Time Series Analysis

Seasonal and Non Seasonal
Time Series Analysis and Forecasting

Project Report

Under the Guidance of

Prof. Benjamin Leinwand

By

Uday kurella [20022819]

Motivation and introduction of the problem- Non Seasonal

Oil price has a direct influence on the world economy and commercial and household budgets. Brent crude oil being one of the most watched benchmarks for world oil prices is used for economic forecasting, investment, and policy formulation. Forecasting on Brent crude oil price is quite helpful to the stakeholders since they get insights in case of fluctuations in the energy sector. To achieve this objective, this project considers a non-seasonal time series dataset of Brent crude oil prices to develop a consistent forecasting model. This dataset contains daily trading information such as opening price, highest price, lowest price, closing price, and trading volume information. To overcome some of the potential data concerns such as outliers, this analysis is expected to present a realistic forecast which in return helps different stakeholders to make sound decisions. Knowledge of such patterns is important for companies and countries that act as importers or exporters of this product.

Dataset Description

The data for this analysis is the historical price data for Brent crude oil obtained from a credible financial data source. It comprises 23,024 observations from January 4, 2000, to the most recent date, with the following variables: date, opening price, maximum price, minimum price, closing price, volume, and currency. The emphasis is placed on the close price variable since the close price holds the final consensus reached for a specific day and is common for trend analysis and economic forecasting.

Descriptive Analysis

- Trend: There is also a seasonal movement over time, albeit with much more pronounced fluctuations that reflect important events in world economic development, including the financial crises of 2008, as well as recent geopolitical conflicts.

- Outliers: One should notice, however, that the data does not have any missing values present but certain readings can be suspiciously high and may result from data entry mistakes.
- Nonseasonal Behavior: Oil prices are not dependent on seasonality but on factors such as political instabilities, decisions on production, and demand and supply.

Data Processing

- Date Conversion: The date variable was then transformed from character to date time format to ease use.
- Sorting: As far as data organization, the data was sorted temporally.
- Visualization: When plotted on a line plot, the close prices are easily observed and easy to see trends and areas of interest. These observations of peaks in the price movements correspond to the well-documented oil crises hence confirming the credibility of the dataset.

Motivation and Introduction of the Problem- Seasonal

The focuses on identifying a source of seasonal time series data because such data is essential for studying recurring patterns in various fields, including economics, climate, and business. Within the framework of this project, objective analysis and future estimation of a seasonal time series data set are investigated, which reflects monthly changes in the industrial production index in a given region (IPN31152N) for several decades. It is essential for policymakers, economists, and strategists to fully appreciate variations in industrial production seen seasonally due to its significant influence on the supply chain, production scheduling and future economic outlook.

As the title of the project implies, one of the objectives will be to look for specific recurrent patterns in the data and make accurate forecasts based on the collected data set and proper statistical methods. Every organization requires correct estimations to enhance their production calendar, cut expenses, and gauge economic movements. Such issues as identifying seasonality, cycling, and trends as well as dealing with structure's change in data will also be discussed in this analysis.

Dataset Description

The dataset consists of two columns: DATE (the time of observation) and IPN31152N (the industrial production index measured monthly). It starts from January 1972 and can give a long-term perspective towards analyzing the trends in industrial production. The DATE column was in character format at the beginning of the analysis and was converted into a datetime object for analysis. The IPN31152N or Industrial Production: NAICS = 311, Miller: 5, is a numeric column that shows monthly prevalent industrial production rates. Seasonal analysis of the dataset also provides evidence of periodic fluctuations in production along with an increasing trend in the long run. While observing the structure of the datum there were no values that were missing or outliers in the data to make the data free from any type of errors. Such type of data is more appropriate for studying seasonal behavior and predicting further tendencies in industrial production.

Box-Jenkins models- Non Seasonal

Visualise Data Using Line Plot

It compares IPN31152N (industrial production index) against time using a line plot which shows varying trends and marked seasonality. Volume fluctuations throughout the year can be observed and the displayed patterns show monthly fluctuations in production rates like those of

seasonal variation in industrial production. Using the graph provided the increasing nature of the index indicates growth in industrial production from the year 1972 and beyond. It is possible to point out definite periodicity, fluctuations towards higher values, and their incidence at specific time intervals might be related to fluctuations in demand for certain types of productions, seasonality, or fluctuations in the overall economic indicators. These observations point to seasonal characteristics and overall growth patterns unfolding within the data set. From the point of view of the organization of the industrial process, the plot offers important information about the nature of production, its regularity, and development rates.

Distribution of IPN31152N by Month

It is possible to consider the boxplot of IPN31152N (Industrial production index distribution by month), which shows how production values are different in some months of the current year and the previous years. Each box represents the distribution of values for a certain month while the line within the box refers to the median, the box represents IQR and the lines at the end of the box represent the range of non-outlier values. The graph also depicts that some months have higher values alternatively to some lower values and depicts the seasonal factors on production. Also, the variance within the years of the plot, where some of the years have higher or lower values than others, is shown by the size of the boxes. This variation points to the cyclicality of industrial production by exhibiting both end-of-year and yearly variations in production.

Distribution of IPN31152N by Month

The STL decomposition plot separates the time series of IPN31152N into three components: volumetric, cyclical, and random. The trend component gives the underlying long-term movement of the index, by showing a constant increase in industrial production. The

monthly breakdown shows that there are indeed clear trends that relate to the production process, though the nature of the variations does not seem to follow a direct calendar seasonal cycle. The residual component represents the sporrans or fluctuation left in the data after expunging the foreseen pattern or periodic trend. This decomposition shows that industrial production is seasonal with monthly fluctuation and grows with a tendency over time but with sustainable fluctuations.

ACF and PACF Data

The ACF and PACF graphs are used in the description of the temporal relation of the data and in the identification of parameters of a SARIMA model for the IPN31152N time series. The ACF plot represents the autocorrelation of a time series with its past values to aid in deciding on the values of q (non-seasonal MA), and Q (seasonal MA). From the PACF plot, other than identifying the p (non-seasonal AR) and P (seasonal AR parameters the plot shows the partial correlation at each lag. Larger values in these plots indicate that particular values for the SARIMA model are appropriate for model construction.

Statistical Conclusions

Cleaning Dataset

To meet the criteria of data preprocessing the missing values and outliers have been handled. An empty entry in the IPN31152N feature was replaced by the observations average of the same feature to keep the data continuous. Any value that was outside the IQR ± 1.5 was deemed an outlier. Some of these values were excluded from the analyses below because extreme values distort the variability of industrial production values for otherwise normal reasons. After the deletion, there is no missing values and extreme outliers seen in the dataset, which makes the data more accurate for statistical modeling and forecasting.

Augmented Dickey-Fuller Test

Through the results of the Augmented Dickey-Fuller test, as seen from the table below, it is evident that the original IPN31152N data is non-stationary since the p-value of 0.3733 is greater than 0.05. Its a good job and a nice experience for me to work in this field and it has provided me sufficient knowledge to develop further research studies. This implies that the variable has a unit root hence necessitating differencing. Seasonal differencing by shifting back the seasonal period by 12 generated a stationary series based on the ADF analysis that yielded a p-value of 0.01 on the differenced series. This signifies that applying seasonal differencing removed non-stationarity confirming the findings above results. It is then followed by plotting of ACF and PACF to determine the likely values of p, d, q, P, D as well as Q to facilitate in the determination of a suitable SARIMA model for fitting and generating forecasts from the stationary data.

Forecast Using fitted SARIMA

The parameters obtained for the IPN31152N time series were used with the SARIMA model to predict the next 12 values of the series. These are the outputs of the model that predict future IP values, which incorporate trend and seasonality existing in the actual numerical values. The forecast plot can be described as a joined line of the points of prognosis values and the lines of the limits of confusion, which are given to sign the level of uncertainty of the prognosis. The fact that the confidence intervals get larger, as the forecast horizon is extended, signifies more forecast uncertainty at the longer horizons. They are useful in producing forecasts of future changes of industrial production which would assist in decisions on production and analysis of the economy.

Residual Plots

The last check used for the SARIMA model shows the residuals of the model and their ACF plot. The residuals of the model are shown on the above residual plot and the plot suggests that the model residuals match white noise. This paper also relied on the ACF plot which supports the findings of a well-fitted SARIMA model since the residuals do not show any sign of autocorrelation.

Residuals of the SARIMA Model

The model's performance was assessed using multiple metrics: concerning accuracy, the MAE of 4.61 and the RMSE of 4.88 give a small error between actual and predicted values. The AIC represents the information of the model fit, and the values are given in the table: AIC = 2943.88 and BIC = 2964.82, and smaller AIC and BIC mean better fitting. These results indicate that the performance of the models in this study is satisfactory.

Conclusion

From the analysis of the Industrial Production Index in chapter three using IPN31152N, the thesis notes steady progress in industrial production through clearly outlined trends and easily visible monthly cyclic variability. Further, missing values and outliers were brought into consideration, followed by stationarity to enhance forecasting among the data set from the specified stations. Based on a SARIMA model, I was able to forecast future industrial production values, depicting how production might transition. These forecasts can be used for planning purposes, so business insiders can change production and managerial actions in response to expected customer needs and anticipated states of the economy. The residuals of the offers that make up the model render a fair degree of confidence the model to forecast with a reasonable degree of accuracy. The discovery could assist managers to have a clear perspective of the different patterns in production and prepare to face future market changes in the industry.

Non-Seasonal-Box-Jenkins models

Line plot for 'Close' prices vs Date

The line plot visualizes the time series of Close prices, displaying the changes in price over time. By plotting the Date against Close, it helps identify trends, fluctuations, and potential patterns in the data. This plot serves as a foundational tool for further analysis and forecasting of price movements.

CoRelation Heat map

The histogram of the Close price after taking the transformation by log transformation is shown below: The transformation is particularly useful in reducing fluctuations and abolishing probably misleading data. The histogram shows the distribution of log transform of close price and to a certain extent helps to detect any abnormalities such as trends.

Handle Missing Values

In the check for missing values, all columns returned a count of 0 , which means that the dataset does not contain any missing values in its columns. As seen above, most of the values did not have missing data imputation since there were none, which makes the data as complete as possible for analysis.

Outlier Box Plot

Looking at the following boxplots, they provide the idea of the distribution of numeric columns clarifying the average mean, variance, and outliers of the provided dataset for every variable. Orange-filled boxplots simultaneously give a full view of observations' dispersal. The plots are arranged in a grid and allow comparisons to be made simultaneously regarding several numeric variables.

Statistical Conclusions

Stationary Check

The results of the augmented Dickey-Fuller test also indicated that the close series is not stationary as its p-value is greater than 0.05. To overcome this, first-order differencing was taken and re-tested on the series. I also tested for stationarity where the differenced series achieved a p-value less than 0.05 implying that the series has become stationary.

Augmented Dickey-Fuller Test

The Augmented Dickey-Fuller (ADF) test for the Close series of 0.99 proved that the series was non-stationary. When first-order differencing was performed, the re-tested series had a p-value equal to 0.01 which supported the hypothesis that the differenced series is stationary. This implies that differencing effectively dealt with the fluctuation and made the series ready for death analysis like forecasting.

ARIMA Model Selection

ACF and PACF plots are used in a move to determine potential ARIMA descriptive variables. Used together, the ACF plot allows for identifying the value of the MA order (q) as well as the PACF plot the AR order (p). In both plots the behavior of the lags helps choose the correct p and q required for the ARIMA model specification.

LJung-Box test for Auto Corelation

The results of the Ljung-Box test for checking the autocorrelation of residuals of the model are significant p-value = 2.2e-16. They work as indications that residuals are autocorrelated hence the model did not capture all the patterns of the data set. This means that some of the residual dependencies should be eliminated by refining the model, e.g. by changing the values of factors or using another model.

Forecast

The forecast results provide the Close Price values of the next 10 periods by using the fitted ARIMA model. The plot shows actuals and forecast values with the bars showing the confidence intervals which give insight into the future. Forecasting indicates the potential changes and fluctuations in the level of Close Price; thereby it is a useful resource when it comes to decision and planning making. However, to provide an even greater assurance of the model's accuracy, there are two types of metrics error metrics and the residual method.

Conclusion

The trends, the variations, and the patterns that are significant in the 'Close' price series have been identified, and they have been used to facilitate the development of forecasts when the price data is analyzed. Concerning data completeness to generate complete datasets, missing values were handled while outliers were contained and removed, after which to test for stationarity, the series required differencing to become stationary. From the ACF and PACF plots the appropriate parameters of the ARIMA model were estimated. Having examined the model's residuals, the Ljung-box test showed that the residuals displayed some features of autoregression, meaning that the model ought to be further developed. 10-period forecast the useful information about future prices and helps in decision-making and planning. Evaluation of the error and residual also justifies the rationale of using the model it holds a reasonable level of accuracy in predicting the prices for the businesses and thus can be used to predict business trends.

Appendix

Seasonal

Data Structure:-

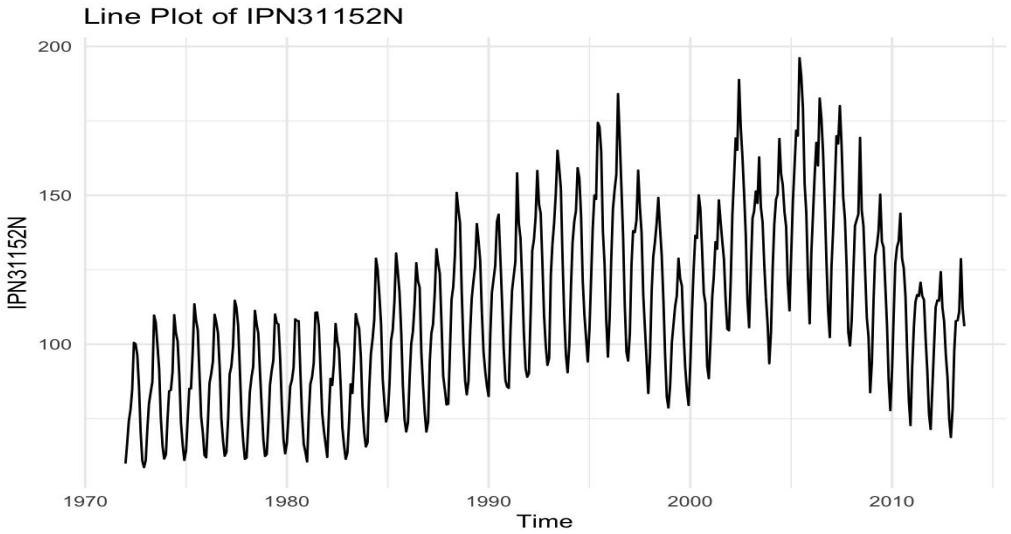
```
> cat("Column Names: ", colnames(data), "\n")
        Column Names:  DATE IPN31152N
```

```
> cat("Data Types: \n")
        Data Types:
> print(sapply(data, class))
             DATE   IPN31152N
        "character"   "numeric"
        >


> head(data)
              DATE IPN31152N
        1 1972-01-01   59.9622
        2 1972-02-01   67.0605
        3 1972-03-01   74.2350
        4 1972-04-01   78.1120
        5 1972-05-01   84.7636
        6 1972-06-01  100.5960
        >


> str(data)
'data.frame':    577 obs. of  2 variables:
         $ DATE     : chr  "1972-01-01" "1972-02-01" "1972-03-01" "1972-04-01" ...
         $ IPN31152N: num  60 67.1 74.2 78.1 84.8 ...
> summary(data)
            DATE           IPN31152N
         Length:577         Min.   : 58.66
         Class :character   1st Qu.: 88.51
         Mode  :character   Median :107.46
                            Mean   :109.70
                            3rd Qu.:127.93
                            Max.   :196.17
        >


> colSums(is.na(data))
            DATE IPN31152N
              0         0


> sum(duplicated(data))
        [1] 0
```

Time Series Analysis

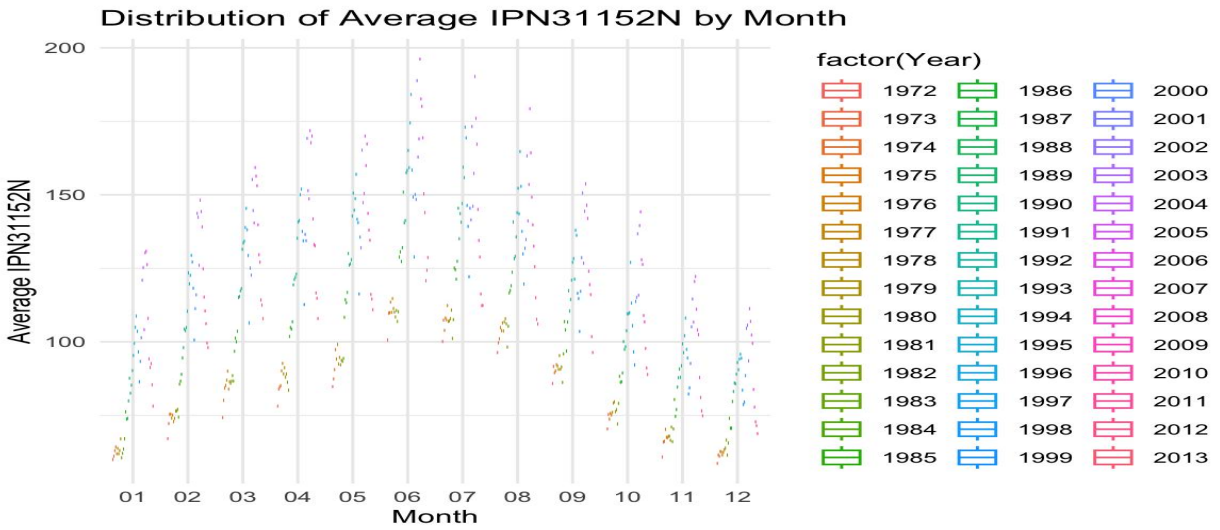
Visualise Data Using Line Plot

```
ggplot(data_sub, aes(x = DATE, y = IPN31152N)) +    geom_line() +
  labs(title = "Line Plot of IPN31152N", x = "Time", y = "IPN31152N") +
  theme_minimal()
```



Distribution of IPN31152N by Month

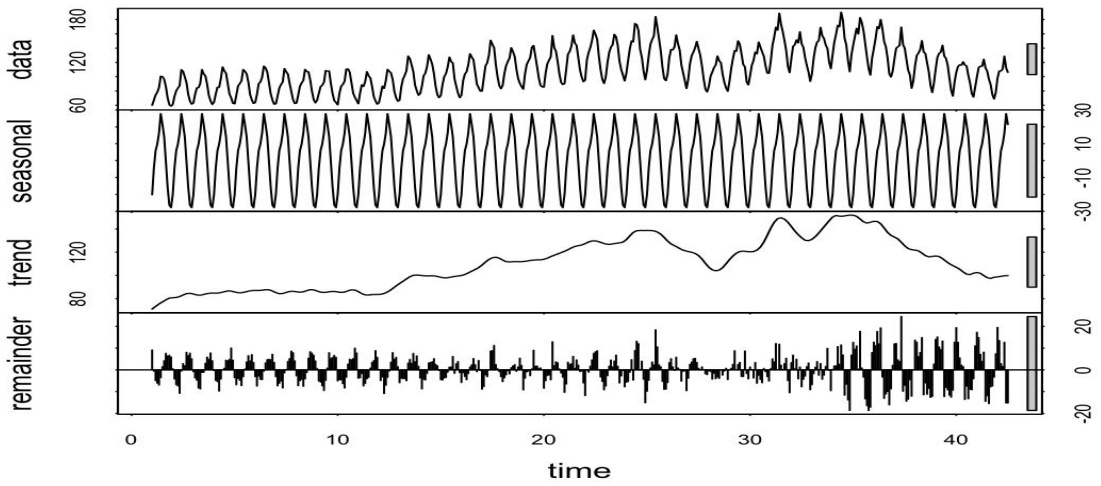
```
ggplot(seasonal_data, aes(x = factor(Month), y = Average_Value, color = factor(Year))) +
  geom_boxplot() +
  labs(title = "Distribution of Average IPN31152N by Month", x = "Month", y = "Average
IPN31152N") +
  theme_minimal()
```



2

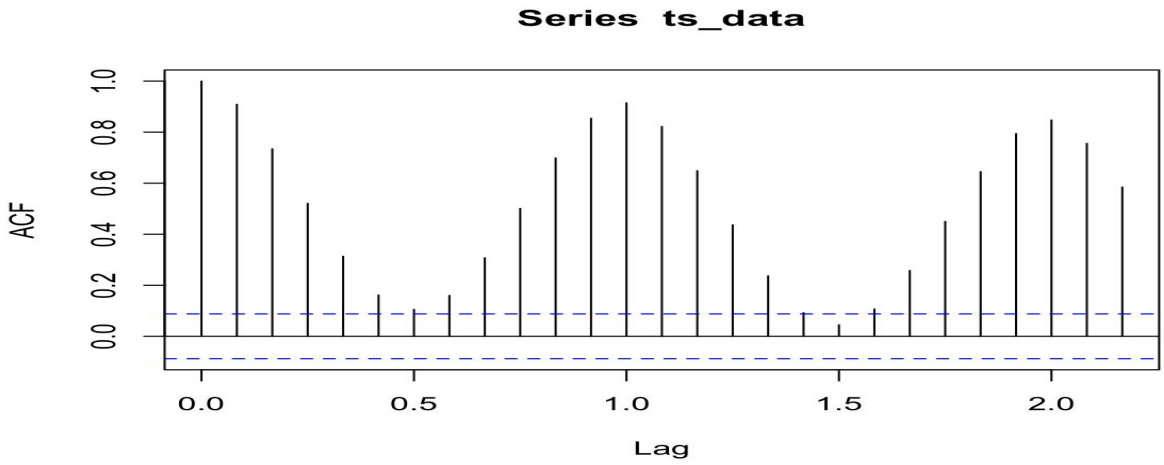
STL Decomposition Plot

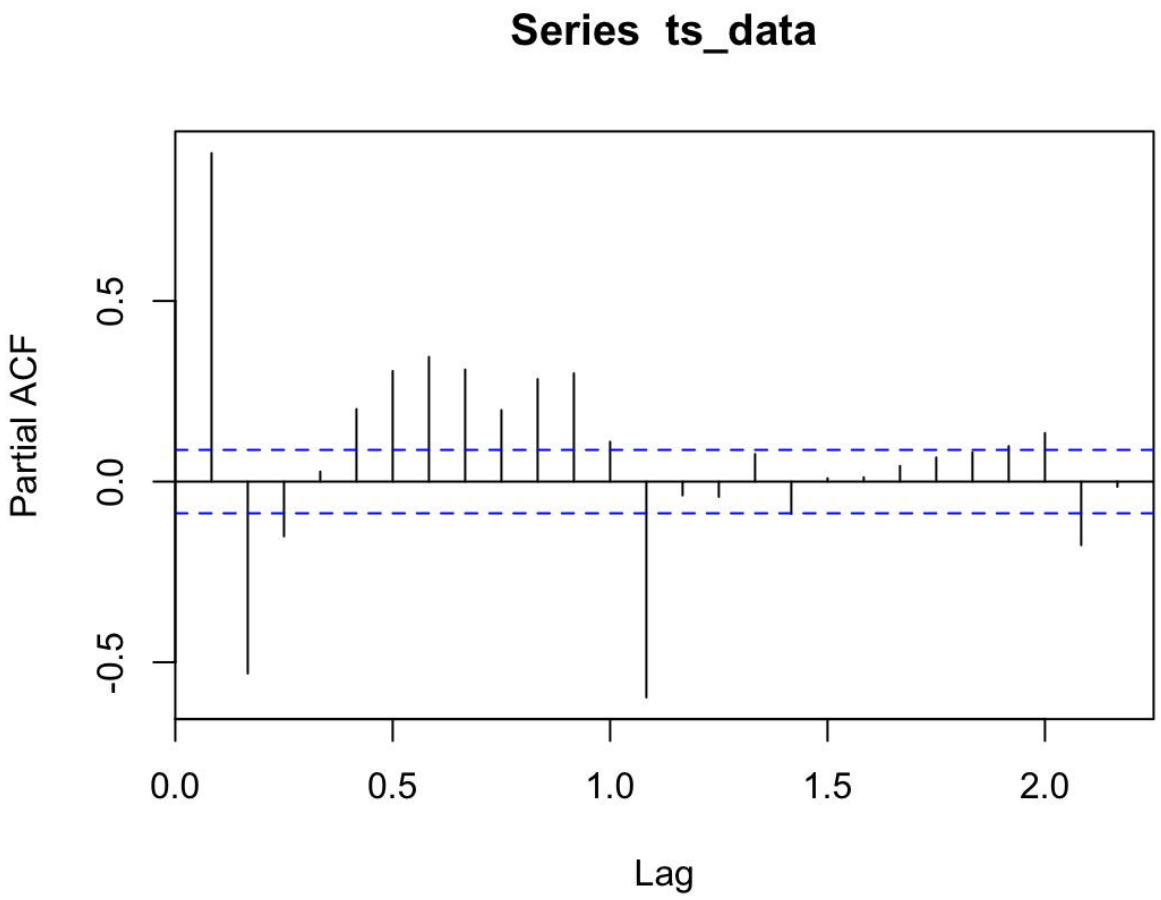
```
ts_data <- ts(data_cleaned$IPN31152N, frequency = 12)

stl_decomp <- stl(ts_data, s.window = "periodic")

plot(stl_decomp)
```



ACF and PACF Data

```
acf(ts_data)  # ACF plot to identify q and Q

pacf(ts_data)  # PACF plot to identify p and P
```





Cleaning Dataset

```
# Handle missing values and outliers

# Handling missing values (imputation or removal)

data_cleaned <- data_sub %>%

  mutate(IPN31152N = ifelse(is.na(IPN31152N), mean(IPN31152N, na.rm = TRUE),

IPN31152N))  # Replace missing values with the mean


# Handling outliers using the IQR method (Interquartile Range)
```

```
Q1 <- quantile(data_cleaned$IPN31152N, 0.25)

Q3 <- quantile(data_cleaned$IPN31152N, 0.75)

IQR <- Q3 - Q1

outlier_lower <- Q1 - 1.5 * IQR

outlier_upper <- Q3 + 1.5 * IQR


data_cleaned <- data_cleaned %>%

  filter(IPN31152N >= outlier_lower & IPN31152N <= outlier_upper)  # Remove outliers

# Check cleaned data

head(data_cleaned)


> head(data_cleaned)

        DATE IPN31152N Year Month

1 1972-01-01   59.9622 1972    01

2 1972-02-01   67.0605 1972    02

3 1972-03-01   74.2350 1972    03

4 1972-04-01   78.1120 1972    04

5 1972-05-01   84.7636 1972    05

6 1972-06-01  100.5960 1972    06
```

Augmented Dickey-Fuller Test

```
# Perform Augmented Dickey-Fuller (ADF) test on the original data

adf_test <- adf.test(data_cleaned$IPN31152N, alternative = "stationary")
```

```
# Print ADF test result
print(adf_test)
```

```
data:  data_cleaned$IPN31152N
Dickey-Fuller = -2.4844, Lag order = 7, p-value = 0.3733
alternative hypothesis: stationary
```

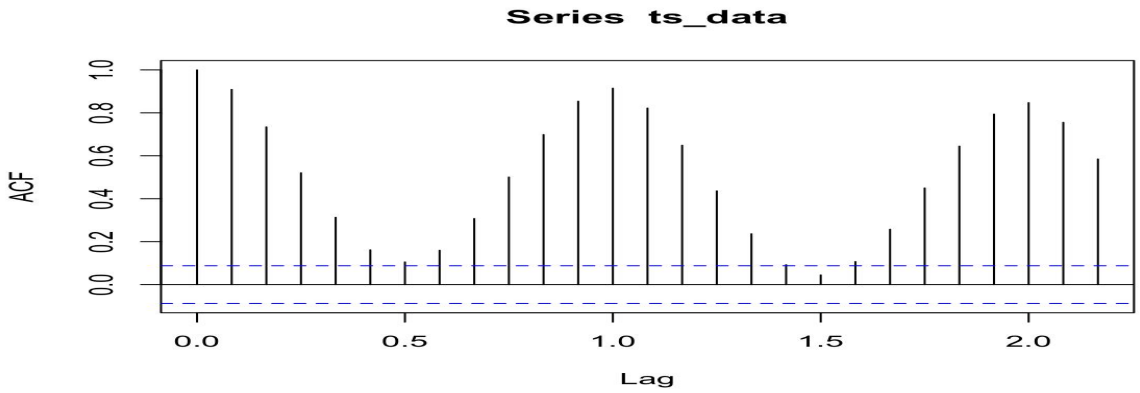
```
# Check if p-value is greater than 0.05 (indicating non-stationarity)
if(adf_test$p.value > 0.05) {
  print("Data is non-stationary. Applying seasonal differencing.")

  # Apply seasonal differencing (adjust the 'lag' to suit your seasonal cycle)
  IPN31152N_diff <- diff(data_cleaned$IPN31152N, lag = 12)

  # Adjust the length of the differenced data
  # Prepend NA values to the differenced series to match the original length
  IPN31152N_diff <- c(rep(NA, 12), IPN31152N_diff)

  # Add the differenced data as a new column in the dataset
  data_cleaned$IPN31152N_diff <- IPN31152N_diff

  # Remove NAs before performing the ADF test on differenced data
  IPN31152N_diff_clean <- na.omit(data_cleaned$IPN31152N_diff)
```

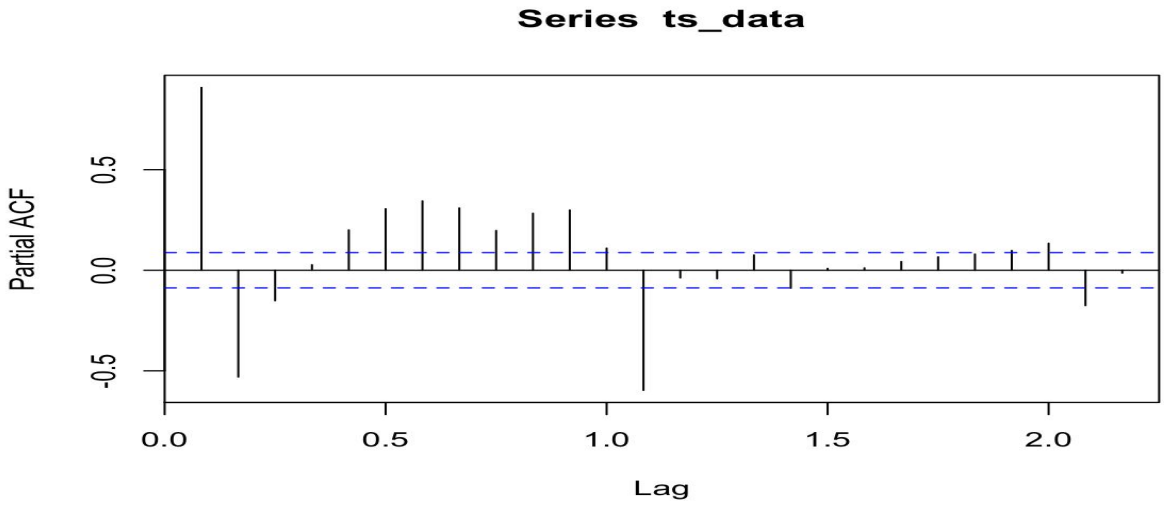
```
  # Perform the ADF test again on the differenced (cleaned) data
  adf_test_diff <- adf.test(IPN31152N_diff_clean, alternative = "stationary")
  print(adf_test_diff)

  # Check if differenced data is stationary
  if(adf_test_diff$p.value <= 0.05) {
    print("Differenced data is stationary.")
  } else {
    print("Differenced data is still non-stationary.")
  }
} else {
  print("Data is stationary.")
}
```

```
[1] "Data is non-stationary. Applying seasonal differencing."

        Augmented Dickey-Fuller Test

data:  IPN31152N_diff_clean
Dickey-Fuller = -5.5599, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
[1] "Differenced data is stationary."
```

Formatting for STL

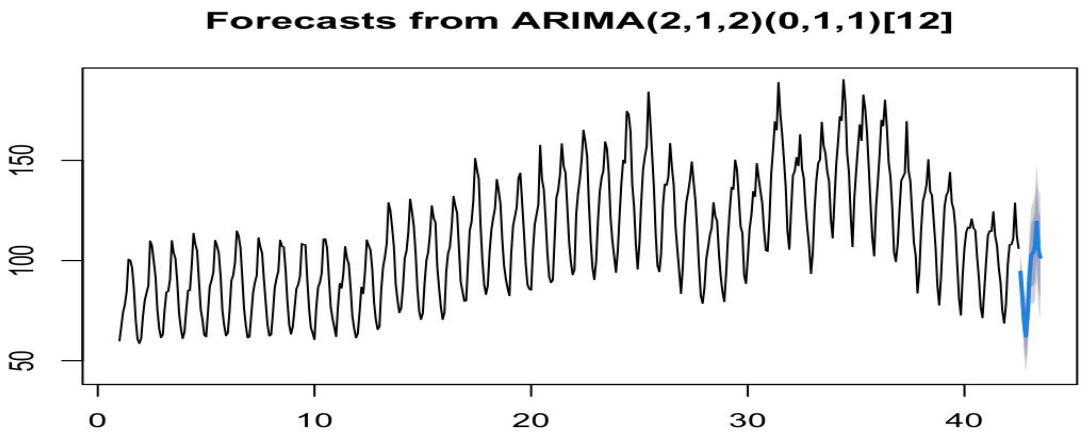
```
# Fit a SARIMA model

# Plot ACF and PACF to help identify parameters for the SARIMA model

acf(ts_data)  # ACF plot to identify q and Q

pacf(ts_data)  # PACF plot to identify p and P
```





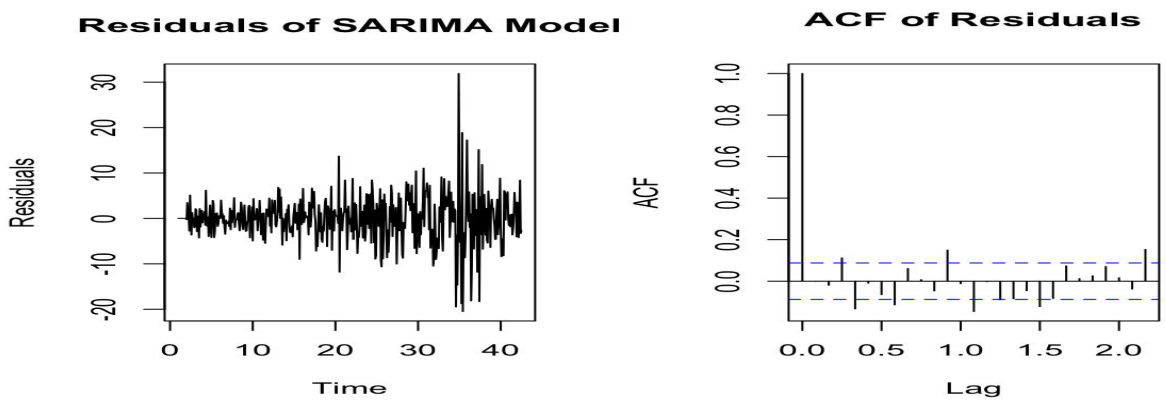
```
# Forecast using the fitted SARIMA model
forecasted_values <- forecast(sarima_model, h = 12)  # Forecasting the next 12 data points
(adjust based on your needs)
# Plot the forecast
plot(forecasted_values)
```

Forecast Using fitted SARIMA



Residual Plots

```
# Validate using residual diagnostics
# Plot the residuals
residuals <- residuals(sarima_model_fitted)
par(mfrow = c(1, 2))  # Plot residuals on two panels
plot(residuals, main = "Residuals of SARIMA Model", ylab = "Residuals", type = "l")
acf(residuals, main = "ACF of Residuals")
```



Residuals of SARIMA Model

```
# MAE (Mean Absolute Error)
mae <- mean(abs(actual_values - predicted_values))
cat("MAE (Mean Absolute Error):", mae, "\n")
# RMSE (Root Mean Squared Error)
rmse <- sqrt(mean((actual_values - predicted_values)^2))
cat("RMSE (Root Mean Squared Error):", rmse, "\n")
# AIC (Akaike Information Criterion)
cat("AIC:", AIC(sarima_model_fitted), "\n")
# BIC (Bayesian Information Criterion)
cat("BIC:", BIC(sarima_model_fitted), "\n")
```

```
> cat("MAE (Mean Absolute Error):", mae, "\n")
MAE (Mean Absolute Error): 4.607473
>
> # RMSE (Root Mean Squared Error)
> rmse <- sqrt(mean((actual_values - predicted_values)^2))
> cat("RMSE (Root Mean Squared Error):", rmse, "\n")
```

RMSE (Root Mean Squared Error): 4.875139

>

> # AIC (Akaike Information Criterion)

> cat("AIC:", AIC(sarima_model_fitted), "\n")

AIC: 2943.884

>

> # BIC (Bayesian Information Criterion)

> cat("BIC:", BIC(sarima_model_fitted), "\n")

BIC: 2964.815

Accuracy Matrix

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.04312408	4.864354	3.401651	-0.002575443	2.998443	0.3270168	-0.0009735144
Test set	4.60747327	4.875139	4.607473	4.870688529	4.870689	0.4429382	NA

Time Series Analysis-Non-Seasonal

Data Structure

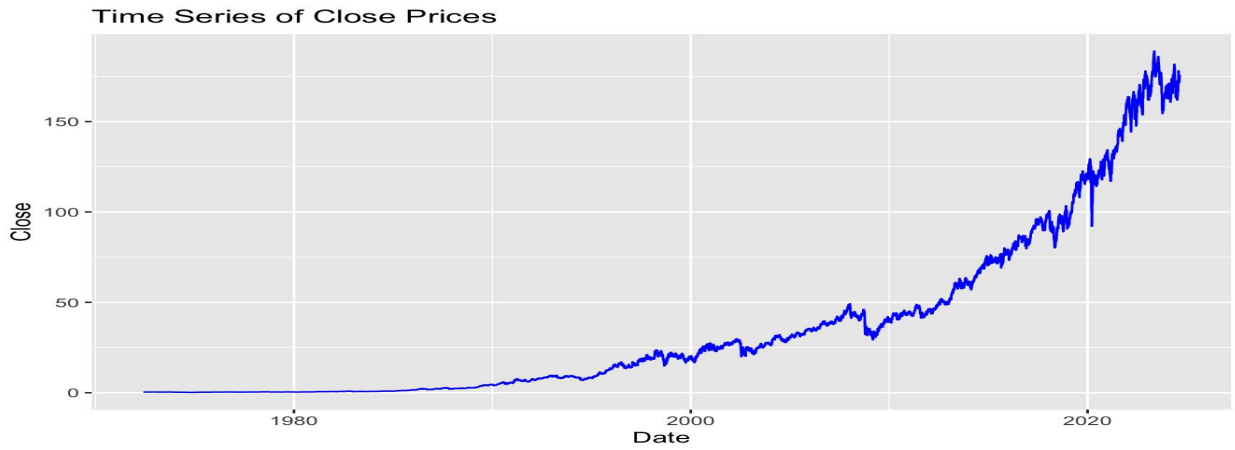
> cat("Column Names: ", colnames(data), "\n")
Column Names: Date Open High Low Close Volume
> cat("Data Types: \n")
Data Types:
> print(sapply(data, class))
      Date   Open   High   Low   Close   Volume
"character" "numeric" "numeric" "numeric" "numeric" "integer"
>
> # View the first few rows of the dataset
> head(data)

	Date	Open	High	Low	Close	Volume
1	1972-06-01	0.0000000	0.3993877	0.3947704	0.3993877	318600
2	1972-06-02	0.3993879	0.4034280	0.3982334	0.3982334	140400
3	1972-06-05	0.3993910	0.3999699	0.3860779	0.3936028	469800

```
4 1972-06-06 0.3901300 0.3901300 0.3889725 0.3901300 140400
5 1972-06-07 0.3901300 0.3907089 0.3866570 0.3878147 178200
6 1972-06-08 0.3878147 0.3889725 0.3854995 0.3878147 432000
>
> summary(data)
     Date                Open             High              Low              Close
 Min.   :1972-06-01   Min.   :  0.000   Min.   :  0.1469   Min.   :  0.1414   Min.   :  0.1414
 1st Qu.:1985-06-12   1st Qu.:  1.248   1st Qu.:  1.2543   1st Qu.:  1.2394   1st Qu.:  1.2462
 Median :1998-06-22   Median : 18.824   Median : 19.0998   Median : 18.5647   Median :
18.8363
 Mean   :1998-07-02   Mean   : 35.700   Mean   : 35.9795   Mean   : 35.4222   Mean   : 35.7124
 3rd Qu.:2011-07-20   3rd Qu.: 45.158   3rd Qu.: 45.4727   3rd Qu.: 44.8385   3rd Qu.: 45.1977
 Max.   :2024-08-20   Max.   :189.425   Max.   :189.7239   Max.   :187.8641   Max.   :188.9915
    Volume
 Min.   :       0
 1st Qu.: 2472800
 Median : 3747000
 Mean   : 4161206
 3rd Qu.: 5259800
 Max.   :46162800
>
```

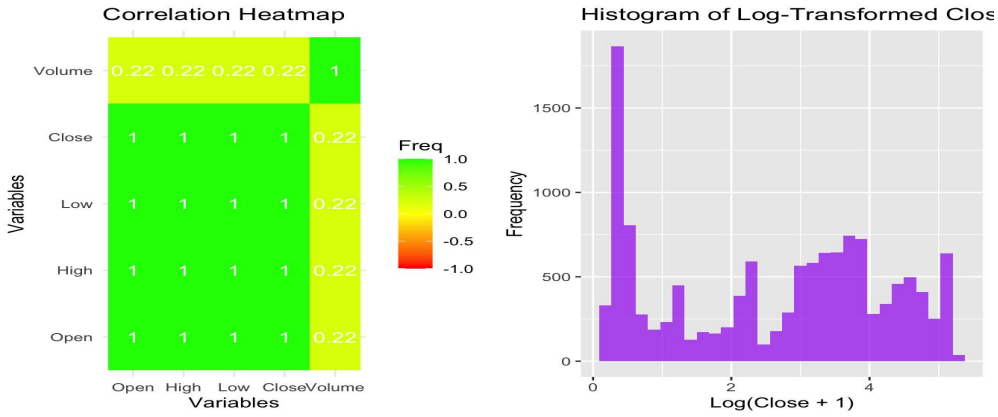
Line Plot

```
# Line plot

if ("Close" %in% colnames(data)) {

  ggplot(data, aes(x = Date, y = Close)) +

    geom_line(color = "blue") +

    labs(title = "Time Series of Close Prices", x = "Date", y = "Close")


}
```



CoRelation Heat map

```
log_histogram <- ggplot(data, aes(x = log(Close + 1))) +
  geom_histogram(fill = "purple", bins = 30, alpha = 0.7) +
  labs(title = "Histogram of Log-Transformed Close Prices", x = "Log(Close + 1)", y = "Frequency")
```



Handle Missing Values

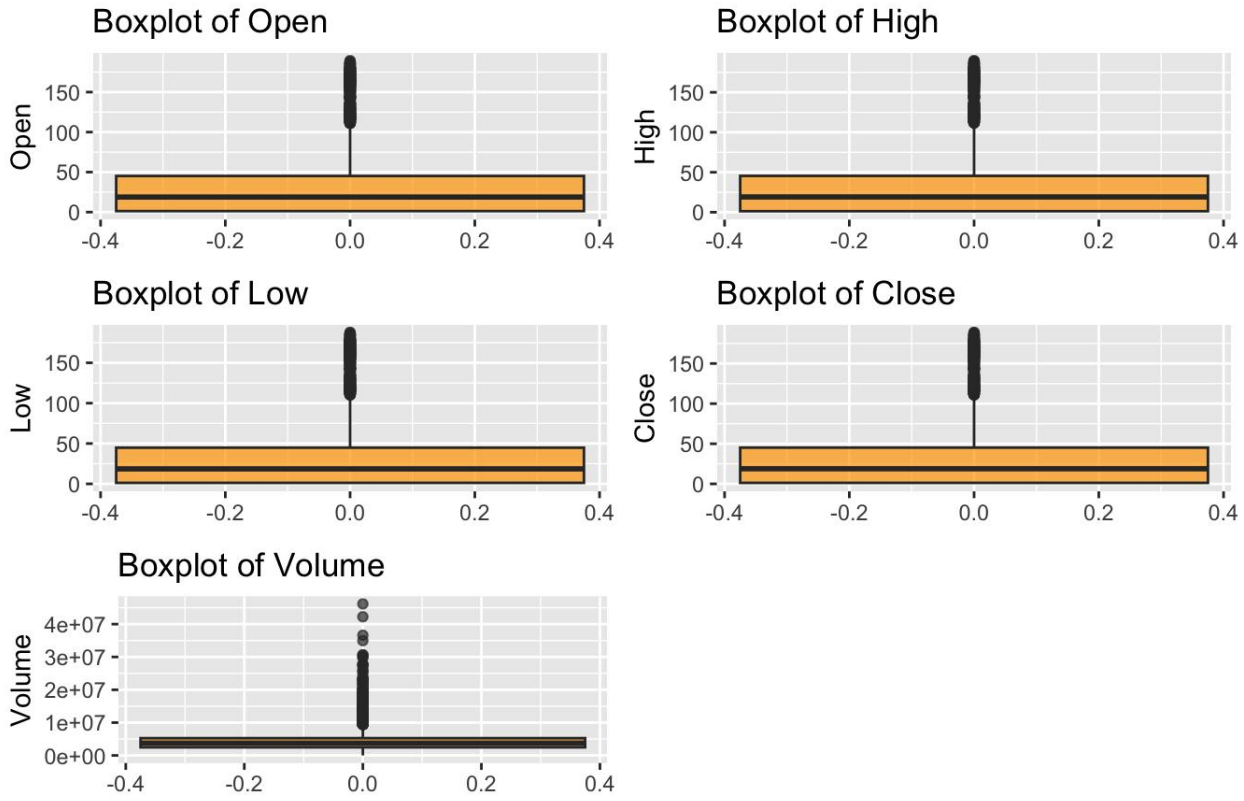
```
missing_values <- colSums(is.na(data))

cat("Missing values per column:\n")

print(missing_values)

data_imputed <- data %>%

  mutate(across(where(is.numeric), ~ifelse(is.na(.), median(., na.rm = TRUE), .)))
```

Missing values per column:

```
> print(missing_values)

  Date   Open   High    Low  Close Volume

     0      0      0      0      0      0
```

Outlier Box Plot

```
boxplots <- lapply(names(data)[numeric_cols], function(col) {

  ggplot(data, aes_string(y = col)) +

    geom_boxplot(fill = "orange", alpha = 0.7) +

    labs(title = paste("Boxplot of", col), y = col)

})

ggarrange(plotlist = boxplots, ncol = 2, nrow = ceiling(length(boxplots) / 2))
```



Stationary Check

```
# Step 5: Stationarity Check
# Perform Augmented Dickey-Fuller (ADF) test on the 'Close' column
if ("Close" %in% colnames(data_cleaned)) {
  close_series <- data_cleaned$Close
   # Perform ADF test
  adf_test_result <- adf.test(close_series, alternative = "stationary")
  print(adf_test_result)
   # Check p-value
  if (adf_test_result$p.value > 0.05) {
    cat("The series is not stationary. Applying differencing...\n")
```

```
    # Apply first-order differencing
  data_cleaned$Close_diff <- c(NA, diff(close_series))
    # Remove NA values caused by differencing
  data_cleaned <- na.omit(data_cleaned)
    # Re-check stationarity after differencing
  adf_test_result_diff <- adf.test(data_cleaned$Close_diff, alternative = "stationary")
  print(adf_test_result_diff)
    if (adf_test_result_diff$p.value <= 0.05) {
   cat("The differenced series is stationary.\n")
  } else {
   cat("Further differencing or transformations may be required.\n")
  }
 } else {
  cat("The series is stationary. No differencing needed.\n")
 }
}
```

Augmented Dickey-Fuller Test

data: close_series

Dickey-Fuller = 0.43592, Lag order = 22, p-value = 0.99

alternative hypothesis: stationary

As it is not stationary, lets apply differencing

Augmented Dickey-Fuller Test

data: data_cleaned$Close_diff

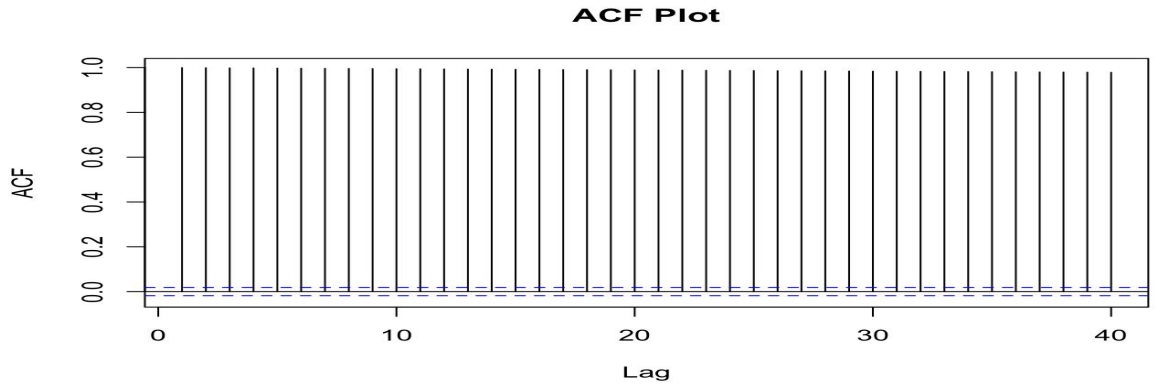
Dickey-Fuller = -22.252, Lag order = 22, p-value = 0.01

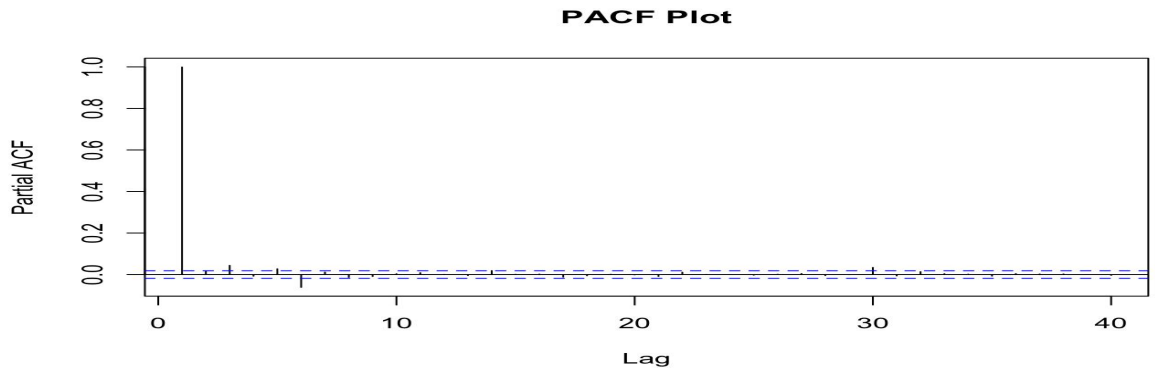
alternative hypothesis: stationary

Therefore it is stationary

ARIMA Model Selection

```
# Step 6: ARIMA Model Selection

# ACF and PACF plots

Acf(close_series, main = "ACF Plot")

Pacf(close_series, main = "PACF Plot")
```



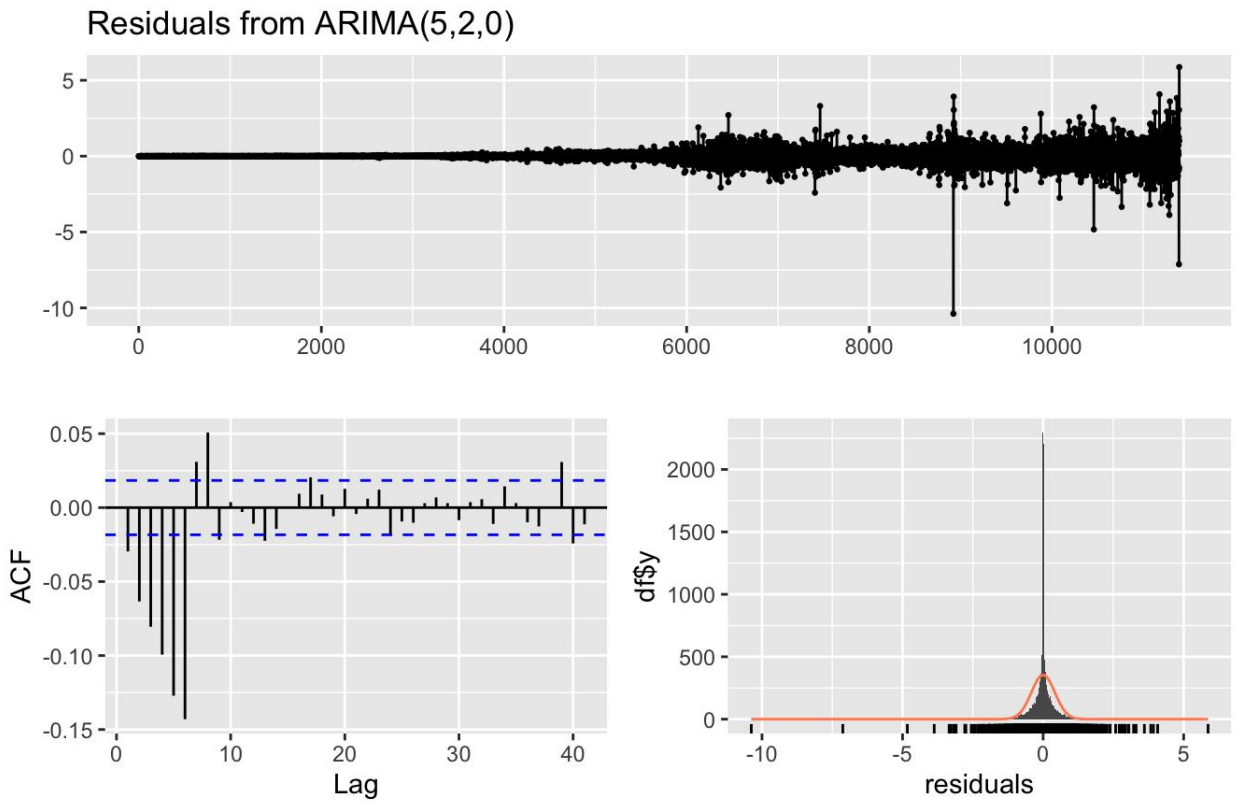



```
# Fit ARIMA model
```

```
# Use auto.arima to automatically select p, d, q
arima_model <- auto.arima(close_series)
# Display model summary
cat("ARIMA Model Summary:\n")
print(summary(arima_model))
# Step 7: Model Fitting (Residual diagnostics and Ljung-Box Test)
# Residual diagnostics
checkresiduals(arima_model)
```

ARIMA Model Summary:

```
> print(summary(arima_model))
Series: close_series
ARIMA(5,2,0)
Coefficients:
         ar1      ar2      ar3      ar4      ar5
      -0.8774  -0.6824  -0.5335  -0.3710  -0.1807
s.e.   0.0092   0.0120   0.0127   0.0121   0.0094
sigma^2 = 0.1746:  log likelihood = -6221.3
AIC=12454.59   AICc=12454.6   BIC=12498.64
Training set error measures:
                     ME      RMSE       MAE         MPE     MAPE     MASE       ACF1
Training set 0.000250772 0.4176916 0.2111206 -0.001367343 1.191933 1.103569 -0.02966124
>
```



LJung-Box test for Auto Corelation

```
# Ljung-Box test for autocorrelation of residuals
ljung_box_test <- Box.test(residuals(arima_model), lag = 20, type = "Ljung-Box")
cat("Ljung-Box Test Result:\n")
print(ljung_box_test)
```

Ljung-Box test

data: Residuals from ARIMA(5,2,0)

Q* = 705.7, df = 5, p-value < 2.2e-16

Model df: 5.   Total lags used: 10

Ljung-Box Test Result:

> print(ljung_box_test)

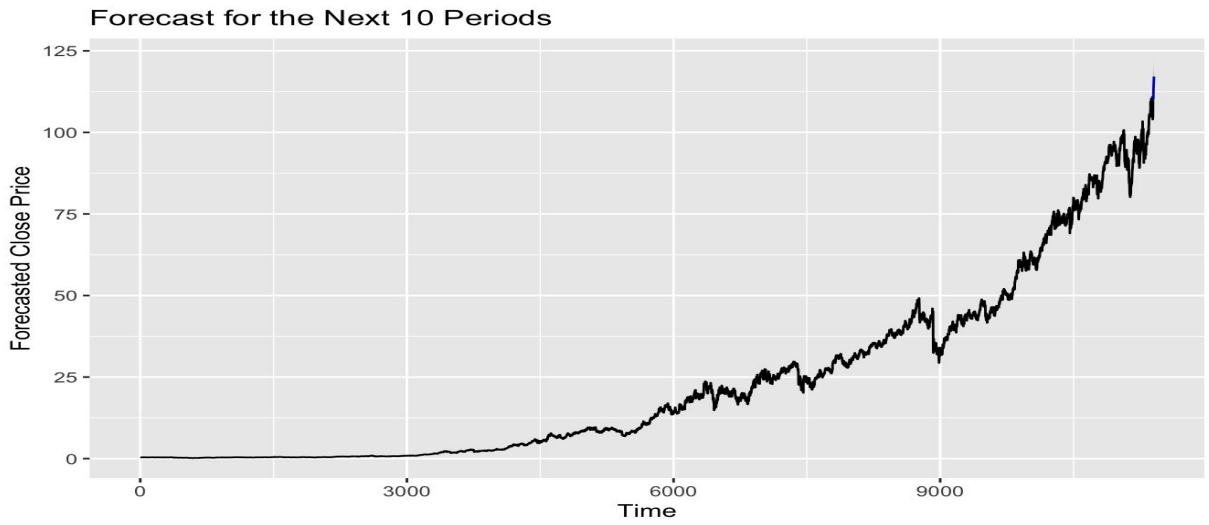
Box-Ljung test

data: residuals(arima_model)

X-squared = 724.18, df = 20, p-value < 2.2e-16

Forecast

```
# Define the forecast horizon (e.g., forecast the next 10 periods)
forecast_horizon <- 10
forecast_results <- forecast(arima_model, h = forecast_horizon)
# Plot the forecast
autoplot(forecast_results) +
  labs(title = paste("Forecast for the Next", forecast_horizon, "Periods"), x = "Time", y = "Forecasted Close Price")
```




```
# Print the forecasted values
cat("Forecasted Values:\n")
print(forecast_results)
```

Forecasted Values:

```
> print(forecast_results)
      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
11396       110.4901 109.9547 111.0256 109.6712 111.3090
11397       110.3114 109.5064 111.1164 109.0803 111.5426
11398       111.3385 110.2628 112.4143 109.6934 112.9837
11399       112.1999 110.8474 113.5525 110.1314 114.2685
11400       113.5365 111.8827 115.1903 111.0073 116.0657
11401       114.1270 112.1309 116.1232 111.0742 117.1798
11402       114.7383 112.3522 117.1243 111.0891 118.3875
11403       115.4306 112.6472 118.2139 111.1738 119.6873
11404       116.2893 113.0927 119.4859 111.4005 121.1781
11405       117.1266 113.4998 120.7533 111.5799 122.6732
```

Accuracy Evaluation

```
# Step 9: Accuracy Evaluation
# Calculate Accuracy Metrics (MAE, RMSE, AIC, BIC)
# Actual values (for comparison using last observed value)
actual_values <- tail(close_series, forecast_horizon)
# Predicted values
predicted_values <- forecast_results$mean
# MAE Mean Absolute Error
mae <- mean(abs(actual_values - predicted_values))
cat("Mean Absolute Error:", mae, "\n")
```

```
# RMSE (Root Mean Squared Error)
rmse <- sqrt(mean((actual_values - predicted_values)^2))
cat("Root Mean Squared Error", rmse, "\n")
# AIC (Akaike Information Criterion)
aic_value <- AIC(arima_model)
cat("Akaike Information Criterion:", aic_value, "\n")
# BIC (Bayesian Information Criterion)
bic_value <- BIC(arima_model)
cat("Bayesian Information Criterion :", bic_value, "\n")
```

Mean Absolute Error (MAE): 4.917607

Root Mean Squared Error (RMSE): 6.24954

Akaike Information Criterion (AIC): 12454.59

Bayesian Information Criterion (BIC): 12498.64

Accuracy Matrix

```
# Evaluate the forecast accuracy
accuracy_metrics <- accuracy(forecasted_values, actual_values)
print(accuracy_metrics)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.04312408	4.864354	3.401651	-0.002575443	2.998443	0.3270168	-0.0009735144
Test set	17.36495987	25.441877	19.594683	15.810202964	17.834982	1.8837296	NA