

Task 9:To analyze and visualize Time Oriented Data

Analysis to identify systemic patterns in the data that help to form trends, cycles or seasonal variances and to forecast the data. - Line Graph, Trend Lines, Area Chart

Tools : Tableau, Language :Python

9a) The dataset set contains various features such as temperature, pressure, humidity, rain, precipitation,etc.

Data set Link: [Delhi Weather Data | Kaggle](#). This dataset contains weather data for New Delhi, India. This data was taken out from wunderground with the help of their easy to use api. It contains various features such as temperature, pressure, humidity, rain, precipitation,etc. To analyze and visualize Time Oriented Data using Line graph, trend lines, area chart.

Aim: To analyze and visualize Time Oriented Data using Line graph, trend lines, area chart.

Algorithm:

- Import necessary libraries: pandas, matplotlib.pyplot, seaborn, numpy, and LinearRegression from sklearn.
- Read the CSV file containing the dataset into a DataFrame using pd.read_csv().
- Preprocess the data by stripping column names, converting 'datetime_utc' column to datetime format, and setting it as the index.
- Plot the temperature data over time using sns.lineplot(), specifying the x-axis as the index of the DataFrame and y-axis as the temperature column ('_tempm').
- Customize the plot by adding a title, labels for the x and y axes, enabling grid lines, rotating x-axis labels for better readability, and displaying the plot using plt.show().

Program:

Line Graph

```
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
from sklearn.linear_model import LinearRegression
```

```

data = pd.read_csv('/content/testset.csv')

data.columns = data.columns.str.strip()

data['datetime_utc'] = pd.to_datetime(data['datetime_utc'])

data.set_index('datetime_utc', inplace=True)

plt.figure(figsize=(10, 6))

sns.lineplot(data=data, x=data.index, y='_tempm') # Assuming '_tempm' is the column you
want to plot

plt.title('Line Graph of Temperature over Time')

plt.xlabel('Date')

plt.ylabel('Temperature (°C)')

plt.grid(True)

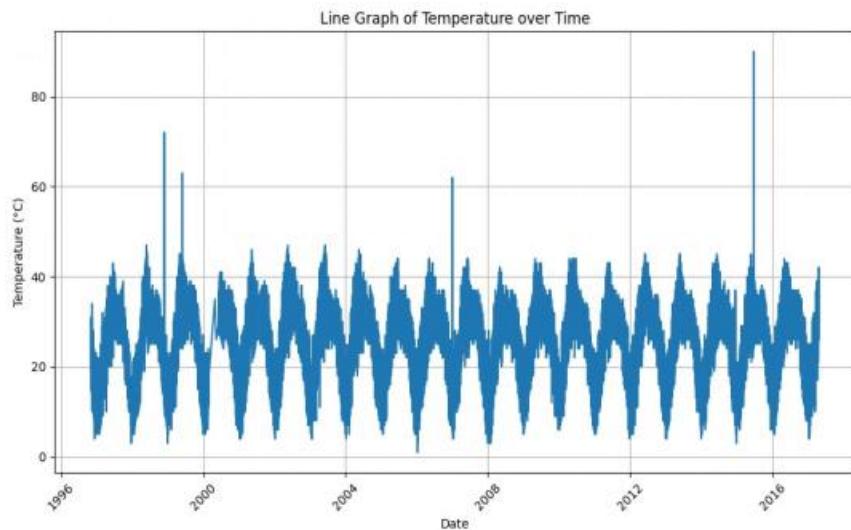
plt.xticks(rotation=45)

plt.tight_layout()

plt.show()

```

Output:



Trend Lines

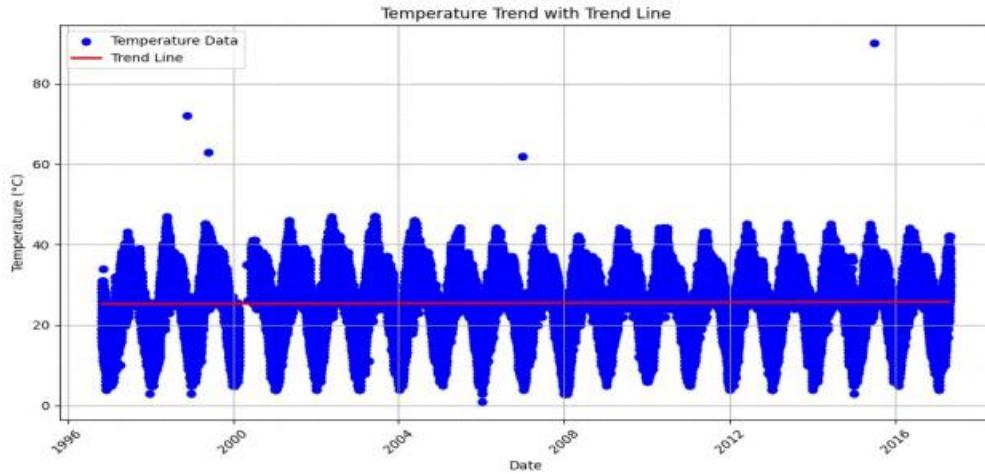
```

import pandas as pd

import numpy as np

```

```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
data = pd.read_csv('/content/testset.csv')
data.columns = data.columns.str.strip()
data['datetime_utc'] = pd.to_datetime(data['datetime_utc'])
data.set_index('datetime_utc', inplace=True)
data['_tempm'].fillna(data['_tempm'].mean(), inplace=True)
x = data.index.astype(int).values.reshape(-1, 1) # Converting datetime to numerical
y = data['_tempm'].values # Assuming '_tempm' is the column for temperature
model = LinearRegression()
model.fit(x, y)
slope = model.coef_[0]
intercept = model.intercept_
plt.figure(figsize=(10, 6))
plt.scatter(data.index, data['_tempm'], color='blue', label='Temperature Data')
plt.plot(data.index, slope * x + intercept, color='red', linestyle='-', linewidth=2, label='Trend Line')
plt.title('Temperature Trend with Trend Line')
plt.xlabel('Date')
plt.ylabel('Temperature (°C)')
plt.grid(True)
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

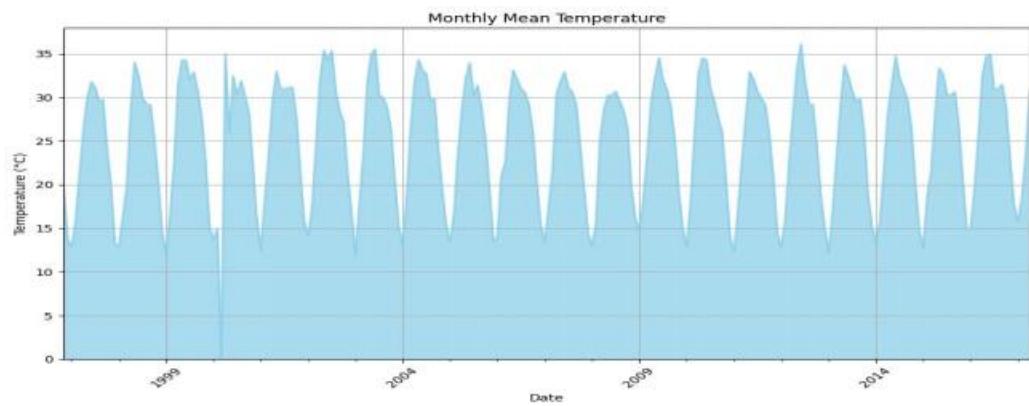


Area Chart:

```

import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('/content/testset.csv')
data.columns = data.columns.str.strip()
data['datetime_utc'] = pd.to_datetime(data['datetime_utc'])
data.set_index('datetime_utc', inplace=True)
monthly_mean_temp = data['_tempm'].resample('M').mean()
plt.figure(figsize=(10, 6))
monthly_mean_temp.plot(kind='area', color='skyblue', alpha=0.7) # Adjust alpha for transparency
plt.title('Monthly Mean Temperature')
plt.xlabel('Date')
plt.ylabel('Temperature (°C)')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



Result: Thus, we successfully implemented the Line graph, Trend lines, Area chart by Time Oriented Data and out is verified