Report

Time Series Analysis

CIA

Definition of Time Series:

A time series is a sequence of data points collected or recorded at specific time intervals. Time series data is often used to track changes over time, such as economic indicators, stock prices, weather data, and more.

Examples:

- Stock Prices: Daily closing prices of a company’s stock.
- Weather Data: Monthly average temperatures over several years.
- Sales Data: Monthly sales revenue for a retail store.
- Economic Indicators: Quarterly GDP growth rates.

Objectives of Time Series Analysis:

- Identify Patterns: Determine any trends, seasonal variations, or cyclic behaviours.
- Forecast Future Values: Predict future values based on historical data.

- Understand Data Characteristics: Analyse and interpret components like trend, seasonality, and noise.
- Model and Analyse Trends: Apply models to understand and project future data points.

Dataset Description:

Name: Monthly Revenue Data

Columns:

- Period: Date or Time Period (e.g., '2021-01-01')
- Revenue: Monthly revenue figures
- Sales_quantity: Number of units sold
- Average_cost: Average cost of goods sold
- The_average_annual_payroll_of_the_region: Average payroll in the region

Perform Exploratory Data Analysis (EDA):

- **1. Load the Data:**
- **2. Inspect the Data:**
- **3. Visualize the Data:**
- **4. Check for Missing Values:**

```r
# View the first few rows
head(Month_Value_1)

# View the structure of the Month_Value_1
str(Month_Value_1)

# Summary of each column
summary(Month_Value_1)

# View the dimensions of the Month_Value_1 (rows, columns)
dim(Month_Value_1)

# Count total missing values
sum(is.na(Month_Value_1))

# Visualize missing Month_Value_1
library(Amelia)
missmap(Month_Value_1, main = "Missing values map")
```

Statistical Analysis:

```r
#Convert the Period column to a Date format for easier manipulation.
Month_Value_1$Period <- as.Date(Month_Value_1$Period, format="%d.%m.%Y")

# Fill missing values with the mean
Month_Value_1$Sales_quantity[is.na(Month_Value_1$Sales_quantity)] <- mean(Month_Value_1$Sales_quantity, na.rm = TRUE)

# Fill missing values with the median
Month_Value_1$Average_cost[is.na(Month_Value_1$Average_cost)] <- median(Month_Value_1$Average_cost, na.rm = TRUE)

summary(Month_Value_1$Revenue)
mean(Month_Value_1$Sales_quantity)
sd(Month_Value_1$Average_cost)
```

Data Visualization:

```
library(ggplot2)
ggplot(Month_Value_1, aes(x = Period, y = Revenue)) +
  geom_line(color = "blue") +
  labs(title = "Revenue Over Time", x = "Period", y = "Revenue")

#b. Bar Plot for Sales Quantity
  #Plot sales quantity to see the monthly distribution.

ggplot(Month_Value_1, aes(x = Period, y = Sales_quantity)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(title = "Sales Quantity Over Time", x = "Period", y = "Sales Quantity")

#c. Boxplot for Average Cost
  #Create a boxplot to check for any outliers in the Average_cost.

ggplot(Month_Value_1, aes(y = Average_cost)) +
  geom_boxplot(fill = "orange") +
  labs(title = "Boxplot of Average Cost", y = "Average Cost")

#d. Scatter Plot: Revenue vs. Sales Quantity
  #Plot the relationship between Revenue and Sales_quantity.

ggplot(Month_Value_1, aes(x = Sales_quantity, y = Revenue)) +
  geom_point(color = "red") +
  labs(title = "Revenue vs. Sales Quantity", x = "Sales Quantity", y = "Revenue")
```

Identify and Interpret Time Series Components:

1. Create a Time Series Object:

```
# Assuming 'Month_Value_1' is your dataset and 'Revenue' is the column of interest
# Create a time series object
ts_Month_Value_1 <- ts(Month_Value_1$Revenue, start = c(2015, 1), frequency = 12)
```

2. Decompose the Time Series:

```
# Decompose the time series (for analysis purposes)
z <- decompose(ts_Month_Value_1)

# Plot the decomposed components
plot(z)
```

Components:

Trend: Long-term movement in the series.

Seasonality: Regular pattern or cycle in the data.

Residuals/Noise: Random variations or errors not explained by trend or seasonality.

Interpretation Example:

Trend: The data might show an upward trend indicating growth in revenue over time.

Seasonality: You might see recurring patterns, such as higher revenue during holiday seasons.

Residuals: These represent the irregular fluctuations not captured by the trend or seasonal components.

Determine Additive or Multiplicative Model:

1. Additive Model:

Use when: The seasonal effect and residuals are constant throughout the series, regardless of the trend.

Model Form:

$$Y_t = T_t + S_t + R_t$$

2. Multiplicative Model:

Use when: The seasonal effect and residuals vary proportionally with the level of the trend.

Model Form:

$$Y_t = T_t * S_t * R_t$$

Justification:

Additive: If the seasonal peaks and troughs remain the same over time.
Multiplicative: If the amplitude of seasonality increases or decreases with the trend.

Eliminate Components and Forecast:

1. **Remove Trend Component:**

```
# Extract the trend component
trend <- z$trend

# Remove the trend component from the original time series
detrended <- ts_Month_Value_1 - trend
```

```
##Justification:
 # - Removing the trend component allows us to analyze the time series without the long-term

# Plot the detrended series
plot(detrended, main = "Detrended Time Series", ylab = "Revenue")
```

2. Apply Holt-Winters Exponential Smoothing:

```
# Apply Holt-Winters exponential smoothing to the original time series
zz <- HoltWinters(ts_Month_Value_1)

# Display the Holt-Winters model
print(zz)

# Plot the fitted model
plot(zz)



plot(zz,main="Holt-Winters Smooth")
zz.pred<-predict(zz,30,prediction.interval = TRUE)
zz.pred

plot(zz,zz.pred, main = "Holt-Winter Forecast")
```

Conclusion:

- **Component Elimination:** Removing the trend allows focusing on seasonal and irregular components.
- **Forecasting:** Holt-Winters smoothing provides forecasts considering both trend and seasonality.

Conclusion:

Dataset Description and Preparation:

The dataset, consisting of monthly revenue and other related variables, was thoroughly explored. Missing values were addressed, and appropriate transformations were applied to ensure data quality and consistency.

Exploratory Data Analysis (EDA):

Through visualizations and summary statistics, we identified key patterns and relationships in the data. The time series plots revealed trends, seasonal variations, and potential outliers, helping to understand the underlying structure of the dataset.

Component Identification:

The time series decomposition provided insights into the trend, seasonality, and residuals. This step allowed us to observe long-term movements, periodic cycles, and random noise in the revenue data.

Model Selection:

We assessed whether an additive or multiplicative model was more appropriate based on the nature of the data. The choice between these models was justified by examining the behavior of seasonal variations relative to the trend.

Component Elimination and Forecasting:

The trend component was removed to isolate and better understand the seasonal and residual patterns. Subsequently, the Holt-Winters exponential smoothing method was applied to forecast future values, capturing both trend and seasonality.

Key Findings:

Trend Analysis: The revenue data exhibited a clear upward trend, indicating overall growth.

Seasonality: Seasonal effects were evident, with notable peaks and troughs corresponding to specific times of the year.

Model Fit: The additive or multiplicative model choice was justified based on observed patterns in the data. For this dataset, the additive model was selected as the seasonal variations remained relatively constant regardless of the trend.

Forecasting Insights:

The Holt-Winters model provided a robust forecast of future revenue, highlighting expected increases and seasonal fluctuations. This forecasting capability can aid in strategic planning and resource allocation, especially during peak and off-peak periods.