Time Series Analysis

Analysis and Prediction of BHP Stocks Price in R

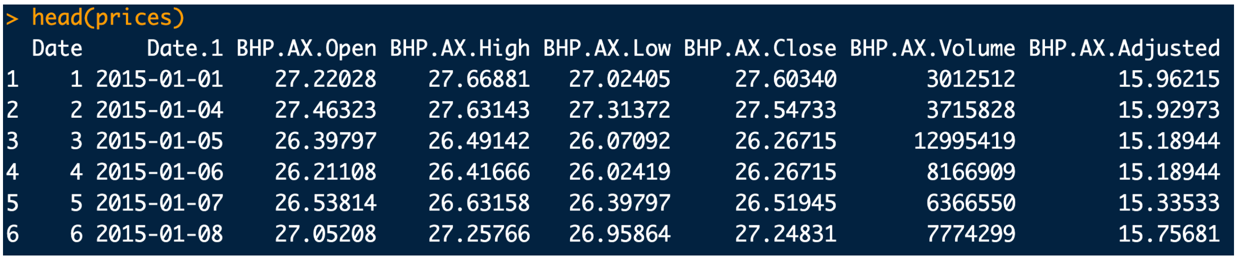
By Amir Mohammadi

Introduction:

Data collected in a chronological sequence is known as time-series data. Often, it is gathered at regular intervals, such as daily, monthly, or annually. By plotting this type of data, patterns such as trends, seasonality, or a mix of both can be observed. Recognizing these patterns can aid in decision-making. For instance, discerning a seasonal pattern in stock prices can help determine the optimal times to buy or sell shares. In R, data decomposition allows for a more detailed examination of these patterns.

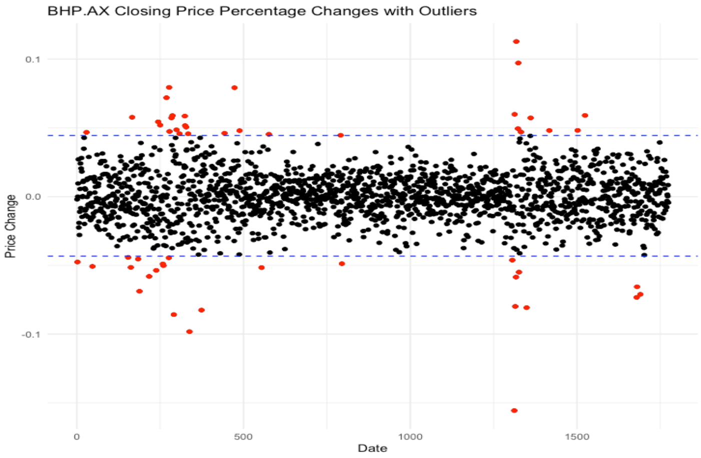
Data

In R, the quantmod package makes it simple to acquire stock prices. This package caters to quantitative traders, helping them effortlessly explore and develop trading models. It relies on Yahoo Finance for stock price data, so it's essential to verify your stock's ticker on Yahoo Finance before using this package. In this example, we'll analyse the stock price of BHP Mining (BHP.AX) from 2015 to the end of 2021.



```
> head(prices)
  Date     Date.1 BHP.AX.Open BHP.AX.High BHP.AX.Low BHP.AX.Close BHP.AX.Volume BHP.AX.Adjusted
1    1 2015-01-01    27.22028    27.66881   27.02405     27.60340       3012512        15.96215
2    2 2015-01-04    27.46323    27.63143   27.31372     27.54733       3715828        15.92973
3    3 2015-01-05    26.39797    26.49142   26.07092     26.26715      12995419        15.18944
4    4 2015-01-06    26.21108    26.41666   26.02419     26.26715       8166909        15.18944
5    5 2015-01-07    26.53814    26.63158   26.39797     26.51945       6366550        15.33533
6    6 2015-01-08    27.05208    27.25766   26.95864     27.24831       7774299        15.75681
```

We can check for null values (non-trading days)and also remove outliers (extreme lows or highs). One common method for detecting outliers is using IQR (Interquartile Range).

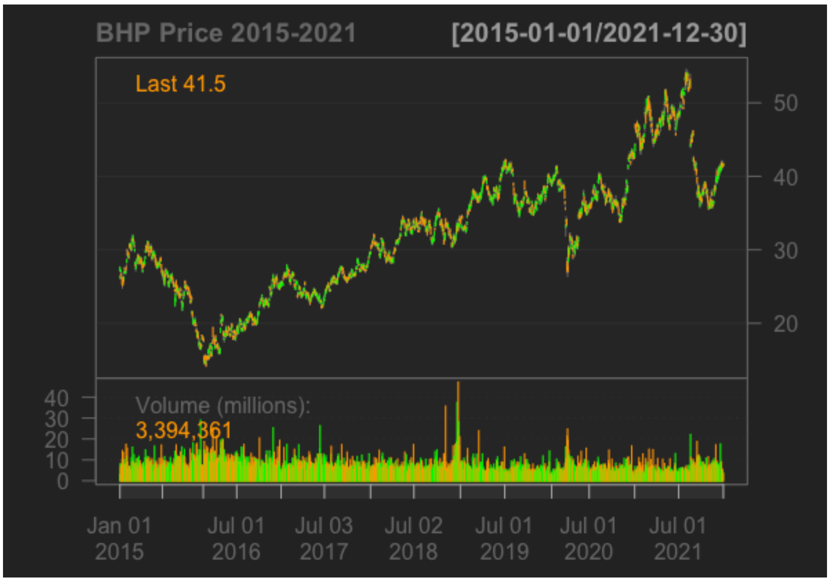


The data we've gathered consists of daily OHLC (Open, High, Low, Close) values. For our analysis, we'll focus on the closing price, which can be extracted using the Cl() function. Closing prices are deemed valuable indicators for evaluating fluctuations in stock prices over time. Some investors, however, prefer using adjusted prices instead of closing prices.

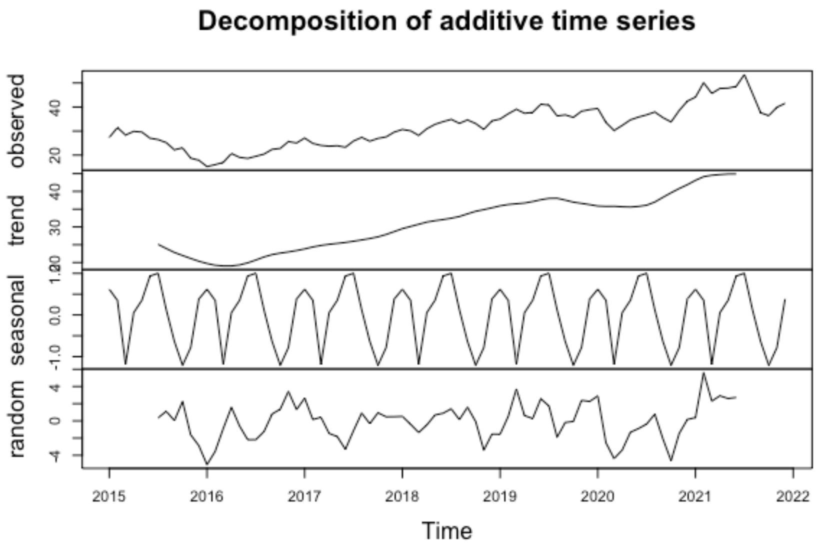
Visualization:

Stock price data can be visualized through static or interactive plots. By doing so, we can examine the data patterns and understand how they may influence our investment decisions.

Static Plot:



In general, the plot reveals an upward trend and seasonal patterns. To further examine the data components, we will decompose it. Our focus will be on the closing price, and we will initially convert the periods to monthly data.

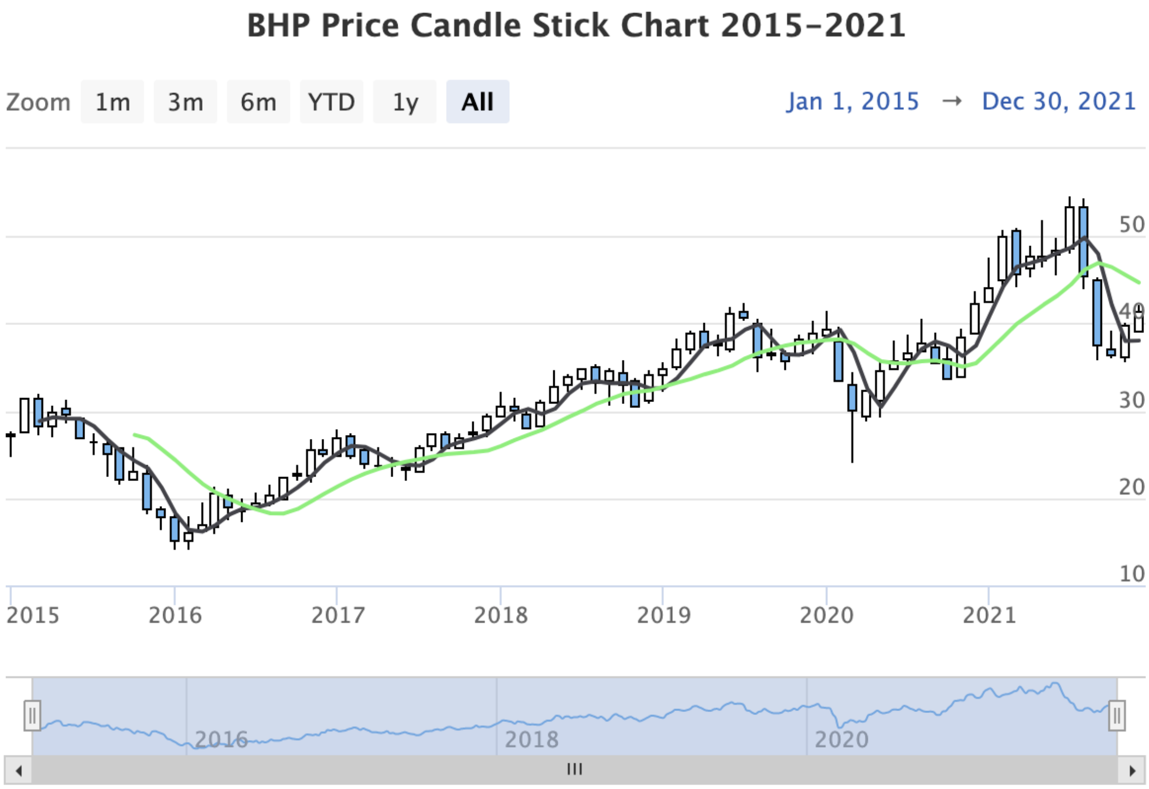


The output shows four plots of our closing price data, which are:

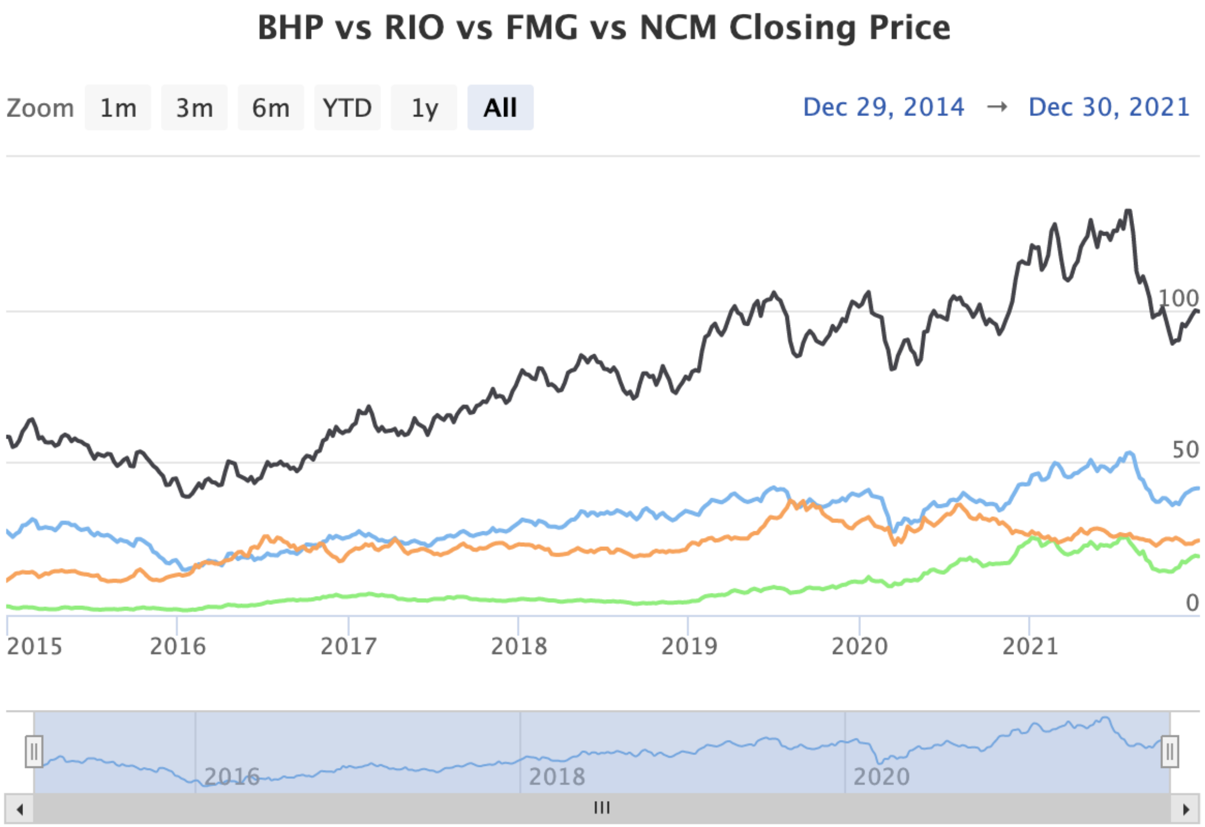
- **Trend:** Refers to long-term shifts in the mean. In this particular plot, we can observe a significant upward trend beginning around late 2015.
- **Seasonal:** The repetitive cyclical variations in the data. The closing price of BHP typically peaked in March and reached its lowest point in December. Based on this pattern, it can be inferred that the optimal time to sell this stock was at the start of the year (particularly in March), while the best time to buy was towards the year's end (specifically in December).
- **Random:** This component represents irregular or random fluctuations not captured by the trend or seasonality. The ongoing Covid-19 pandemic serves as an example of a factor causing such random fluctuations. When the random component is dominant in the data, forecasting becomes more challenging to accomplish accurately. As a result, this article only utilizes data up to 2019.

Interactive Plot:

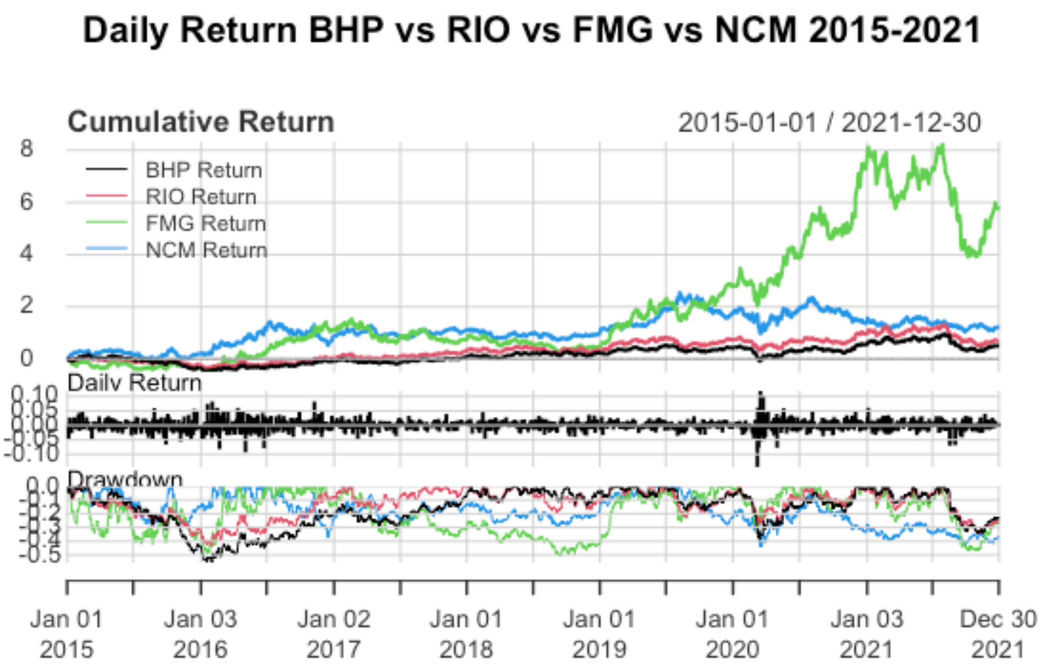
Employing this type of visualization is highly beneficial when examining data details. Additional indicators, such as the Single Moving Average (SMA), are easier to interpret in interactive plots compared to static ones. These plots can be used to emphasize key technical analysis factors that impact our decisions, such as golden or death crosses.



This chart can also be employed to compare our stock price with other stocks in the same sector and the stock index. Such visualization aids in analysing our stock's performance relative to the market. Deciding whether to continue investing in our stock as opposed to other stocks in the same sector can be made after comparing and conducting further analysis.



Examining returns is also crucial for investors, as they naturally want to avoid stocks that yield poor returns on their investments. The return period can vary depending on the chosen timeframe, such as daily, weekly, or monthly returns.



The concept of forecasting time-series data involves utilizing past and present data to make predictions about the future. While no forecasting model can guarantee 100% accuracy, the results are often valuable for making future-related decisions. Numerous models and algorithms can be employed for time-series data forecasting; however, this discussion will focus on the Naive Method and the ARIMA model.

A. Splitting the Data:

Before forecasting our time-series data, we first divide it into training and testing sets. This allows us to evaluate the model's performance in predicting data not used during training. In simpler terms:

Training data: Data used to fit the model

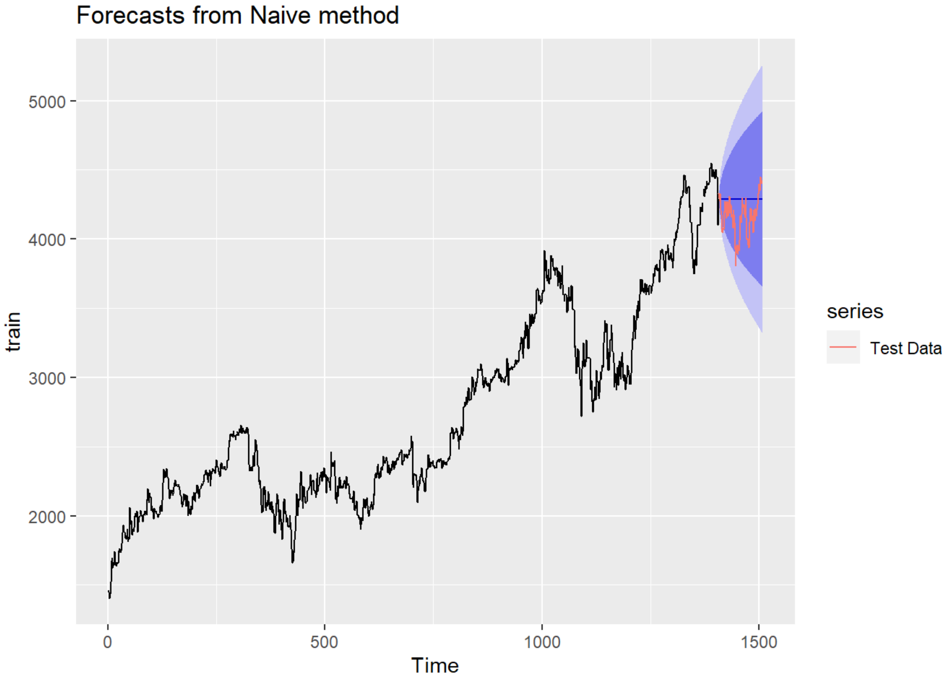
Testing data: Data used to evaluate the model

As analysts, we must determine a split percentage that adequately represents both the training and testing data without incurring excessive computational costs in model training. The most common split ratio is 80% training to 20% testing, although this depends on the specific data and the purpose of the forecast.

In this case, we want to forecast 100 days of stock price in our dataset. Therefore, we use the 100 last observations as the test data while using the remaining data as the train data.

B. Naïve Model:

The Naive Method is a forecasting approach that uses the last observation as the forecast result for our data. It serves as a baseline model in forecasting. If the performance of our chosen model is worse than that of the Naive Method, it would be preferable not to use our model.



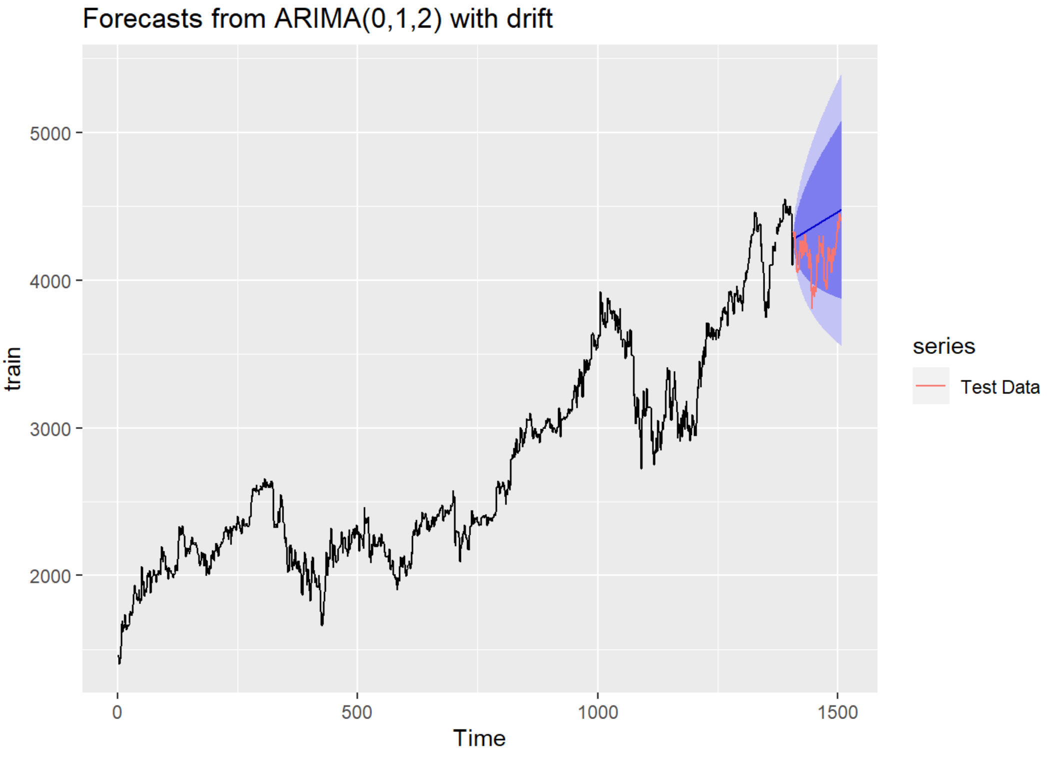
The blue line represents the mean of our prediction, while the darker and lighter shaded areas signify the 80% and 95% confidence intervals, respectively. Comparing these results to the actual test data reveals noticeable differences between the two.

C. ARIMA Model:

The Auto-Regressive Integrated Moving Average (ARIMA) model combines the Auto-Regressive (AR) model, integration through differencing, and the Moving Average (MA) model. The AR model establishes the relationship between an observation and its lagged observations. To utilize the ARIMA model, the time-series data must be stationary, which can be achieved by differencing the data. Finally, the MA model describes the relationship between an observation and the residual errors of the moving average model on the lagged observations.

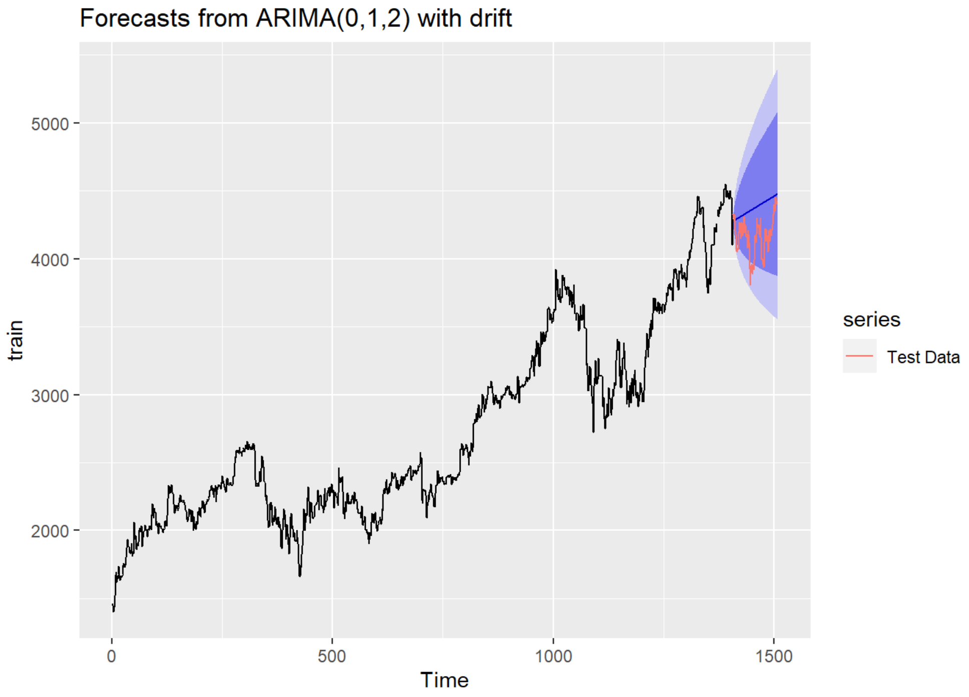
ARIMA models are typically denoted by ARIMA(p,d,q), where p, d, and q represent positive parameters. The order of AR (p) refers to the number of lag observations in the model. The parameter d indicates the number of times the actual data must be differenced to achieve stationarity, with a maximum of 2 times in most cases. Lastly, the order of MA (q) corresponds to the size of the moving average window.

Determining suitable values for the ARIMA parameters can be challenging. However, in R, we have the auto.arima() function that simplifies this task. Using this function, we can obtain two types of ARIMA models: non-seasonal ARIMA (as previously discussed) and seasonal ARIMA. In non-seasonal ARIMA, the seasonal component of the data is not considered, whereas in seasonal ARIMA, it is included. We will assess which model yields better forecast results for our data.



The non-seasonal ARIMA model produces results that show an upward trend in the data. Nevertheless, when compared to the actual test data, there remain noticeable differences between the two.

Interestingly, including or excluding the seasonal component in our data using the auto.arima() function produces the same results. This indicates that the seasonal aspect of our data is not significant, and therefore, the non-seasonal ARIMA model is the best ARIMA model based on auto.arima() for our dataset.



D. Forecast Evaluation:

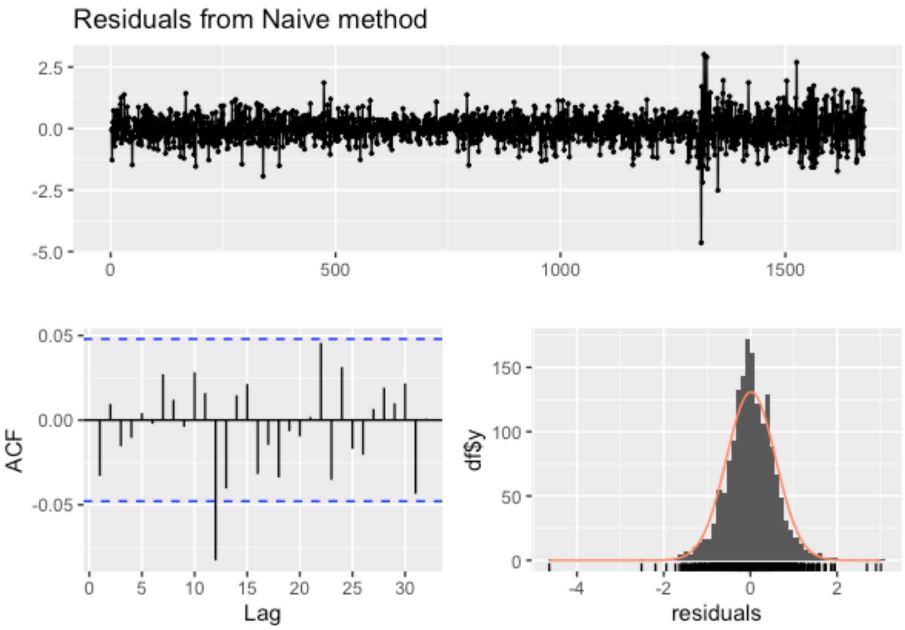
Once the data has been forecasted, the final step is to evaluate the predictions. Forecast evaluation involves verifying if the residuals meet the residual assumptions and comparing accuracy metrics.

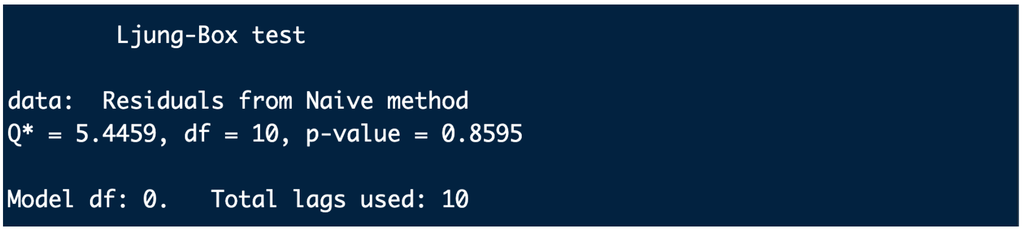
Residual Checks:

A residual is the difference between the forecasted data and the actual data. A good model should have randomly distributed residuals without any apparent pattern. The following residual assumptions must be met:

- Normally distributed (mean = 0), which can be checked using a normal curve. The normal curve should have a bell shape.
- Possessing constant variance, which can be verified using a residual plot. Constant variance is indicated by the data's consistent fluctuations.
- Lacking autocorrelation, which can be assessed using an ACF (Autocorrelation Function) plot and the Ljung Box test. Autocorrelation can be detected in the ACF plot when lines extend beyond the upper or lower bounds. In the Ljung Box test, to satisfy this assumption, the p-value must be greater than 0.05. If the results of the ACF plot and Ljung Box test differ, we prefer to rely on the Ljung Box test's outcome.

Residuals of Naïve Model



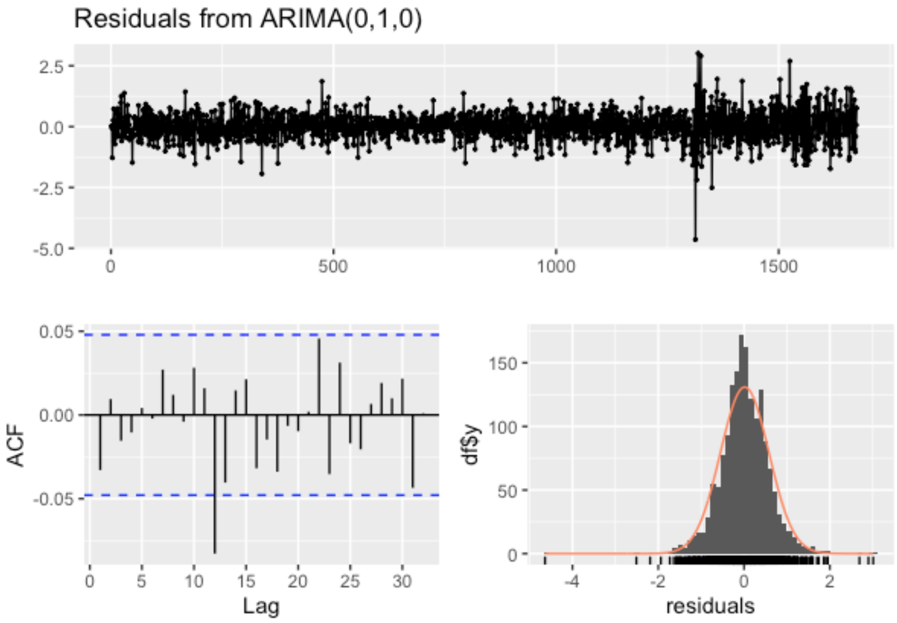


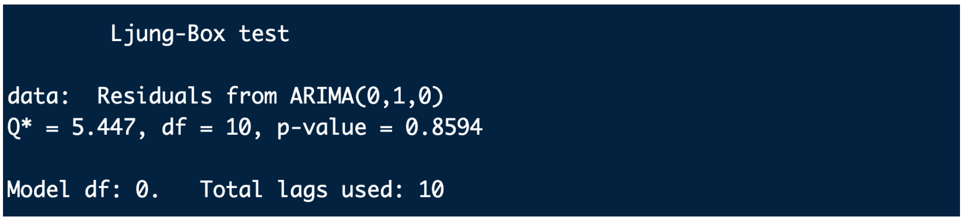
```
        Ljung-Box test

data:  Residuals from Naive method
Q* = 5.4459, df = 10, p-value = 0.8595

Model df: 0.   Total lags used: 10
```

Residuals of ARIMA Model





```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,0)
Q* = 5.447, df = 10, p-value = 0.8594

Model df: 0.   Total lags used: 10
```

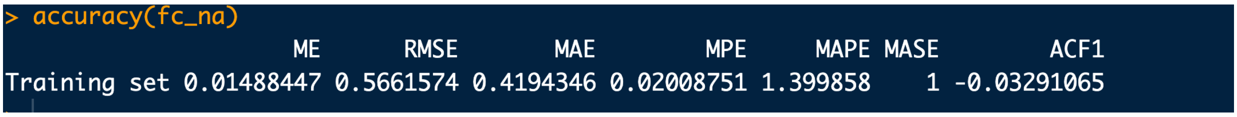
Based on the results, we can see that both the Naive Method and ARIMA model satisfy the residual assumptions. We can now check the accuracy of these two models.

Accuracy Metrics Comparison

There are numerous accuracy metrics used in forecasting. In this discussion, we will compare the Root Mean Squared Error (RMSE) of both models.

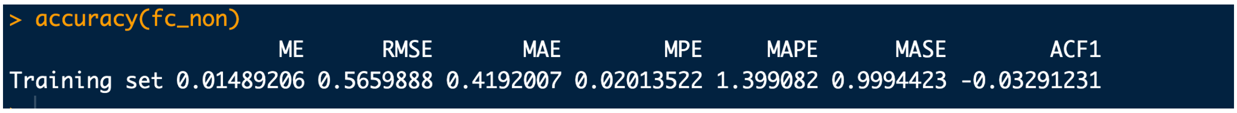
Root Mean Squared Error (RMSE): The standard deviation of the residuals, which measures how spread out the residuals are. A lower RMSE value indicates a better result.

Naïve Method Accuracy:



```
> accuracy(fc_na)
                     ME      RMSE       MAE        MPE     MAPE MASE        ACF1
Training set 0.01488447 0.5661574 0.4194346 0.02008751 1.399858    1 -0.03291065
```

ARIMA Model Accuracy:



```
> accuracy(fc_non)
                     ME      RMSE       MAE        MPE     MAPE      MASE        ACF1
Training set 0.01489206 0.5659888 0.4192007 0.02013522 1.399082 0.9994423 -0.03291231
```

Based on this we can see that both models have a high and very similar accuracy.

Conclusion

Time-series data is pervasive and often influences our decisions. Besides stock prices, which we discussed throughout this article, there are numerous other examples. For a business owner of a department store, sales data is a crucial time-series dataset that drives business decisions. Analysing and forecasting sales data can assist a businessperson in making informed decisions.

Decomposing time-series data provides a more detailed view of the patterns in the data. Analysing these patterns can help with decision-making. However, in R, decomposition can only be performed on ts objects with specified frequencies.

After analysing time-series data, we often want to forecast it to help inform future decisions. Keep in mind, however, that forecasting results can never be 100% accurate, so it's important to have contingency plans. Forecasting can be done using various models and algorithms. In this article, we discussed the Naive Method and ARIMA model.

The Naive Method is an excellent baseline tool and a straightforward approach, as it uses the last observation as the forecast result. While this method can sometimes be useful (e.g., a salesperson targeting the same sales for the next month as the current month), it is generally recommended to use more robust approaches for other cases. In contrast, the ARIMA model (including the seasonal variant) can be an effective model for forecasting more fluctuating data. However, this model is only applicable to data that can be differenced to achieve stationarity. There are many other models that may yield better results for your dataset, so don't hesitate to explore further!