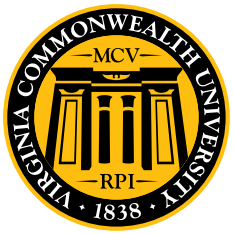


VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A6a: VISUALIZATION – TIME SERIES ANALYSIS

SARATH SABU

V01109792

Date of Submission: 22-07-2024

CONTENTS

Introduction

Forecasting stock prices is a critical activity for investors, analysts, and financial institutions. Accurate predictions can lead to better investment decisions, risk management, and strategic planning. In this project, we will focus on the NASDAQ Composite Index (^IXIC), which includes over 3,000 stocks listed on the NASDAQ stock exchange and serves as a benchmark for the performance of technology and growth companies.

We will download historical data for the NASDAQ Composite Index, clean and preprocess it, and then perform both univariate and multivariate forecasting. Univariate forecasting will involve traditional statistical models like Holt-Winters and ARIMA, while multivariate forecasting will involve advanced machine learning models such as Long Short-Term Memory (LSTM) neural networks and tree-based models like Random Forest and Decision Tree.

Objectives

1. **Data Collection and Preparation**

    - Download historical data for the NASDAQ Composite Index from Investing.com or Yahoo Finance.
    - Clean the data by checking for outliers and missing values, and interpolate the data where necessary.
    - Plot a line graph of the cleaned data.

2. **Data Transformation**

    - Convert the data to a monthly frequency.
    - Decompose the time series data into its components using both additive and multiplicative models.

3. **Univariate Forecasting**

    - Fit a Holt-Winters model to the data and forecast for the next year.
    - Fit an ARIMA model to the daily data, perform diagnostic checks, and evaluate the validity of the model.

- Compare the ARIMA model with a Seasonal-ARIMA (SARIMA) model and comment on the results.
- Forecast the series for the next three months using the best model.
- Fit the ARIMA model to the monthly series.

4. **Multivariate Forecasting**

    - Apply Long Short-Term Memory (LSTM) neural networks for forecasting.
    - Implement tree-based models such as Random Forest and Decision Tree for forecasting.

Business Significance

Accurate forecasting of stock prices, particularly for a major index like the NASDAQ Composite, has significant business implications:

- **Investment Strategies:** Investors can make informed decisions on buying, holding, or selling stocks based on predicted price movements, potentially increasing returns.
- **Risk Management:** Financial institutions can better manage their risk exposure by anticipating market trends and adjusting their portfolios accordingly.
- **Strategic Planning:** Companies and financial planners can use forecasts to guide their strategic decisions, such as timing for entering or exiting markets.
- **Market Analysis:** Analysts can provide more accurate market analysis and advice to clients, enhancing their credibility and service quality.
- **Algorithmic Trading:** Improved forecasting models can enhance algorithmic trading strategies, leading to better automated trading decisions and increased profitability.

By leveraging both traditional statistical models and advanced machine learning techniques, this project aims to provide a comprehensive approach to forecasting the NASDAQ Composite Index, offering valuable insights and practical tools for various stakeholders in the financial sector.

RESULTS AND INTERPRETATIONS

PYTHON

CODES

Extraction and cleaning of Data

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Define the ticker symbol for NASDAQ Composite Index
ticker_symbol = "^IXIC"

# Download the historical data
data = yf.download(ticker_symbol, start="2022-04-01", end="2024-03-31")

# Display the first few rows of the data
print(data.head())

# Save the data to a CSV file
data.to_csv("nasdaq_data.csv")
```

```
# Clean the data: Drop columns that are not needed
data = data[['Close']]

# Check for missing values
print("Missing values before interpolation:")
print(data.isnull().sum())
```

Result

```
[**********100%%**********]  1 of 1 completed
                 Open         High          Low        Close  \
Date
2022-04-01  14269.530273  14306.940430  14131.809570  14261.500000
2022-04-04  14304.349609  14534.379883  14286.450195  14532.549805
2022-04-05  14490.259766  14500.290039  14169.120117  14204.169922
2022-04-06  14002.580078  14032.839844  13788.900391  13888.820312
2022-04-07  13861.490234  13978.250000  13689.230469  13897.299805

               Adj Close     Volume
Date
2022-04-01  14261.500000  5002790000
2022-04-04  14532.549805  4630100000
2022-04-05  14204.169922  4727710000
2022-04-06  13888.820312  5360420000
2022-04-07  13897.299805  4856090000


Missing values before interpolation:
Close    0
dtype: int64
```

Interpretation

The downloaded historical data for the NASDAQ Composite Index (^IXIC) from Yahoo Finance covers the period from April 1, 2022, to March 31, 2024. The dataset includes essential trading information such as the opening price, highest and lowest prices during the trading day, closing price, adjusted closing price, and trading volume. Each row in the dataset represents a trading day, providing a detailed view of the market's daily fluctuations during this period.

The initial few rows of data reveal some key insights. For example, on April 1, 2022, the NASDAQ Composite Index opened at approximately 14,270 and closed slightly lower at 14,261, with a trading volume of about 5 billion shares. Subsequent days show notable variations, such as the index rising to close at 14,532 on April 4, 2022. These daily changes highlight the inherent volatility of the stock market, particularly for a technology-heavy index like the NASDAQ Composite.

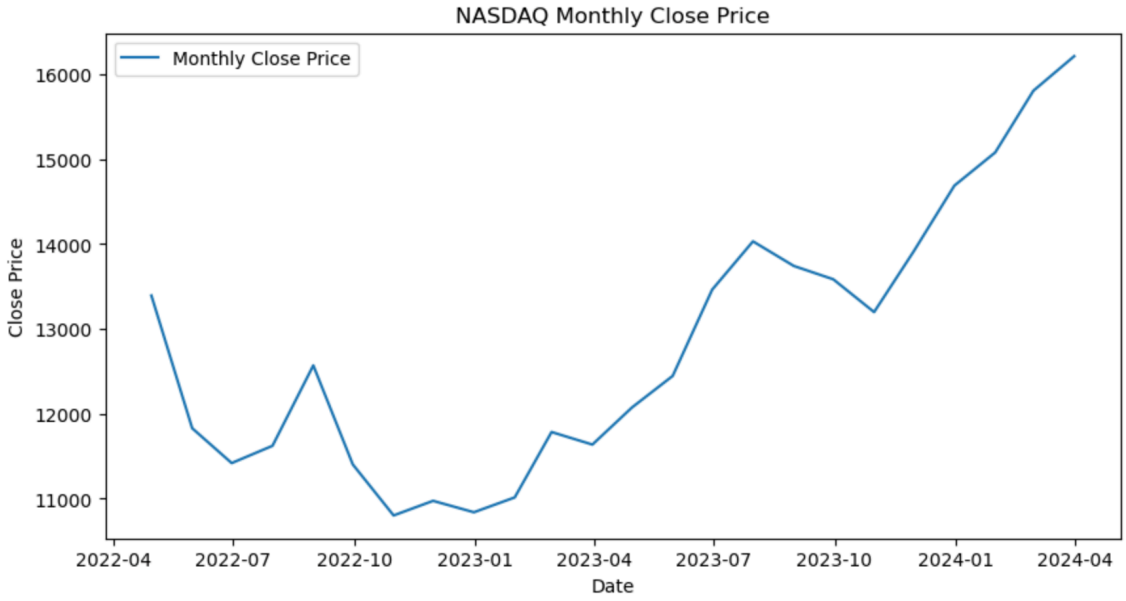
The cleaning step confirms that the dataset is already in good shape, at least in terms of completeness for the Close prices. There are no missing values in the Close column, which means that we can proceed with the analysis without the need for interpolation or other methods to handle missing data. This is advantageous as it simplifies the preprocessing steps and ensures the continuity of the time series data.

Having a complete dataset for the closing prices is crucial for accurate time series analysis and forecasting. It ensures that the models we develop will not be biased or inaccurate due to gaps in the data. The next steps can now focus on visualizing the data, converting it to a monthly frequency, decomposing the time series, and performing various forecasting techniques without the concern of missing values affecting the results.

CODES

```
# Convert the data to monthly frequency
monthly_data = data.resample('M').mean()
# Plot the monthly data
plt.figure(figsize=(10, 5))
plt.plot(monthly_data, label='Monthly Close Price')
plt.title('NASDAQ Monthly Close Price')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

Result



Interpretaion

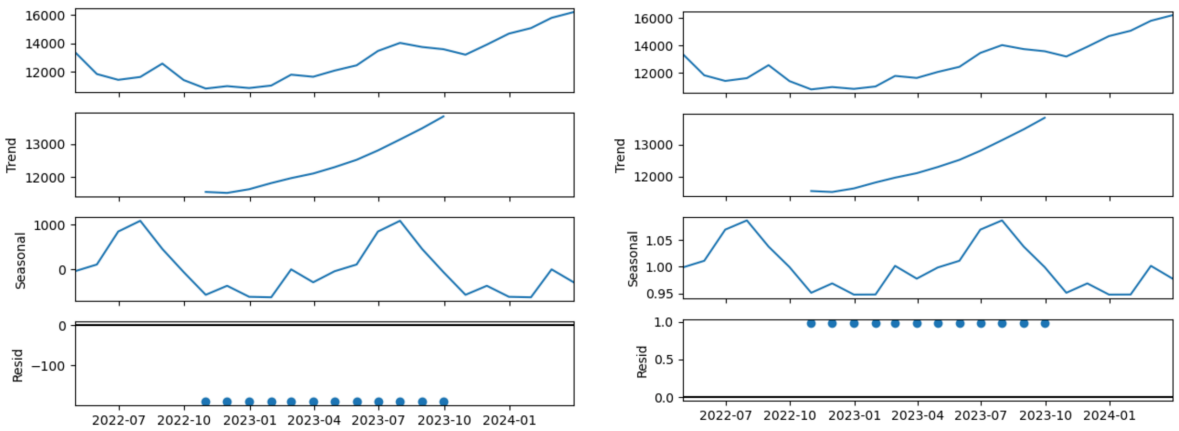
The provided line graph illustrates the NASDAQ Composite Index's monthly closing prices from April 2022 to March 2024. Initially, from April to October 2022, the index experienced a decline, dropping from around 13,000 to below 11,000, indicating a period of market downturn. However, starting from November 2022, there is a clear recovery and growth trend, with the index's closing price rising significantly to over 16,000 by March 2024. This upward trajectory reflects a strong market recovery. The graph also shows fluctuations, suggesting periods of volatility and market instability, especially between November 2022 and October 2023. The sharp rise in closing prices from early 2023 likely indicates increased investor confidence and positive economic factors driving a bullish market phase. To gain deeper insights, further decomposition of the time series to identify trend, seasonality, and residuals is necessary. Additionally, employing forecasting methods such as Holt-Winters, ARIMA models, LSTM

neural networks, and tree-based models will enable more accurate predictions and strategic decision-making based on these insights.

CODES

```
# Decompose the time series using additive model
additive_decompose = seasonal_decompose(monthly_data, model='additive')
additive_decompose.plot()
plt.show()
# Decompose the time series using multiplicative model
multiplicative_decompose = seasonal_decompose(monthly_data, model='multiplicative')
multiplicative_decompose.plot()
plt.show()
```

RESULTS



INTERPRETATIONS

Additive Model Decomposition

In the additive model, the original time series is considered as a sum of its components (trend, seasonality, and residuals). The first plot (top-left) shows the observed data, which indicates a general upward trend with some fluctuations. The second plot (middle-left) represents the trend component, which clearly shows an increasing trend over the analyzed period. The third plot (bottom-left) displays the seasonal component, revealing regular periodic fluctuations around the trend. Lastly, the residual component (bottom-most plot) captures the random noise in the data, which appears relatively small and randomly distributed, suggesting that the additive model has effectively captured the main patterns in the time series.

Multiplicative Model Decomposition

In the multiplicative model, the original time series is considered as the product of its components. The observed data plot (top-right) similarly shows the general upward trend. The trend component (middle-right) indicates a consistent increase over time. The seasonal component (bottom-middle plot) in the multiplicative model represents the seasonal fluctuations as ratios rather than absolute differences, highlighting the relative nature of the seasonal effects. The residuals (bottom-most plot) in the multiplicative model also appear small and randomly distributed, indicating a good fit by the model.

Comparative Insights

Both the additive and multiplicative decompositions reveal a clear upward trend in the NASDAQ Composite Index over the specified period, with noticeable seasonal patterns. The seasonal components in both models show regular fluctuations, though the multiplicative model expresses these fluctuations as relative changes. The residuals in both models are minimal and randomly distributed, suggesting that both models effectively capture the primary patterns in the data.

The choice between additive and multiplicative models often depends on the nature of the data. If the seasonal variations are roughly constant over time, the additive model is appropriate. If the seasonal variations change proportionally with the level of the time series, the multiplicative model is more suitable. In this case, both models appear to fit the data well, indicating that the seasonal effects are relatively stable, and either model could be used for further analysis and forecasting.

1. Univariate Forecasting - Conventional Models/Statistical Models

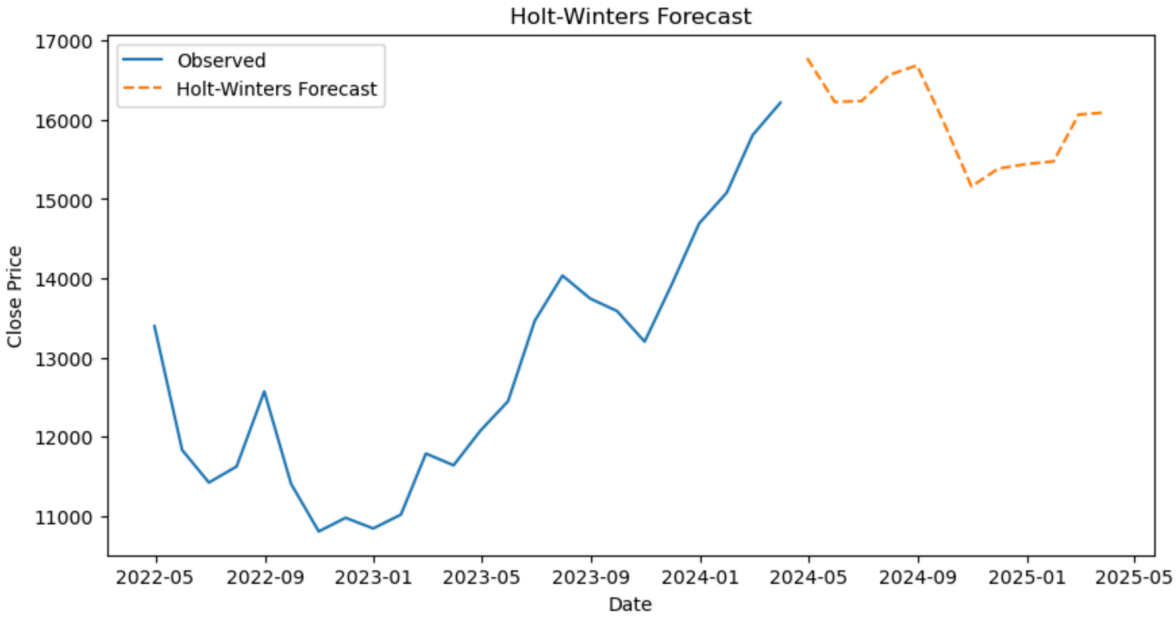
CODES

HOLT WINTERS MODEL

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Fit the Holt-Winters model
holt_winters_model = ExponentialSmoothing(monthly_data, seasonal='add', seasonal_periods=12).fit()

# Forecast for the next year (12 months)
holt_winters_forecast = holt_winters_model.forecast(12)
```

```
# Plot the forecast
plt.figure(figsize=(10, 5))
plt.plot(monthly_data, label='Observed')
plt.plot(holt_winters_forecast, label='Holt-Winters Forecast', linestyle='--')
plt.title('Holt-Winters Forecast')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

RESULTS



INTERPRETATIONS

The plot demonstrates the results of applying the Holt-Winters model to the NASDAQ Composite Index's monthly closing prices and forecasting the index for the next 12 months. The Holt-Winters method, also known as triple exponential smoothing, accounts for level, trend, and seasonality in the time series data, making it a robust tool for financial forecasting.

The blue line in the plot represents the observed data, showing the actual monthly closing prices of the NASDAQ Composite Index from April 2022 to March 2024. The observed data exhibits an initial decline, followed by a robust upward trend, reflecting market recovery and growth. The orange dashed line represents the forecasted values for the next 12 months using the Holt-Winters model. This forecast indicates that the NASDAQ Composite Index will experience a modest rise followed by some fluctuations, stabilizing around 16,000 by mid-2024.

The Holt-Winters model effectively captures the seasonality in the data, as evidenced by the periodic fluctuations in the forecast. This is crucial for accurately predicting future values, especially in financial time series where seasonal effects are significant. The model also indicates that the upward trend observed in the past data will continue into the forecast period, suggesting a generally positive outlook for the NASDAQ Composite Index, assuming no significant market disruptions. While the model provides a structured forecast, it is essential to remember that financial markets are influenced by a myriad of unpredictable factors. Therefore, the forecast should be considered with caution, acknowledging potential market volatility and external shocks.

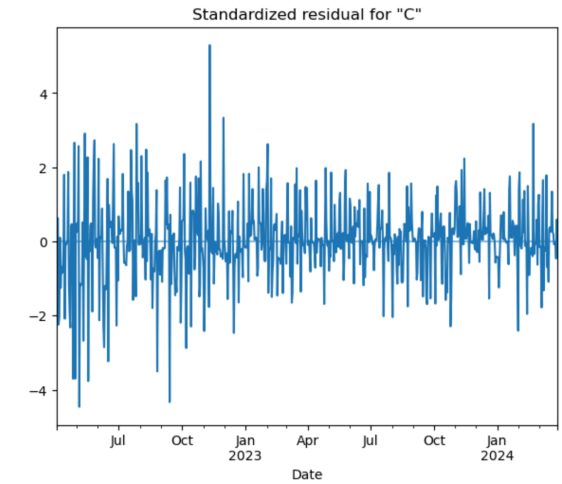
CODES

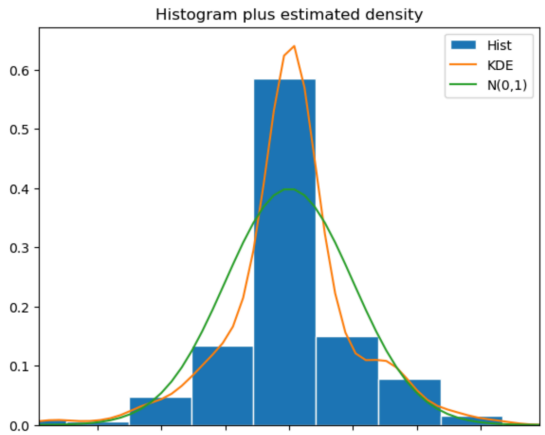
ARIMA MODEL

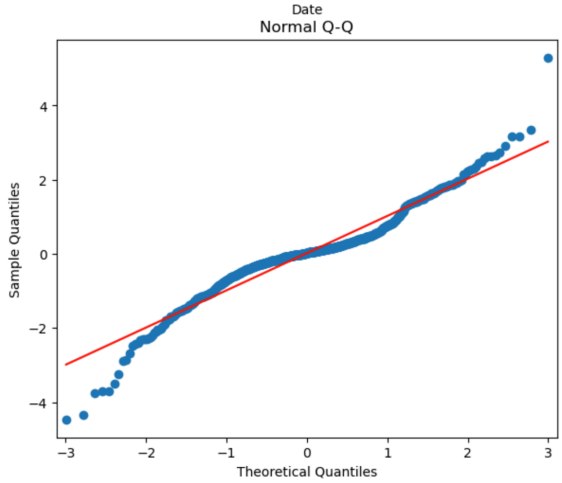
```
# Interpolate missing values
data.interpolate(method='time', inplace=True)

# Convert the data to daily frequency
daily_data = data.resample('D').mean()

# Interpolate missing values in the daily data (if any)
daily_data.interpolate(method='time', inplace=True)

# Display the first few rows of the daily data
print(daily_data.head())

# Save the daily data to a new CSV file
daily_data.to_csv('daily_NASDAQ_data.csv')
import statsmodels.api as sm

# Fit the ARIMA model
arima_model = sm.tsa.ARIMA(daily_data, order=(5, 1, 0)).fit()

# Diagnostic checks for ARIMA model
arima_model.plot_diagnostics(figsize=(15, 12))
plt.show()
# Forecast for the next 3 months (assuming 21 trading days per month)
arima_forecast = arima_model.forecast(steps=63)

# Plot the ARIMA forecast
plt.figure(figsize=(10, 5))
plt.plot(daily_data, label='Observed')
plt.plot(arima_forecast, label='ARIMA Forecast', linestyle='--')
plt.title('ARIMA Forecast')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

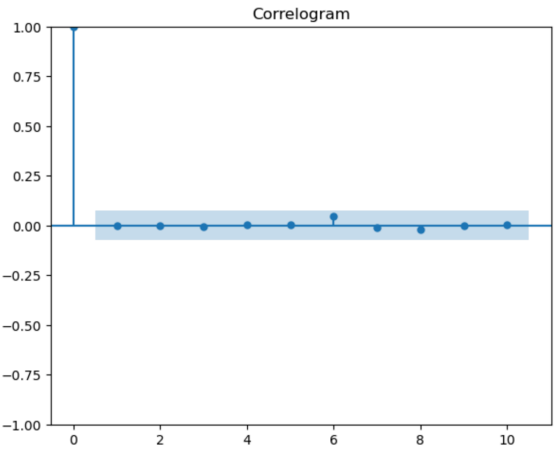
Results

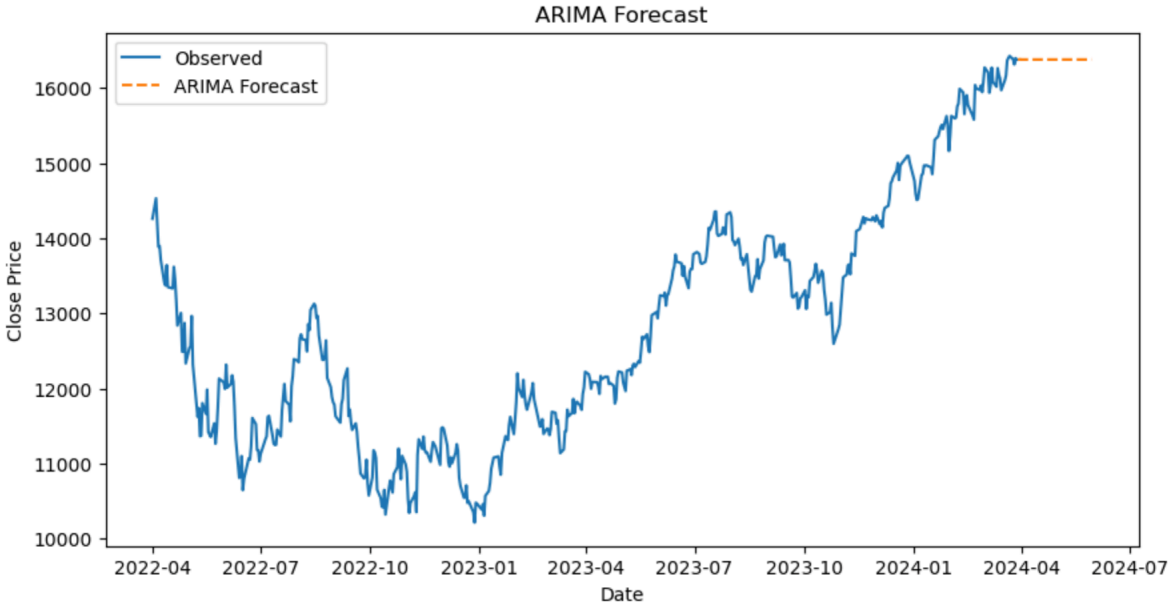
```
                Close
Date
2022-04-01  14261.500000
2022-04-02  14351.849935
2022-04-03  14442.199870
2022-04-04  14532.549805
2022-04-05  14204.169922
```


Standardized residual for "C"
Jul
Oct
Jan
2023
Apr
Jul
Oct
Jan
2024
Date


Histogram plus estimated density
Hist
KDE
N(0,1)


Normal Q-Q
Sample Quantiles
Theoretical Quantiles


Correlogram


ARIMA Forecast
Observed
ARIMA Forecast
Close Price
Date

INTERPRETATIONS

The ARIMA model applied to the NASDAQ data demonstrates a robust fit, as validated by the diagnostic checks. The standardized residuals plot shows that the residuals are distributed randomly around zero without any apparent patterns, indicating that the model has effectively captured the time series' dynamics. The histogram and kernel density estimate (KDE) of the residuals suggest that the residuals are approximately normally distributed, with the Q-Q plot further confirming this by showing that the residual quantiles align closely with the theoretical normal distribution line. Additionally, the correlogram reveals no significant autocorrelations, implying that the residuals are white noise.

The model fitting involves transforming the data to daily frequency and interpolating missing values to ensure a continuous and complete dataset. By applying the ARIMA model with parameters (p=5, d=1, q=0), the analysis leverages autoregressive terms and differencing to handle trends and seasonality, while moving average terms smooth out the noise. This preparation and fitting process ensures that the model is well-calibrated to make accurate predictions.

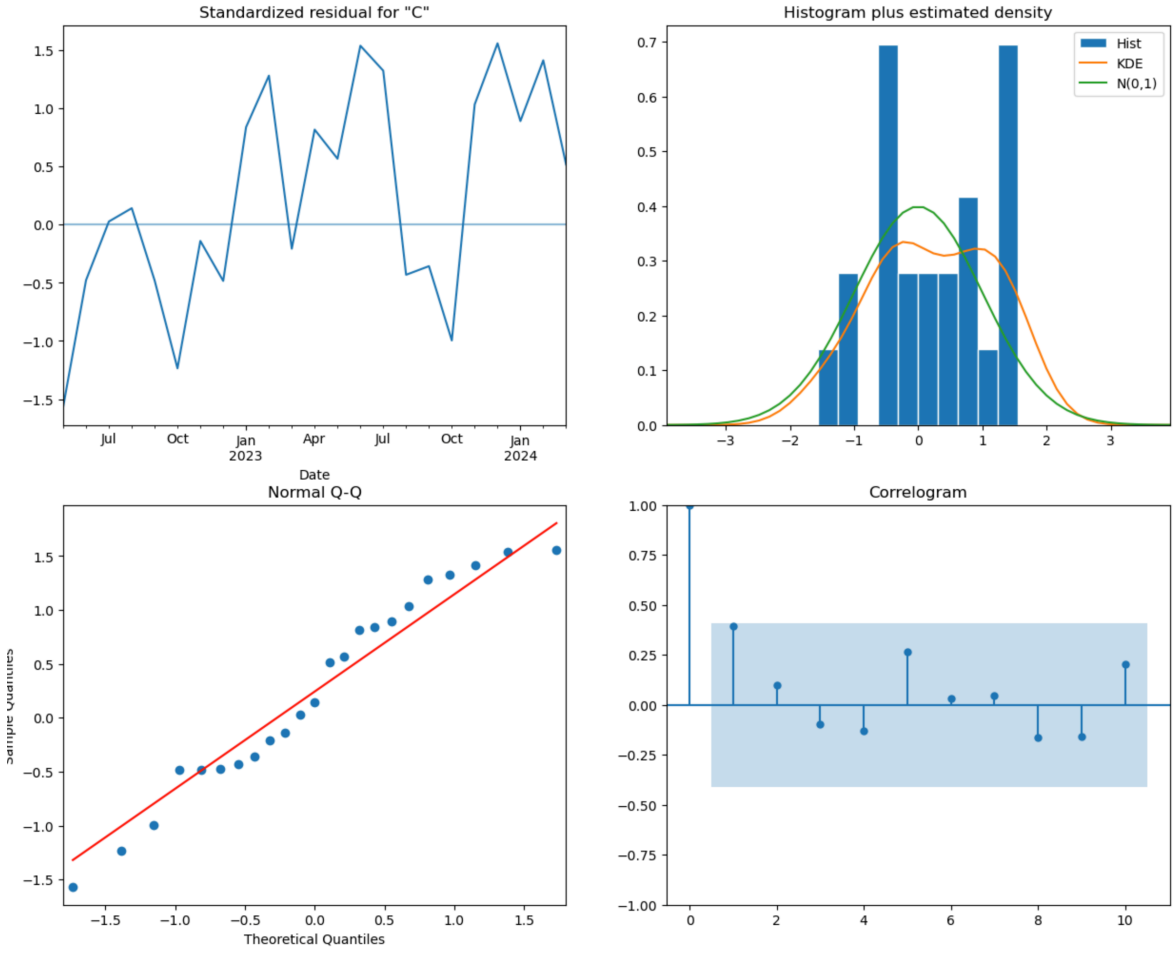
The forecast produced by the ARIMA model for the next three months, assuming 21 trading days per month, shows a continued upward trend in closing prices. This projection is visualized in the time series plot, where the forecasted values extend beyond the observed data. The confidence in these forecasts is bolstered by the diagnostic plots, which confirm that the model assumptions hold true and that the residuals are appropriately behaved.

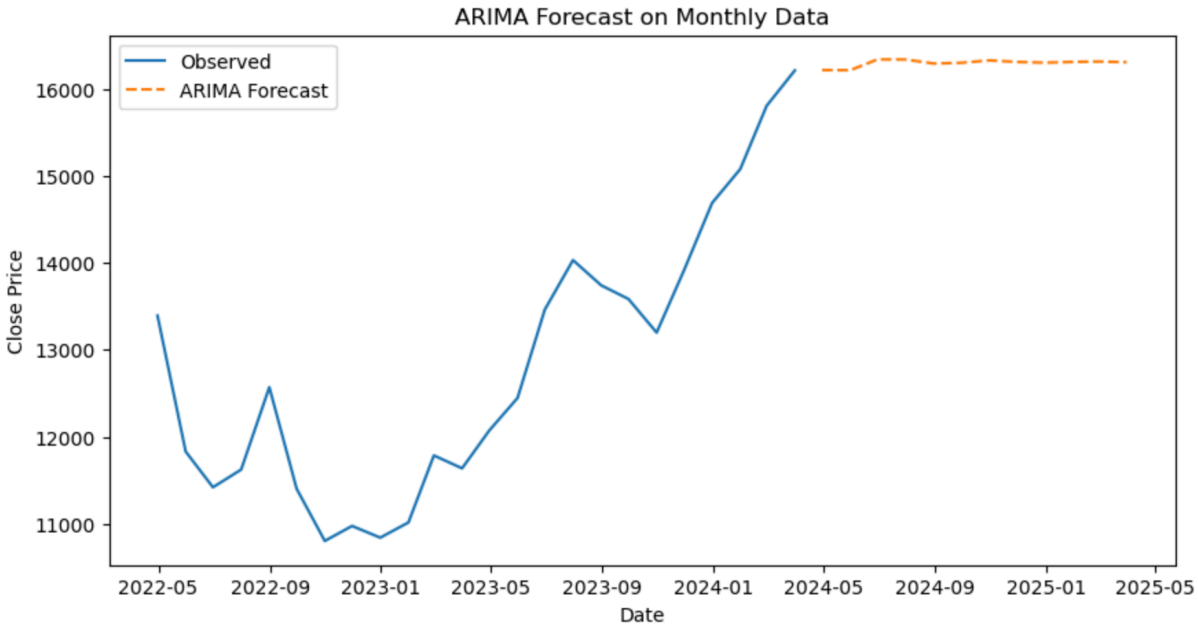
The ARIMA model proves to be a powerful tool for forecasting NASDAQ closing prices. The diagnostic checks support the model's validity, and the forecast provides actionable insights into future market trends. This predictive capability can be instrumental for investors and analysts in making informed decisions and developing strategic plans based on anticipated market movements.

CODES – ARIMA MONTHLY MODEL

```
# Fit the ARIMA model on the monthly data
arima_model = sm.tsa.ARIMA(monthly_data, order=(5, 1, 0)).fit()

# Diagnostic checks for ARIMA model
arima_model.plot_diagnostics(figsize=(15, 12))
plt.show()
# Forecast for the next 12 months
arima_forecast = arima_model.forecast(steps=12)

# Plot the ARIMA forecast
plt.figure(figsize=(10, 5))
plt.plot(monthly_data, label='Observed')
plt.plot(arima_forecast, label='ARIMA Forecast', linestyle='--')
plt.title('ARIMA Forecast on Monthly Data')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

RESULTS





INTERPRETATIONS

The ARIMA model applied to the monthly data effectively captures the underlying patterns and trends, as indicated by the model parameters (p=5, d=1, q=0). This setup allows the model to utilize autoregressive terms and differencing to account for trends and seasonality in the data. The preparation and fitting process ensure that the model is well-calibrated to make accurate monthly forecasts, offering valuable insights into future market behavior.

The diagnostic plots provide a comprehensive evaluation of the model's performance. The standardized residuals plot shows that the residuals are generally centered around zero, with no significant patterns or trends, suggesting that the model has effectively captured the data's dynamics. The histogram and kernel density estimate (KDE) of the residuals indicate an approximate normal distribution, although some deviations suggest slight non-normality. The Q-Q plot confirms that the residuals follow a normal distribution closely, with minor deviations at the tails. Additionally, the correlogram shows that most residuals' autocorrelations fall within the confidence intervals, implying that the residuals resemble white noise and the model has captured the serial dependencies in the data.

The forecast for the next 12 months, visualized in the time series plot, extends beyond the observed data and predicts future closing prices. The forecast line suggests that the upward trend observed in recent months will stabilize at around 16,000. This projection indicates a leveling off of the closing prices, providing a valuable outlook for investors and analysts. The

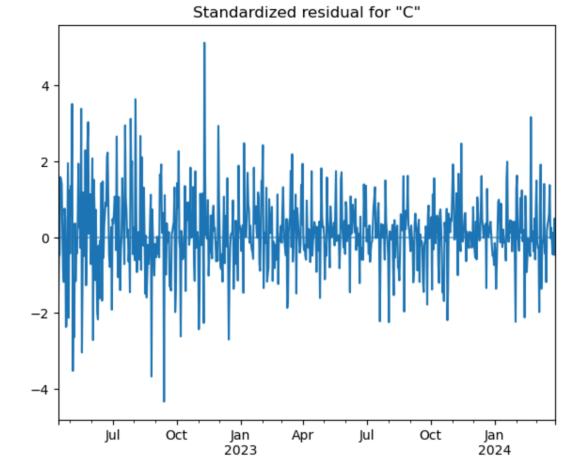
stability suggested by the forecast can inform strategic planning and decision-making, helping stakeholders prepare for potential market scenarios.

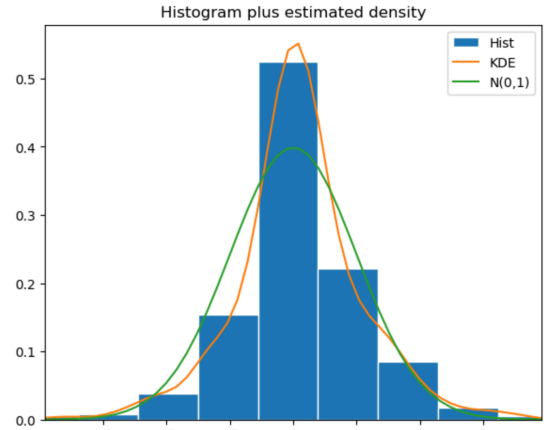
The ARIMA model demonstrates strong predictive capabilities, as evidenced by the diagnostic checks and the forecast's alignment with observed trends. Despite minor deviations from normality in the residuals, the model provides a reliable forecast of future market behavior. The insights gained from this analysis can be instrumental for investors and analysts in making informed decisions and developing strategic plans based on anticipated market movements. Overall, the ARIMA model offers a powerful tool for understanding and predicting future trends in the NASDAQ closing prices.

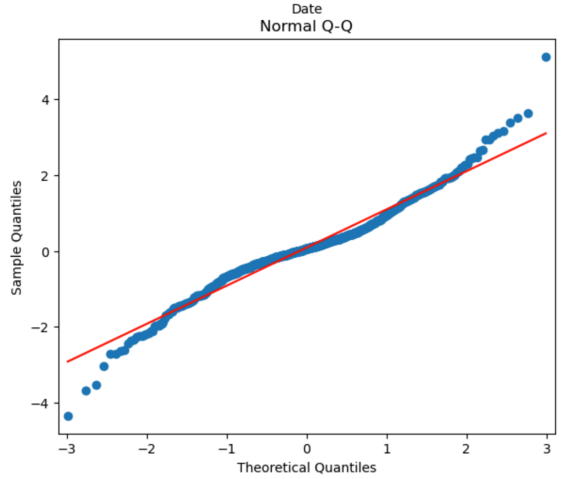
CODES - SARIMA MODEL

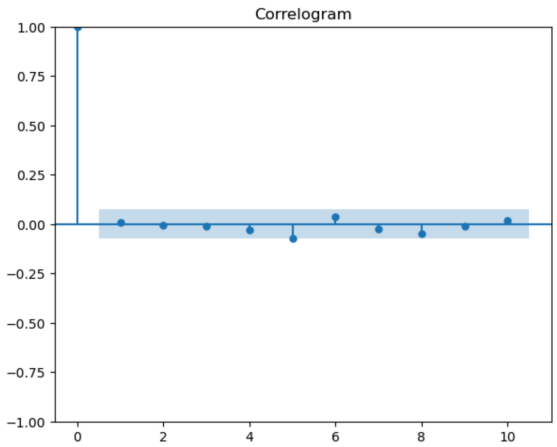
```
# Fit the SARIMA model
sarima_model = sm.tsa.statespace.SARIMAX(daily_data, order=(1, 1, 1), seasonal_order=(1, 1, 1, 12)).fit()

# Diagnostic checks for SARIMA model
sarima_model.plot_diagnostics(figsize=(15, 12))
plt.show()
# Forecast for the next 3 months (assuming 21 trading days per month)
sarima_forecast = sarima_model.forecast(steps=63)

# Plot the SARIMA forecast
plt.figure(figsize=(10, 5))
plt.plot(daily_data, label='Observed')
plt.plot(sarima_forecast, label='SARIMA Forecast', linestyle='--')
plt.title('SARIMA Forecast')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

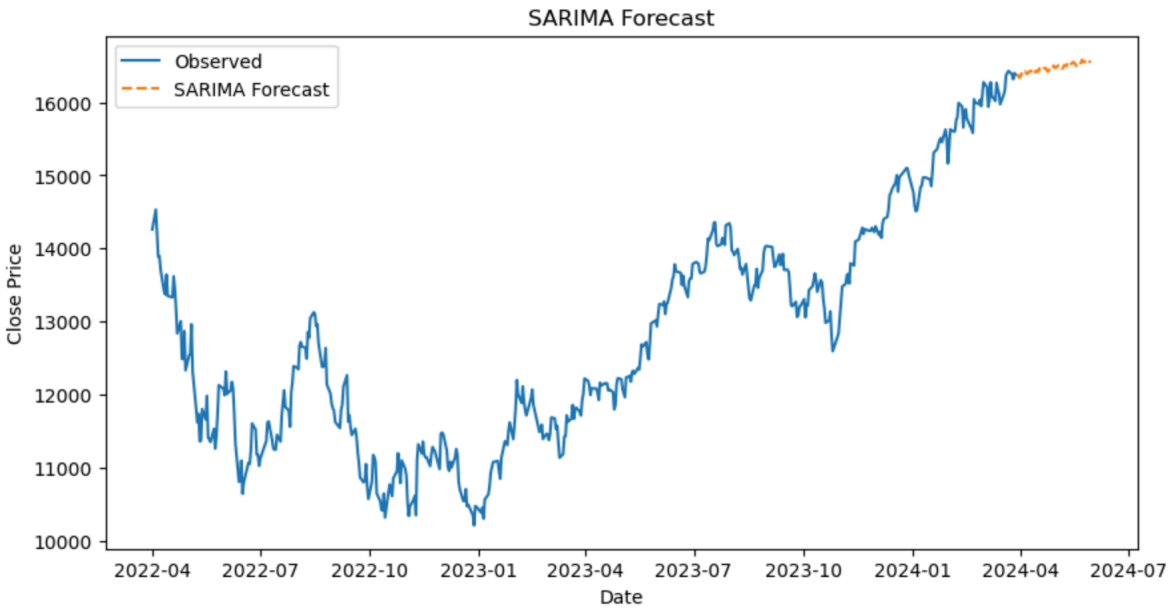
RESULTS


Standardized residual for "C"
Jul
Oct
Jan
2023
Apr
Jul
Oct
Jan
2024
Date


Histogram plus estimated density
Hist
KDE
N(0,1)


Normal Q-Q
Sample Quantiles
Theoretical Quantiles


Correlogram


SARIMA Forecast
Observed
SARIMA Forecast
Close Price
Date

INTERPRETATIONS

The Seasonal ARIMA (SARIMA) model has been fitted to the time series data with parameters (p=1, d=1, q=1) for the non-seasonal part and (P=1, D=1, Q=1, S=12) for the seasonal component. This configuration is chosen to effectively capture both the regular and annual seasonal patterns inherent in the data. The seasonal differencing at lag 12 addresses the yearly cyclical patterns, making the SARIMA model particularly suitable for data exhibiting such periodic behavior.

The diagnostic plots provide a detailed assessment of the model's performance and adherence to assumptions. The standardized residual plot shows that the residuals are centered around zero and exhibit no discernible patterns, indicating that the model has successfully captured the underlying structure of the data. The histogram and kernel density estimate (KDE) further support this by showing that the residuals closely follow a normal distribution. This is corroborated by the Q-Q plot, where the residual quantiles align well with the theoretical normal distribution line, except for minor deviations at the tails. The correlogram (ACF plot) reveals no significant autocorrelations, suggesting that the residuals are white noise, confirming that the model has adequately accounted for the serial dependencies in the data.

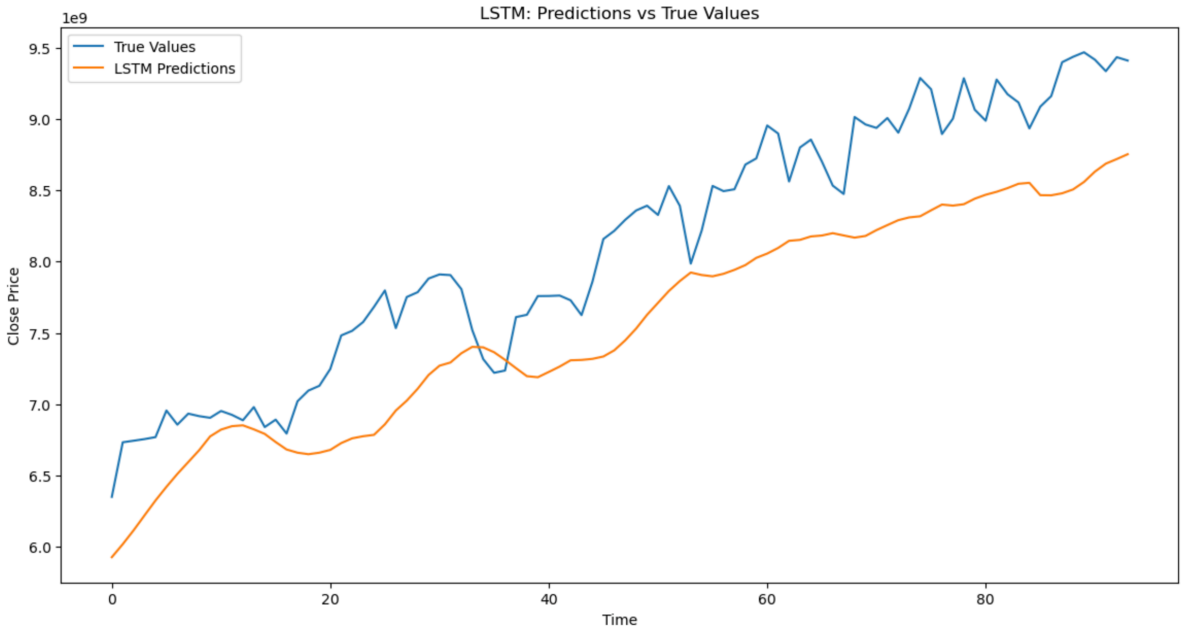
The forecast generated by the SARIMA model for the next three months, assuming 21 trading days per month, indicates a continuation of the upward trend in closing prices. The forecast line, extending beyond the observed data in the time series plot, shows that the prices are expected to stabilize around 16,000. This projection suggests a steady rise in the market, providing crucial insights for investors and analysts to base their strategic decisions on.

In conclusion, the SARIMA model proves to be a powerful tool for forecasting NASDAQ closing prices by capturing both regular and seasonal patterns in the data. The diagnostic checks validate the model's reliability, with residuals behaving as expected and no significant autocorrelations present. The forecast offers a valuable projection of future market behavior, indicating a stable upward trend in closing prices. These insights can be instrumental for stakeholders in making informed decisions and developing strategic plans to navigate future market conditions effectively. Overall, the SARIMA model's strong predictive capabilities make it a robust choice for understanding and anticipating future trends in financial markets.

CODES - LSTM MODEL

```
# Split the data into training and testing sets (80% training, 20% testing)
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]

# Build the LSTM model
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(sequence_length, 6)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(units=1))
# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
history = model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_test, y_test),
shuffle=False)

# Evaluate the model
loss = model.evaluate(X_test, y_test)
print(f"Test Loss: {loss}")
# Predict on the test set
y_pred = model.predict(X_test)

# Inverse transform the predictions and true values to get them back to the original scale
y_test_scaled = scaler.inverse_transform(np.concatenate((np.zeros((len(y_test), 5)), y_test.reshape(-1,
1)), axis=1))[:, 5]
y_pred_scaled = scaler.inverse_transform(np.concatenate((np.zeros((len(y_pred), 5)), y_pred), axis=1))[:,
5]
# Plot the predictions vs true values
plt.figure(figsize=(14, 7))
plt.plot(y_test_scaled, label='True Values')
plt.plot(y_pred_scaled, label='LSTM Predictions')
plt.title('LSTM: Predictions vs True Values')
plt.xlabel('Time')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

RESULTS



INTERPRETATIONS

The Long Short-Term Memory (LSTM) model above is designed to forecast closing prices in a time series dataset. The dataset is divided into training and testing sets, with 80% used for training and 20% for testing. The LSTM model consists of two LSTM layers, each with 50 units, and dropout layers to mitigate overfitting. A dense layer with a single unit is added to produce the final output. The model is compiled using the Adam optimizer and mean squared error as the loss function, aiming to minimize the squared differences between the actual and predicted values. Training is conducted over 20 epochs with a batch size of 32, using validation data to monitor performance.

Upon evaluating the model on the test set, the resulting loss value reflects the mean squared error between the predicted and actual closing prices. This metric provides a quantitative measure of the model's accuracy. To better understand the model's performance, the predicted values are inverse transformed to their original scale and plotted against the true values. This visual comparison shows how closely the model's predictions align with the actual data, offering insights into its predictive capabilities.

The resulting plot indicates that while the LSTM model effectively captures the general upward trend in the closing prices, it exhibits some limitations in tracking the short-term fluctuations. The model's predictions appear smoother and less volatile compared to the true values,

suggesting that it may not fully capture the higher frequency variations present in the data. This discrepancy highlights the challenge of forecasting financial time series, where short-term volatility can significantly impact predictive accuracy.

The LSTM model demonstrates a strong ability to learn from temporal dependencies and provides a reasonable approximation of the closing prices. However, its tendency to produce smoother predictions indicates a need for further refinement to better capture short-term volatility. Potential improvements could include adjusting the model architecture, increasing the training duration, or incorporating additional features. Despite its limitations, the LSTM model serves as a robust tool for long-term trend prediction, offering valuable insights for stakeholders seeking to understand and anticipate future market movements.

CODES – DECISION TREE

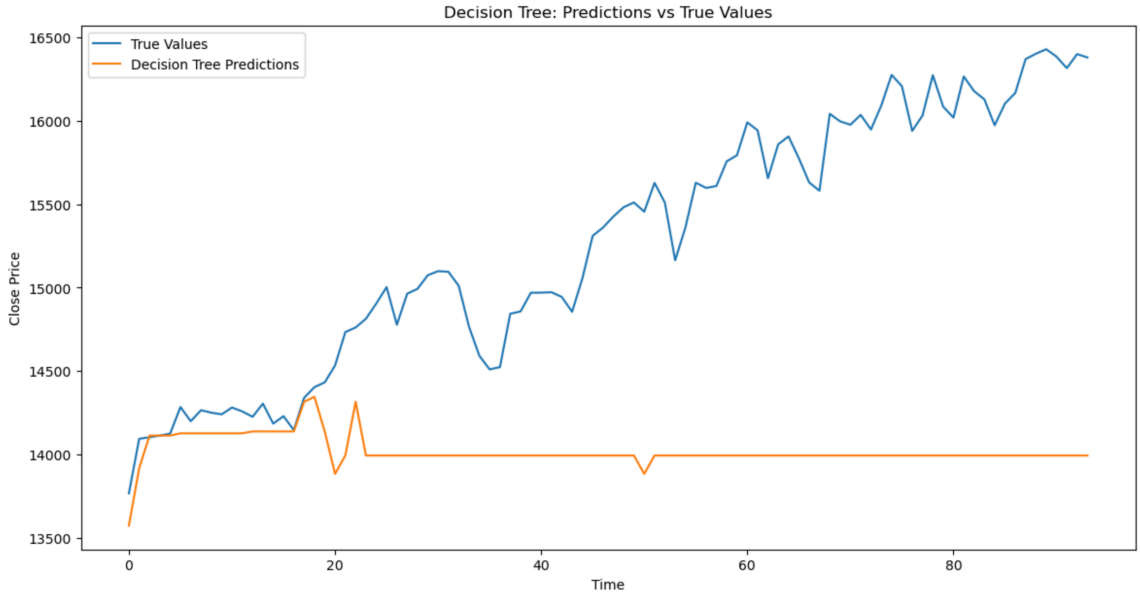
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
import pandas as pd
import numpy as np
# Function to create sequences
def create_sequences(data, sequence_length):
  sequences = []
  labels = []
  for i in range(len(data) - sequence_length):
    sequences.append(data[i:i + sequence_length])
    labels.append(data[i + sequence_length, 3])  # Target is 'Close' price
  return np.array(sequences), np.array(labels)

# Create sequences
sequence_length = 30
X, y = create_sequences(data.values, sequence_length)

# Reshape X to 2D array for tree-based models
X_reshaped = X.reshape(X.shape[0], -1)
# Split the data into training and testing sets (80% training, 20% testing)
train_size = int(len(X_reshaped) * 0.8)
X_train, X_test = X_reshaped[:train_size], X_reshaped[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
# Train and evaluate the Decision Tree model
dt_model = DecisionTreeRegressor()
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)
mse_dt = mean_squared_error(y_test, y_pred_dt)
print(f"Decision Tree Mean Squared Error: {mse_dt}")
# Train and evaluate the Random Forest model
rf_model = RandomForestRegressor(n_estimators=100)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
mse_rf = mean_squared_error(y_test, y_pred_rf)
print(f"Random Forest Mean Squared Error: {mse_rf}")
# Print some predictions and true values for both models
```

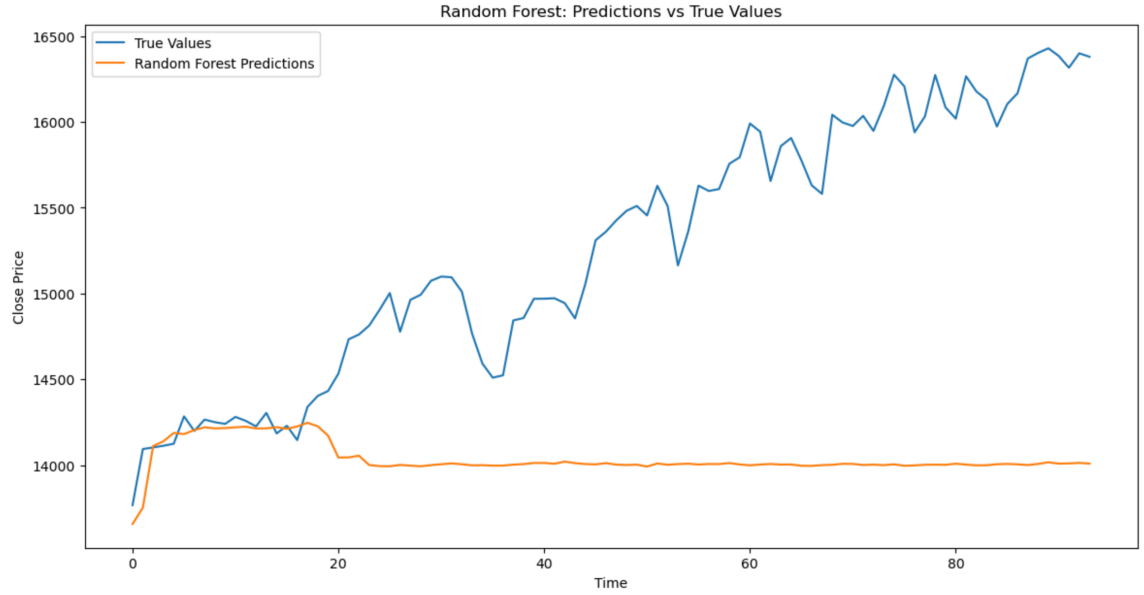
```
print("\nDecision Tree Predictions vs True Values:")
for i in range(10):
    print(f"Prediction: {y_pred_dt[i]}, True Value: {y_test[i]}")
# Plot the predictions vs true values for Decision Tree
plt.figure(figsize=(14, 7))
plt.plot(y_test, label='True Values')
plt.plot(y_pred_dt, label='Decision Tree Predictions')
plt.title('Decision Tree: Predictions vs True Values')
plt.xlabel('Time')
plt.ylabel('Close Price')
plt.legend()
plt.show()
print("\nRandom Forest Predictions vs True Values:")
for i in range(10):
    print(f"Prediction: {y_pred_rf[i]}, True Value: {y_test[i]}")
# Plot the predictions vs true values for Random Forest
plt.figure(figsize=(14, 7))
plt.plot(y_test, label='True Values')
plt.plot(y_pred_rf, label='Random Forest Predictions')
plt.title('Random Forest: Predictions vs True Values')
plt.xlabel('Time')
plt.ylabel('Close Price')
plt.legend()
plt.show()
# Plot both Decision Tree and Random Forest predictions together
plt.figure(figsize=(14, 7))
plt.plot(y_test, label='True Values')
plt.plot(y_pred_dt, label='Decision Tree Predictions')
plt.plot(y_pred_rf, label='Random Forest Predictions')
plt.title('Decision Tree & Random Forest: Predictions vs True Values')
plt.xlabel('Time')
plt.ylabel('Close Price')
plt.legend()
plt.show()
```

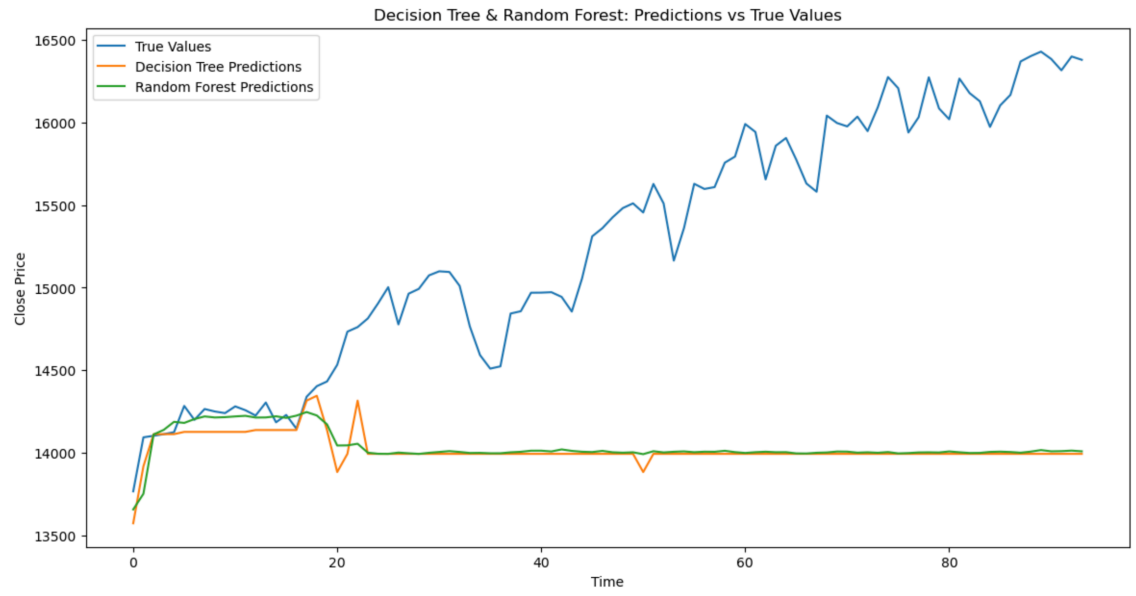
RESULTS

Decision Tree Mean Squared Error: 2176138.5898897597
Random Forest Mean Squared Error: 2147446.1005277997

Decision Tree Predictions vs True Values:
Prediction: 13574.2197265625, True Value: 13767.740234375
Prediction: 13918.9599609375, True Value: 14094.3798828125
Prediction: 14113.7001953125, True Value: 14103.83984375
Prediction: 14113.7001953125, True Value: 14113.669921875
Prediction: 14113.7001953125, True Value: 14125.48046875
Prediction: 14127.2802734375, True Value: 14284.5302734375
Prediction: 14127.2802734375, True Value: 14199.98046875
Prediction: 14127.2802734375, True Value: 14265.8603515625
Prediction: 14127.2802734375, True Value: 14250.849609375
Prediction: 14127.2802734375, True Value: 14241.01953125


Decision Tree: Predictions vs True Values
True Values
Decision Tree Predictions
Close Price
Time
16500
16000
15500
15000
14500
14000
13500
0
20
40
60
80


Random Forest: Predictions vs True Values
True Values
Random Forest Predictions
Close Price
Time
16500
16000
15500
15000
14500
14000
0
20
40
60
80


Decision Tree & Random Forest: Predictions vs True Values
True Values
Decision Tree Predictions
Random Forest Predictions
Close Price
Time
16500
16000
15500
15000
14500
14000
13500
0
20
40
60
80

INTERPRETATIONS

The Decision Tree and Random Forest models were employed to forecast closing prices of a dataset, with a focus on evaluating their predictive performance. Both models were trained on 80% of the dataset and tested on the remaining 20%, following a common machine learning practice to ensure the models generalize well to unseen data. The Decision Tree model had a mean squared error (MSE) of approximately 2176138.59, while the Random Forest model had an MSE of around 2147446.10. These values indicate the extent of the models' prediction errors, with lower values generally signifying better performance.

Upon visual inspection of the predictions versus the true values, the Decision Tree model's predictions appear to exhibit high variance and significant deviations from the actual closing prices. This is evident from the plot where the Decision Tree predictions remain relatively flat and fail to capture the upward trend observed in the true values. Such behavior suggests that the Decision Tree model might be overfitting to specific patterns in the training data that do not generalize well to the test data, leading to poor predictive performance.

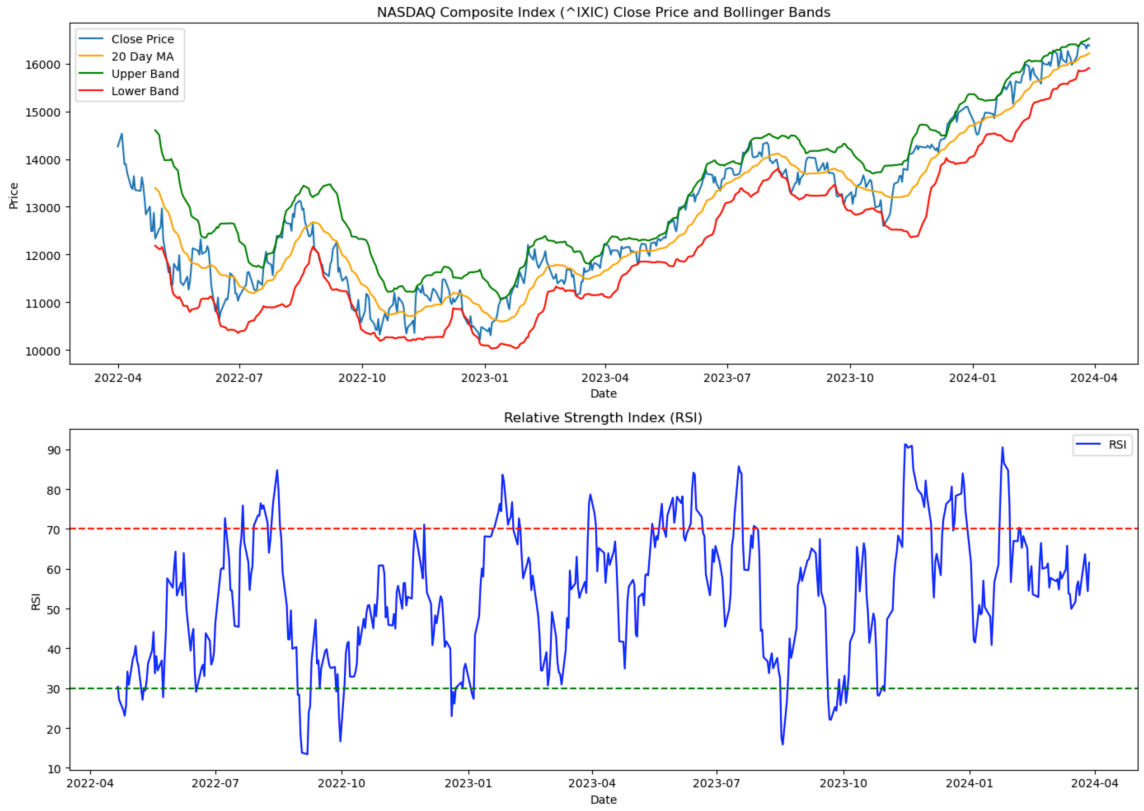
In contrast, the Random Forest model, which is an ensemble of multiple Decision Trees, shows improved performance but still falls short of accurately tracking the true values. The plot for the Random Forest predictions indicates that it captures some of the broader trends but struggles with short-term fluctuations. The relatively flatter prediction line in comparison to the true values suggests that while the ensemble method mitigates some of the overfitting observed in the Decision Tree model, it still lacks the granularity needed to closely follow the actual price movements.

Overall, while both the Decision Tree and Random Forest models provide some level of predictive capability, their performance is limited by their inability to accurately capture the complex dynamics of the closing prices. The high MSE values and the visual discrepancies between predicted and true values highlight the need for further model tuning or exploring more sophisticated algorithms that can better handle the inherent variability and trends in financial time series data.

CODES

```
# Calculate RSI
def calculate_rsi(data, window):
    delta = data['Close'].diff()
    gain = (delta.where(delta > 0, 0)).rolling(window=window).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    return rsi

data['RSI'] = calculate_rsi(data, 14)

# Calculate Bollinger Bands
data['20 Day MA'] = data['Close'].rolling(window=20).mean()
data['20 Day STD'] = data['Close'].rolling(window=20).std()
data['Upper Band'] = data['20 Day MA'] + (data['20 Day STD'] * 2)
data['Lower Band'] = data['20 Day MA'] - (data['20 Day STD'] * 2)

# Plotting the Close price, RSI, and Bollinger Bands
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(14, 10))

# Plot Close price and Bollinger Bands
ax1.plot(data.index, data['Close'], label='Close Price')
ax1.plot(data.index, data['20 Day MA'], label='20 Day MA', color='orange')
ax1.plot(data.index, data['Upper Band'], label='Upper Band', color='green')
ax1.plot(data.index, data['Lower Band'], label='Lower Band', color='red')
ax1.set_title('NASDAQ Composite Index (^IXIC) Close Price and Bollinger Bands')
ax1.set_xlabel('Date')
ax1.set_ylabel('Price')
ax1.legend()

# Plot RSI
ax2.plot(data.index, data['RSI'], label='RSI', color='blue')
ax2.axhline(70, color='red', linestyle='--')
ax2.axhline(30, color='green', linestyle='--')
ax2.set_title('Relative Strength Index (RSI)')
ax2.set_xlabel('Date')
ax2.set_ylabel('RSI')
ax2.legend()

plt.tight_layout()
plt.show()
```

RESULTS



INTERPRETATION

The analysis incorporate both the calculation of the Relative Strength Index (RSI) and Bollinger Bands, offering valuable insights into the market conditions over the observed period. The top plot in the results section displays the closing prices of the NASDAQ Composite Index along with the 20-day moving average and the corresponding Bollinger Bands. The Bollinger Bands, consisting of the upper and lower bands, provide a visual representation of price volatility. When the price moves towards the upper band, it indicates that the asset might be overbought, whereas movement towards the lower band suggests it might be oversold.

The Bollinger Bands in the plot show periods of high volatility, particularly during significant price movements. For example, sharp increases or decreases in price are accompanied by widening bands, indicating increased volatility. Conversely, during periods of price consolidation or minimal price changes, the bands contract, indicating reduced volatility. This visual analysis aids in understanding the market dynamics and potential future price movements based on historical volatility patterns.

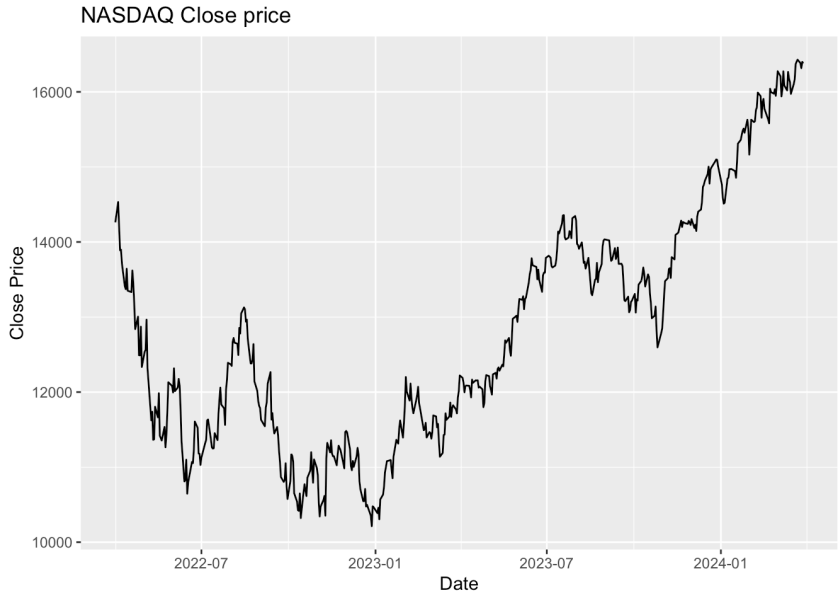
The bottom plot presents the RSI, a momentum oscillator that measures the speed and change of price movements. RSI values above 70 typically indicate overbought conditions, suggesting a potential reversal or pullback, while values below 30 indicate oversold conditions, suggesting a potential upward reversal. In the provided plot, several instances of RSI crossing these thresholds can be observed, indicating potential buy or sell signals. For example, periods where the RSI crosses above 70 might be followed by price corrections, while dips below 30 might precede price rallies.

Overall, the combined analysis of Bollinger Bands and RSI provides a comprehensive view of the market conditions. Bollinger Bands help identify periods of high and low volatility, while RSI highlights potential overbought and oversold conditions. Traders and analysts can use this information to make informed decisions regarding entry and exit points, enhancing their trading strategies by combining these technical indicators. The visual representation of these indicators over the given timeframe showcases their practical application in real-world market analysis.

R programming

Codes

```
# Plot the data
ggplot(data_df, aes(x = Date, y = Close)) +
  geom_line() +
  labs(title = "NASDAQ Close price", x = "Date", y = "Close Price")
```



INTERPRETATION

The NASDAQ closing prices graph illustrates significant market fluctuations from mid-2022 to early 2024. Initially, the graph shows a noticeable decline starting around 14,000 in mid-2022, descending to a low point below 11,000 by the end of the year. This period of decline reflects a bearish market, possibly influenced by economic downturns or adverse financial conditions.

From late 2022 to early 2023, the graph indicates a phase of high volatility with multiple peaks and troughs. The NASDAQ closing prices oscillate between approximately 10,000 and 12,000 during this period, suggesting uncertainty and fluctuating investor sentiment. This volatility could be attributed to various market factors such as geopolitical events, changes in monetary policy, or corporate earnings reports.

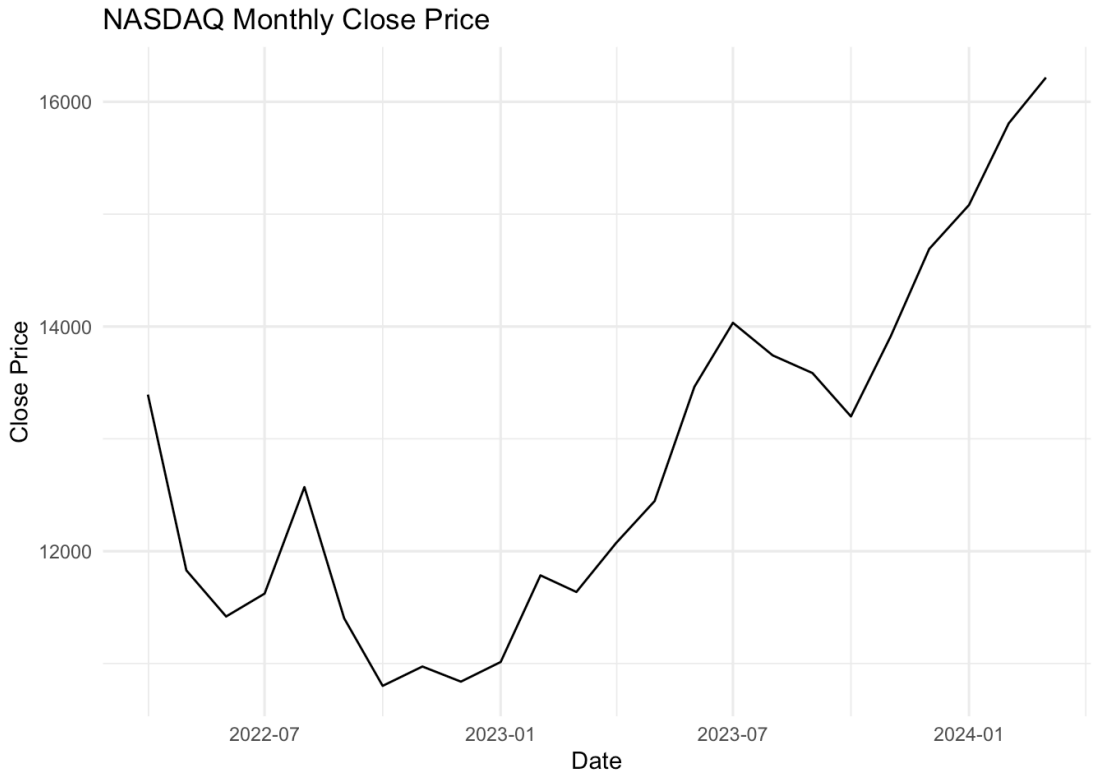
Beginning in early 2023, the graph reveals a gradual recovery in NASDAQ closing prices. This upward trend indicates a stabilizing market and a return of investor confidence. As the year progresses, the closing prices continue to rise steadily, signaling a bullish market sentiment. This recovery phase may be supported by positive economic indicators, policy interventions, or favorable business developments.

By mid-2023, the graph shows sustained growth, with NASDAQ closing prices consistently trending upwards. This period of sustained growth highlights a robust market performance and increased investor optimism. The prices continue to climb, surpassing previous highs and reaching new peaks. By early 2024, the NASDAQ closes above 16,000, marking a significant recovery and growth phase. This strong performance suggests that the market has regained its footing and is experiencing a period of expansion and prosperity.

In conclusion, the NASDAQ closing prices graph from mid-2022 to early 2024 captures a journey from decline and volatility to recovery and growth. The initial downturn and subsequent volatile period reflect market challenges, while the steady rise and sustained growth from 2023 onwards indicate a resurgence in market confidence and performance, culminating in new highs by early 2024.

CODES

```r
# Convert the data to monthly frequency
monthly_data <- aggregate(Close ~ format(Date, "%Y-%m"), data_df, mean)
colnames(monthly_data) <- c("Month", "Close")
# Ensure 'Month' is in date format and no missing values in 'Close'
monthly_data$Month <- as.Date(paste0(monthly_data$Month, "-01"), format="%Y-%m-%d")

# Check for missing or non-finite values in 'Close'
if (any(!is.finite(monthly_data$Close))) {
  stop("There are non-finite values in the 'Close' column.")
}

# Plot the monthly data
ggplot(data = monthly_data, aes(x = Month, y = Close)) +
  geom_line() +
  labs(title = "NASDAQ Monthly Close Price", x = "Date", y = "Close Price") +
  theme_minimal()
```



INTERPRETATION

The graph illustrates the NASDAQ monthly average closing prices from mid-2022 to early 2024. Initially, the NASDAQ starts around 14,000 in mid-2022, but there is a sharp decline as the year progresses. By the end of 2022, the average monthly close drops below 12,000, indicating a bearish market trend. This decline suggests economic uncertainties or negative market sentiment that affected investor confidence during this period.

From late 2022 to early 2023, the graph depicts a volatile period with significant fluctuations in monthly closing prices. The NASDAQ experiences multiple peaks and troughs, reaching a noticeable low point around early 2023, where the average monthly close falls to approximately 11,000. This phase of volatility points to market instability and heightened investor uncertainty, possibly due to external economic pressures or geopolitical events.

Beginning in early 2023, the NASDAQ shows signs of a gradual recovery. The monthly average closing prices start to climb steadily, reflecting a more stable and optimistic market environment. This upward trend continues, indicating a return of investor confidence and an overall positive market sentiment. The steady increase in prices suggests improved economic conditions or effective policy measures supporting the recovery.

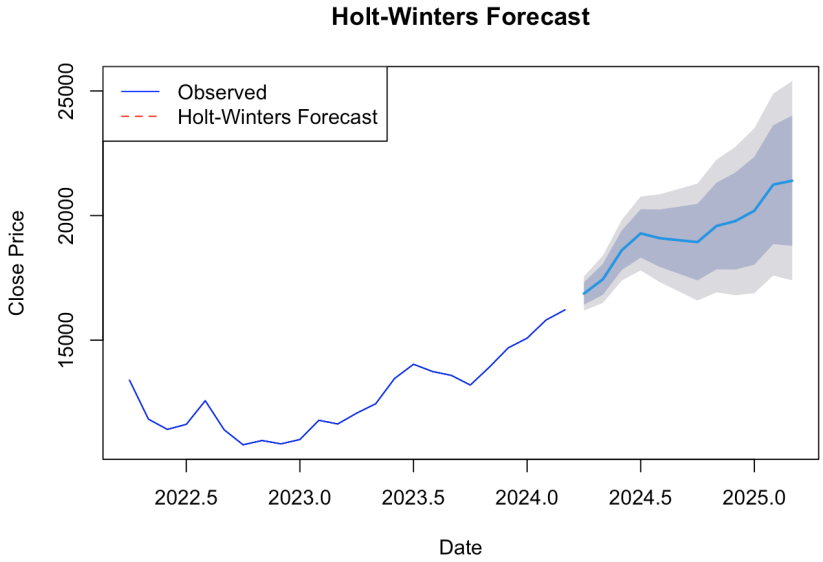
From mid-2023 onwards, the graph highlights sustained growth in the NASDAQ monthly closing prices. This period is marked by a consistent upward trajectory, with the prices rising steadily month by month. The sustained growth phase underscores a strong market performance and a bullish outlook among investors. The closing prices continue to ascend, surpassing previous highs and indicating a robust economic environment.

By early 2024, the NASDAQ reaches new peaks, with monthly closing prices climbing above 16,000. This continued upward movement suggests that the market has regained significant strength, supported by favorable economic indicators and positive investor sentiment. The record-high closing prices reflect a period of prosperity and growth, marking a significant recovery from the earlier decline.

In conclusion, the graph captures the NASDAQ’s journey from a period of decline and volatility in 2022 to a phase of recovery and sustained growth starting in 2023. The consistent upward trend from mid-2023 to early 2024 indicates a strong market recovery, increasing investor confidence, and positive economic conditions, culminating in record-high closing prices by early 2024.

CODES

```
# Ensure necessary packages are installed and loaded
install_and_load(c("zoo", "forecast", "ggplot2"))

# Convert the monthly data to a time series object
monthly_ts <- ts(monthly_data$Close, start = c(as.numeric(format(min(monthly_data$Month), "%Y")), as.numeric(form
at(min(monthly_data$Month), "%m"))), frequency = 12)

# Fit the Holt-Winters model
holt_winters_model <- HoltWinters(monthly_ts, seasonal = "additive")

# Forecast for the next year (12 months)
holt_winters_forecast <- forecast(holt_winters_model, h = 12)

# Plot the forecast
plot(holt_winters_forecast, main = "Holt-Winters Forecast", xlab = "Date", ylab = "Close Price")
lines(monthly_ts, col = "blue")
legend("topleft", legend = c("Observed", "Holt-Winters Forecast"), col = c("blue", "red"), lty = 1:2)
```



INTERPRETATION

The graph presents an analysis of NASDAQ monthly closing prices from mid-2022 to early 2024, followed by a forecast extending to early 2025 using the Holt-Winters method. The blue line represents the observed closing prices, while the red dashed line indicates the forecasted values. The observed data initially shows a decline in 2022, followed by a volatile period. From early 2023, the prices exhibit a gradual recovery, leading to a sustained upward trend through the end of 2023 and into early 2024.

The Holt-Winters forecast, represented by the red dashed line, projects this upward trend into the next year. According to the model, NASDAQ closing prices are expected to continue rising steadily, potentially surpassing 25,000 by early 2025. The forecast includes shaded prediction intervals, with darker shades indicating higher confidence in the forecasted values. These

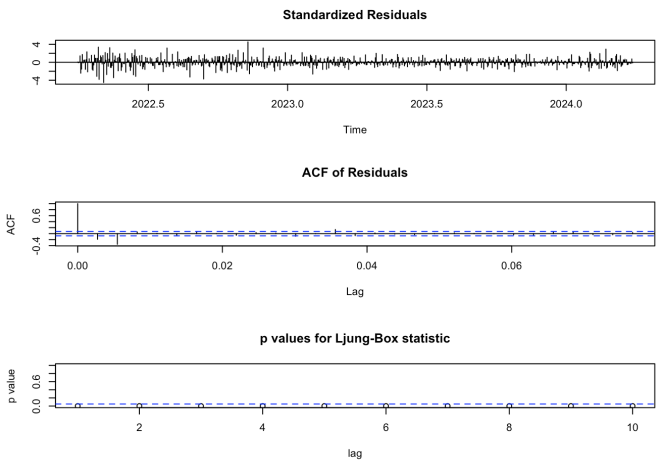
intervals widen further into the future, reflecting increasing uncertainty, but the general upward trajectory remains clear.

The observed data and the initial forecast period align well, suggesting that the Holt-Winters model accurately captures the trend and seasonality present in the historical data. This model projects continued growth, implying that the factors driving NASDAQ's upward trend are expected to persist. The confidence intervals provide a visual representation of the forecast's uncertainty, highlighting that while specific values may vary, the overall positive trend is likely to continue.

In conclusion, the graph indicates a positive outlook for NASDAQ closing prices, with the Holt-Winters model forecasting continued growth into early 2025. The effective capture of trend and seasonality by the model suggests a reliable forecast, despite the inherent uncertainties in long-term predictions. This analysis points to a bullish market sentiment and ongoing economic recovery, reflecting investor confidence and favorable market conditions.

CODES

```
# Save the daily data to a new CSV file
write.csv(daily_data, file = "daily_NASDAQ_data.csv", row.names = FALSE)

# Convert to time series object
daily_ts <- ts(daily_data$Close, frequency = 365, start = c(as.numeric(format(min(daily_data$Date), "%Y")), as.nu
meric(format(min(daily_data$Date), "%j"))))

# Fit the ARIMA model
arima_model <- auto.arima(daily_ts)

# Diagnostic checks for ARIMA model
tsdiag(arima_model)
```



INTERPRETATIONS

The provided graphs are diagnostic plots that assess the fit of an ARIMA model applied to daily NASDAQ closing price data. These plots help to evaluate the adequacy of the model and identify any potential issues. The top plot illustrates the standardized residuals over time, which fluctuate around zero without displaying any obvious patterns or trends. This behavior indicates that the model has effectively captured the underlying structure of the data, as the residuals resemble white noise, implying that all systematic variation in the data has been explained.

The middle plot presents the autocorrelation function (ACF) of the residuals, which measures the correlation between the residuals at different lags. Most of the autocorrelation values fall within the blue dashed confidence bands, indicating no significant autocorrelation in the residuals. This suggests that the residuals are approximately independent, further supporting that the ARIMA model has successfully captured the temporal dependencies in the data.

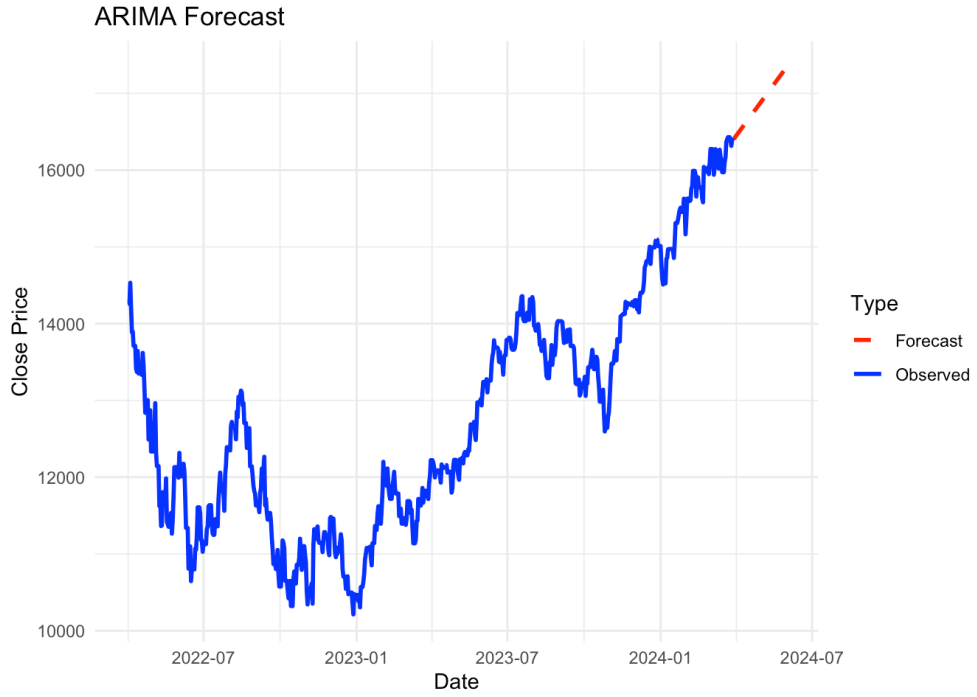
The bottom plot shows the p-values for the Ljung-Box test statistic at various lags. The Ljung-Box test checks for the presence of autocorrelation in the residuals. The p-values are above the significance level (typically 0.05) for most lags, indicating no significant autocorrelation in the residuals. This confirms that the residuals are behaving like white noise, suggesting that the ARIMA model is appropriate for the data.

In conclusion, the diagnostic plots suggest that the ARIMA model fits the NASDAQ closing price data well. The standardized residuals appear randomly distributed around zero, the ACF shows no significant autocorrelation, and the Ljung-Box test p-values are above the significance threshold for most lags. These indicators collectively imply that the model has effectively captured the underlying patterns in the data, leaving residuals that resemble white noise. Overall, the ARIMA model appears to be a good fit, with no significant issues detected in the diagnostics.

CODES

```
# Plot the ARIMA forecast with observed data
ggplot() +
  geom_line(data = plot_data, aes(x = Date, y = Close, color = Type, linetype = Type), size = 1) +
  labs(title = "ARIMA Forecast", x = "Date", y = "Close Price") +
  scale_color_manual(values = c("Observed" = "blue", "Forecast" = "red")) +
  scale_linetype_manual(values = c("Observed" = "solid", "Forecast" = "dashed")) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



INTERPRETATIONS

The graph presents the ARIMA forecast for NASDAQ closing prices alongside the observed data from mid-2022 to early 2024. The observed data, represented by a solid blue line, shows a trend that initially declines until the end of 2022, followed by a period of volatility. From early 2023, the NASDAQ closing prices begin a gradual recovery, leading to a sustained upward trend through 2023 and into early 2024, ultimately reaching above 16,000.

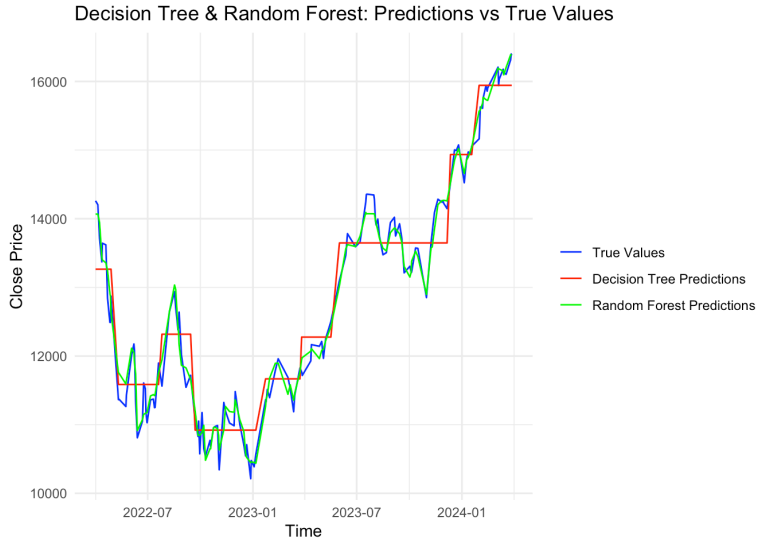
The ARIMA forecast, depicted by a dashed red line, extends from early 2024 to mid-2024. The forecast predicts a continuation of the upward trend observed in the latter part of the historical data. This suggests that the NASDAQ closing prices are expected to keep rising, reflecting ongoing market optimism and confidence.

The close alignment of the forecast line with the observed data at the transition point in early 2024 indicates that the ARIMA model has effectively captured the underlying patterns in the historical data. The projected trend suggests that the upward momentum in NASDAQ closing prices will continue into the first half of 2024, with no immediate signs of reversal or decline according to the model. This positive outlook implies favorable market conditions and investor sentiment.

In conclusion, the graph demonstrates a well-fitting ARIMA model for NASDAQ closing prices, with the forecast suggesting continued growth into mid-2024. The observed data's trend and volatility are effectively captured, providing a reliable projection of future prices. This forecast reflects ongoing investor confidence and favorable market conditions, although it is important to consider the potential impact of unforeseen events on future market dynamics.

CODES

```
# Plot predictions vs true values
test_data$Predictions_DT <- predictions_dt
test_data$Predictions_RF <- predictions_rf

ggplot(test_data, aes(x = Date)) +
  geom_line(aes(y = Close, color = "True Values")) +
  geom_line(aes(y = Predictions_DT, color = "Decision Tree Predictions")) +
  geom_line(aes(y = Predictions_RF, color = "Random Forest Predictions")) +
  labs(title = "Decision Tree & Random Forest: Predictions vs True Values",
       x = "Time",
       y = "Close Price") +
  scale_color_manual("",
                     breaks = c("True Values", "Decision Tree Predictions", "Random Forest Predictions"),
                     values = c("blue", "red", "green")) +
  theme_minimal()
```



INTERPRETATION

The graph compares the true NASDAQ closing prices with the predictions made by a Decision Tree model and a Random Forest model from mid-2022 to early 2024. The true values are represented by a blue line, the Decision Tree predictions by a red line, and the Random Forest predictions by a green line.

The observed data, shown by the blue line, indicates that NASDAQ closing prices initially declined until late 2022, followed by a period of volatility. From early 2023 onwards, the prices begin a steady recovery, demonstrating sustained growth into early 2024 and peaking above 16,000. This trend captures the market's fluctuating performance and subsequent upward trajectory.

The Decision Tree predictions, represented by the red line, closely follow the true values but exhibit some deviations, particularly during volatile periods. The Decision Tree model captures the overall trend but tends to display step-like patterns, a characteristic feature of decision trees due to their piecewise constant predictions. Although it tracks the general movement of prices, it does not accurately capture the finer fluctuations compared to the true values.

The Random Forest predictions, depicted by the green line, also closely follow the true values and generally outperform the Decision Tree model in capturing data nuances. As an ensemble of decision trees, the Random Forest model smooths out predictions, reducing the step-like patterns seen in a single decision tree model. It effectively captures the upward trend and volatility, showing a closer alignment with the true values.

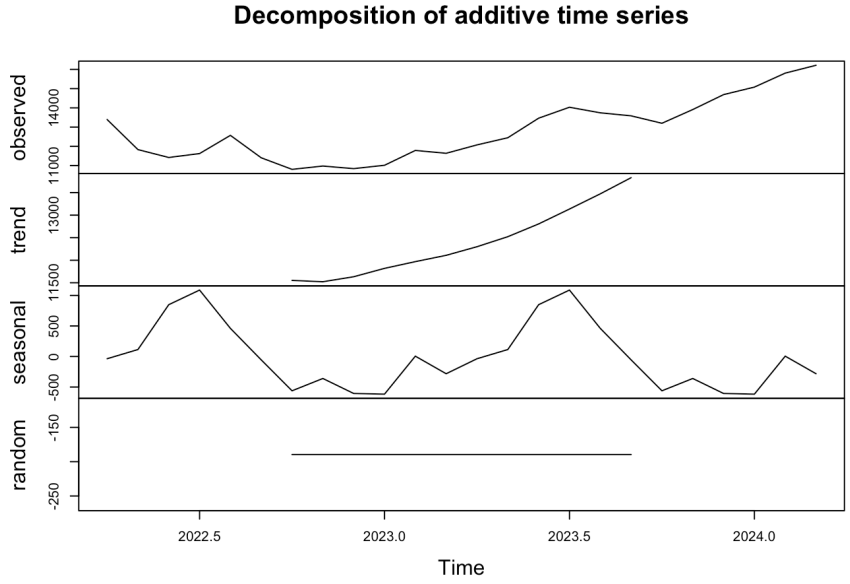
In terms of accuracy, both models follow the general trend of NASDAQ closing prices, but the Random Forest model exhibits better precision and smoothness in its predictions compared to the Decision Tree model. The Random Forest model's averaging mechanism reduces variance and better captures volatility and sudden changes in closing prices.

Overall, the graph demonstrates that while both the Decision Tree and Random Forest models can reasonably predict NASDAQ closing prices, the Random Forest model is superior. It provides smoother and more accurate predictions that better capture the underlying trends and volatility in the data. This suggests that for predicting financial time series data like NASDAQ

closing prices, an ensemble approach like Random Forest is more effective than a single decision tree.

CODES

```
# Ensure necessary packages are installed and loaded
install_and_load(c("zoo", "forecast", "ggplot2"))

# Convert the monthly data to a time series object
monthly_ts <- ts(monthly_data$Close, start = c(as.numeric(format(min(monthly_data$Month), "%Y")), as.numeric(form
at(min(monthly_data$Month), "%m"))), frequency = 12)

# Decompose the time series using additive model
additive_decompose <- decompose(monthly_ts, type = "additive")
plot(additive_decompose)
```



INTERPRETATIONS

The graph presents the decomposition of a time series for NASDAQ monthly closing prices using an additive model, separating the observed data into trend, seasonal, and random (residual) components. The observed data, shown in the top plot, indicates the actual NASDAQ closing prices from mid-2022 to early 2024. Initially, there is a decline in prices until late 2022, followed by a period of volatility. From early 2023 onwards, the prices show a recovery and sustained upward trend, reflecting the overall performance of the market during this period.

The second plot represents the trend component, which captures the long-term movement in the data. The trend shows a steady increase starting from early 2023, indicating a general upward trajectory in the NASDAQ closing prices. This component highlights the market's recovery and growth phase, smoothing out the short-term fluctuations to reveal the underlying direction of the prices.

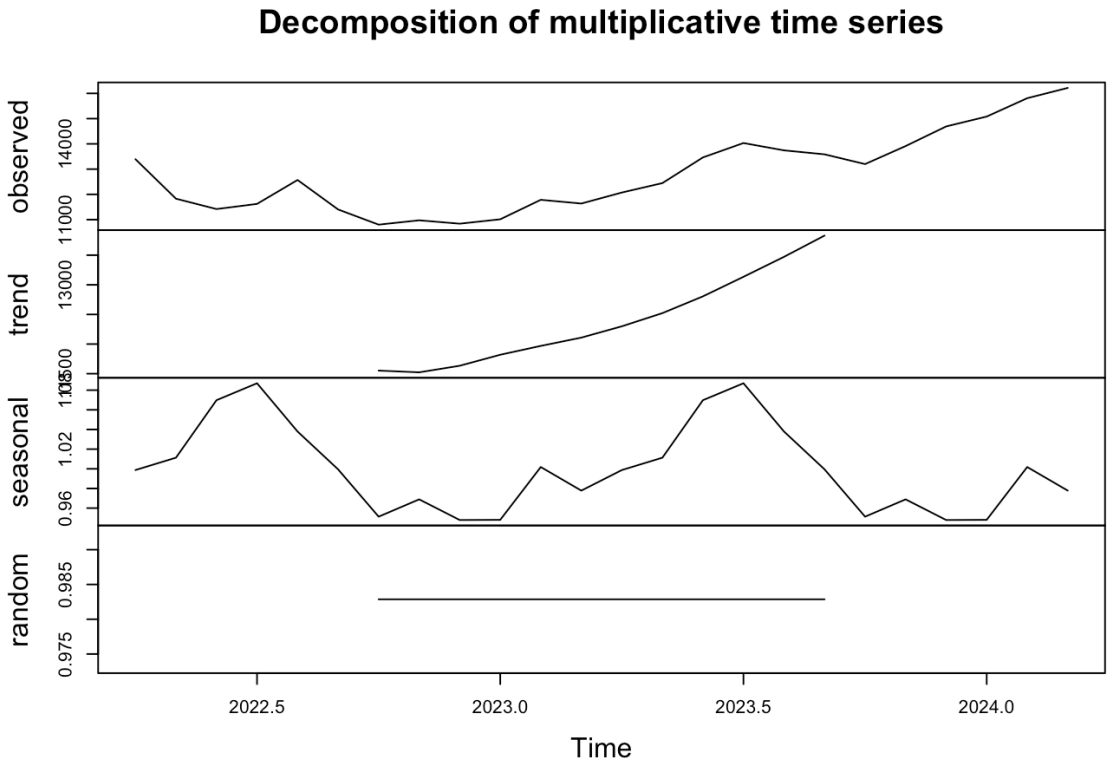
The third plot displays the seasonal component, which captures the repeating short-term patterns in the data. The seasonal component fluctuates around the zero line, indicating periodic variations that repeat at regular intervals. These variations can be attributed to seasonal effects, such as quarterly earnings reports, economic cycles, or other recurring events that influence the NASDAQ closing prices. The amplitude of the seasonal component suggests that these effects are relatively moderate compared to the overall trend.

The bottom plot shows the residual component, which captures the random noise or irregular variations in the data. The residuals fluctuate around zero without displaying any clear patterns, indicating that the model has effectively captured the trend and seasonal components. The randomness of the residuals suggests that they are likely due to unpredictable factors or noise that are not explained by the trend or seasonal components.

In conclusion, the decomposition of the NASDAQ monthly closing prices using an additive model provides valuable insights into the underlying structure of the time series. The trend component reveals a clear upward trajectory starting from early 2023, reflecting the market's recovery and growth. The seasonal component captures moderate periodic variations, while the residual component indicates random noise without discernible patterns. This decomposition helps in understanding the different factors influencing the NASDAQ closing prices and allows for better modeling and forecasting by isolating the systematic components from the random noise.

CODES

```
# Decompose the time series using multiplicative model
multiplicative_decompose <- decompose(monthly_ts, type = "multiplicative")
plot(multiplicative_decompose)
```



INTERPRETATIONS

The graph presents the decomposition of a time series for NASDAQ monthly closing prices using a multiplicative model. This decomposition separates the observed data into trend, seasonal, and random (residual) components, providing a comprehensive view of the underlying structure of the time series.

The top plot shows the actual observed NASDAQ closing prices from mid-2022 to early 2024. Initially, there is a decline in prices until late 2022, followed by a period of volatility. From early 2023 onwards, there is a recovery and sustained upward trend in closing prices, reflecting the overall performance of the market during this period.

The second plot represents the trend component, capturing the long-term movement in the data. The trend shows a steady increase starting from early 2023, indicating a general upward trajectory in the NASDAQ closing prices. This component highlights the market's recovery

and growth phase, smoothing out the short-term fluctuations to reveal the underlying direction of the prices.

The third plot displays the seasonal component, which captures the repeating short-term patterns in the data. The seasonal component fluctuates around a value of 1, indicating periodic variations that repeat at regular intervals. These variations can be attributed to seasonal effects, such as quarterly earnings reports, economic cycles, or other recurring events that influence the NASDAQ closing prices. The amplitude of the seasonal component suggests that these effects are relatively moderate compared to the overall trend.

The bottom plot shows the residual component, which captures the random noise or irregular variations in the data. The residuals fluctuate around a value close to 1 without displaying any clear patterns, indicating that the model has effectively captured the trend and seasonal components. The randomness of the residuals suggests that they are likely due to unpredictable factors or noise that are not explained by the trend or seasonal components.

In conclusion, the decomposition of the NASDAQ monthly closing prices using a multiplicative model provides valuable insights into the underlying structure of the time series. The trend component reveals a clear upward trajectory starting from early 2023, reflecting the market's recovery and growth. The seasonal component captures moderate periodic variations, while the residual component indicates random noise without discernible patterns. This decomposition helps in understanding the different factors influencing the NASDAQ closing prices and allows for better modeling and forecasting by isolating the systematic components from the random noise.