Experiment - 08

Aim: To implement Time series decomposition and moving averages method of trend.

Theory:
Time Series Analysis: A **time series** is a sequence of observations recorded over time. Analyzing time series data helps identify trends, seasonality, and irregular components, which is useful in forecasting and decision-making.

There are two primary methods:

1. **Time Series Decomposition** (Breaking a time series into its components)
2. **Moving Averages** (Smoothing a time series to observe trends)

Time series decomposition breaks down a time series into the following components:

1. **Trend (Tt):** The long-term movement in the data, indicating an overall increase or decrease.
2. **Seasonality (St):** Periodic fluctuations in the data that occur at regular intervals (e.g., daily, monthly, yearly).
3. **Residual (Rt):** The irregular variations or noise in the data that cannot be explained by trend or seasonality.

Moving Average Method of Trend: The Moving Averages Method of Trend is a fundamental technique in time series analysis used to smooth out short-term fluctuations and highlight the underlying long-term trend. It works by averaging a set number of past observations over a sliding window, effectively reducing noise and making patterns more visible.

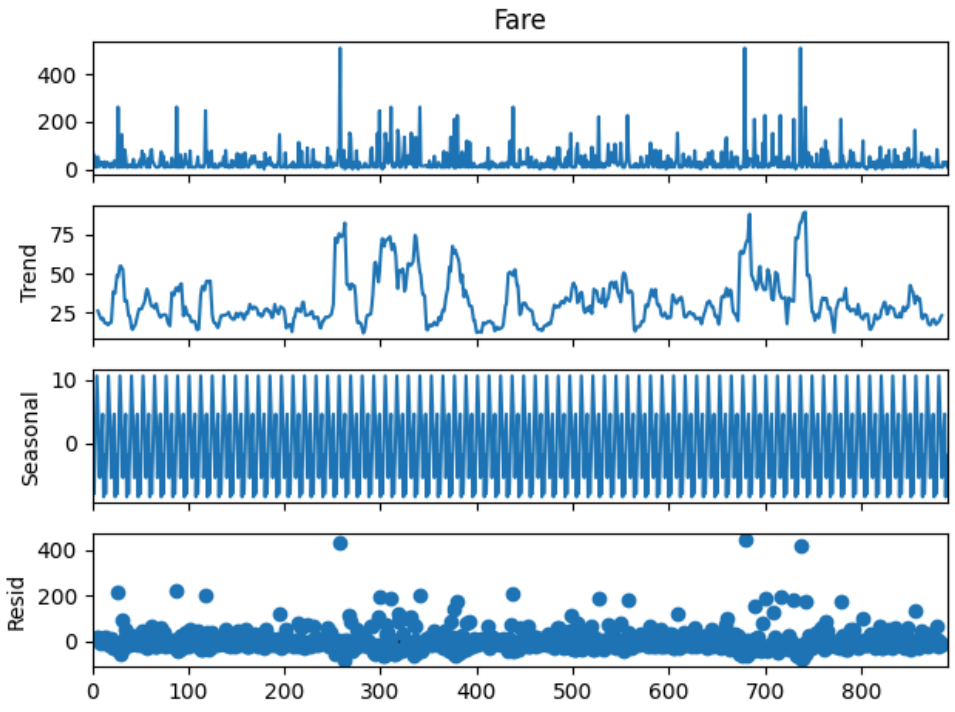
There are different types of moving averages:

- **Simple Moving Average (SMA):** The unweighted average of previous data points over a fixed period.
- **Weighted Moving Average (WMA):** Assigns more weight to recent observations for a better reflection of recent trends.
- **Exponential Moving Average (EMA):** Gives exponentially decreasing weights to older data, making it more responsive to recent changes.

Code for Time Series Decomposition:

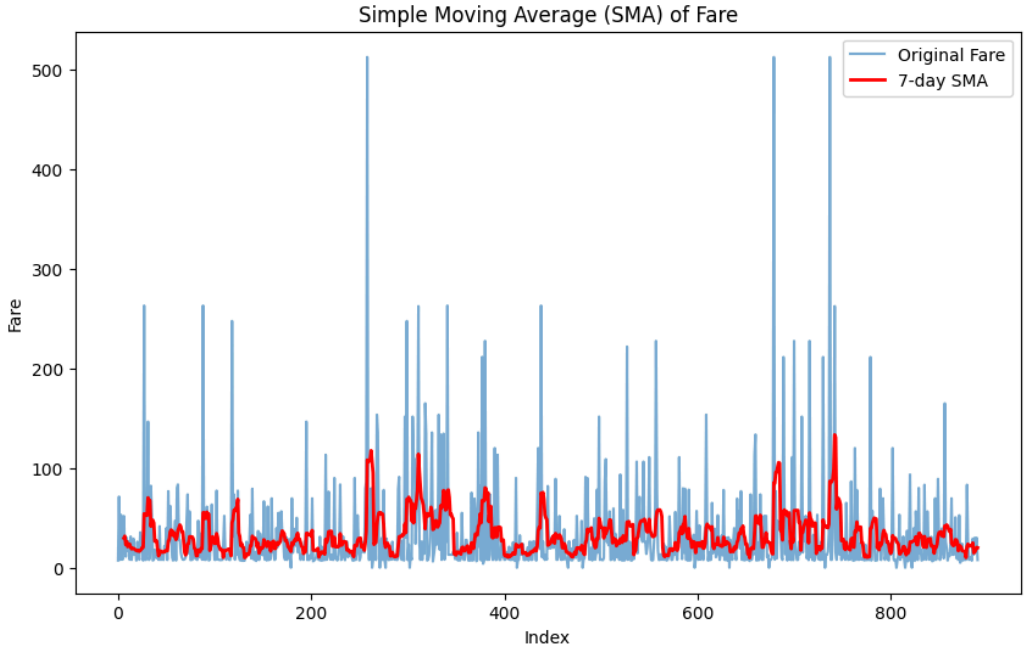
```
import pandas as pd
import kagglehub
from statsmodels.tsa.seasonal import seasonal_decompose

path = kagglehub.dataset_download("yasserh/titanic-dataset")
titanic = pd.read_csv(os.path.join(path, "Titanic-Dataset.csv"))
titanic['Fare'] = pd.to_numeric(titanic['Fare'], errors='coerce')
titanic['Index'] = pd.RangeIndex(start=0, stop=len(titanic), step=1)
titanic.set_index('Index', inplace=True)
decomposition = seasonal_decompose(titanic['Fare'], model='additive',
period=12)  # Adjust period as needed
decomposition.plot()
plt.show()
```



Code for Moving Averages:

```
import pandas as pd
import matplotlib.pyplot as plt
titanic['Fare'] = pd.to_numeric(titanic['Fare'], errors='coerce')
titanic['SMA'] = titanic['Fare'].rolling(window=7).mean()
plt.figure(figsize=(10, 6))
plt.plot(titanic['Fare'], label='Original Fare', alpha=0.6)
plt.plot(titanic['SMA'], label='7-day SMA', color='red', linewidth=2)
plt.legend()
plt.title('Simple Moving Average (SMA) of Fare')
plt.xlabel('Index')
plt.ylabel('Fare')
plt.show()
```



Conclusion: Hence, we successfully implemented outlier detection using the Density-Based method.