Project Report

Time Series Analysis Of ITC stock

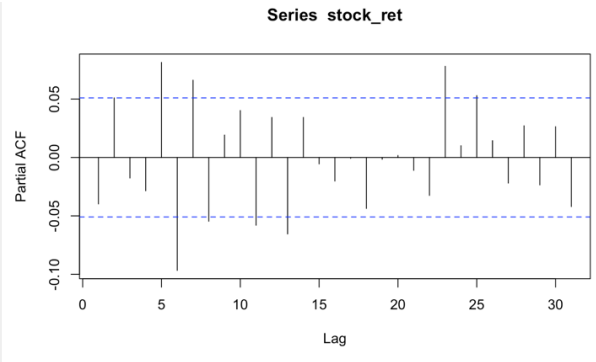
Project – 2

Submitted by:
Sujeet Kumar
MBA (BA) – 39A

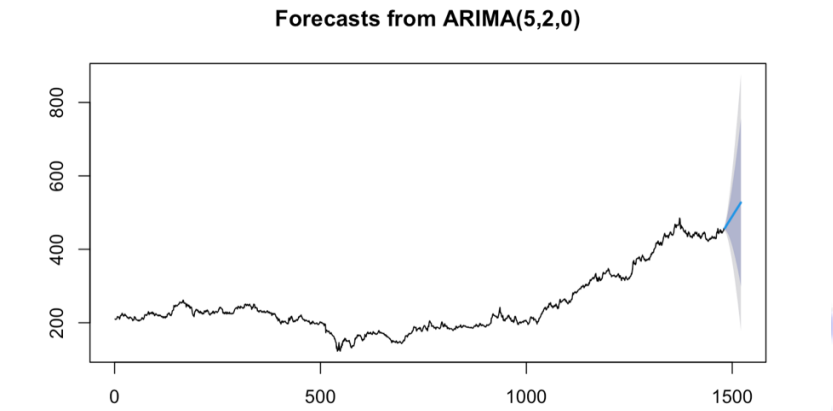
The presented R code delves deeply into the time series characteristics of **ITC** stock prices, offering a multifaceted approach to understanding its behavior over the period from **January 1, 2018, to December 31, 2023**.

1. Data Retrieval: Utilizing the quantmod package, the code extracts historical ITC stock prices from Yahoo Finance. By isolating the closing prices, a time series object is formed, setting the stage for a rigorous analysis.

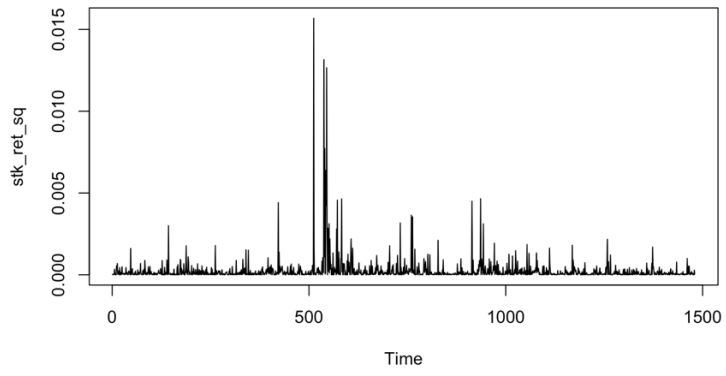
2. Stationarity Testing: To assess the stationarity of the stock prices and returns, the Augmented Dickey-Fuller (ADF) test is employed. The findings reveal that the raw stock prices exhibit a non-stationary behavior, necessitating further differencing. Conversely, the log returns demonstrate stationarity, indicating that this transformation is crucial for a more stable time series analysis.



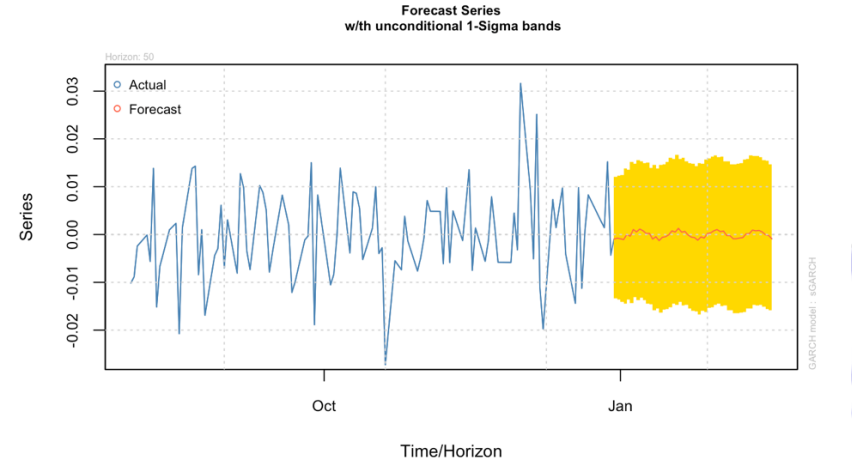
3. Autocorrelation Analysis: Auto ARIMA modeling and Ljung-Box tests are utilized to scrutinize autocorrelation patterns within the time series. The code generates autocorrelation and partial autocorrelation functions for both stock prices and returns, unraveling potential serial dependencies.



4. ARIMA Modeling: The Auto ARIMA algorithm is employed to identify suitable models for stock returns and prices individually. The ARIMA(3,0,2) model with a non-zero mean is selected for returns, capturing the underlying temporal patterns. Simultaneously, the ARIMA(5,2,0) model is deemed appropriate for modeling stock prices, considering the intricacies of the time series.



5. GARCH Modeling: Two GARCH models are fitted to the residuals obtained from the ARIMA models. These GARCH models incorporate mean components based on ARFIMA specifications, allowing for a more nuanced understanding of conditional variance dynamics. The results of this step reveal the presence of volatility clustering and heteroskedasticity in the stock returns.



6. Forecasting: In the final phase, the code generates forecasts for the next 50 periods using the selected GARCH model. The forecast encompasses predicted returns and associated volatility estimates, offering a glimpse into potential future stock behavior and associated uncertainties.

7. Volatility Clustering Tests: To validate the GARCH model's efficacy, the code conducts Box tests and ARCH tests, scrutinizing the presence of volatility clustering or heteroskedasticity in the squared residuals series.

In conclusion, this comprehensive time series analysis equips investors and analysts with a robust framework for deciphering the temporal patterns, stationarity nuances, and volatility dynamics of ITC stock. The forecasting component augments this understanding, facilitating more informed investment decisions in the dynamically evolving stock market landscape.

R code

```
---
title: "Sujeet ITC stock Time Series Analysis"
output: html_notebook
---

This is our second assignment for RSTM subject.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```{r}
# I am running the whole test in ITC stock
# # Required Packages
packages = c('quantmod','car','forecast','tseries','FinTS',  'rugarch','utf8','ggplot2')
# # Install all Packages with Dependencies on first run
# install.packages(packages, dependencies = TRUE)
# # Load all Packages
lapply(packages, require, character.only = TRUE)
# 0.1. Fetch Single Stock/Index Data
getSymbols(Symbols = 'ITC.NS',
        src = 'yahoo',
        from = as.Date('2018-01-01'),
        to = as.Date('2023-12-31'),
        periodicity = 'daily')

stock_price = na.omit(ITC.NS$ITC.NS.Adjusted)
class(stock_price) # Time-Series Object

stock_ret = na.omit(diff(log(stock_price))) # Stock Returns
plot(stock_price)
plot(stock_ret)#this is stock return of ITC
```
```{r}
# Augmented Dickey-Fuller (ADF) Test for Stationarity with Stock Data

adf_test_stk_price = adf.test(stock_price); adf_test_stk_price  #  Non-Stationary

adf_test_stk_ret = adf.test(stock_ret); adf_test_stk_ret # Stationary
```
```{r}
# Ljung-Box Test for Autocorrelation - Stock Data
# ********************************************

lb_test_stk_ret = Box.test(stock_ret); lb_test_stk_ret # Inference : Stock Difference (Stationary) Time-Series is Autocorrelated

# Autocorrelation Function (ACF) | Partial Autocorrelation Function (PACF)
# ***************************************************************************
```
```

```
acf(stock_price) # ACF of Stock Price
pacf(stock_price) # PACF of Stock Price

acf(stock_ret) # ACF of Stock Return (Stationary) Series
pacf(stock_ret) # PACF of Stock Return (Stationary) Series
```
```{r}
# Auto ARIMA
arma_pq_stk_ret = auto.arima(stock_ret); arma_pq_stk_ret
arma_pq_stk = auto.arima(stock_price); arma_pq_stk
```
```{r}
# Ljung-Box Test for Autocorrelation - Model Residuals
# ***************************************************
lb_test_arma_pq_stk_ret = Box.test(arma_pq_stk_ret$residuals); lb_test_arma_pq_stk_ret

# Forecasting with ARIMA Models
# ***********************************
# jj_ds_f11 = predict(arma11, n.ahead = 40)
# plot(jj_ds_f11)
# lines(jj_ds_f11$pred, col = 'blue')
# lines(jj_ds_f11$pred + 2 * jj_ds_f11$se, col = 'red')
# lines(jj_ds_f11$pred - 2 * jj_ds_f11$se, col = 'red')

stock_ret_fpq = forecast(arma_pq_stk_ret, h = 40)
plot(stock_ret_fpq)

stock_fpq = forecast(arma_pq_stk, h = 40)
plot(stock_fpq)
```
```{r}
# Test for Volatility Clustering or Heteroskedasticity: Box Test
stk_ret_sq = arma_pq_stk_ret$residuals^2 # Return Variance (Since Mean Returns is approx. 0)
plot(stk_ret_sq)
stk_ret_sq_box_test = Box.test(stk_ret_sq, lag = 10) # H0: Return Variance Series is Not Serially
Correlated
stk_ret_sq_box_test # Inference : Return Variance Series is Heteroskedastic (Has Volatility Clustering)

# Test for Volatility Clustering or Heteroskedasticity: ARCH Test
stk_ret_arch_test = ArchTest(arma_pq_stk_ret$residuals, lags = 10) # H0: No ARCH Effects
stk_ret_arch_test # Inference : Return Series is Heteroskedastic (Has Volatility Clustering)
```
```{r}
# GARCH Model
garch_model1 = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1,1)),
mean.model = list(armaOrder = c(0,0), include.mean = TRUE))
nse_ret_garch1 = ugarchfit(garch_model1, data = arma_pq_stk_ret$residuals); nse_ret_garch1

garch_model2 = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1,1)),
mean.model = list(armaOrder = c(4,5), include.mean = FALSE))
nse_ret_garch2 = ugarchfit(garch_model2, data = arma_pq_stk_ret$residuals); nse_ret_garch2
```

 # Test for Volatility Clustering or Heteroskedasticity: ARCH Test
gar_resd = residuals(nse_ret_garch2)^2
stk_ret_arch_test1 = ArchTest(gar_resd, lags = 1) # H0: No ARCH Effects
stk_ret_arch_test1 # Inference : Return Series is Heteroskedastic (Has Volatility Clustering)
```
```{r}
*garch_modelf = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1,1)),*
*mean.model = list(armaOrder = c(4,5), include.mean = FALSE))*
*stk_ret_garch = ugarchfit(garch_modelf, data = stock_ret); stk_ret_garch*
```
```{r}
*# GARCH Forecast*
*stk_ret_garch_forecast1 = ugarchforecast(stk_ret_garch, n.ahead = 50); stk_ret_garch_forecast1*
*plot(stk_ret_garch_forecast1)*
```

*Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.*

*When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).*

*The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.*


```