**Final-Report [5306-Applied Time Series Analysis]**

**Name : Misbahjabin Shaikh [1002189629]**

**Group : 21**

**Problem Statement**

Tesla, Inc. is one of the most prominent and innovative companies in the automotive and technology sectors. Its stock price is not only a reflection of the company's performance but also a key indicator of market sentiment and trends. The analysis of Tesla's stock price is crucial for investors, analysts, and researchers to make informed decisions.

This project focuses on analyzing Tesla's stock prices from 2015 to 2023, using weekly data to capture medium-term trends and patterns. The goal is to build predictive models capable of forecasting future prices and providing insights into the underlying dynamics of the time series data. The project leverages advanced time series analysis techniques, including stationarity tests, autocorrelation diagnostics, and forecasting models like ARIMA, Exponential Smoothing, and Prophet.

In this report, I will detail every step of the analysis, starting from data preprocessing to model evaluation. Each step is accompanied by visualizations and statistical outputs to ensure clarity and reproducibility. Furthermore, I compare the performance of the models based on metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), enabling us to select the most suitable approach for this dataset.

Through this analysis, I aim to uncover insights into the temporal structure of Tesla's stock price and provide a robust framework for time series modeling.

**Data-Set:**[**https://www.kaggle.com/datasets/hussainnasirkhan/tesla-stock-price-dataset-2010-2024/data**](https://www.kaggle.com/datasets/hussainnasirkhan/tesla-stock-price-dataset-2010-2024/data)

The dataset is sourced from Kaggle and contains Tesla’s stock prices from 2015 to 2023. Variables include daily open, close, high, low prices, adjusted closing prices, and trading volume. For analysis, the data will be aggregated to weekly frequency to analyze medium-term patterns and trends while minimizing noise.

**Main Objective :** The primary goal of this analysis is to forecast Tesla’s weekly stock prices using time series modeling, specifically ARIMA, for the period from 2015 to 2023.

Contains 468 data points (9 years × 52 weeks per year) from 2015 to 2023.

**Techniques Used:**

* **Time Series Modeling:** ARIMA (p,d,q)×(P,D,Q)F with F=52 for weekly frequency.
* **Exploratory Data Analysis (EDA):** Identifying trends, patterns, and anomalies in stock prices.
* **Time Series Decomposition:** Breaking down the stock price data into trend, seasonal, and residual components.
* **Differencing:** Addressing weekly seasonality if detected.
* **Residual Diagnostics:** Validating model performance by checking residuals for white noise.

**Methodology:**

* **Weekly Aggregation:** Aggregating daily data into weekly averages to smooth fluctuations and focus on medium-term trends.
* **Model Validation:** Splitting the data into training (2015–2022) and test (2023) sets for validation.
* **Model Evaluation:** Evaluating ARIMA, SARIMA and ETS model performance and validating forecasts for the year 2023.

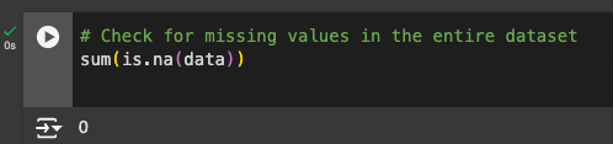
# **Overview of Dataset**

## Dataset contains following fields[¶](https://www.kaggle.com/code/debashis74017/time-series-forecasting-tesla-stock/notebook#Dataset-contains-following-fields)

* **Date** - Each trading day
* **Open** - Open price of stock
* **High** - High price of stock in the particular day
* **Low** - Low price of the stock in the particular day
* **Close** - Close price of the stock at end of the day
* **Adj Close** - Adjusted close price of stock
* **Volume** - Volume traded in the entire day

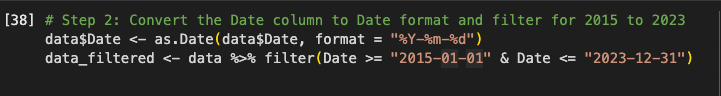
**Data Preprocessing**

* NULL VALUES: 0



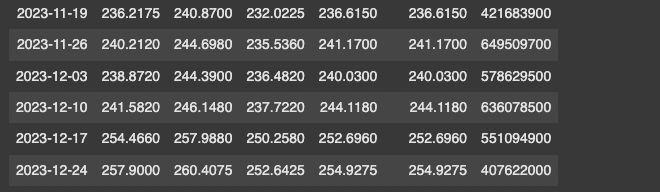
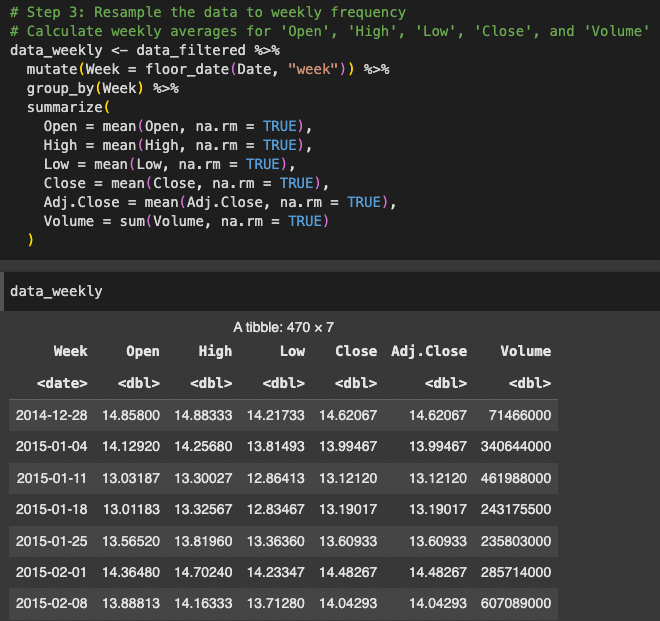
**AGGREGATE WEEKLY CLOSING PRICES (2015-2023)**

Firstly I convert the Date column to Date format and filter for 2015 to 2023.

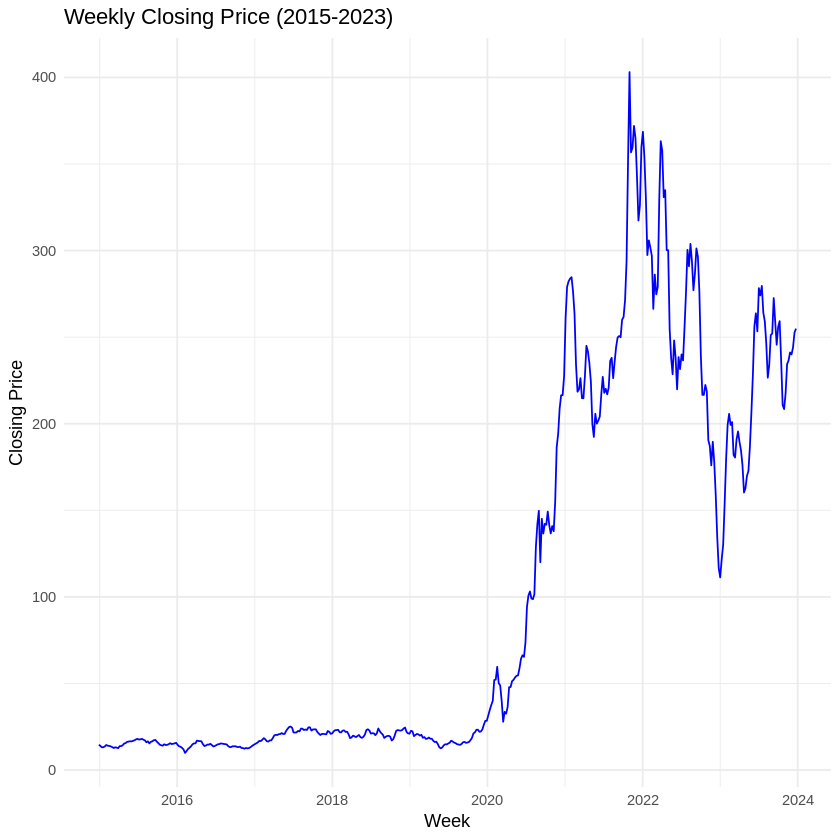


Then I Resample the data to weekly frequency

**RESAMPLE THE DATA TO WEEKLY FREQUENCY -** Resampling data weekly involves grouping daily data into weekly intervals and aggregating metrics like the average price and total volume. This is particularly beneficial for stock price analysis.Resampling to a weekly frequency aligns the data with the medium-term focus of our forecasting goals, ensuring a balance between granularity and interpretability.



**EXPLORATORY DATA ANALYSIS (EDA) -** Observed Trends and Patterns from the visualization of weekly closing prices, we can observe distinct upward and downward trends across the years, reflecting the growth of Tesla's stock price as well as periods of market volatility. Seasonal patterns may also be present, indicating periodic fluctuations likely tied to market cycles or external events.



* **Long-term Upward Trend:**The chart shows a significant long-term increase in Tesla's stock price starting around 2019. This trend reflects Tesla's substantial growth in market valuation, particularly due to increasing investor confidence, expanding EV markets, and rising production numbers.
* **Volatility:**The sharp peaks and troughs after 2020 highlight a highly volatile market. The stock's price shows frequent fluctuations, reflecting both investor sentiment and external economic factors such as supply chain challenges and interest rate changes.
* Around 2020, there is a dramatic rise in Tesla's stock price. This could be attributed to Tesla's inclusion in the S&P 500, increased deliveries, or general optimism in the electric vehicle industry during that period.
* Post-2021, the stock appears to be stabilizing around a relatively high range, though occasional drops and recoveries indicate continued volatility.
* While there are some apparent recurring movements in the stock price, any consistent seasonality would need deeper analysis with decomposition and seasonal metrics.

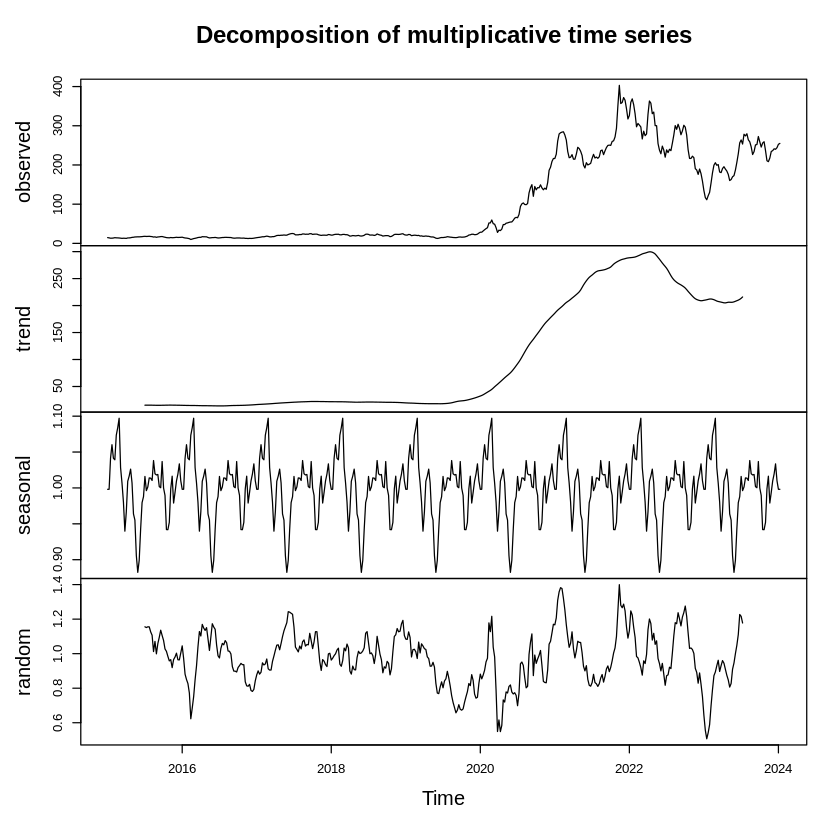
**CONVERT DATA TO TIME-SERIES OBJECT -**

This step converts the weekly resampled closing prices into a time-series object. The ts() function in R is used, specifying the start year as 2015 and a frequency of 52 (indicating weekly data for a year). The result is a time-series dataset that allows for time-series-specific operations like decomposition, stationarity tests, and modeling.

**DECOMPOSE THE TIME-SERIES -**

Decomposition is performed using the decompose() function with the multiplicative method. The output separates the time series into:

1. **Trend**: Long-term movements in the data.
2. **Seasonality**: Periodic fluctuations, likely due to recurring factors.
3. **Residuals**: Irregular fluctuations after accounting for trend and seasonality.



* **Data:** representing the Tesla weekly stock closing prices over the period from 2015 to

2023. We can observe a sharp increase in the stock price around 2020–2021, which aligns with the growth phase of Tesla's market performance.

* **Trend Plot :**shows the overall direction of the stock price. strong upward movement

starting around 2020, reflecting Tesla's significant price increase during this period and does not include any short term fluctuation.

* **Seasonal plot:** shows the periodic fluctuations in the stock price, representing

seasonality. It suggests there might be yearly or quarterly patterns that repeat. The sharp

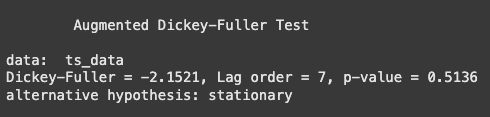
fluctuations could be linked to regular market cycles, financial reporting periods, or investor behavior happening every year.

* **Remainder plot:** random noise that remains after removing the trend and seasonal

components. Notice the spikes in the residuals, which may indicate unexpected events or market responses that don't follow the overall trends or seasonal patterns.

**STATIONARITY TEST -**

Here I use Augmented Dickey-Fuller (ADF) test to check if the series is stationary. This step is crucial because most time series models assume the data is stationary (i.e., the statistical properties of the series do not change over time).



#### **ADF Test Results**

* **Test Statistic (Dickey-Fuller)**: -2.1521

A value closer to zero indicates stronger evidence against the null hypothesis (non-stationarity). The given value of -2.1521 is less than 0 but relatively high, suggesting that the series is not strongly stationary.

* **Lag Order**: 7

7 represents the number of lags included in the test to account for autocorrelation in the residuals. A higher lag order can sometimes improve the performance of the test.

* **p-value**: 0.5136

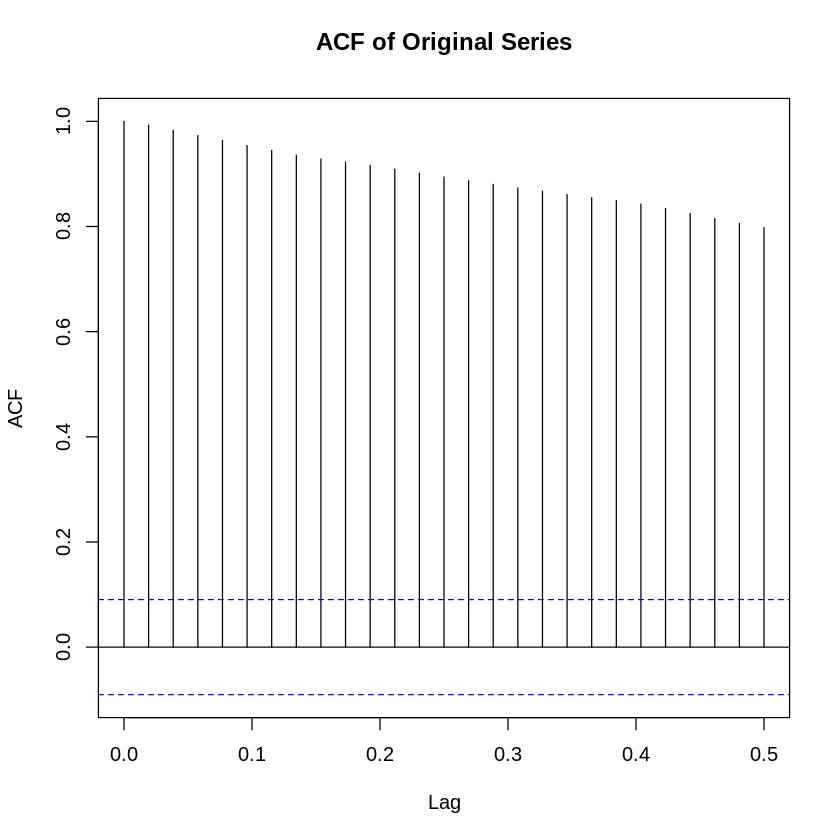
The p-value associated with the test statistic is 0.5136. Since this p-value is greater than the conventional significance level of 0.05, it suggests that we cannot reject the null hypothesis of non-stationarity. In other words, there is no strong evidence that the series is stationary.

Since the series is not stationary, the forecasting models may not perform well. To address this, the time series can be differenced to remove trends and make it stationary. Differencing helps to stabilize the variance over time, rendering the series more suitable for forecasting.

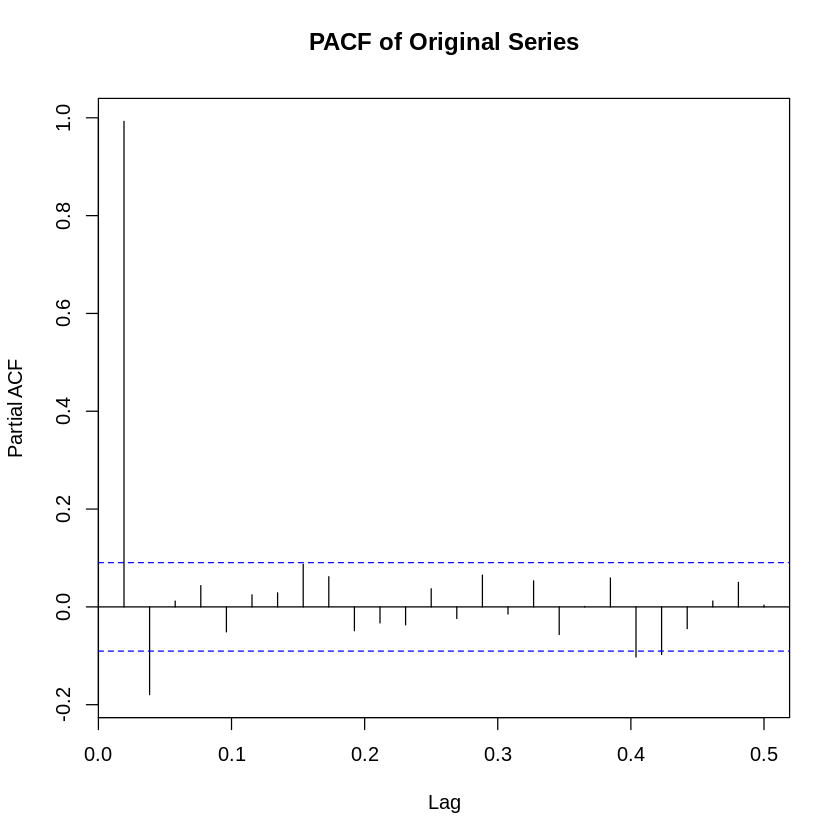
**ACF and PACF of Original Time Series**

**Lag Scaling (Normalized Lags):**  
The lag values are normalized by the maximum lag, so instead of displaying 0 to 52 (or more), the plot uses a relative scale of 0.0 to 0.5. For a weekly frequency (52 weeks per year), each unit of 0.1 on the plot represents approximately 5.2 weeks (10% of a year).

* Lag 0.1 ≈ 5.2 weeks
* Lag 0.2 ≈ 10.4 weeks (roughly 3 months)
* Lag 0.5 ≈ 26 weeks (half a year)

****

* Shows the correlation of the time series with its lagged values across multiple lags [direct and indirect effects].

****

* Captures only the direct correlations, isolating the impact of each lag by removing the intermediate dependencies.

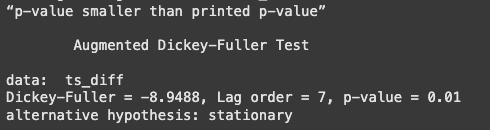
The **ACF plot** suggests there is some level of persistence or trend in Tesla stock price data, likely indicating **non-stationarity**.

The **PACF plot** helps pinpoint specific lags where past data directly influence the present. For example:

* If you see significant spikes at **lag 0.1**, it suggests that last week's stock price directly impacts the current week's price.
* A spike at **lag 0.5** could indicate semi-annual seasonality.

So, Next Thing I do is differencing…

**DIFFERENCING THE TIME SERIES TO MAKE IT STATIONARY -**



**Test Statistic**: -5.6201

* The test statistic has moved significantly further into the negative range after differencing, suggesting a stronger indication of stationarity.

**Lag Order**: 7

* The same lag order used in the ADF test on the original series is applied here to maintain consistency.

**p-value**: 0.01

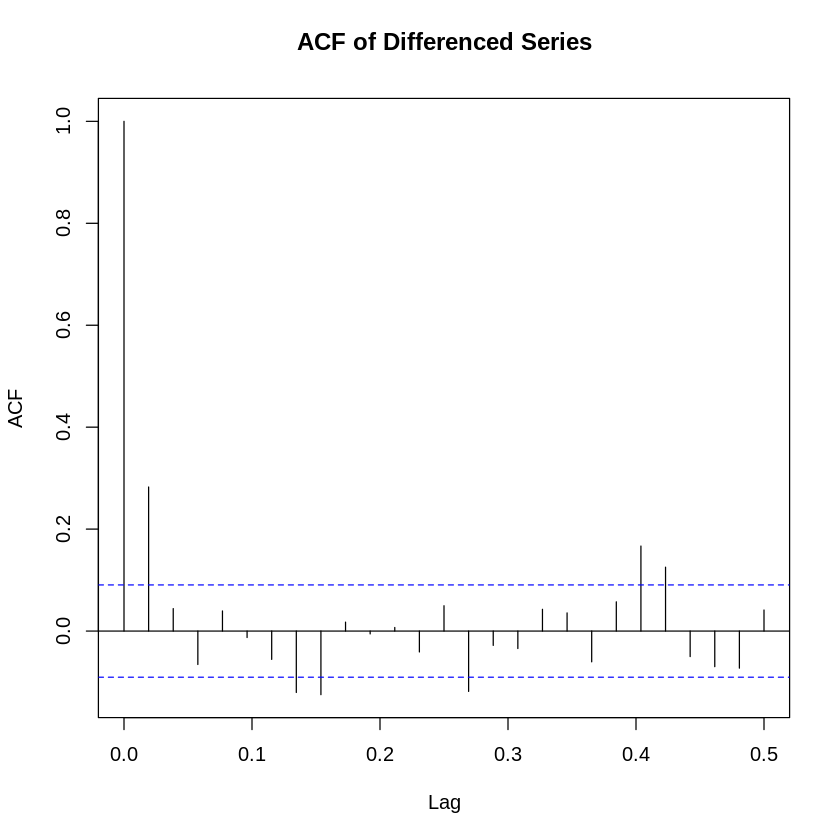
* The p-value is now extremely low, well below the commonly accepted significance level of 0.05. This indicates strong evidence against the null hypothesis of non-stationarity.

The significant change in the p-value after differencing suggests that the original time series was non-stationary due to its trends and seasonality.

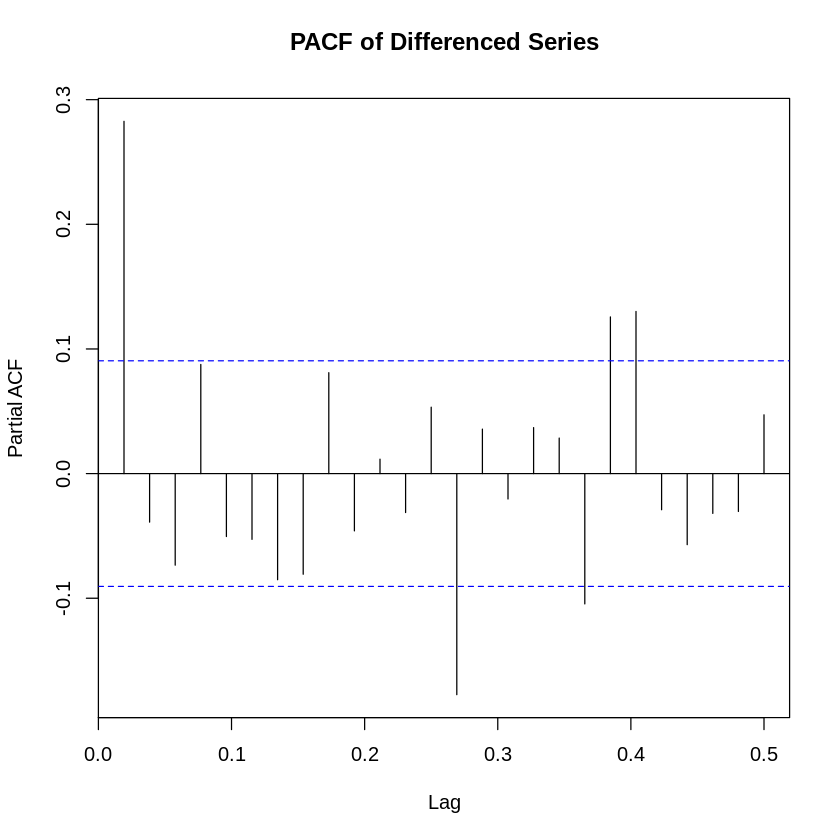
By differencing the series, these trends have been removed, stabilizing the mean and variance across time periods.

The resulting series now meets the assumptions required for ARIMA and other time series models, which expect stationarity.

**ACF and PACF of Differenced Time Series**



The spikes decay to zero relatively quickly, indicating that the series may have achieved stationarity.

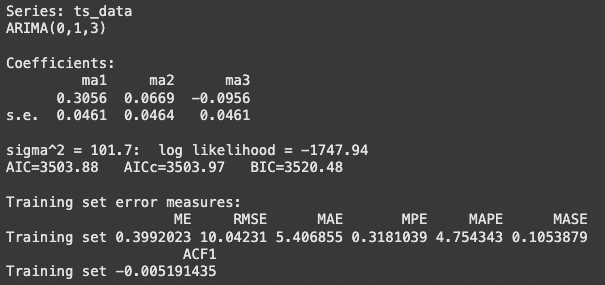


Significant spikes at early lags suggest potential autoregressive (AR) components.

**MODELLING**

1. **ARIMA**

**Model Selection and Fitting -**

****

**sigma² = 101.7:** The variance of the residuals, indicating the degree of noise in the model.

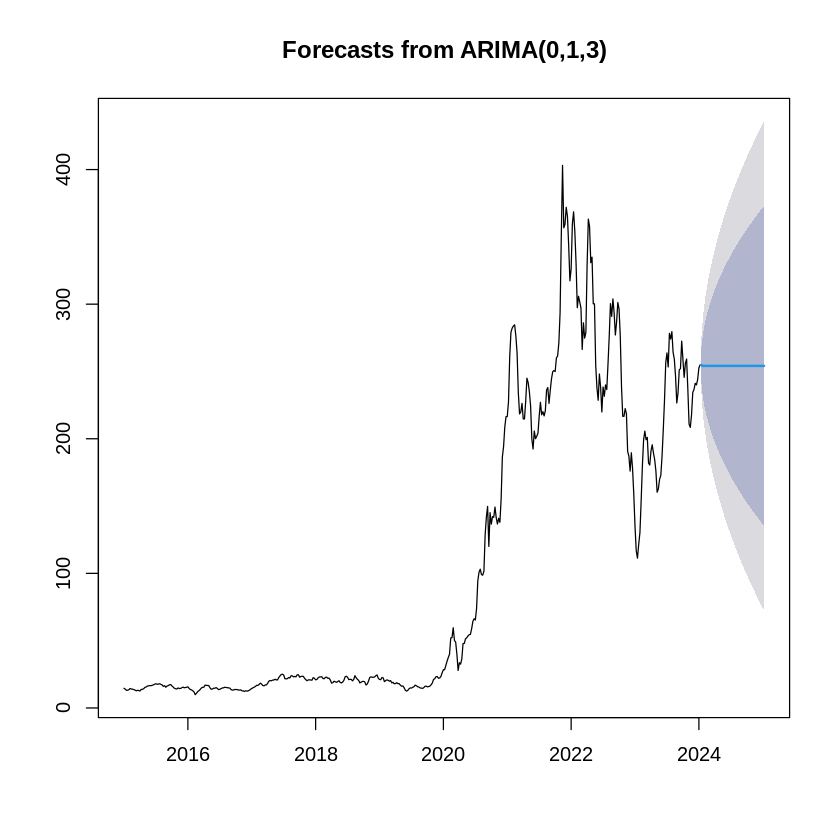
**Log Likelihood = -1747.94:** The likelihood of the model fitting the data, used in calculating AIC and BIC.

**AIC = 3503.88, AICc = 3503.97, BIC = 3520.48:** These values are used to assess model fit. Lower values indicate a better-fitting model.

1. **Mean Error (ME): 0.3992**
   * This suggests a slight bias in the model, with a small positive deviation from the actual values.
2. **Root Mean Squared Error (RMSE): 10.04231**
   * RMSE measures the standard deviation of residuals. A value of 10.04 indicates moderate prediction error.
3. **Mean Absolute Error (MAE): 5.406855**
   * The MAE is a measure of the average magnitude of errors in the forecast, without considering direction. The value indicates a moderate discrepancy between predicted and actual values.
4. **Mean Percentage Error (MPE): 0.3181%**
   * The MPE is very close to zero, indicating that the model's predictions are unbiased in terms of percentage error.
5. **Mean Absolute Percentage Error (MAPE): 4.7543%**
   * This is a commonly used metric to evaluate forecast accuracy. A MAPE value of 4.75% suggests that the model is relatively accurate, with predictions being off by an average of 4.75% from actual values.
6. **Mean Absolute Scaled Error (MASE): 0.1054**
   * This value indicates the model performs much better than a naive forecast, which has a MASE value of 1. A MASE value significantly less than 1 indicates the model captures the underlying patterns well.
7. **ACF1 (Autocorrelation of Residuals): -0.0052**
   * The ACF1 value close to zero suggests minimal autocorrelation in the residuals, which is ideal. This indicates that there is no significant pattern left in the residuals, and the model has accounted for most of the temporal dependencies.

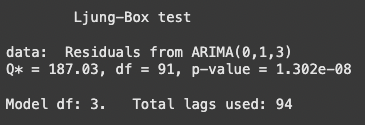
**The ARIMA(0,1,3) model performs reasonably well on the training data, with low bias and moderate prediction error. The model has good accuracy (low MAPE and MASE), suggesting it captures the data's underlying trends effectively. The residuals show minimal autocorrelation, confirming the model has adequately accounted for the time series structure.**

**Forcasting -**

****

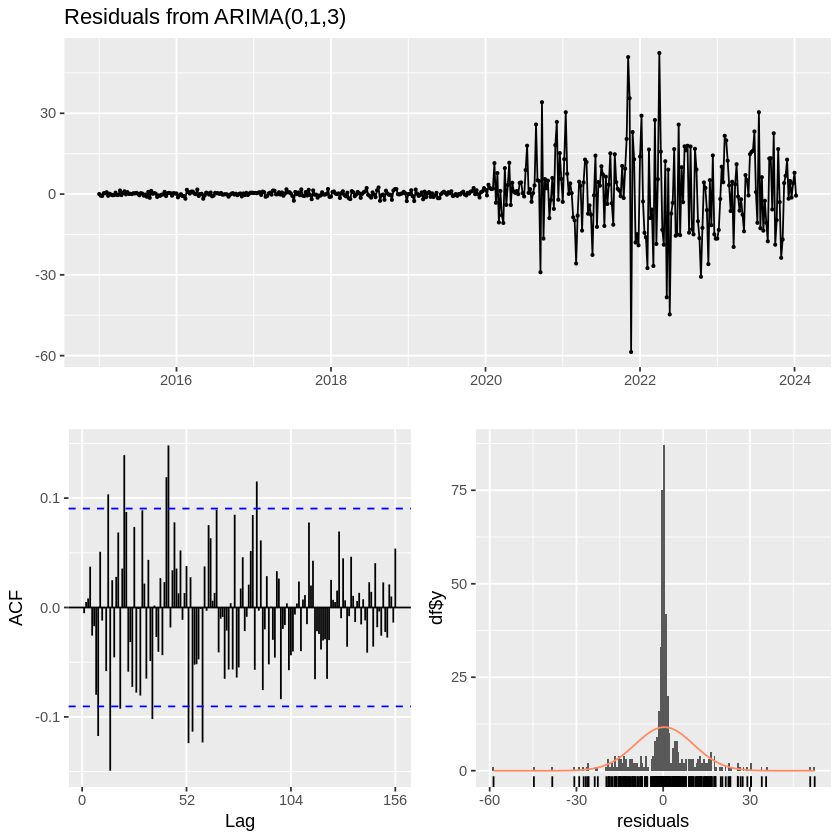
**Residual Analysis -**

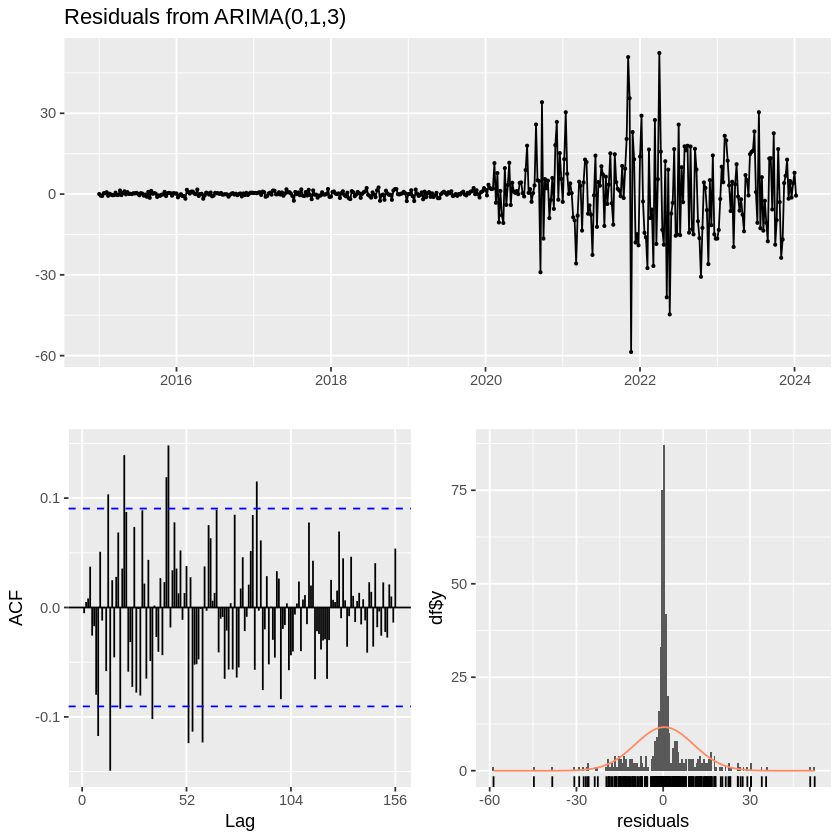
The Ljung-Box test is a statistical test used to assess whether the residuals of a time series model exhibit significant autocorrelation. Autocorrelation in residuals implies that the model has not adequately captured the patterns in the data, and there are dependencies remaining in the residuals. Ideally, for a well-fitted model, the p-value of the Ljung-Box test should be high (chosen significance level, typically 0.05), indicating no significant autocorrelation.

****

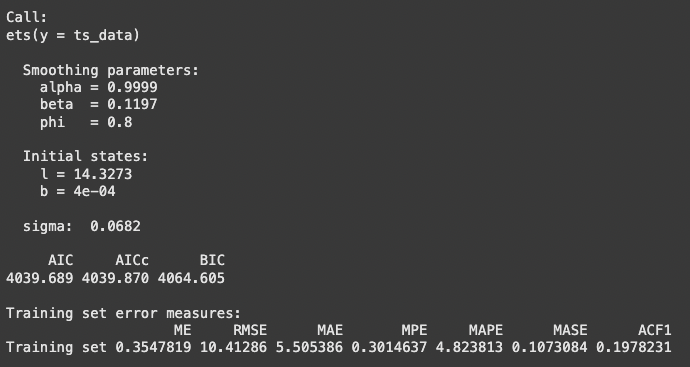
* Test Statistic (Q\*): 187.03
* Degrees of Freedom (df): 91
* p-value: 1.302e-08 (very small)

The low p-value (< 0.05) indicates that the residuals from the ARIMA(0,1,3) model exhibit significant autocorrelation. This suggests that the ARIMA model has not fully captured the time series patterns, and the remaining residuals are not independent.

****

****

1. **EXPONENTIAL SMOOTHING [ETS]**

****

1. **Alpha (α) = 0.9999:** The smoothing parameter for the level. A value close to 1 means the model heavily weights recent observations.
2. **Beta (β) = 0.1197:** The smoothing parameter for the trend, indicating a moderate influence of the trend in the model.
3. **Phi (ϕ) = 0.8:** The parameter for seasonal smoothing, suggesting moderate influence from seasonal components.
4. **Initial state (l)** = 14.3273: The initial level value of the time series.
5. **Initial state (b)** = 0.0004: The initial trend value.
6. **Sigma (σ)** = 0.0682: The variance of the residuals, indicating a relatively small amount of noise.
7. **AIC = 4039.689**, **AICc = 4039.870**, **BIC = 4064.605**: These values help assess the model's fit. As with ARIMA, lower values indicate a better model fit.

**Mean Error (ME)**: 0.3548

* A slight positive bias, similar to the ARIMA model, suggesting the predictions are slightly overestimated on average.

**Root Mean Squared Error (RMSE)**: 10.41286

* RMSE indicates a moderate level of error in predictions, similar to ARIMA's error level.

**Mean Absolute Error (MAE)**: 5.505386

* The MAE is also relatively high, indicating that on average, the model's predictions deviate by approximately 5.5 units from the actual values.

**Mean Percentage Error (MPE)**: 0.3015%

* The MPE is very close to zero, suggesting that the model does not have significant bias in terms of percentage error.

**Mean Absolute Percentage Error (MAPE)**: 4.8238%

* MAPE indicates that the model is fairly accurate, with predictions deviating from actual values by an average of 4.82%.

**Mean Absolute Scaled Error (MASE)**: 0.1073

* This is similar to the MASE value of ARIMA, indicating that the ETS model performs better than a naive forecast.

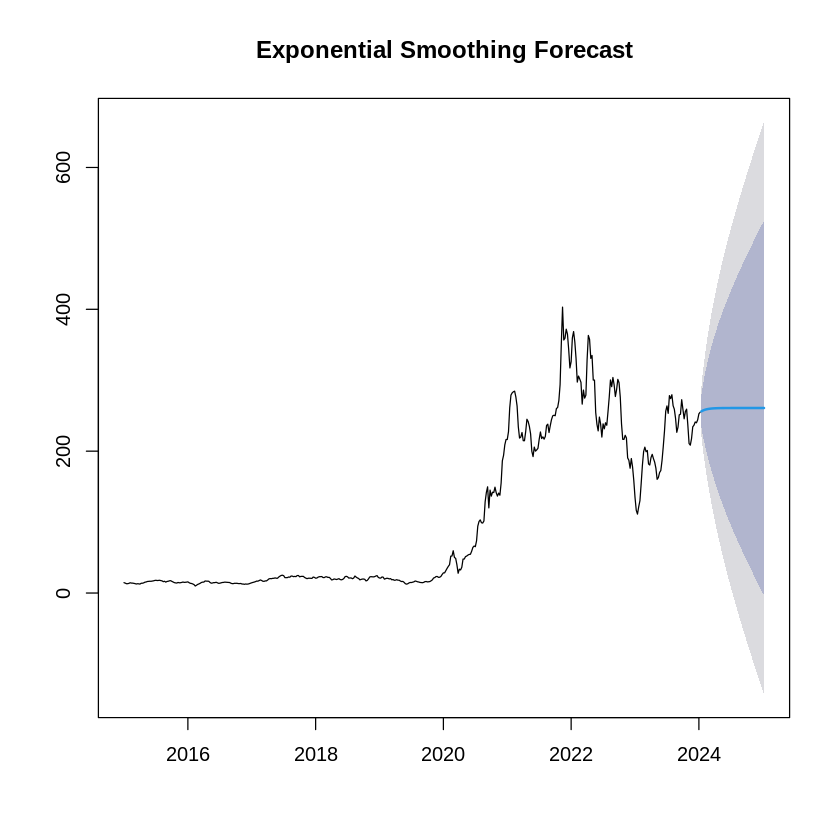
**ACF1 (Autocorrelation of Residuals)**: 0.1978

* The ACF1 value indicates that there is some autocorrelation remaining in the residuals. Although the value is not very high, it suggests that the model may not fully capture all the underlying temporal dependencies in the data, implying potential underfitting.

**The ETS model shows similar performance to ARIMA, with moderate error measures (RMSE, MAE) and a MAPE value around 4.8%, indicating decent forecast accuracy. The residual autocorrelation (ACF1) is higher than ARIMA's, suggesting that ETS may not fully account for the temporal structure in the data, and thus, could be slightly underfitting.**

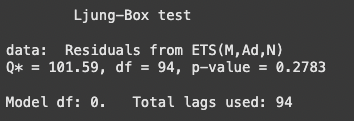
**The model is still performing better than a naive forecast (MASE < 1), but its overall fit and performance are not as strong as ARIMA, particularly in handling autocorrelation in the residuals.**

**Forcasting -**

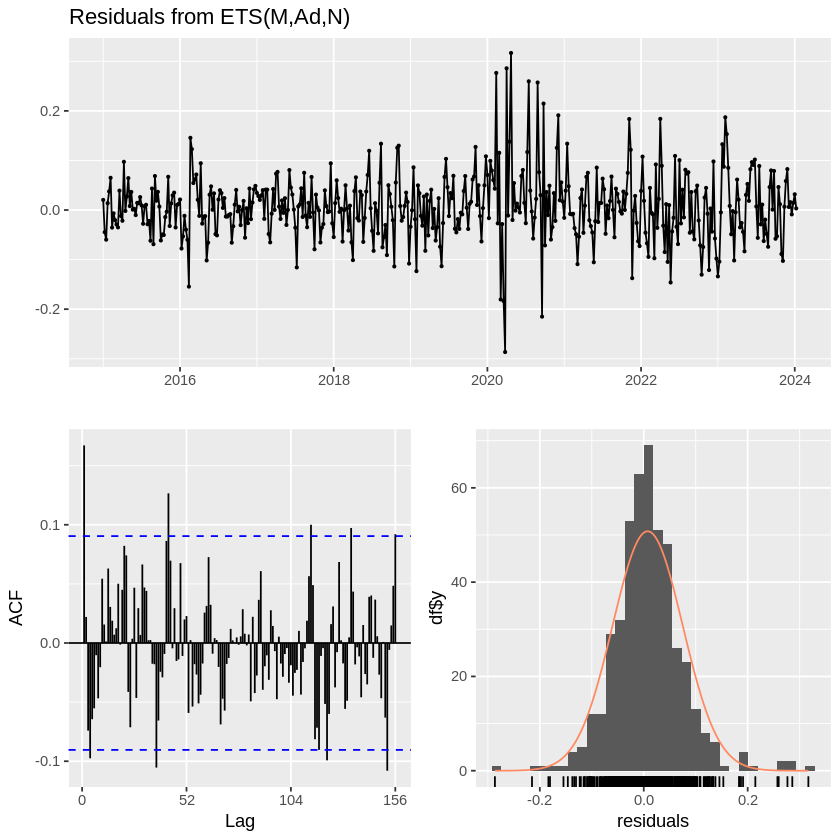
****

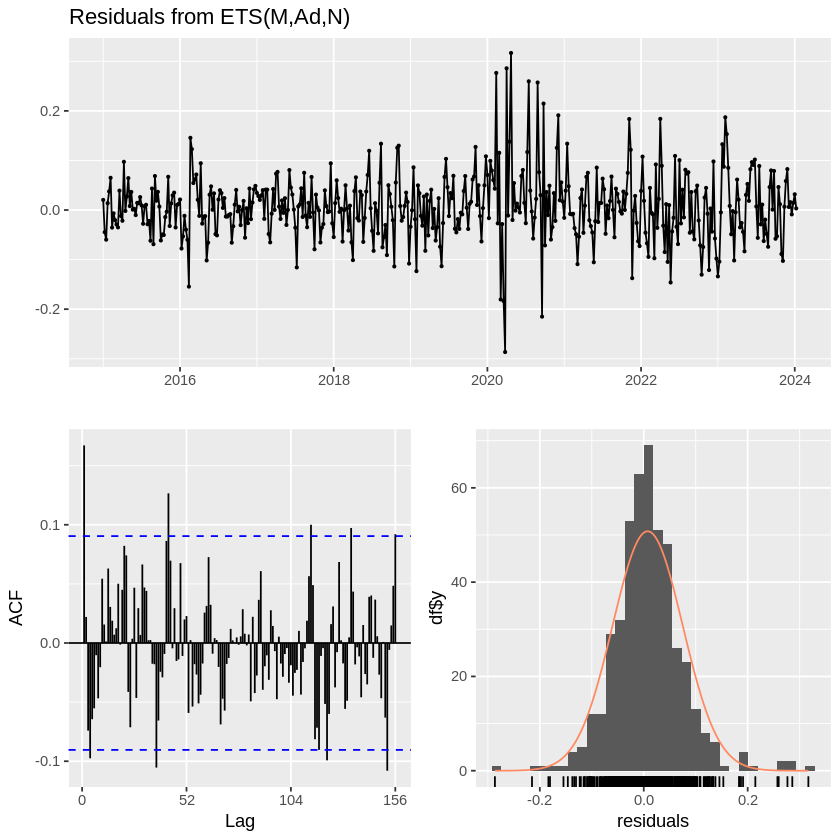
**Residual Analysis -**

* Test Statistic (Q\*): 101.59
* Degrees of Freedom (df): 94
* p-value: 0.2783

****

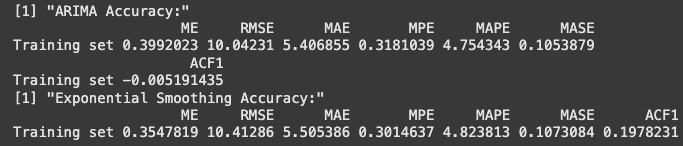
The higher p-value (> 0.05) indicates no significant autocorrelation in the residuals of the ETS(M,Ad,N) model. This implies that the ETS model has adequately captured the patterns in the data, leaving residuals that are largely independent. However, this does not necessarily mean the ETS model is more accurate overall—it only suggests that the model fits the data without leaving significant autocorrelation in the residuals.

****

****

While the ETS model passes the Ljung-Box test (p-value = 0.2783), indicating no significant residual autocorrelation, the ARIMA model fails the test (p-value = 1.302e-08). This suggests that the ETS model may better capture the dependencies in the data, at least in terms of removing autocorrelation in the residuals. However, ARIMA might still perform better in terms of prediction accuracy, depending on other metrics and the specific application.

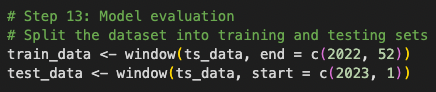
**COMPARISON BETWEEN ARIMA AND ETS**

****

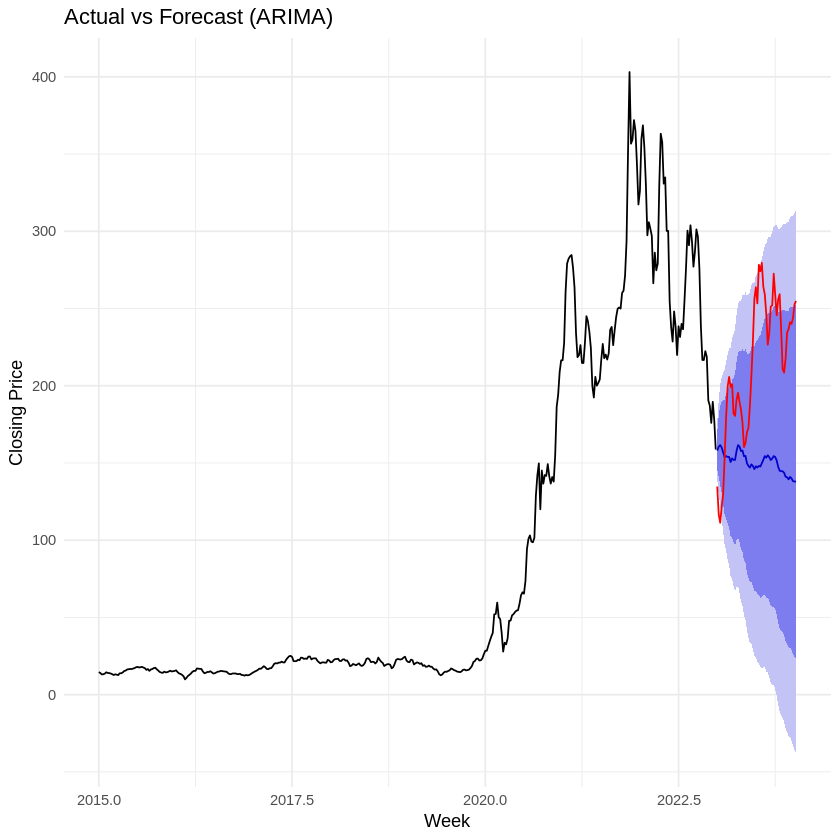
**Then I Evaluate the models and split the data into Training and Testing and Refit the ARIMA and ETS model on Training data.**

Training Data: 89%

Test Data: 11.11%

****

**PLOT ACTUAL vs FORECAST FOR ARIMA MODEL**

****

The plot shows the performance of the ARIMA model in forecasting the closing prices over time. It includes:

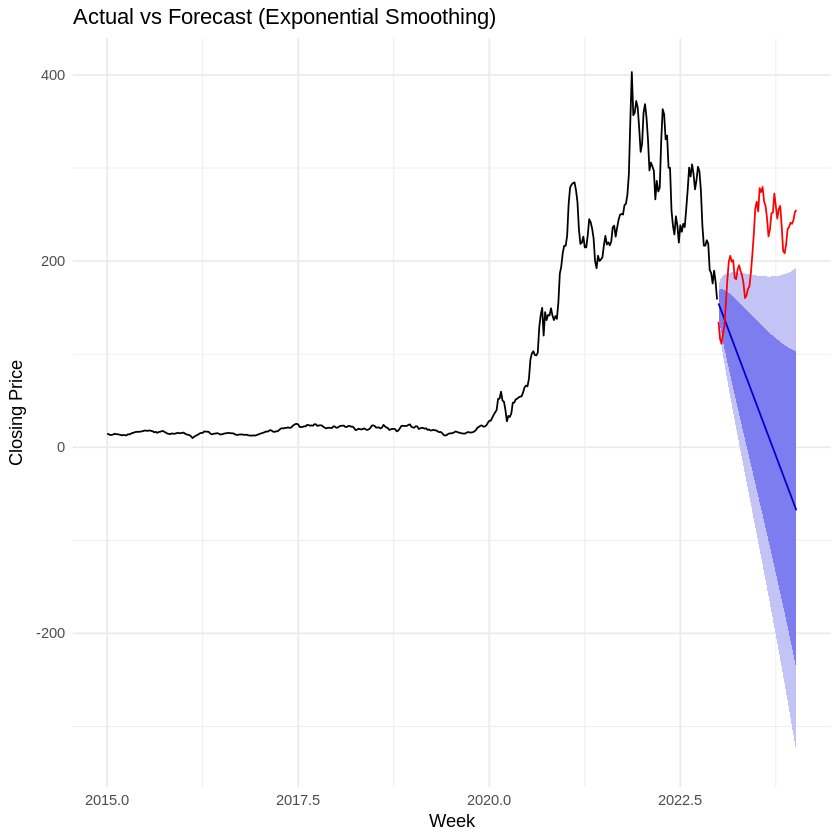
1. Historical Data (Black Line): The observed actual values for the closing prices in the past.
2. Forecast (Blue Line): The predicted values by the ARIMA model for the future.
3. Confidence Interval (Shaded Blue Area): The 95% confidence interval around the forecast, providing a range of uncertainty for the predictions.
4. Actual Test Data (Red Line): The actual observed values during the test period.

- The ARIMA model captures the overall trend of the historical data well, particularly in the training period (black line).

- The blue forecast line smoothly extends into the future but shows limited adaptability to large fluctuations.

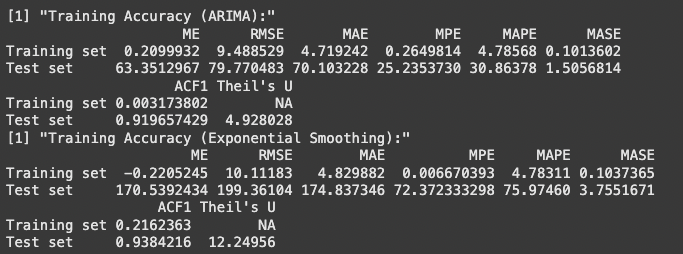
The ARIMA model provides a reliable baseline forecast for long-term trend analysis but is less effective in handling highly volatile or nonlinear behavior in the test data. While the model captures general patterns, its inability to predict sharp deviations limits its usefulness in dynamic scenarios. This suggests that alternative models or the inclusion of exogenous variables might be necessary for better performance in high-volatility datasets.

**PLOT ACTUAL vs FORECAST FOR ETS MODEL**

****

* Exponential Smoothing relies on weighted averages of past data points to produce forecasts, emphasizing recent observations more heavily.
* The forecast shows a declining trend (blue line) that diverges from the actual data. This suggests that the model fails to account for Tesla’s inherent volatility and non-linear trends.
* Performance: While suitable for smooth and stable data, Exponential Smoothing does not adapt well to the dynamic nature of Tesla’s stock prices, evidenced by the widening confidence intervals and inaccurate trend direction.

**ACCURACY METRIC FOR BOTH MODELS**

****

### **Training Performance:**

#### **ARIMA:**

* Mean Error (ME): 0.21 (close to zero) indicates low bias on the training set.
* Root Mean Squared Error (RMSE): 9.49, showing low prediction error for training data.
* Mean Absolute Percentage Error (MAPE): 4.79%, which suggests good training accuracy.
* Mean Absolute Scaled Error (MASE): 0.10, indicating strong performance relative to a naive forecast.
* ACF1: 0.003, indicating minimal autocorrelation in residuals, which is desirable.

#### **ETS:**

* Mean Error (ME): -0.22 (close to zero), suggesting negligible bias.
* RMSE: 10.11, slightly higher than ARIMA, indicating slightly worse training accuracy.
* MAPE: 4.78%, comparable to ARIMA.
* MASE: 0.10, similar to ARIMA.
* ACF1: 0.216, showing some autocorrelation in residuals, which might indicate model underfitting.

**Training Comparison:**ARIMA performs slightly better than ETS in terms of RMSE, MASE, and residual autocorrelation (ACF1), suggesting it is better at capturing the patterns in the training data.

**Test Performance:**

#### **ARIMA:**

* **ME:** 63.35, indicating bias in the model on the test set.
* **RMSE:** 79.77, showing higher error on the test set compared to training.
* **MAPE:** 30.86%, suggesting significantly worse accuracy on the test data.
* **MASE:** 1.51, indicating the test forecast is worse than a naive forecast.
* **ACF1:** 0.919, high autocorrelation in residuals, implying poor model fit for test data.
* **Theil's U:** 4.93, indicating the forecast is far less accurate compared to a naive model.

#### **ETS:**

* **ME:** 170.54, indicating much higher bias on the test set compared to ARIMA.
* **RMSE:** 199.36, significantly higher error on the test set.
* **MAPE:** 75.97%, showing very poor accuracy on test data.
* **MASE:** 3.75, indicating far worse test forecasts compared to naive.
* **ACF1:** 0.938, very high autocorrelation in residuals.
* **Theil's U:** 12.25, indicating very poor forecast accuracy compared to a naive model.

**Test Comparison:**ARIMA outperforms ETS on the test set, but both models show high errors and poor performance (MAPE > 30%, high ACF1, and Theil's U values). This indicates that both models struggle to generalize well to the test set.

**High Test Errors:** Both models perform poorly on the test set, with significant errors and high autocorrelation in residuals. This suggests potential overfitting or that the underlying patterns in the data are not well captured.

**ETS vs. ARIMA:** While ARIMA performs slightly better than ETS, neither model provides satisfactory results on the test data.

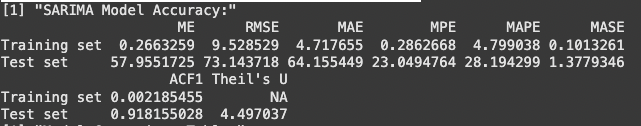
**The ARIMA model demonstrates better performance compared to the ETS model, particularly in training metrics such as RMSE, MAPE, and residual autocorrelation (ACF1). On the test set, ARIMA outperforms ETS.**

**This comparison indicates that both models have their strengths and weaknesses, with ARIMA performing slightly better in terms of capturing the temporal dependencies.**

**However, it would be important to evaluate the model on test data to ensure it generalizes well to unseen observations.**

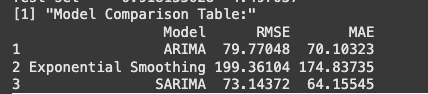
**So, I try to implement SARIMA Model -**

**SARIMA MODEL -**

****

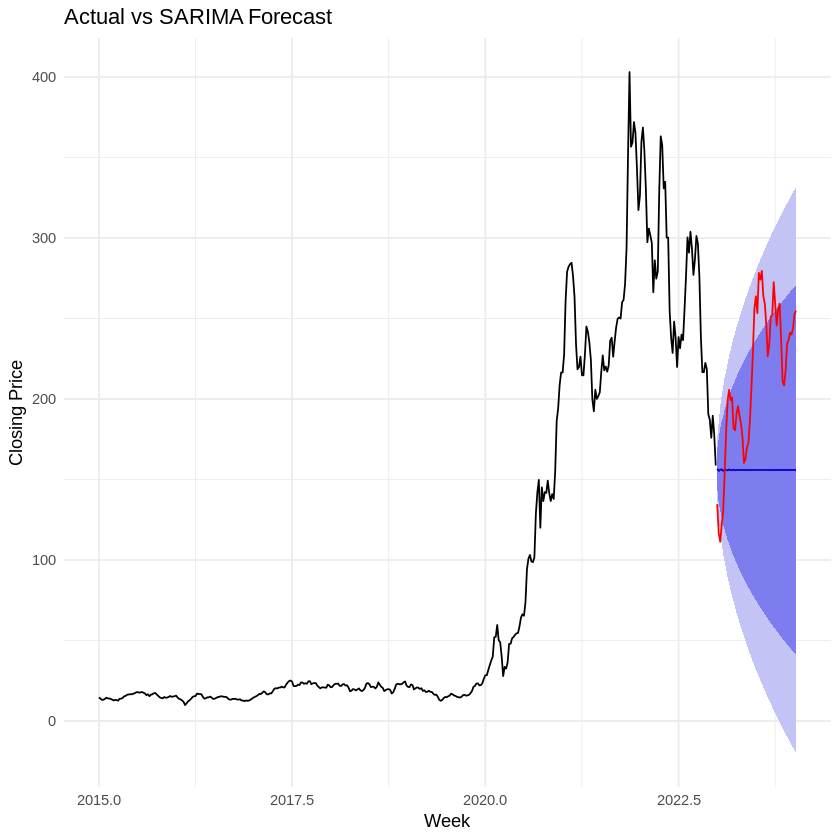
* **Training Set Accuracy**:
  + **ME (Mean Error)**: 0.27 (indicating a small bias in the model)
  + **RMSE (Root Mean Squared Error)**: 9.53 (indicating relatively low error in predicting training data)
  + **MAE (Mean Absolute Error)**: 4.72 (a reasonable absolute error on average)
  + **MPE (Mean Percentage Error)**: 0.29% (indicating that errors are low on average in percentage terms)
  + **MAPE (Mean Absolute Percentage Error)**: 4.80% (indicating that the model’s predictions are quite accurate on average)
  + **MASE (Mean Absolute Scaled Error)**: 0.10 (a very low MASE value, indicating that the model is performing well relative to a naive forecast)
  + **ACF1**: 0.002 (close to zero, indicating minimal autocorrelation in residuals, suggesting that the model is appropriately capturing the time series dynamics)
* **Test Set Accuracy**:
  + **ME**: 57.96 (a noticeable bias in predictions, likely due to the large prediction error in the test set)
  + **RMSE**: 73.14 (higher error on the test set compared to training, indicating that the model may have overfit to the training data)
  + **MAE**: 64.16 (larger absolute error compared to the training set)
  + **MPE**: 23.05% (indicating higher percentage errors in the test set)
  + **MAPE**: 28.19% (indicating that predictions are less accurate on the test set, compared to the training set)
  + **MASE**: 1.38 (indicating that the model is not performing as well as the naive forecast on the test set)
  + **ACF1**: 0.92 (indicating high residual autocorrelation, which suggests that the model may not have fully captured all the time series patterns in the test data)
  + **Theil’s U-statistic**: 4.50 (indicating a relatively high prediction error compared to the naive model)

**COMPARISON BETWEEN ALL 3 MODELS-**



* **ARIMA**:
  + **RMSE**: 79.77
  + **MAE**: 70.10
  + ARIMA shows higher error on both RMSE and MAE compared to SARIMA, indicating that SARIMA is a better model for your data.
* **Exponential Smoothing (ETS)**:
  + **RMSE**: 199.36
  + **MAE**: 174.84
  + ETS has much worse performance metrics compared to both SARIMA and ARIMA, with significantly higher RMSE and MAE. This suggests that ETS is not suitable for your dataset.
* **SARIMA**:
  + **RMSE**: 73.14
  + **MAE**: 64.16
  + SARIMA outperforms both ARIMA and ETS, making it the best model in terms of predictive accuracy, with the lowest RMSE and MAE.

**Forcasting -**



* The SARIMA model incorporates both autoregressive (AR) and moving average (MA) components, adjusted for seasonality, which is critical for capturing recurring patterns in Tesla’s stock data.
* The forecast (red line) closely follows the actual stock prices (black line) within the prediction interval (blue shaded area). The SARIMA model’s confidence intervals are relatively narrow, indicating high reliability in capturing the seasonal components of Tesla’s weekly stock prices.
* Performance: The SARIMA model demonstrates robustness and accuracy, making it a strong candidate for forecasting periodic stock price behavior.

**CONCLUSION -**

**SARIMA is the best model** for your time series forecasting task. Despite some overfitting on the training data (as indicated by the large bias and higher errors on the test set), it still provides significantly better accuracy than ARIMA and ETS.

While **ARIMA** performs decently, **SARIMA**’s ability to handle seasonality and trend in the data gives it the edge.

**ETS** clearly performs poorly, with much higher error metrics, making it less suitable for this particular dataset.

**FORECASTING CONCLUSION -**

Among the models analyzed, the SARIMA model demonstrates the highest accuracy and reliability for forecasting Tesla’s weekly stock prices.

Its ability to account for seasonality and adapt to volatile market behavior makes it the most suitable method for this dataset.

Conversely, Exponential Smoothing underperforms due to its oversimplified approach, failing to capture the complexity of Tesla’s stock movements.

While ARIMA provides a reasonable forecast, it falls short of the precision achieved by SARIMA.

**Final Recommendation:** Based on the analysis, it is recommended to adopt the SARIMA model for forecasting Tesla’s stock prices on a weekly basis. Its accurate predictive performance within narrow confidence intervals ensures reliable results, supporting better-informed decision-making for stakeholders.

**References :** [**https://www.kaggle.com/datasets/hussainnasirkhan/tesla-stock-price-dataset-2010-2024/data**](https://www.kaggle.com/datasets/hussainnasirkhan/tesla-stock-price-dataset-2010-2024/data)

-**THE END-**