

7a. Write a Python Program to draw Time Series using Plotly libraries.

1. Introduction.

- The given Python program utilizes the Plotly library, specifically Plotly Express, to create a time series line chart depicting a typical engineering student's daily activities. The program uses sample data, including dates, studying hours, and sleeping hours, to visualize the time series trend of these activities over a period of five days.

2. Program code.

```
# Import necessary libraries
import plotly.express as px
import pandas as pd
from plotly.offline import plot

# Sample data for a typical engineering student's daily activities as time series
data = {
    'Date': pd.date_range(start='2024-01-01', periods=5, freq='D'),
    'Studying_Hours': [4, 3, 2, 5, 1],
    'Sleep_Hours': [7, 6, 7, 8, 7]
}

# Create a DataFrame from the sample data
df = pd.DataFrame(data)

# Plot the Student Daily Studying and Sleeping Time Series using Plotly Express
fig = px.line(df, x='Date',
              y=['Studying_Hours', 'Sleep_Hours'],
              color_discrete_sequence=['blue', 'green'],
              title="Engineering Student's Studying and Sleeping Time Series",
              labels={'value': 'Hours', 'variable': 'Activity'})

# Customize the legend
fig.update_layout(legend_title_text='Activities')

# Show the plot
plot(fig, filename='Student_statistics.html')
```

3. Explanation of the code

3a. Set up the necessary libraries for creating plot.

- *import plotly.express as px*

This line imports the Plotly Express module and gives it the alias px. Plotly Express is a high-level interface for creating various types of interactive plots in Python. It simplifies the process of creating charts and graphs.

- *import pandas as pd*

This line imports the Pandas library and gives it the alias pd. Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrames, which are used to organize and structure data in tabular form.

- *from plotly.offline import plot*

This line imports the plot function from Plotly's offline module. The plot function is used to save the plot as an HTML file. The offline module allows you to create and save plots without an internet connection.

3b. Creates data set using python dictionary

```
data = {  
  
    'Date': pd.date_range(start='2024-01-01', periods=5, freq='D'),  
  
    'Studying_Hours': [4, 3, 2, 5, 1],  
  
    'Sleep_Hours': [7, 6, 7, 8, 7]  
}
```

- 'Date' Key:

'Date': This key is associated with the values generated using pd.date_range.

- `pd.date_range(start='2024-01-01', periods=5, freq='D')`

This generates a date range starting from '2024-01-01' with a frequency of one day ('D') for a total of 5 periods. The result is a Pandas DatetimeIndex representing five consecutive days.

- 'Studying_Hours': This key is associated with a list of integers representing the number of studying hours for each corresponding date.

[4, 3, 2, 5, 1]: These values represent the studying hours for each day in the date range.

- 'Sleep_Hours': This key is associated with a list of integers representing the number of sleeping hours for each corresponding date.

[7, 6, 7, 8, 7]: These values represent the sleeping hours for each day in the date range.

3c. Create a Pandas DataFrame

- **df = pd.DataFrame(data)**

This line of code transforms the data dictionary into a structured tabular format, making it easier to analyze and visualize the information. The resulting DataFrame (df) can be used for various data manipulation and exploration tasks.

	Date	Studying_Hours	Sleep_Hours
0	2024-01-01	4	7
1	2024-01-02	3	6
2	2024-01-03	2	7
3	2024-01-04	5	8
4	2024-01-05	1	7

3d. Configure the line plot

```
fig = px.line(df, x='Date',
```

```
    y=['Studying_Hours', 'Sleep_Hours'],
```

```
    color_discrete_sequence=['blue', 'green'],
```

```
    title="Engineering Student's Daily Studying and Sleeping Time Series",
```

```
    labels={'value': 'Hours', 'variable': 'Activity'}))
```

This above code segment effectively configures the line chart, specifying the data source, x-axis, y-axis, colors, title, and labels. The resulting fig object can be further customized or directly displayed using fig.show().

- `px.line`: This function creates a line chart using Plotly Express.
- `df`: The Pandas DataFrame (`df`) is the data source for the chart.
- `x='Date'`: The 'Date' column from the DataFrame is used as the x-axis values.
- `y=['Studying_Hours', 'Sleep_Hours']`: The 'Studying_Hours' and 'Sleep_Hours' columns from the DataFrame are used as the y-axis values for two lines on the chart.
- `color_discrete_sequence=['blue', 'green']`:

This parameter specifies the colors of the lines. The line for 'Studying_Hours' is colored blue, and the line for 'Sleep_Hours' is colored green.

- `title="Engineering Student's Daily Studying and Sleeping Time Series"`:

This sets the title of the chart .

- `labels={'value': 'Hours', 'variable': 'Activity'}`:

This parameter allows you to customize the axis labels. It sets the y-axis label to 'Hours' and the legend label to 'Activity'.

- `fig`:

The resulting object (`fig`) is a Plotly Express figure that represents the line chart. It contains all the information about the chart's layout, data, and visual properties.

3e. Customizing the legend title in the Plotly figure

`fig.update_layout(legend_title_text='Activities')`

- ***fig.update_layout***:

This method is used to update the layout properties of the Plotly figure (`fig`).

It allows you to customize various aspects of the chart layout, such as titles, legends, axes, and more.

- ***legend_title_text='Activities'***:

The `legend_title_text` parameter is specific to the legend in the chart.

In this case, it sets the title text of the legend to 'Activities'. The legend is the area on the chart that typically identifies the colors or styles associated with different elements, such as lines or markers.

3f. Save and display the Plotly figure (fig) as an HTML file

- `plot(fig, filename='Student_statistics.html')`

This saves the interactive line chart represented by fig as an HTML file with the specified filename. This HTML file can be opened in a web browser, and users can interact with the chart, zoom in, pan, and view data points. This is particularly useful for sharing interactive visualizations without the need for an internet connection.

4. Output

- The program generates a line chart where the x-axis represents dates, and there are two lines for studying and sleeping hours with distinct colors. The legend indicates which color corresponds to each activity.



5. Real-Time Applications of Plotly Library for Plotting Time Series Data

Plotly is widely used in various domains for plotting time series data in real-time applications, such as:

- Finance:

Plotting stock prices over time to analyze trends, volatility, and make informed investment decisions.

- IoT and Sensor Data:

Visualizing sensor data collected over time to monitor and optimize industrial processes or environmental conditions.

- Health Monitoring:

Tracking and visualizing patient health metrics over time for healthcare professionals to analyze and make medical decisions.

- Energy Consumption:

Analyzing and visualizing energy consumption patterns to optimize usage and reduce costs.

- Climate and Weather Monitoring:

Visualizing time series data related to climate and weather conditions for meteorological analysis and predictions.