860
Lag[2*(p+q)+(p+q)-1][5]    0.8542  0.8917
Lag[4*(p+q)+(p+q)-1][9]    1.8636  0.9206
d.o.f=2
```

```
Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]    0.1915 0.500 2.000  0.6617
ARCH Lag[5]    0.6690 1.440 1.667  0.8329
ARCH Lag[7]    1.5299 2.315 1.543  0.8153

Nyblom stability test
------------------------------------
Joint Statistic:  50.186
Individual Statistics:
mu     0.21120
ma1    0.05159
omega  5.78501
alpha1 0.21589
beta1  0.20806

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 <dbl>	prob <dbl>	sig <chr>
Sign Bias	0.39309346	0.69430726	
Negative Sign Bias	1.84914679	0.06463659	*
Positive Sign Bias	0.09166085	0.92697995	
Joint Effect	6.87122602	0.07611749	*

4 rows

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     68.65    1.541e-07
2    30     78.99    1.629e-06
3    40    110.05    1.077e-08
4    50     97.70    4.400e-05


Elapsed time : 0.8885021
```

- **Forecasting: And then we would have foretasted the data using the following code**
    - The forecast horizon is 50 periods.
    - For each forecasted period, the predicted value of the series (stock return) and the corresponding standard deviation (sigma) are provided.

Hide

```
# GARCH Forecast
nse_ret_garch_forecast = ugarchforecast(stock_return_Garch_model_1, n.ahead = 50); nse_ret_ga
rch_forecast
```

```
*---------------------------------*
*       GARCH Model Forecast      *
*---------------------------------*
Model: sGARCH
Horizon: 50
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=2023-12-29]:
        Series   Sigma
T+1  0.0003924 0.01085
T+2  0.0005068 0.01106
T+3  0.0005068 0.01125
T+4  0.0005068 0.01143
T+5  0.0005068 0.01159
T+6  0.0005068 0.01174
T+7  0.0005068 0.01188
T+8  0.0005068 0.01200
T+9  0.0005068 0.01212
T+10 0.0005068 0.01223
T+11 0.0005068 0.01233
T+12 0.0005068 0.01242
T+13 0.0005068 0.01250
T+14 0.0005068 0.01258
T+15 0.0005068 0.01266
T+16 0.0005068 0.01272
T+17 0.0005068 0.01279
T+18 0.0005068 0.01284
T+19 0.0005068 0.01290
T+20 0.0005068 0.01295
T+21 0.0005068 0.01300
T+22 0.0005068 0.01304
T+23 0.0005068 0.01308
T+24 0.0005068 0.01312
T+25 0.0005068 0.01315
T+26 0.0005068 0.01318
T+27 0.0005068 0.01321
T+28 0.0005068 0.01324
T+29 0.0005068 0.01327
T+30 0.0005068 0.01329
T+31 0.0005068 0.01331
T+32 0.0005068 0.01333
T+33 0.0005068 0.01335
T+34 0.0005068 0.01337
T+35 0.0005068 0.01339
T+36 0.0005068 0.01340
T+37 0.0005068 0.01342
T+38 0.0005068 0.01343
T+39 0.0005068 0.01345
T+40 0.0005068 0.01346
T+41 0.0005068 0.01347
T+42 0.0005068 0.01348
T+43 0.0005068 0.01349
T+44 0.0005068 0.01350
```

```
T+45 0.0005068 0.01351
T+46 0.0005068 0.01351
T+47 0.0005068 0.01352
T+48 0.0005068 0.01353
T+49 0.0005068 0.01353
T+50 0.0005068 0.01354
```

Hide

```
plot(nse_ret_garch_forecast)
```

```
Make a plot selection (or 0 to exit):

1:   Time Series Prediction (unconditional)
2:   Time Series Prediction (rolling)
3:   Sigma Prediction (unconditional)
4:   Sigma Prediction (rolling)
```

Hide

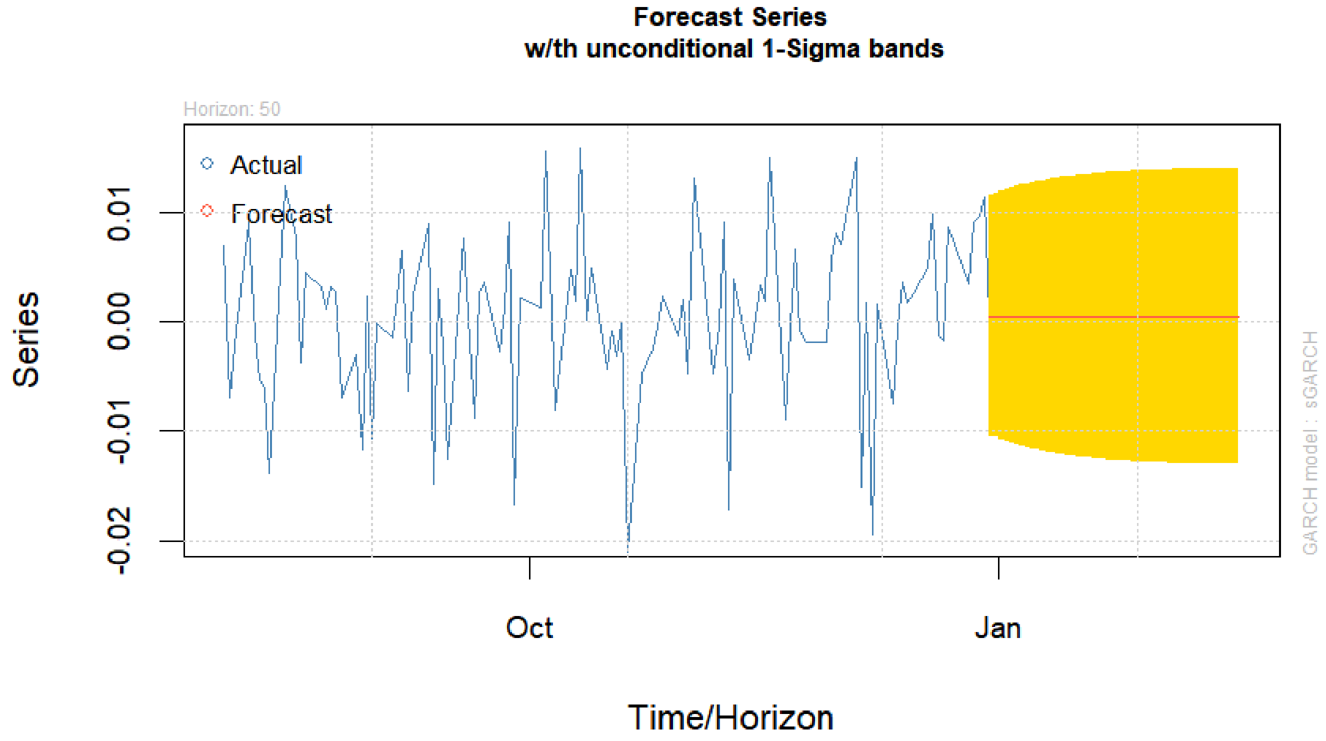
```
1
```

```
Make a plot selection (or 0 to exit):

1:   Time Series Prediction (unconditional)
2:   Time Series Prediction (rolling)
3:   Sigma Prediction (unconditional)
4:   Sigma Prediction (rolling)
```

Hide

```
3
```

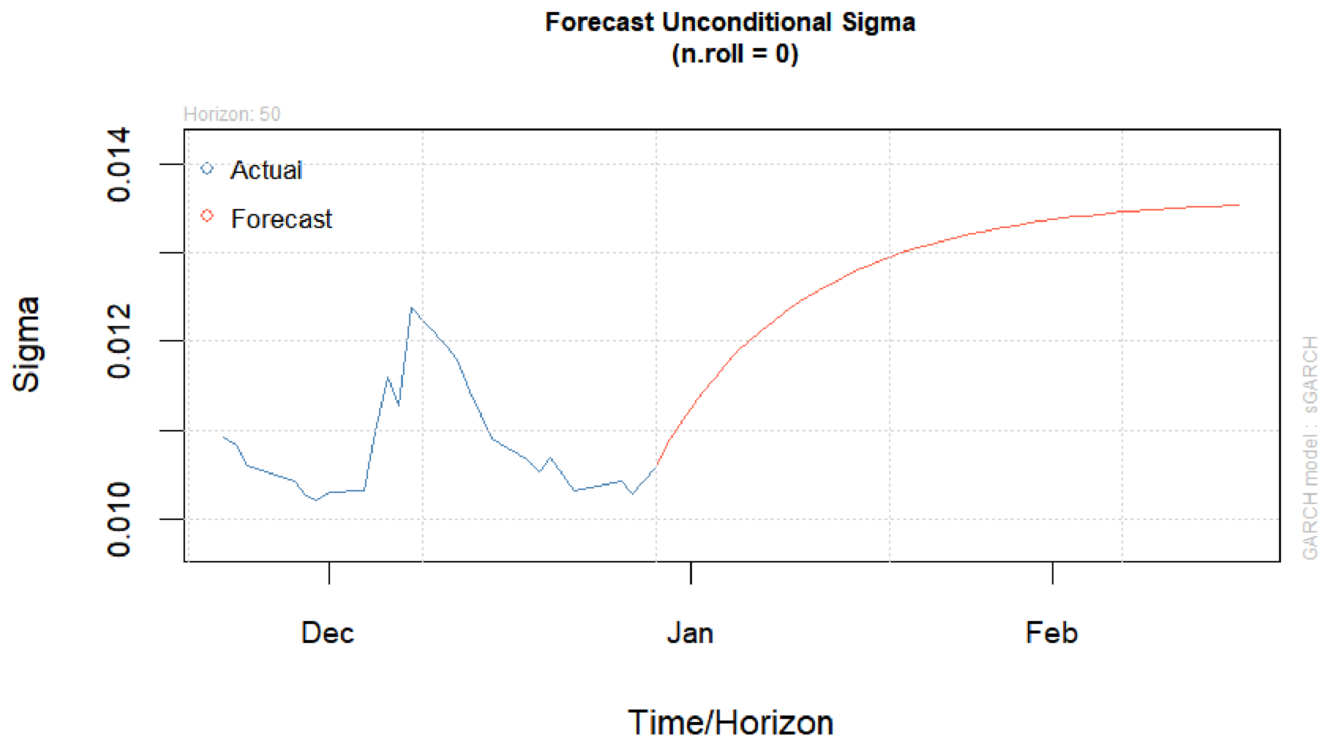



```
Make a plot selection (or 0 to exit):

1:   Time Series Prediction (unconditional)
2:   Time Series Prediction (rolling)
3:   Sigma Prediction (unconditional)
4:   Sigma Prediction (rolling)
```

Hide

```
0
```


Forecast Unconditional Sigma
(n.roll = 0)
Horizon: 50
Actual
Forecast
Sigma
0.010
0.012
0.014
Dec
Jan
Feb
Time/Horizon
GARCH model : sGARCH