**Demand and Forecast Group Assignment**

# G24044 Shubhendu Dandapat

# Time Series Forecasting using R

## 1. Introduction

This assignment demonstrates the application of time series forecasting and regression analysis on the monthly stock price data of IBM. The process involves data preprocessing, model building, diagnostic testing, and forecasting future values using ARIMA models with external regressors.

## 2. Libraries Used

lmtest: For regression diagnostics  
tseries: For time series stationarity tests  
forecast: For ARIMA modeling and forecasting  
dplyr: For data manipulation

## 3. Data Preprocessing

The data is first transformed by converting the date column and extracting month-year combinations. Then it is grouped by month to compute monthly averages for Open, High, Low, Close, and Volume.

## 4. Regression Models

Two linear models are created to predict the average Close price:  
Model 1: Includes Open, High, Low, Volume  
Model 2: Excludes Volume  
Model 2 is preferred based on AIC values.

## 5. Stationarity Testing

Stationarity is tested using KPSS and ADF tests. The Close prices are non-stationary initially, but become stationary after differencing.

## 6. ARIMA Modeling

Box-Cox transformation is applied and two ARIMA models are fitted:  
Model with Open, High, Low, Volume as regressors  
Model without Volume  
Residuals are tested using Box test to ensure no autocorrelation.

## 7. Forecasting and Output

Forecasts are generated using the test set and plotted. Outputs are exported to CSV files for further use.

## 8. R Code

library(lmtest)  
library(tseries)  
library(forecast)  
library(dplyr)

* These packages support time series analysis, forecasting, regression testing, and data wrangling.

**Data Processing**

data$Date <- as.Date(data$Date)

data$Month <- format(data$Date, "%Y-%m")

* Converts the Date column to proper Date format.
* Extracts year-month for monthly aggregation.

data2 <- data %>%

group\_by(Month) %>%

summarise(

Open\_Avg = mean(Open, na.rm = TRUE),

High\_Avg = mean(High, na.rm = TRUE),

Low\_Avg = mean(Low, na.rm = TRUE),

Close\_Avg = mean(Close, na.rm = TRUE),

Volume\_Avg = mean(Volume, na.rm = TRUE)

)

* Groups daily data by month and computes monthly **average** of Open, High, Low, Close, and Volume

draft <- data2 %>%

summarise(

Open = Open\_Avg,

High = High\_Avg,

Low = Low\_Avg,

Close = Close\_Avg,

Volume = Volume\_Avg

)

**Train-Test Split**

train <- draft[1:100,]

test <- draft[101:144,]

* Splits draft into training and testing sets: first 100 rows for training, remaining 44 for testing.

**Linear Regression Models**

model\_1 <- lm(Close ~ Open + High + Low + Volume, data=draft)

model\_2 <- lm(Close ~ Open + High + Low, data=draft)

* Fits two regression models to predict Close price.

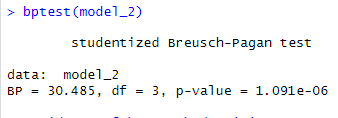
AIC(model\_1) # -145.904

AIC(model\_2) # -147.3206

* Comparing AIC values
* Model 2 (without Volume) is preferred (lower AIC = better fit).

**Diagnostic Tests**

bptest(model\_2) # Breusch-Pagan test for heteroskedasticity



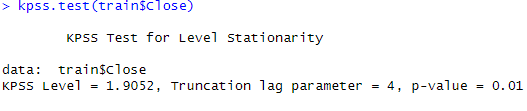
* Indicates heteroskedasticity is present.

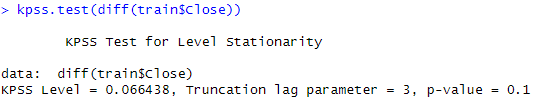
kpss.test(train$Close) # Tests for trend stationarity

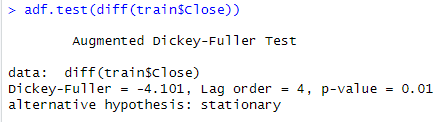
adf.test(train$Close) # Augmented Dickey-Fuller test for unit root

kpss.test(diff(train$Close))

adf.test(diff(train$Close))







Since the p-value (0.1) is greater than 0.05, we **fail to reject the null hypothesis**. Therefore, we **do not have enough evidence to conclude that the 1 time series is non-stationary , hence it is stationary**

Stationarity checks:

* Close is non-stationary initially.
* After differencing, becomes stationary

**Box-Cox Transformation and ARIMA Modeling**

value\_bc <- BoxCox.lambda(train$Close) # Box-Cox λ



Box-Cox transformation helps stabilize variance.

reg <- cbind(train$Close, train$Open, train$High, train$Low,train$volume)

model\_final <- auto.arima(train$Close, xreg=reg, lambda=value\_bc, d=1)

* Fits ARIMA with external regressors and Box-Cox transformation.

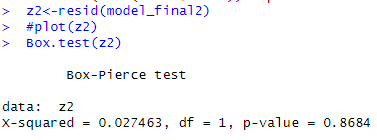
reg2 <- cbind(train$Close, train$Open, train$High, train$Low)

model\_final2 <- auto.arima(train$Close, xreg=reg2, lambda=value\_bc, d=1)

* Second model drops Volume to compare performance.

**Model Residual Checks**

Box.test(resid(model\_final2))



* Checks for autocorrelation in residuals. No significant autocorrelation = good model.

**Forecasting**

new <- cbind(test$Close, test$Open, test$High, test$Low)

output <- forecast(model\_final2, xreg=new2)

plot(forecast(model\_final2, xreg=new2))

* Forecasts future Close prices using model\_final and test data.
* Plots the forecast.

A graph with a line and a blue line

AI-generated content may be incorrect.

After performing the Time series forecasting we obtained the new forecasted values

Here are the links for data : <https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231/code>

Here is the github link for code : <https://github.com/shubhendu7astra/dbf_xlri/blob/main/ibm_demand_forecasting.R>