**Implement Programs on DATA VISUALIZATIONS**

**EX:No.2**

**DATE: 31/01/25**

# AIM: To program to implement data visualising using time series data.

# CODE AND DESCRIPTION:

# LINE PLOT

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

weather\_data = pd.read\_csv("/content/weatherHistory.csv")

print(weather\_data.head())

weather\_data["Formatted Date"] = pd.to\_datetime(weather\_data["Formatted Date"])

weather\_data.set\_index("Formatted Date", inplace=True)

plt.figure(figsize=(8, 6))

plt.plot(weather\_data.index, weather\_data["Temperature (C)"], label="Temperature (C)", color="blue")

plt.xlabel("Date")

plt.ylabel("Temperature (Celsius)")

plt.title("Temperature Over Time")

plt.legend()

plt.show()

#BOX PLOT

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

weather\_data = pd.read\_csv("/content/weatherHistory.csv")

weather\_data["Formatted Date"] = pd.to\_datetime(weather\_data["Formatted Date"], errors="coerce", utc=True)

weather\_data = weather\_data.dropna(subset=["Formatted Date"])

weather\_data["Month"] = weather\_data["Formatted Date"].dt.month\_name()

plt.figure(figsize=(8, 6))

sns.boxplot(x="Month", y="Temperature (C)", data=weather\_data, palette="coolwarm", order=[

"January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December"

])

plt.xlabel("Month")

plt.ylabel("Temperature (Celsius)")

plt.title("Temperature Distribution by Month")

plt.xticks(rotation=45)

plt.show()

#HEAT MAP

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

weather\_data = pd.read\_csv("/content/weatherHistory.csv")

# Create a lagged version of the 'Temperature (C)' column (lag = 1)

weather\_data["Temperature Lag1"] = weather\_data["Temperature (C)"].shift(1)

# Drop NaN values (created by the shift)

weather\_data = weather\_data.dropna(subset=["Temperature Lag1"])

# Create the lag scatter plot

plt.figure(figsize=(8, 6))

plt.scatter(weather\_data["Temperature Lag1"], weather\_data["Temperature (C)"], alpha=0.6)

plt.title("Lag Scatter Plot: Temperature vs Temperature (Lag 1)")

plt.xlabel("Temperature (Lag 1) [Previous Hour]")

plt.ylabel("Temperature (C) [Current Hour]")

plt.show()

#SCATTER PLOT

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

weather\_data = pd.read\_csv("/content/weatherHistory.csv")

weather\_data["Formatted Date"] = pd.to\_datetime(weather\_data["Formatted Date"], errors="coerce", utc=True)

weather\_data = weather\_data.dropna(subset=["Formatted Date"])

weather\_data["Month"] = weather\_data["Formatted Date"].dt.month\_name()

weather\_data["Hour"] = weather\_data["Formatted Date"].dt.hour

heatmap\_data = weather\_data.pivot\_table(

values="Temperature (C)",

index="Hour",

columns="Month",

aggfunc="mean"

)

plt.figure(figsize=(10, 8))

sns.heatmap(heatmap\_data, annot=True, cmap="coolwarm", fmt=".1f", linewidths=0.5)

plt.title("Average Temperature by Hour and Month")

plt.xlabel("Month")

plt.ylabel("Hour of the Day")

plt.show()

#AUTOCORRELATION PLOT

import pandas as pd

import matplotlib.pyplot as plt

import statsmodels.api as sm

# Assuming weather\_data is already loaded and preprocessed

# Calculate autocorrelation

autocorrelation = sm.tsa.acf(weather\_data["Temperature (C)"], nlags=48) # Calculate for 48 lags (2 days)

# Plot autocorrelation

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

plt.plot(autocorrelation, marker='o')

plt.title("Autocorrelation Plot of Temperature")

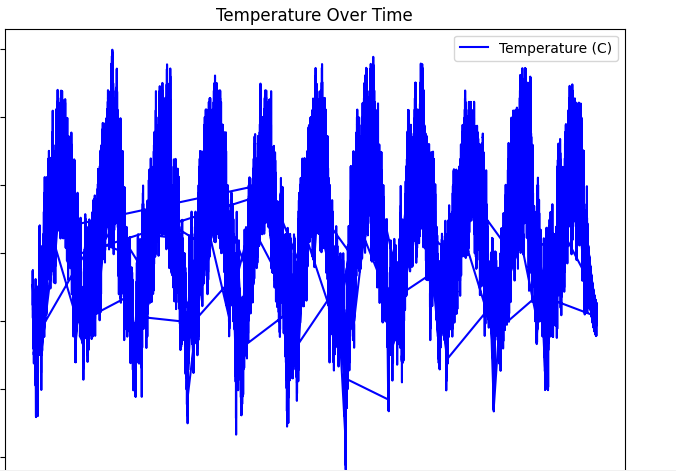
plt.xlabel("Lag (Hours)")

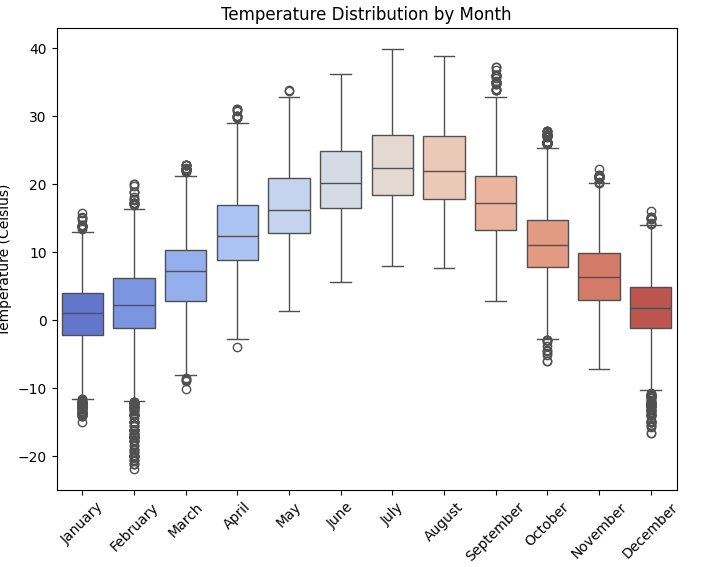
plt.ylabel("Autocorrelation")

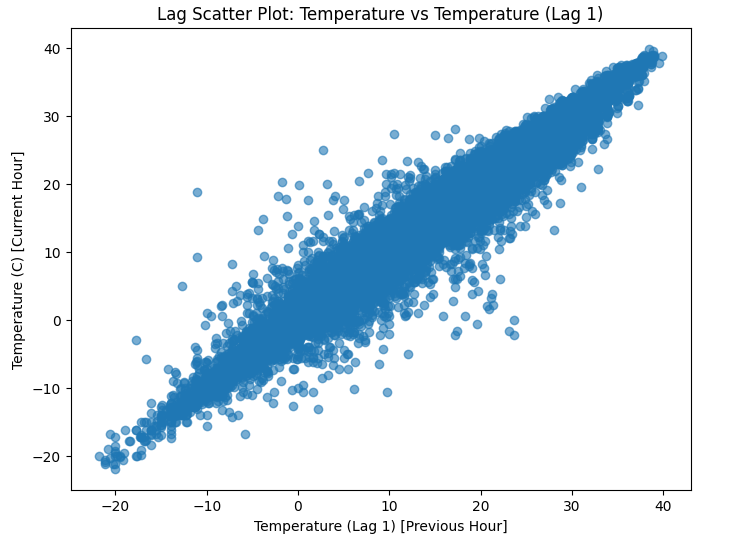
plt.axhline(y=0, color='gray', linestyle='--') # Add horizontal line at 0

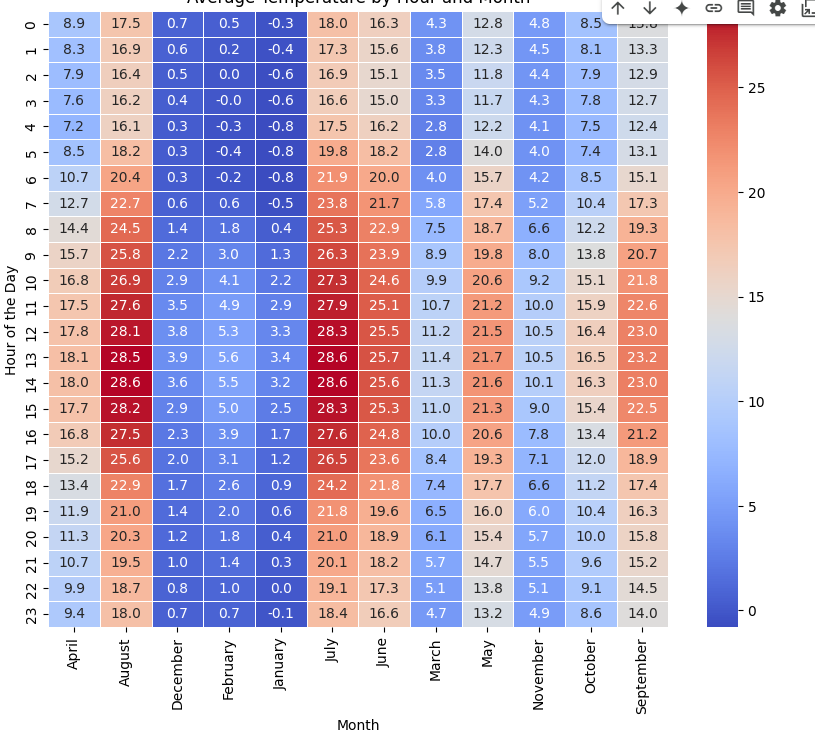
plt.show()

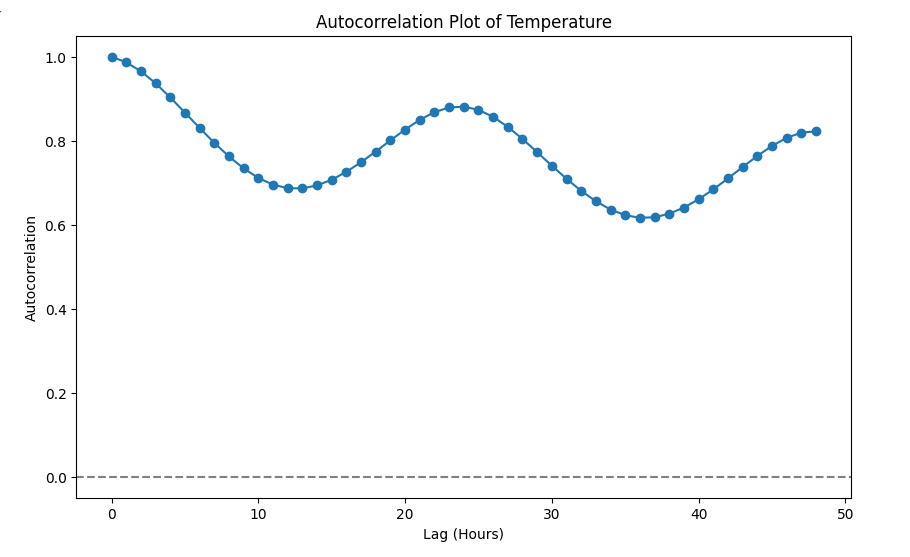
# OUTPUT











**RESULT:**

Thus, the program has been completed and verified successfully.