WEEK 4: Lab Activity: Time Series Analysis of Student Engagement Over a Semester

Objective:

Analyse weekly student engagement data (e.g., attendance rate, login count) using time series techniques in Python and Tableau.

Part A: Python-Based Time Series Analysis

Dataset: weekly_attendance.csv

Step 1: Import Libraries and Load Data

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('student_engagement.csv')

# Display the first few rows
print(df.head())
```

Step 2: Convert 'Week' to Date Format

```
# Convert 'Week' column to datetime format
df['Week'] = pd.to_datetime(df['Week'])

# Set 'Week' as the index for time series analysis
df.set_index('Week', inplace=True)
```

```
# Confirm the index change
print(df.index)
```

Step 3: Plot Raw Time Series Data

```
# Plot attendance rate over time
plt.figure(figsize=(10, 5))
plt.plot(df.index, df['AttendanceRate'], marker='o', label='Attendance Rate')
plt.title('Weekly Attendance Rate')
plt.xlabel('Week')
plt.ylabel('Attendance (%)')
plt.grid(True)
plt.legend()
plt.show()

# Plot login count over time
plt.figure(figsize=(10, 5))
plt.plot(df.index, df['LoginCount'], marker='s', color='orange', label='Login Count')
plt.title('Weekly Login Count')
plt.xlabel('Week')
plt.ylabel('Logins')
plt.grid(True)
plt.legend()
plt.show()
```

Step 4: Apply Rolling Average (Smoothing)

```
# Add rolling averages (window = 3 weeks)
df['AttendanceRate_Smooth'] = df['AttendanceRate'].rolling(window=3).mean()
df['LoginCount_Smooth'] = df['LoginCount'].rolling(window=3).mean()
```

```
# Plot smoothed attendance
plt.figure(figsize=(10, 5))
plt.plot(df.index, df['AttendanceRate'], label='Original')
plt.plot(df.index, df['AttendanceRate_Smooth'], label='Smoothed', linestyle='--')
plt.title('Smoothed Attendance Rate')
plt.xlabel('Week')
plt.ylabel('Attendance (%)')
plt.legend()
plt.grid(True)
plt.show()
```

Step 5: Simple Linear Regression Prediction

```
from sklearn.linear_model import LinearRegression
import numpy as np

# Prepare data for regression
df = df.dropna()  # Drop rows with NaN from rolling average
X = np.arange(len(df)).reshape(-1, 1)  # Time as numeric
y = df['AttendanceRate'].values

# Fit linear regression model
model = LinearRegression()
model.fit(X, y)

# Predict future attendance
future_X = np.arange(len(df)+4).reshape(-1, 1)  # Predict 4 weeks ahead
future_y = model.predict(future_X)

# Plot prediction
```

```
plt.figure(figsize=(10, 5))
plt.plot(df.index, y, label='Actual Attendance')
plt.plot(pd.date_range(df.index[0], periods=len(future_y), freq='W'), future_y, label='Predicted', linestyle='--')
plt.title('Attendance Forecast')
plt.xlabel('Week')
plt.ylabel('Attendance (%)')
plt.legend()
plt.grid(True)
plt.show()
```

Part B: Tableau-Based Time Series Analysis

Tools: Tableau Desktop or Tableau Public

Step-by-Step Instructions:

1. **Open Tableau and Connect to CSV**

   Click “Connect to Data” → Choose student_engagement.csv.

2. **Set Date Field**

   Drag Week to Columns → Right-click → Convert to Date.

3. **Create Line Charts**

   Drag AttendanceRate to Rows → Choose “Line” chart.

   Repeat for LoginCount.

4. **Add Rolling Average**

   Click on AttendanceRate → Create Calculated Field: WINDOW_AVG(SUM([AttendanceRate]), -2, 0)

   Add it to the chart as a second line.

5. **Forecasting**

   Right-click on chart → "Forecast" → "Show Forecast".

   Tableau will auto-predict future values based on trend.

6. **Dashboard Creation**

   Combine both charts into a dashboard.

   Add filters for module or student group if available.

Learning Outcomes

- Understand time series structure and smoothing techniques.
- Apply regression for forecasting.
- Use Tableau's built-in forecasting and visualisation tools.
- Interpret trends to support academic decision-making.