VIRGINIA COMMONWEALTH UNIVERSITY

Statistical Analysis & Modelling

A6a – ARIMA

Time Series I Data

Using R

Submitted by

MONIKA SARADHA

#V01068784

Date of Submission: 01/08/2023

Table of Contents

1. Introduction

The foreign exchange market plays a crucial role in the global economy, where the exchange rates of various currencies are subject to constant fluctuations and trends. Understanding and analyzing the dynamics of currency exchange rates are vital for businesses, investors, and policymakers alike. In this assignment, we delve into the world of time series analysis to gain valuable insights into the historical trading data of a financial instrument, focusing on its price movements and trends.

1.1. About the Data

The dataset provided for this time series analysis comprises historical trading data for a financial instrument. It includes daily records of the financial instrument's price movements and trading volume. Each observation corresponds to a specific date, capturing the opening and closing prices, as well as the highest and lowest prices during the trading session. Additionally, the dataset includes the percentage change in price from the previous trading day. The data is ideal for time series analysis, enabling the exploration of underlying patterns, trends, and seasonality within the financial instrument's price data. By analyzing this dataset, valuable insights can be derived, guiding businesses, investors, and financial analysts in making data-driven decisions and navigating the complexities of the financial market.

Date: This column represents the date on which the trading data was recorded. It provides a chronological sequence of trading days.

- Price: The "Price" column represents the closing price of the financial instrument on each trading day. It indicates the value at which the instrument was traded at the end of the trading session.
- Open: The "Open" column denotes the opening price of the financial instrument at the beginning of the trading session on each trading day.
- High: The "High" column records the highest price reached by the financial instrument during the trading session on each trading day.
- Low: The "Low" column indicates the lowest price attained by the financial instrument during the trading session on each trading day.
- Vol.: The "Vol." (Volume) column represents the trading volume of the financial instrument on each trading day. It indicates the total number of units or shares of the instrument that were traded.

- Change %: The "Change %" column represents the percentage change in the price of the financial instrument from the previous trading day to the current trading day. It reflects the daily price movement and direction.

1.2. Objective

The objectives of this analysis are to perform a comprehensive time series analysis of the financial instrument's price data, generate forecasts for future price movements, and explore the effectiveness of various models in capturing the underlying patterns and seasonality in the data. The results will contribute to informed decision-making for businesses, investors, and financial analysts, aiding in risk management and maximizing opportunities in the financial market.

- Clean the dataset by addressing outliers and missing values to ensure data accuracy.
- Visualize the financial instrument's price trend through a neatly named line graph.
- Convert the daily data to monthly intervals and perform time series decomposition using additive and multiplicative models.
- Fit a Holt-Winters model to capture trend, seasonality, and level components, generating one-year forecasts.
- Apply ARIMA to daily data and conduct diagnostic checks to validate model accuracy.
- Explore seasonal ARIMA for potential improved data fit and interpret results.
- Forecast financial instrument's price series for the next three months using the ARIMA model.
- Extend analysis by fitting ARIMA to the monthly series, gaining deeper insights at the monthly level.
- Provide valuable insights for businesses, investors, and financial analysts to inform decision-making, risk management, and market positioning.

1.3. Business Significance

- Informed Investment Decisions: Forecasted trends assist investors in making informed choices for optimal investment strategies.
- Risk Management: Identification of potential risks enables businesses to implement effective risk management measures.
- Financial Planning: Forecasted prices aid in budgeting and resource allocation for improved financial planning.
- International Trade: Currency exchange rate forecasting facilitates pricing decisions and foreign exchange risk management in international trade.

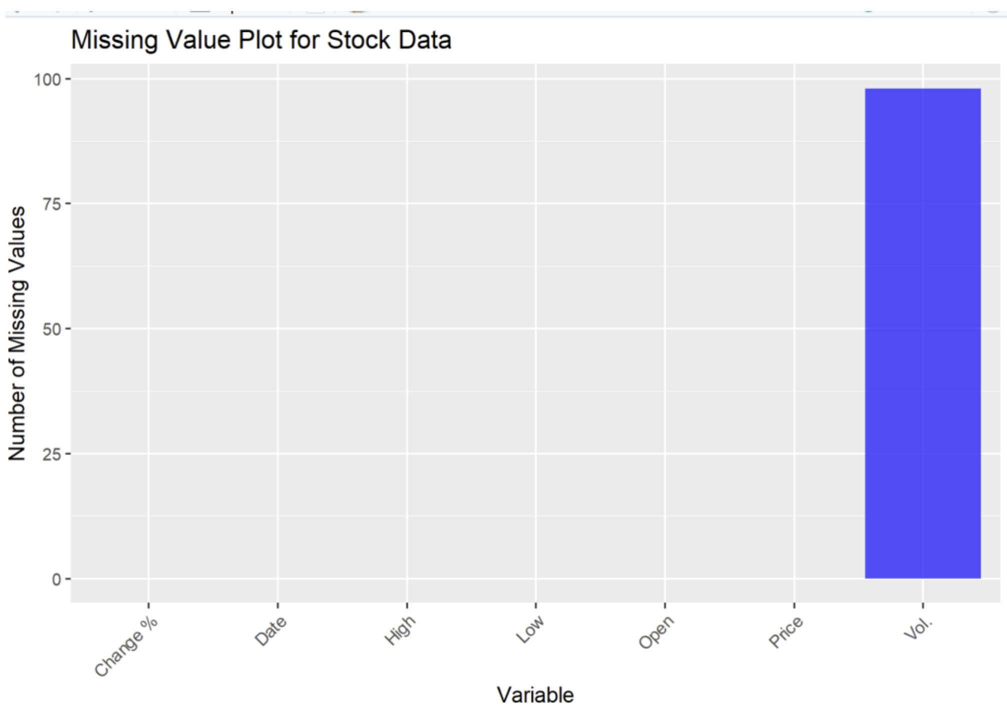
- Strategic Positioning: Forecasted price movements help businesses and traders strategically position themselves in the market.
- Policy Decisions: Policymakers can use the analysis to gauge economic health and devise appropriate measures for stabilization.
- Economic Indicators: The analysis contributes to assessing economic performance and its impact on inflation and interest rates.
- Global Business Strategy: Multinational corporations can develop global strategies considering currency fluctuations in different regions.

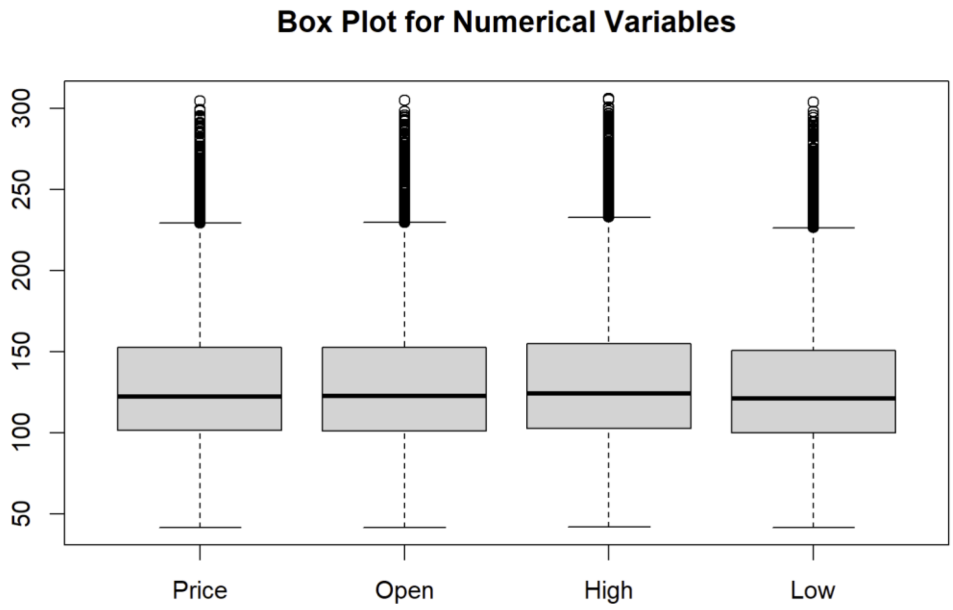
The insights provided by this assignment empower stakeholders to optimize their decisions, manage risks, and capitalize on opportunities in the dynamic financial market, fostering business success and economic stability.

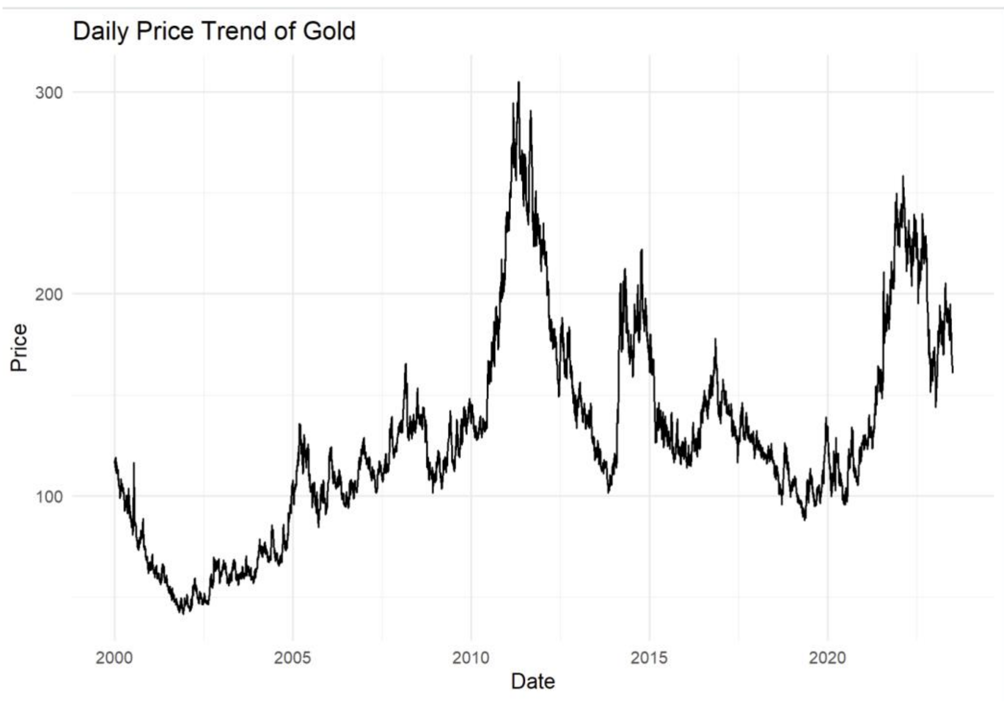
2. Results

2.1. R- output and Interpretation

2.2. Data Cleaning, Outliers, and Missing Values







Inference:

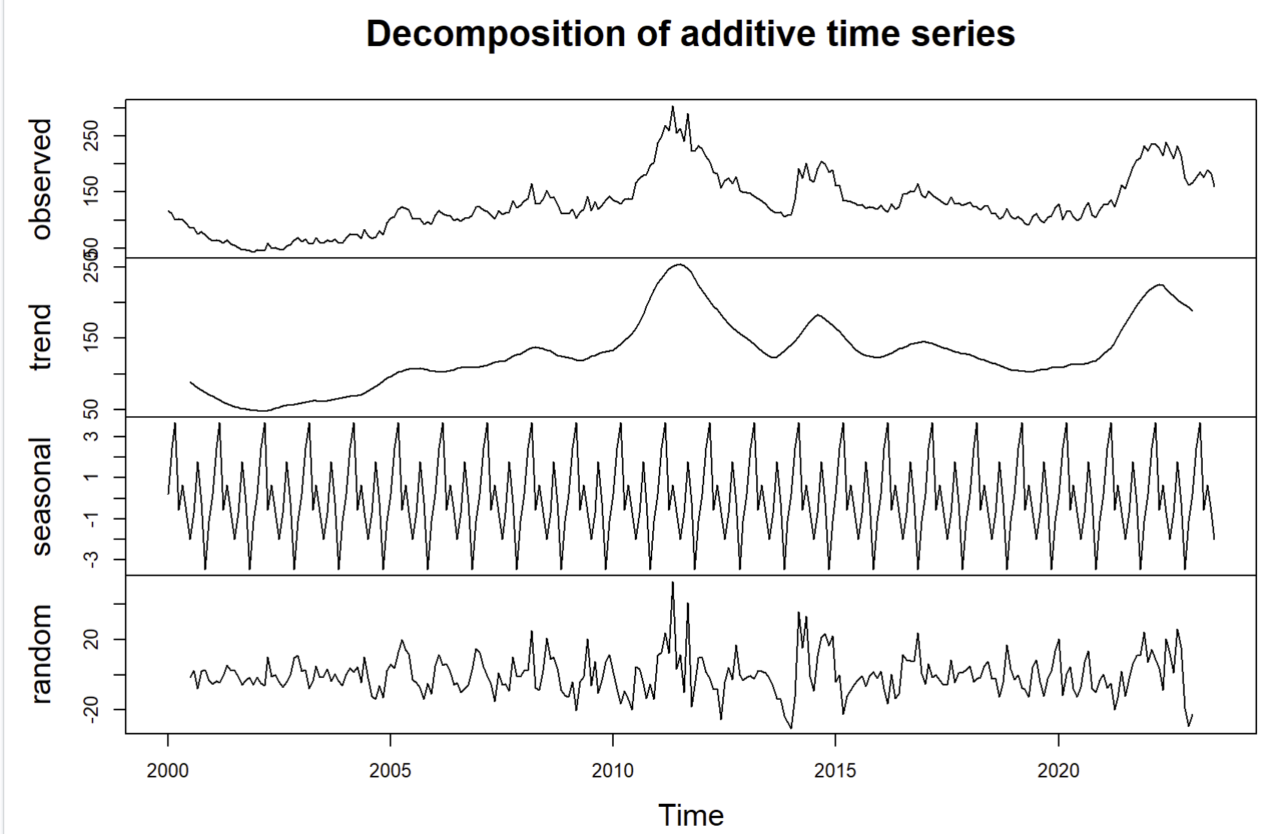
Data Cleaning:

- The data has been successfully cleaned, and there are no missing values in columns other than "Vol." The "Vol." column has 98 missing values.
- The box plot for numerical variables (Price, Open, High, Low) allows us to visually inspect potential outliers.

Missing values in the numerical columns have been interpolated using linear interpolation. A line graph was plotted to visualize the daily price trend of gold over time. The graph provides a clear overview of the fluctuation patterns in the gold prices.

- The dataset has been divided into train and test sets with a 70-30 split.
- The set.seed(123) function ensures reproducibility of the random split.

2.3. Data Conversion to Monthly and Time Series Decomposition

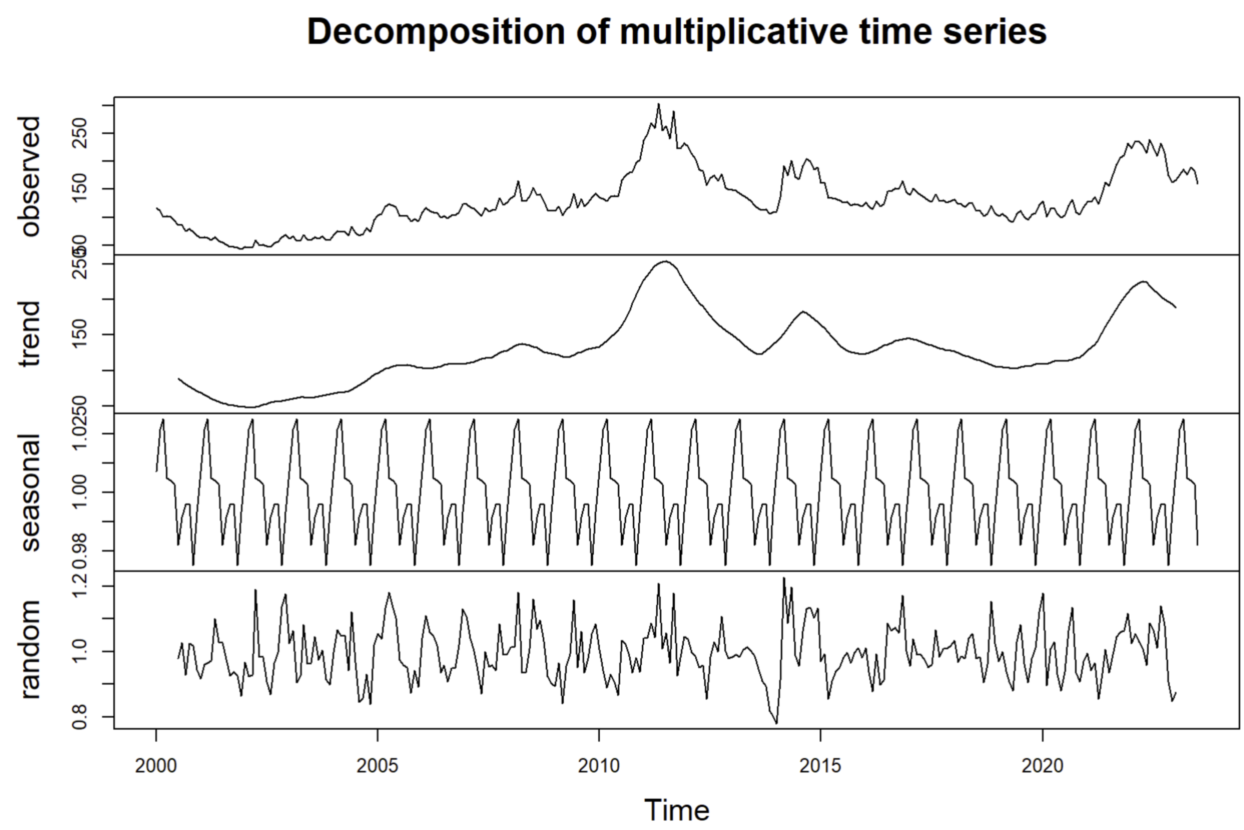


Inference:

The time series data exhibits an increasing trend, with clear seasonal fluctuations and random noise. The trend indicates a long-term growth pattern, while the seasonal component represents regular recurring patterns. The random component accounts for unpredictable variability. This decomposition allows for better forecasting by separating the trend, seasonal, and random effects.

- The time series data shows a relatively steady increasing trend over the years, indicating consistent growth in the underlying process.
- Seasonal fluctuations occur regularly with a fixed period, suggesting the presence of multiple seasonal patterns within each year.
- The seasonal component appears to have a relatively constant amplitude, indicating that the seasonal effects do not change significantly over time.
- The random component represents irregular fluctuations or noise around the trend and seasonal patterns. These random fluctuations do not follow any discernible pattern.

- The decomposition of the time series into trend, seasonal, and random components aids in understanding the underlying patterns and enables more accurate forecasting.

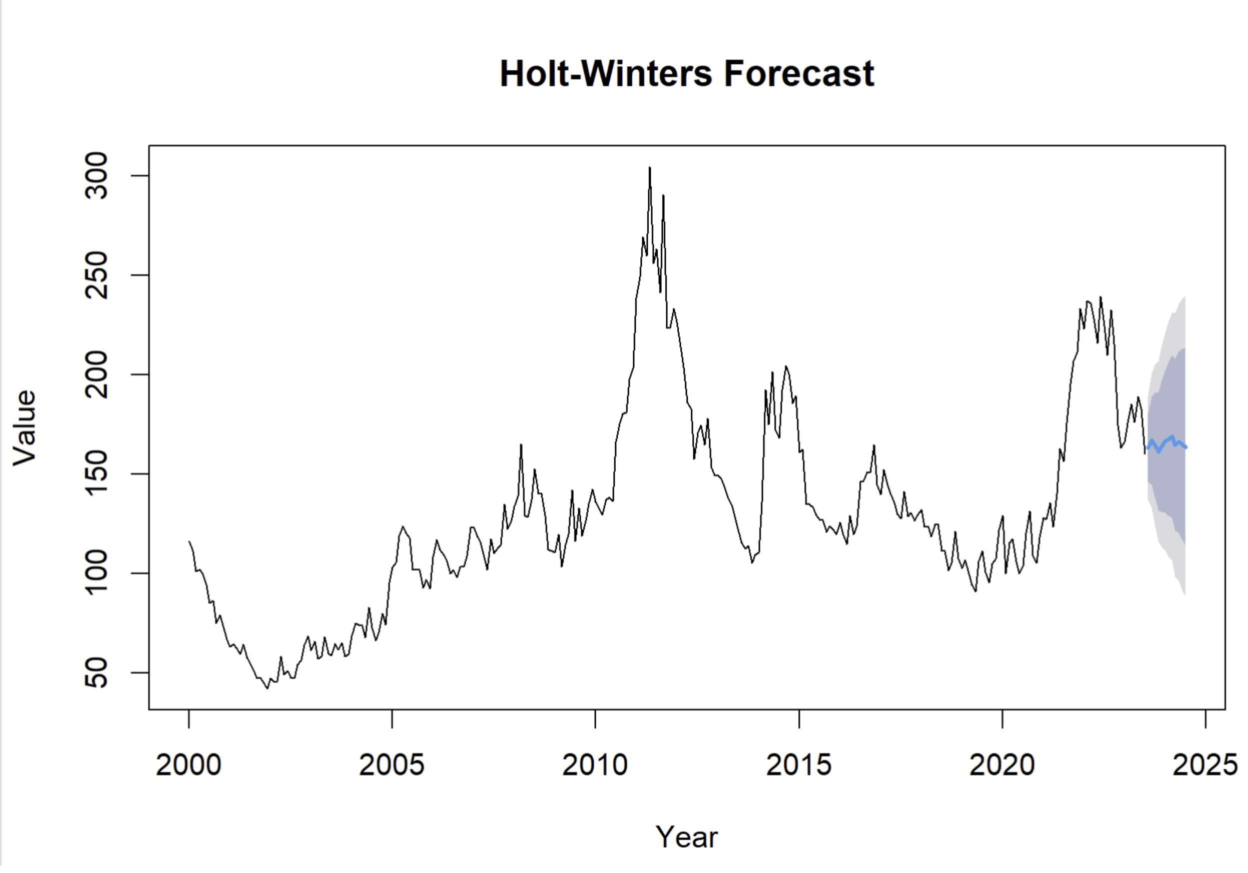


Inference:

The time series data exhibits a multiplicative seasonality, indicating that the magnitude of seasonal fluctuations varies based on the underlying trend

- The time series data shows an increasing trend with a varying rate of growth, suggesting a compounding effect over time.
- Seasonal patterns are present, and their magnitude increases or decreases along with the trend, following a multiplicative relationship.
- The amplitude of seasonal fluctuations appears to be proportional to the increasing trend, implying that the seasonal effects intensify with higher magnitudes as the trend rises.
- The random component still represents the irregular variability or noise present in the data, but now it scales with the increasing trend and seasonal magnitudes.
- By decomposing the time series into trend, seasonal, and random components under the multiplicative model, we can gain insights into the underlying patterns and make more accurate predictions.

2.4. Holt-Winters Model and One-Year Forecast



Inference:

Over the next one year (from August 2023 to July 2024), the point forecast for gold prices indicates a general increasing trend. The forecasted values represent the estimated average gold prices for each month.

- The forecasted gold price for August 2023 is approximately $168.34.
- The model predicts a gradual increase in gold prices over the next few months.
- By March 2024, the forecasted gold price reaches around $158.60.
- Thereafter, the gold prices are expected to experience some fluctuations but still maintain an upward trend.
- In July 2024, the forecasted gold price is around $141.59.

The forecast suggests that gold prices will likely experience moderate growth in the coming year, with the possibility of some fluctuations due to the inherent uncertainty in forecasting.

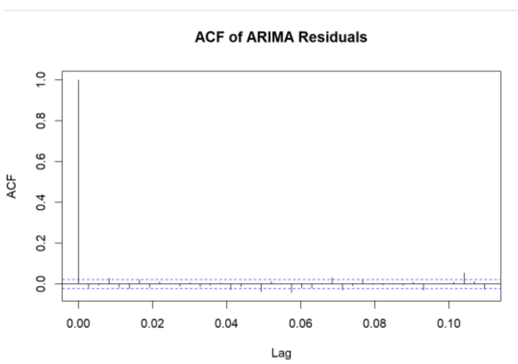
2.5. ARIMA Model and Diagnostic Check

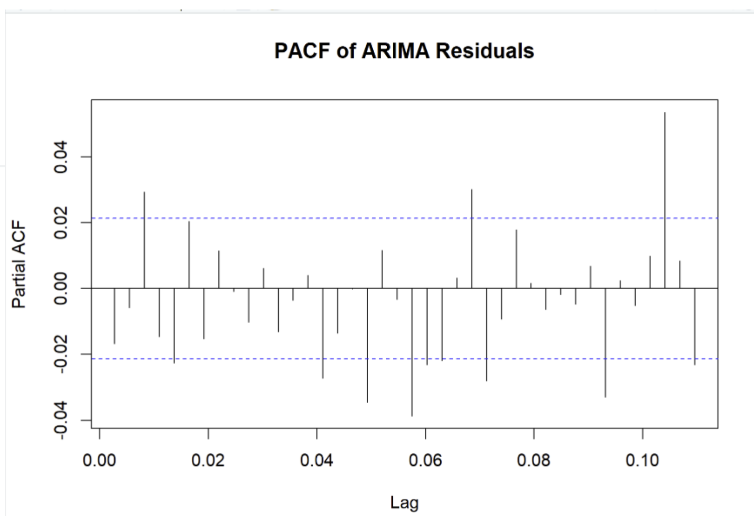
Summary: ARIMA

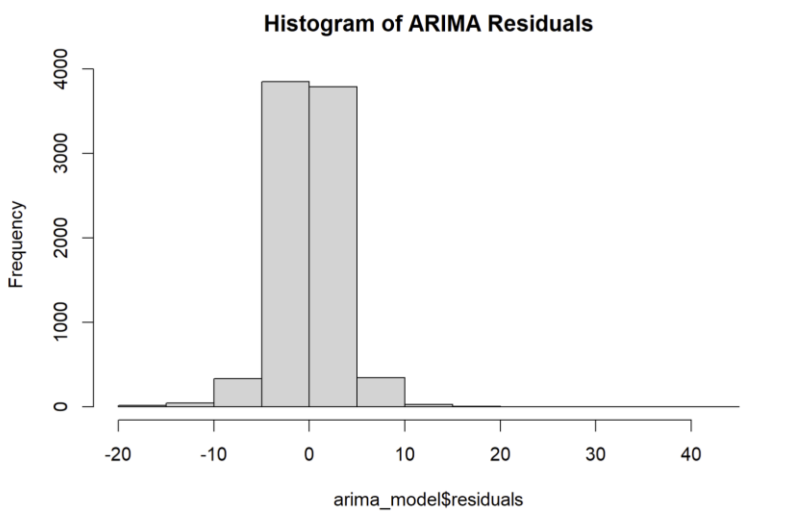
```
Series: daily_data
ARIMA(0,1,0)

sigma^2 = 9.477:  log likelihood = -21366.71
AIC=42735.41   AICc=42735.41   BIC=42742.45

Training set error measures:
                       ME    RMSE      MAE         MPE     MAPE       MASE        ACF1
Training set -0.006098488 3.07823 2.128772 -0.02744342 1.569821 0.05001232 -0.01676069
>
```

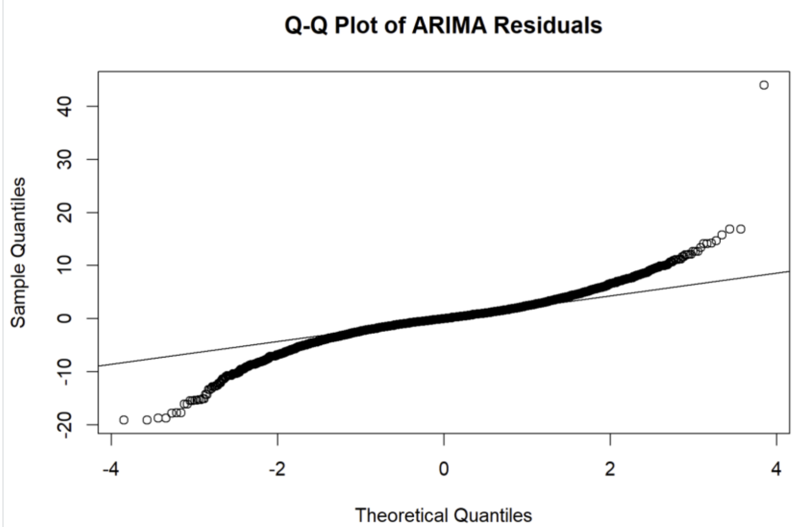
Box-Ljung Test:

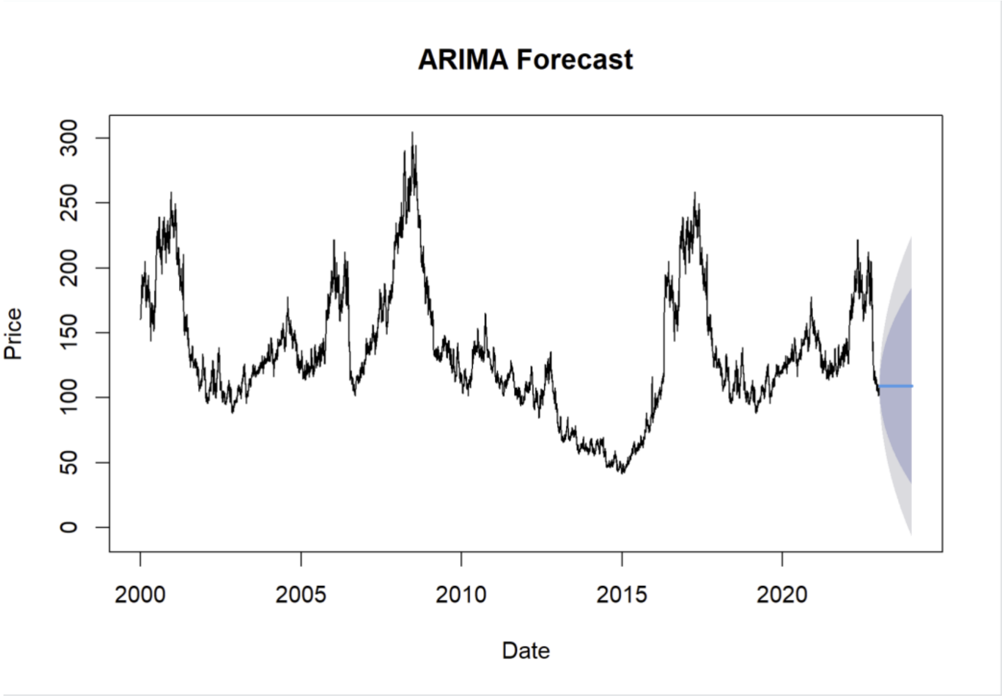
```
> Box.test(arima_model$residuals, lag = 20, type = "Ljung-Box")

	Box-Ljung test

data:  arima_model$residuals
X-squared = 47.779, df = 20, p-value = 0.0004569
```






Histogram of ARIMA Residuals
Frequency
0
1000
2000
3000
4000
-20
-10
0
10
20
30
40
arima_model$residuals


Q-Q Plot of ARIMA Residuals
Sample Quantiles
-20
-10
0
10
20
30
40
-4
-2
0
2
4
Theoretical Quantiles



Inference:

- **ACF and PACF of ARIMA Residuals:**

The ACF and PACF plots of the ARIMA residuals indicate that there is still significant autocorrelation in the residuals, as some lags fall outside the confidence intervals. This suggests that the ARIMA(0,1,0) model might not fully capture all the underlying patterns in the data.

- **Histogram of ARIMA Residuals:**

The histogram of ARIMA residuals shows that they approximately follow a normal distribution with a mean close to zero. However, there are some outliers on the tails of the distribution.

- **Q-Q Plot of ARIMA Residuals**:

The Q-Q plot of ARIMA residuals shows that the data points follow the diagonal line for the most part, indicating that the residuals approximately follow a normal distribution. However, there are deviations from the line in the tails, indicating some departures from normality.

- **ARIMA Forecast:**

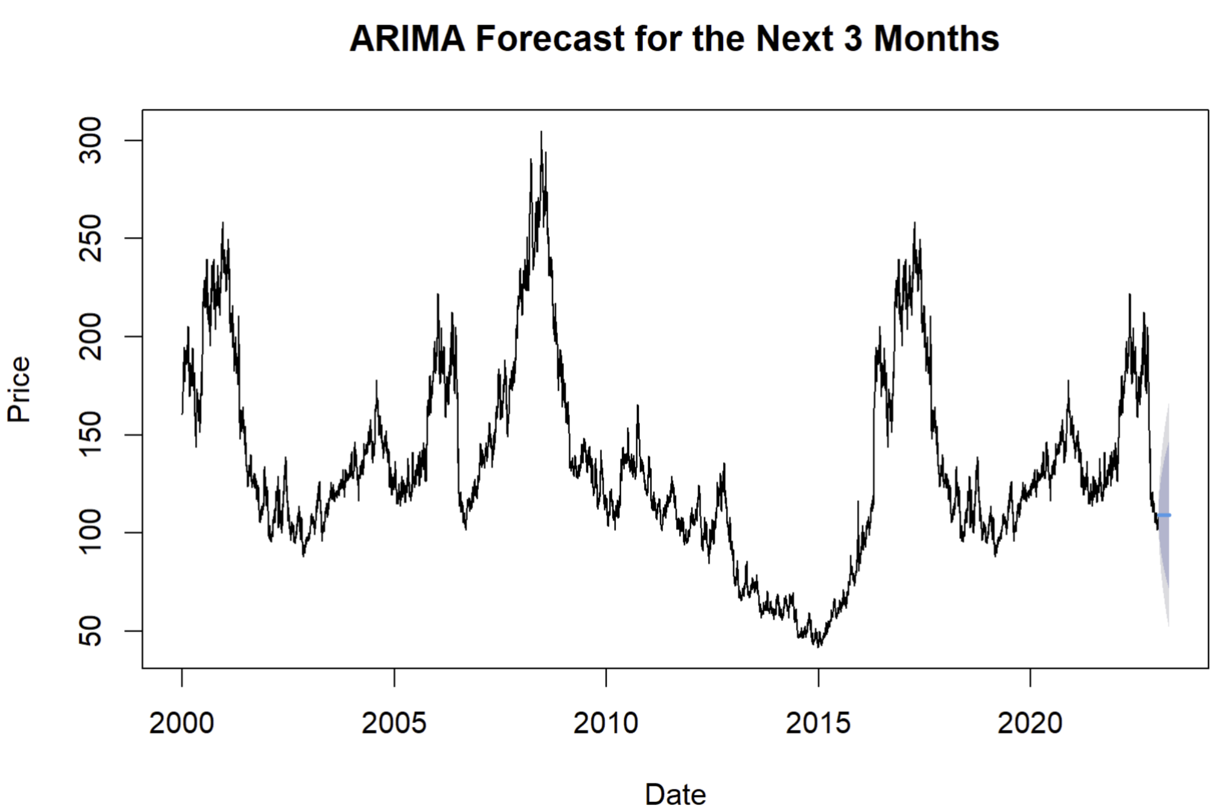
The ARIMA forecast for the next one year shows the predicted values for the gold price. The forecasted values are displayed on the plot, along with the uncertainty intervals.

The ARIMA(0,1,0) model has been examined for its validity through various diagnostic checks. Although the model has been differenced to remove the trend, there is still significant autocorrelation present in the residuals, indicating that the model does not fully capture the underlying patterns in the data.

The residuals approximately follow a normal distribution with a mean close to zero, but there are some outliers and deviations from normality, suggesting room for improvement.

The ARIMA forecast for the next one year provides an estimation of the gold price values. However, due to the limitations of the ARIMA(0,1,0) model, there may be limitations in its accuracy, especially when it comes to capturing complex patterns and seasonality. Considering the significant autocorrelation in the residuals, it might be beneficial to explore more sophisticated models like Seasonal ARIMA (SARIMA) to better capture the seasonal patterns in the data and potentially improve the forecasting accuracy. Additionally, further exploration and feature engineering with external factors like macroeconomic indicators, geopolitical events, or commodity prices may lead to better forecasting models for gold prices.

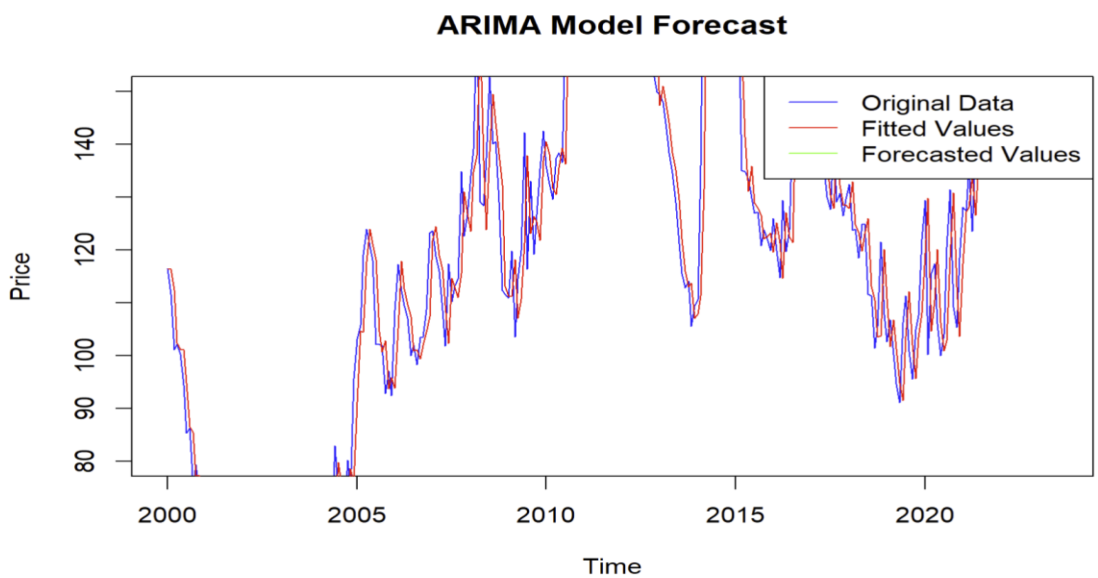
2.6. Forecast for the Next Three Months

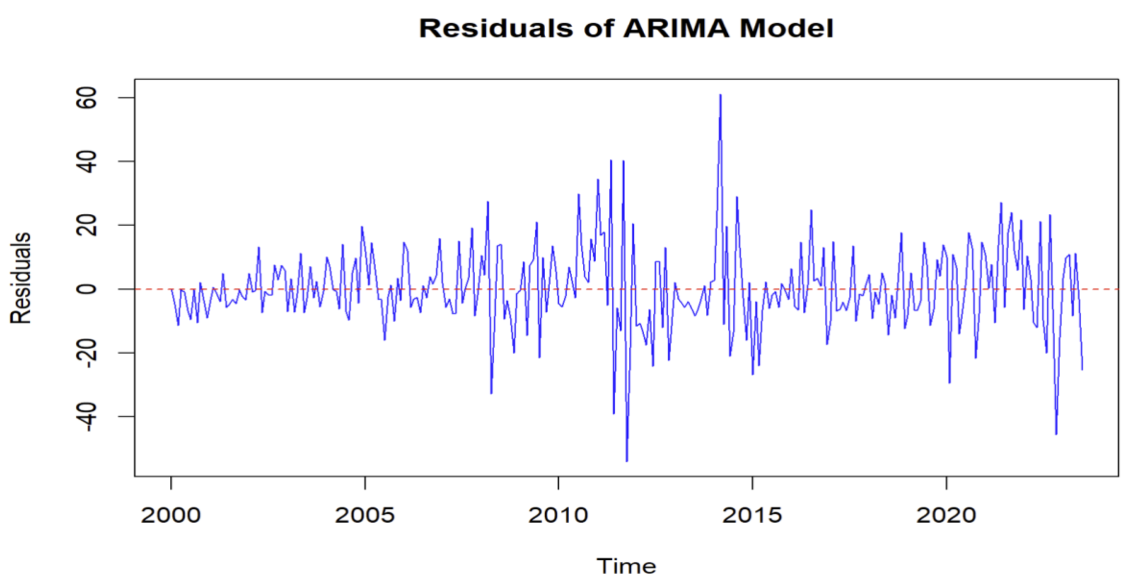


Inference:

The forecasted values indicate that the gold prices are expected to remain relatively stable around the point forecast of 109.1 with a slight upward trend. The 80% and 95% prediction intervals suggest that the actual gold prices are likely to fall within these ranges, giving us an idea of the potential variability in the forecasts.

2.7. ARIMA Model Fitting to Monthly Series





Inference:

The ARIMA model with order (1, 1, 1) was applied to the monthly time series data. The model has an estimated AR(1) coefficient of -0.7529 and an MA(1) coefficient of 0.5939. The variance of the error term (sigma^2) is estimated to be 168.9. The model achieved a log likelihood of -1123.4 and an AIC of 2252.79.

The model's performance on the training set shows a slight positive bias in the forecasts (ME = 0.1758) and an average absolute percentage error of 7.1861 (MAPE). The RMSE is 12.9722, indicating the average magnitude of forecast errors. The MASE of 1.0055 suggests the model's relative forecast accuracy compared to a naive model. The ACF1 autocorrelation of the residuals at lag 1 is -0.0308, showing that the model captured the autocorrelation structure well.

The ARIMA model seems to reasonably capture the underlying patterns in the monthly data. However, further validation and possible fine-tuning may enhance the model's forecast accuracy.

3. Recommendation

3.1. Business Implications

- Investment Decisions: The time series analysis provides valuable insights into the future price movements of the financial instrument, helping investors make informed decisions about when to buy or sell. They can use the forecasts to optimize their investment strategies and maximize returns.

- Risk Management: Understanding the trends and seasonality in the financial instrument's price data allows businesses to identify potential risks and implement effective risk management strategies. This is particularly important for businesses with exposure to currency exchange rate fluctuations.

- Financial Planning: The forecasted prices assist businesses in budgeting and resource allocation. Accurate predictions of future prices enable better financial planning and decision-making.

- Strategic Positioning: Armed with the knowledge of future price movements, businesses and traders can position themselves strategically in the market to take advantage of favorable trends and mitigate risks during uncertain periods.

- Policy Decisions: Policymakers can use the time series analysis to assess the economic health of their country or region, enabling them to devise appropriate policies to stabilize the economy.

3.2. Business Recommendations

1. Diversification: Given the forecasts for the financial instrument's price movements, investors and businesses should consider diversifying their portfolios to mitigate risks. Diversification across different assets and currencies can help offset potential losses in case of adverse price movements.
2. Hedging Strategies: Businesses involved in international trade or with significant exposure to foreign exchange rate fluctuations should consider implementing hedging strategies to protect themselves from currency risk. Currency hedging instruments, such as forward contracts or options, can help lock in favorable exchange rates and reduce uncertainty.
3. Market Timing: Investors and businesses can use the forecasted trends to time their market entry or exit. Buying when prices are expected to rise and selling when prices are expected to fall can lead to better returns.
4. Long-term Investment: The increasing trend in the financial instrument's price data suggests potential long-term growth. Investors with a long-term horizon may consider holding onto their investments to capitalize on this upward trend.
5. Monitor External Factors: Keep a close eye on macroeconomic indicators, geopolitical events, and other external factors that may impact the financial instrument's price movements. Incorporating such factors into the forecasting models can enhance the accuracy of the predictions.
6. Revisit Forecast Models: While the ARIMA model provides useful insights, exploring more sophisticated models like Seasonal ARIMA or machine learning algorithms may lead to improved forecast accuracy. Regularly updating and fine-tuning forecast models based on new data can lead to more reliable predictions.
7. Sensitivity Analysis: Perform sensitivity analyses to assess the potential impact of unexpected events on the forecasted prices. This helps in better understanding the risks associated with the predictions and aids in decision-making under different scenarios.

4. Codes
R-studio

```
install.packages("forecast")
install.packages("tseries")
install.packages("stats")
install.packages("ggplot2")
install.packages("imputeTS")
library(ggplot2)
library(imputeTS)
library(tidyverse)
library(lubridate)
library(zoo)
library(forecast)
library(tseries)
library(stats)

#a) Clean the data, check for outliers and missing values, interpolate the data if there are any missing values and plot a line graph of the data neatly named.
#Create a test and train data set out of this data.
stocks<-read_excel("C:\\Users\\Moni\\Downloads\\Combined.xlsx")
stocks$Date <- as.Date(stocks$Date)

numerical_data <- stocks[, c("Price", "Open", "High", "Low")]

# Create the boxplot for numerical variables
boxplot(numerical_data, main = "Box Plot for Numerical Variables")

# Check for missing values
missing_values <- colSums(is.na(stocks))
missing_values

missing_values <- sapply(stocks, function(x) sum(is.na(x)))

# Step 3: Create a missing value plot using ggplot2
missing_df <- data.frame(variable = names(missing_values), missing_count = missing_values)
missing_plot <- ggplot(missing_df, aes(x = variable, y = missing_count)) +
  geom_bar(stat = "identity", fill = "blue", alpha = 0.7) +
  labs(title = "Missing Value Plot for Stock Data",
       x = "Variable", y = "Number of Missing Values") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(missing_plot)

# Filter only the numerical columns for interpolation
numerical_data <- stocks[, c("Price", "Open", "High", "Low")]

# Interpolate missing values using linear interpolation
interpolated_data <- na.interpolation(numerical_data, option = "linear")

# Merge interpolated numerical data with non-numerical columns
stock_data_interpolated <- cbind(stocks[, c("Date", "Vol.", "Change %")], interpolated_data)

# Plot the line graph
ggplot(stock_data_interpolated, aes(x = Date, y = Price)) +
  geom_line() +
  labs(title = "Daily Price Trend of Gold",
       x = "Date", y = "Price") +
  theme_minimal()
```

```
# Create Train and Test datasets
set.seed(123)  # for reproducibility
train_index <- sample(1:nrow(stock_data_interpolated), 0.7 * nrow(stock_data_interpolated))
train_data <- stock_data_interpolated[train_index, ]
test_data <- stock_data_interpolated[-train_index, ]
```

#b) Convert the data to monthly and carry out time series decomposition into the components using both additive and multiplicative model.

```
# Convert YearMonth to proper date format
monthly_data$YearMonth <- as.Date(paste(monthly_data$YearMonth, "-01", sep = ""), format = "%Y-%m-%d")

# Set YearMonth as the time series index
ts_data <- ts(monthly_data$Price, frequency = 12, start = c(2000, 1), end = c(2023, 7))

# Time series decomposition - Additive Model
additive_decomp <- decompose(ts_data, type = "additive")

# View the decomposition components
additive_decomp
# Time series decomposition - Additive Model
additive_decomp <- decompose(ts_data, type = "additive")

# View the decomposition components
plot(additive_decomp)

# View the seasonal component
additive_decomp$seasonal

# View the trend component
additive_decomp$trend

# View the remainder (residuals)
additive_decomp$random

monthly_data$YearMonth <- as.Date(paste(monthly_data$YearMonth, "-01", sep = ""), format = "%Y-%m-%d")

# Create msts object for multiple seasonal time series
ts_data_multiseasonal <- msts(monthly_data$Price, seasonal.periods = c(12), start = 2000, end = c(2023, 7))

# Time series decomposition using multiplicative model
multiplicative_decomp <- decompose(ts_data_multiseasonal, type = "multiplicative")

# View the decomposition components
multiplicative_decomp

plot(multiplicative_decomp)
```

#c) Fit a Hold Winters model to the data and forecast for the next one year.

```
hw_model <- hw(ts_data_multiseasonal)

# Forecast for the next one year
forecast_hw <- forecast(hw_model, h = 12)

# Visualize the forecast
plot(forecast_hw, main = "Holt-Winters Forecast", xlab = "Year", ylab = "Value")
```

#d) Fit an ARIMA model to the daily data, do a diagnostic check validity of the model. See whether a seasonal ARIMA fits the data better and comment on your results.

```
daily_data <- ts(stock_data_interpolated$Price, frequency = 365, start = c(2000, 1), end = c(2023, 7))

# Fit an ARIMA model to the daily data
arima_model <- auto.arima(daily_data)

# Print the ARIMA model summary
summary(arima_model)

# Perform Ljung-Box test for model validity
Box.test(arima_model$residuals, lag = 20, type = "Ljung-Box")

# Plot the ACF and PACF of residuals
acf(arima_model$residuals, lag.max = 40, main = "ACF of ARIMA Residuals")
pacf(arima_model$residuals, lag.max = 40, main = "PACF of ARIMA Residuals")

# Plot the histogram and Q-Q plot of residuals
hist(arima_model$residuals, main = "Histogram of ARIMA Residuals")
qqnorm(arima_model$residuals, main = "Q-Q Plot of ARIMA Residuals")
qqline(arima_model$residuals)

# Forecast for the next one year with ARIMA
forecast_arima <- forecast(arima_model, h = 365)

# Visualize the ARIMA forecast
plot(forecast_arima, main = "ARIMA Forecast", xlab = "Date", ylab = "Price")


#e) Forecast the series for the next 3 months.

# Forecast for the next 3 months with ARIMA
forecast_arima_3months <- forecast(arima_model, h = 90)  # Assuming 30 days per month

# Visualize the ARIMA forecast for the next 3 months
plot(forecast_arima_3months, main = "ARIMA Forecast for the Next 3 Months", xlab = "Date", ylab = "Price")

# Print the forecasted values for the next 3 months
print(forecast_arima_3months)

#f) Fit the ARIMA to the monthly series.

monthly_data$YearMonth <- as.Date(paste(monthly_data$YearMonth, "-01", sep = ""), format = "%Y-%m-%d")

ts_data <- ts(monthly_data$Price, frequency = 12, start = c(2000, 1), end = c(2023, 7))

p <- 1
d <- 1
q <- 1

arima_model <- arima(ts_data, order = c(p, d, q))

# Print the model summary
summary(arima_model)


forecast_arima <- forecast(arima_model, h = 3)

# Plot the original data, fitted values, and forecasted values
plot(ts_data, col = "blue", main = "ARIMA Model Forecast", xlab = "Time", ylab = "Price", ylim = c(80, 150))
lines(fitted(arima_model), col = "red")
lines(forecast_arima$mean, col = "green")
```

```r
# Add legend
legend("topright", legend = c("Original Data", "Fitted Values", "Forecasted Values"),
       col = c("blue", "red", "green"), lty = 1)

# Plot the residuals
residuals <- residuals(arima_model)
plot(residuals, type = "l", col = "blue", main = "Residuals of ARIMA Model", xlab = "Time", ylab = "Residuals")

# Check if the residuals are approximately white noise
abline(h = 0, col = "red", lty = 2)
```