


Objective: Visualize temperature trends using weather data.

You work for a cutting-edge environmental organization tracking climate change data. Your mission is to visualize temperature trends over time to see how the world's climate has shifted. With tools like Matplotlib and Seaborn, you will create meaningful visualizations to share with researchers and scientists globally.

```
!unzip /content/GlobalLandTemperaturesByCity.csv.zip
```

 Archive: /content/GlobalLandTemperaturesByCity.csv.zip
inflating: GlobalLandTemperaturesByCity.csv

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```


```
import pandas as pd

# Load the dataset
df = pd.read_csv("/content/GlobalLandTemperaturesByCity.csv")

# Convert the date column to datetime format
df["dt"] = pd.to_datetime(df["dt"])

# Select relevant columns
df = df[["dt", "AverageTemperature"]].dropna()
df.rename(columns={"dt": "Date", "AverageTemperature": "Temperature"}, inplace=True)

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



	Date	Temperature
0	1743-11-01	6.068
5	1744-04-01	5.788
6	1744-05-01	10.644
7	1744-06-01	14.051
8	1744-07-01	16.082

```
import pandas as pd

# Load dataset
df = pd.read_csv("/content/GlobalLandTemperaturesByCity.csv")

# Convert date column to datetime format
df["dt"] = pd.to_datetime(df["dt"])

# Select relevant columns
df = df[["dt", "AverageTemperature"]].dropna()
df.rename(columns={"dt": "Date", "AverageTemperature": "Temperature"}, inplace=True)

# Sort by date
df = df.sort_values(by="Date")

# Display first few rows
```

```
print(df.head())
```

```
↗
```

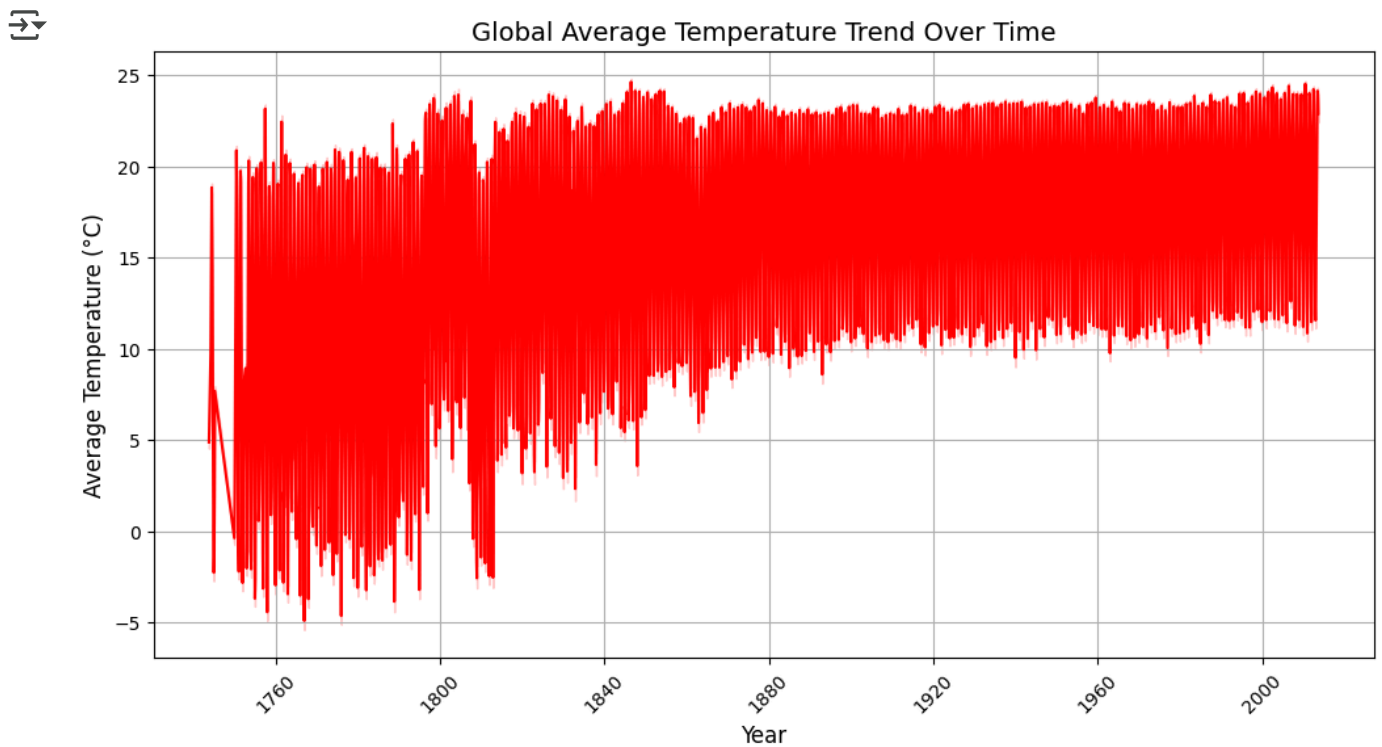
	Date	Temperature
0	1743-11-01	6.068
7409909	1743-11-01	-2.971
6528013	1743-11-01	5.593
5497681	1743-11-01	-3.011
2645561	1743-11-01	1.939

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(12, 6))
sns.lineplot(x=df["Date"], y=df["Temperature"], color="red")
```

```
plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Global Average Temperature Trend Over Time", fontsize=14)
```

```
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



```
print(df.columns)
```

```
↗ Index(['date', 'temperature'], dtype='object')
```

```
import pandas as pd
```

```
# Load dataset
df = pd.read_csv("/content/GlobalLandTemperaturesByCity.csv")
```

```
# Check first few rows
print(df.head())

# Print all column names
print(df.columns)
```

```
↔
      dt AverageTemperature AverageTemperatureUncertainty City \
0  1743-11-01             6.068                1.737 Århus
1  1743-12-01             NaN                NaN Århus
2  1744-01-01             NaN                NaN Århus
3  1744-02-01             NaN                NaN Århus
4  1744-03-01             NaN                NaN Århus

      Country Latitude Longitude
0  Denmark   57.05N   10.33E
1  Denmark   57.05N   10.33E
2  Denmark   57.05N   10.33E
3  Denmark   57.05N   10.33E
4  Denmark   57.05N   10.33E
Index(['dt', 'AverageTemperature', 'AverageTemperatureUncertainty', 'City',
      'Country', 'Latitude', 'Longitude'],
      dtype='object')
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime if not already done
df["dt"] = pd.to_datetime(df["dt"])

# Filter for Aurangabad
df_city = df[df["City"] == "Aurangabad"]

# Extract year from 'dt' column
df_city["Year"] = df_city["dt"].dt.year

# Group by Year and take average temperature
df_city_grouped = df_city.groupby("Year")["AverageTemperature"].mean().reset_index()

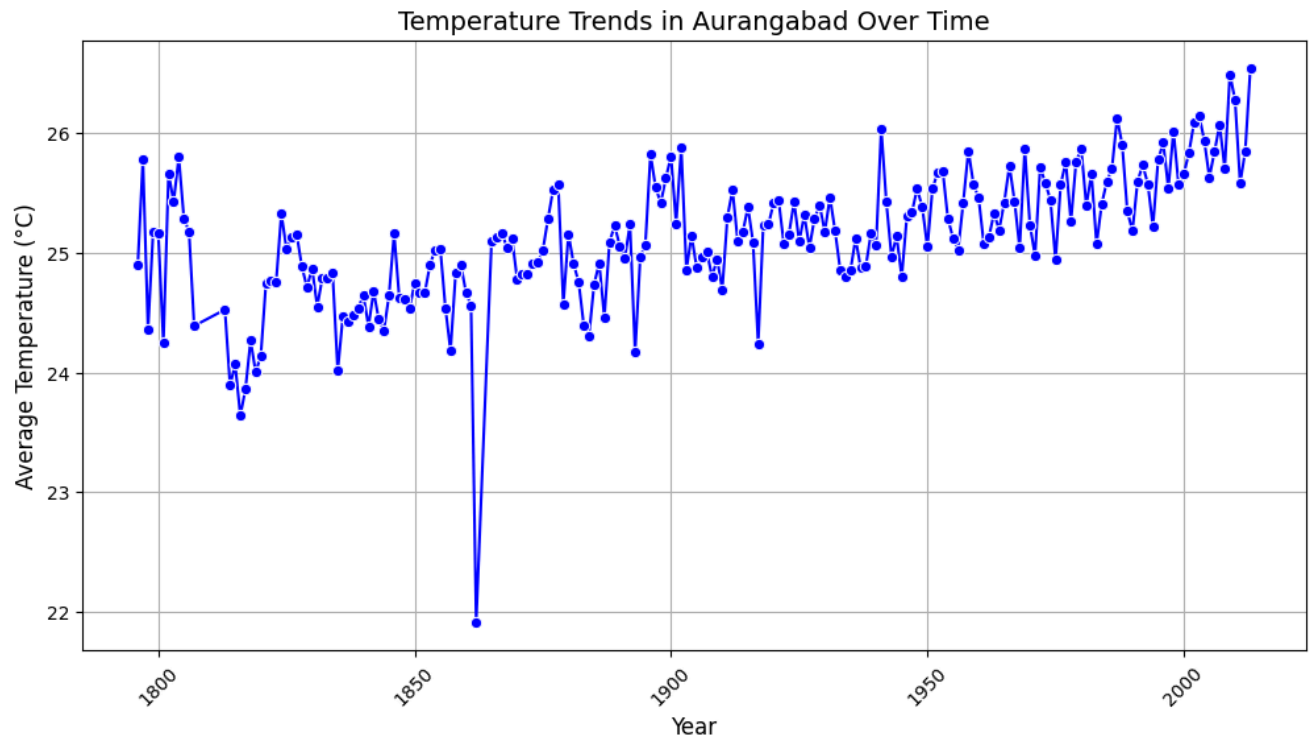
# Plot the temperature trends
plt.figure(figsize=(12, 6))
sns.lineplot(x=df_city_grouped["Year"], y=df_city_grouped["AverageTemperature"], color="blue", marker='x')

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in Aurangabad Over Time", fontsize=14)

plt.xticks(rotation=45) # Rotate for better readability
plt.grid(True)
plt.show()
```

➡ <ipython-input-21-40ba09934ccf>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ir
df_city["Year"] = df_city["dt"].dt.year



```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime if not already done
df["dt"] = pd.to_datetime(df["dt"])

# Filter for India (Case-sensitive correction)
df_city = df[df["Country"] == "India"].copy() # Using `.copy()` to avoid warning

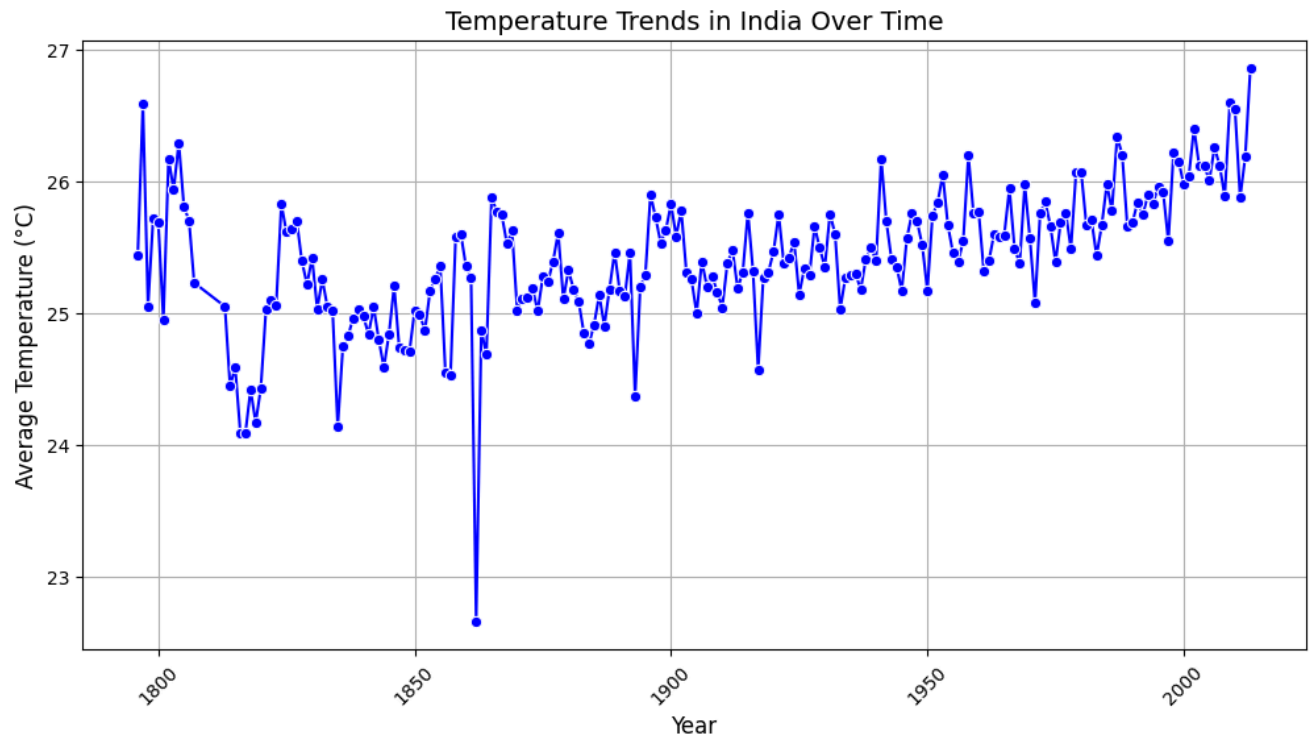
# Extract year from 'dt' column
df_city.loc[:, "Year"] = df_city["dt"].dt.year # Use .loc to avoid warnings

# Group by Year and take average temperature
df_city_grouped = df_city.groupby("Year")["AverageTemperature"].mean().reset_index()

# Plot the temperature trends
plt.figure(figsize=(12, 6))
sns.lineplot(x=df_city_grouped["Year"], y=df_city_grouped["AverageTemperature"], color="blue", marker='o')

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in India Over Time", fontsize=14) # Fixed title

plt.xticks(rotation=45) # Rotate for better readability
plt.grid(True)
plt.show()
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime if not already done
df["dt"] = pd.to_datetime(df["dt"])

# Filter for India (Case-sensitive correction)
df_city = df[df["Country"] == "United States"].copy() # Using `.copy()` to avoid warning

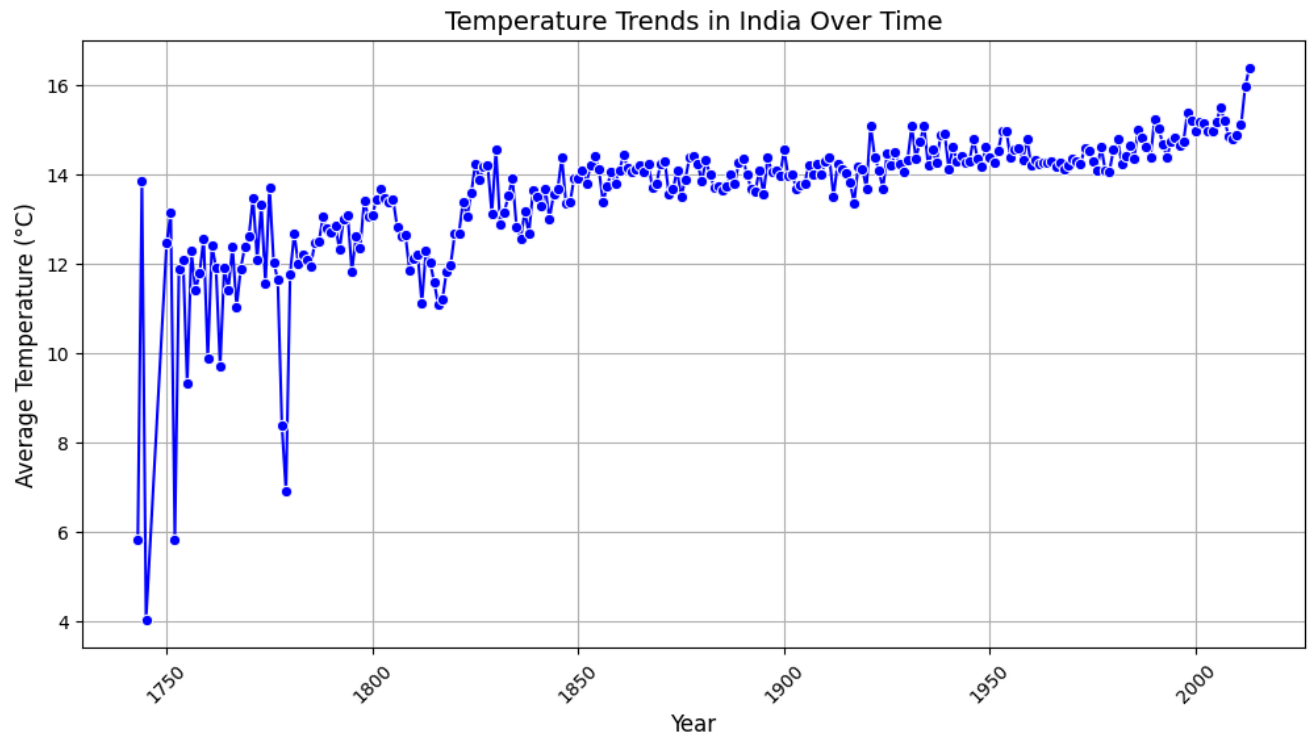
# Extract year from 'dt' column
df_city.loc[:, "Year"] = df_city["dt"].dt.year # Use .loc to avoid warnings

# Group by Year and take average temperature
df_city_grouped = df_city.groupby("Year")["AverageTemperature"].mean().reset_index()

# Plot the temperature trends
plt.figure(figsize=(12, 6))
sns.lineplot(x=df_city_grouped["Year"], y=df_city_grouped["AverageTemperature"], color="blue", marker='o')

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in India Over Time", fontsize=14) # Fixed title

plt.xticks(rotation=45) # Rotate for better readability
plt.grid(True)
plt.show()
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime
df["dt"] = pd.to_datetime(df["dt"])

# Filter for India (Ensure correct case-sensitive match)
df_country = df[df["Country"] == "India"].copy() # Use .copy() to avoid warnings

# Extract year from 'dt' column
df_country["Year"] = df_country["dt"].dt.year

# Group by Year and take average temperature
df_grouped = df_country.groupby("Year")["AverageTemperature"].mean().reset_index()

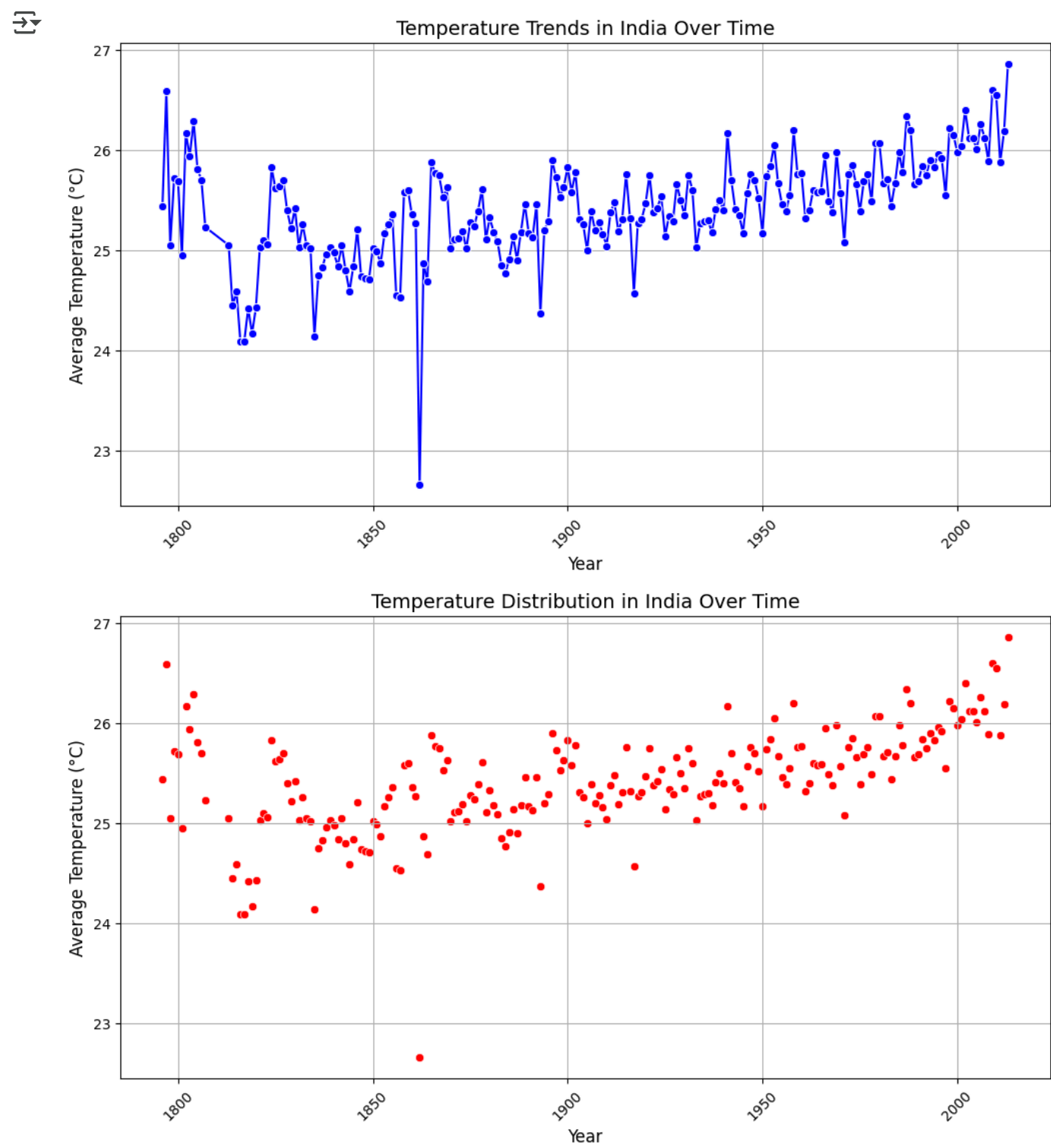
## ● **Line Chart: Temperature Trends Over Time**
plt.figure(figsize=(12, 6))
sns.lineplot(x=df_grouped["Year"], y=df_grouped["AverageTemperature"], color="blue", marker="o")

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in India Over Time", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

## ● **Scatter Plot: Temperature Distribution Over Time**
plt.figure(figsize=(12, 6))
sns.scatterplot(x=df_grouped["Year"], y=df_grouped["AverageTemperature"], color="red")

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Distribution in India Over Time", fontsize=14)
plt.xticks(rotation=45)
```

```
plt.grid(True)
plt.show()
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime
df["dt"] = pd.to_datetime(df["dt"])
```

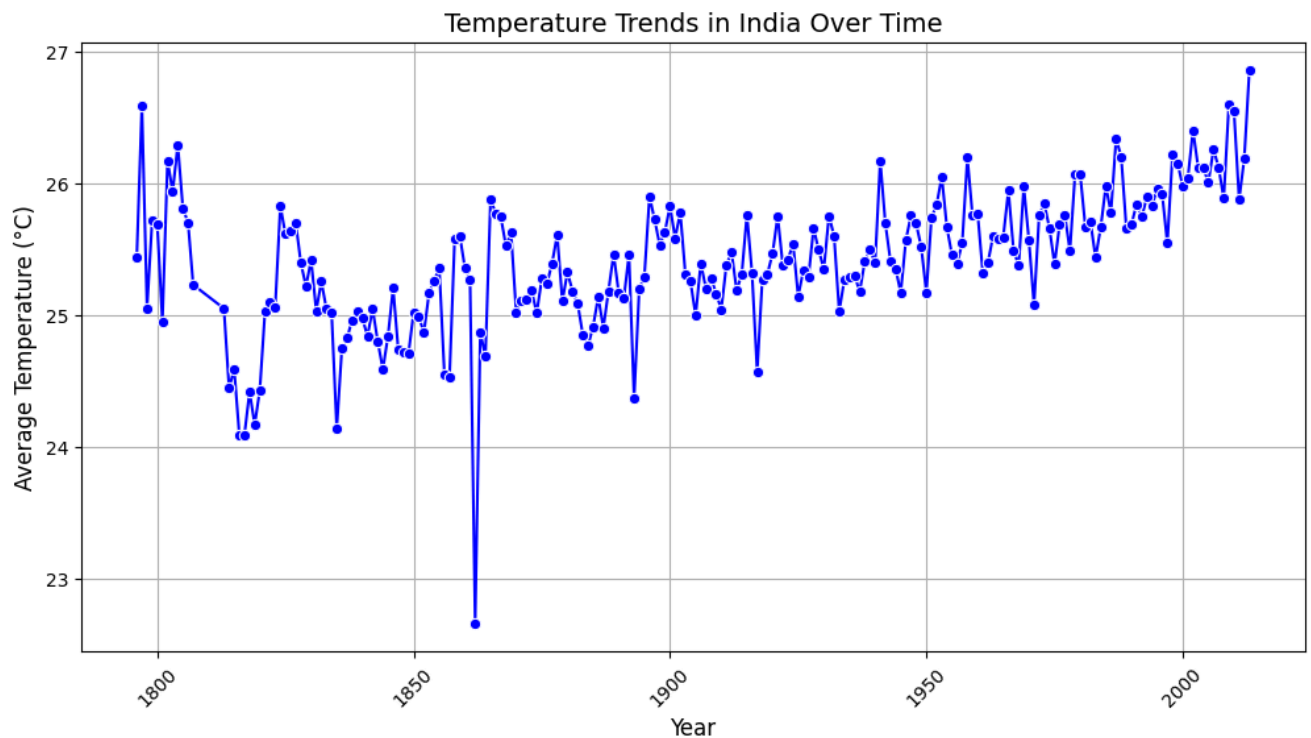
```
# Filter for India (Ensure correct case-sensitive match)
df_country = df[df["Country"] == "India"].copy() # Use .copy() to avoid warnings

# Extract year from 'dt' column
df_country["Year"] = df_country["dt"].dt.year

# Group by Year and take average temperature
df_grouped = df_country.groupby("Year")["AverageTemperature"].mean().reset_index()

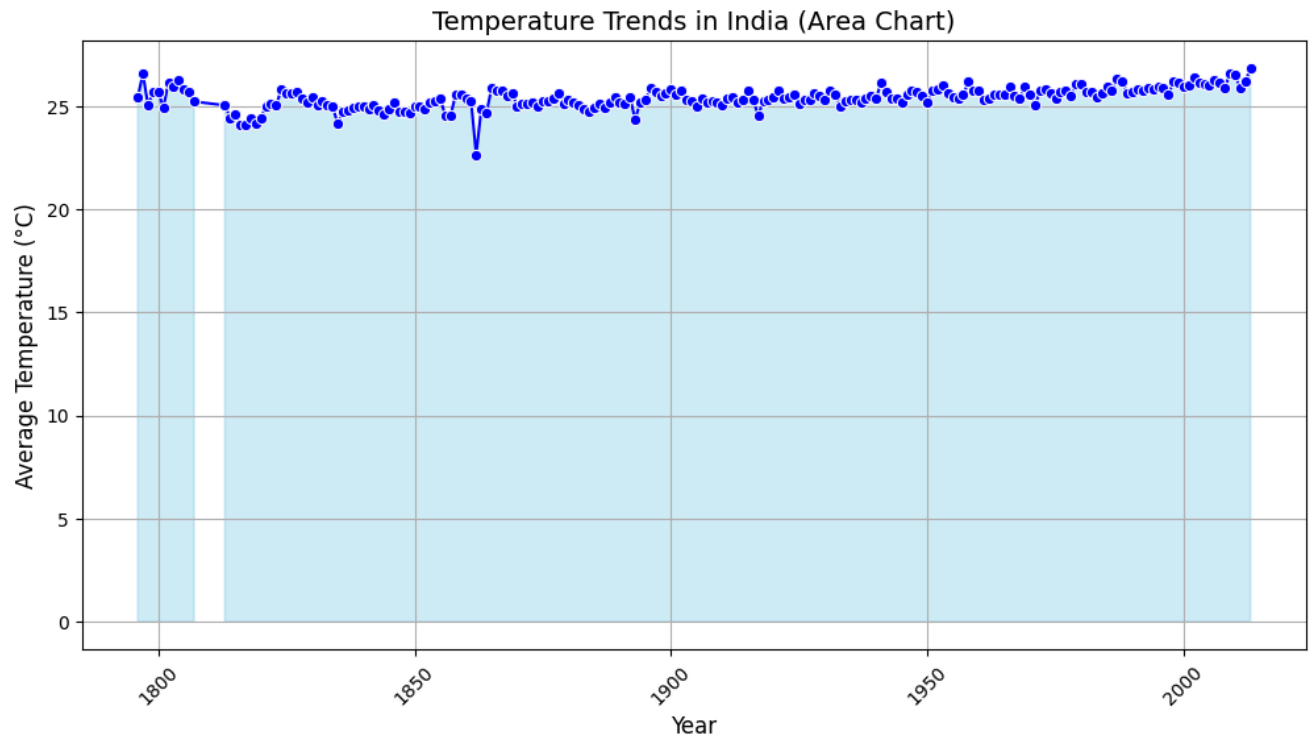
## ● **Line Chart: Temperature Trends Over Time**
plt.figure(figsize=(12, 6))
sns.lineplot(x=df_grouped["Year"], y=df_grouped["AverageTemperature"], color="blue", marker="o")

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in India Over Time", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



```
plt.figure(figsize=(12, 6))
plt.fill_between(df_grouped["Year"], df_grouped["AverageTemperature"], color="skyblue", alpha=0.4)
sns.lineplot(x=df_grouped["Year"], y=df_grouped["AverageTemperature"], color="blue", marker="o")

plt.xlabel("Year", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in India (Area Chart)", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime
df["dt"] = pd.to_datetime(df["dt"])

# Filter for India
df_country = df[df["Country"] == "India"].copy()

# Extract year
df_country["Year"] = df_country["dt"].dt.year

# Group by Decades (10-year bins)
df_country["Decade"] = (df_country["Year"] // 10) * 10 # Grouping into decades
df_grouped = df_country.groupby("Decade")["AverageTemperature"].mean().reset_index()

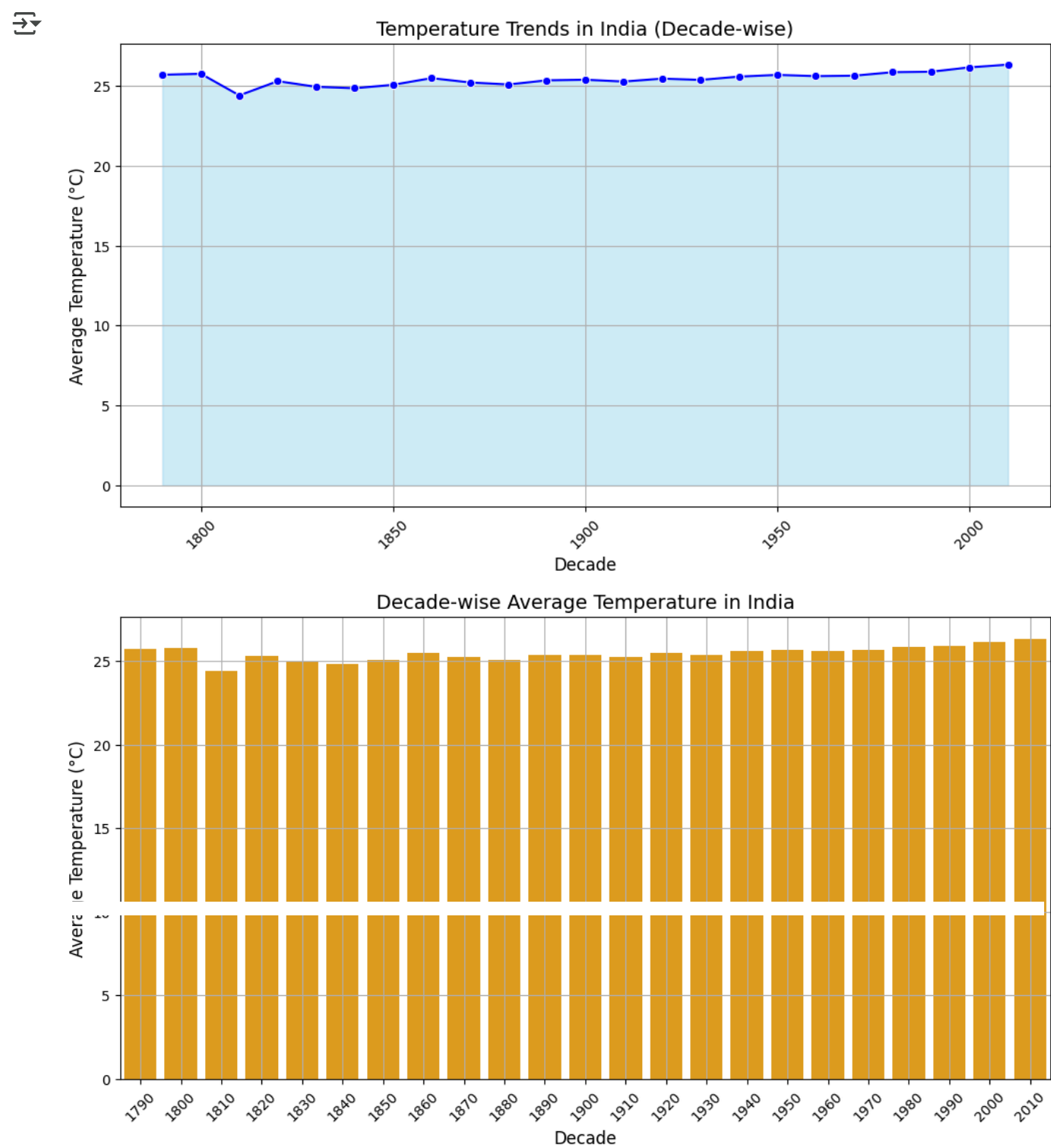
## ● **1. Area Chart (Grouped by Decades)**
plt.figure(figsize=(12, 6))
plt.fill_between(df_grouped["Decade"], df_grouped["AverageTemperature"], color="skyblue", alpha=0.4)
sns.lineplot(x=df_grouped["Decade"], y=df_grouped["AverageTemperature"], color="blue", marker="o")

plt.xlabel("Decade", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
plt.title("Temperature Trends in India (Decade-wise)", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

## ● **2. Bar Chart (Grouped by Decades)**
plt.figure(figsize=(12, 6))
sns.barplot(x=df_grouped["Decade"], y=df_grouped["AverageTemperature"], color="orange")

plt.xlabel("Decade", fontsize=12)
plt.ylabel("Average Temperature (°C)", fontsize=12)
```

```
plt.title("Decade-wise Average Temperature in India", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Convert 'dt' to datetime if not already done
```

```

df["dt"] = pd.to_datetime(df["dt"])

# Filter for India
df_country = df[df["Country"] == "India"].copy()

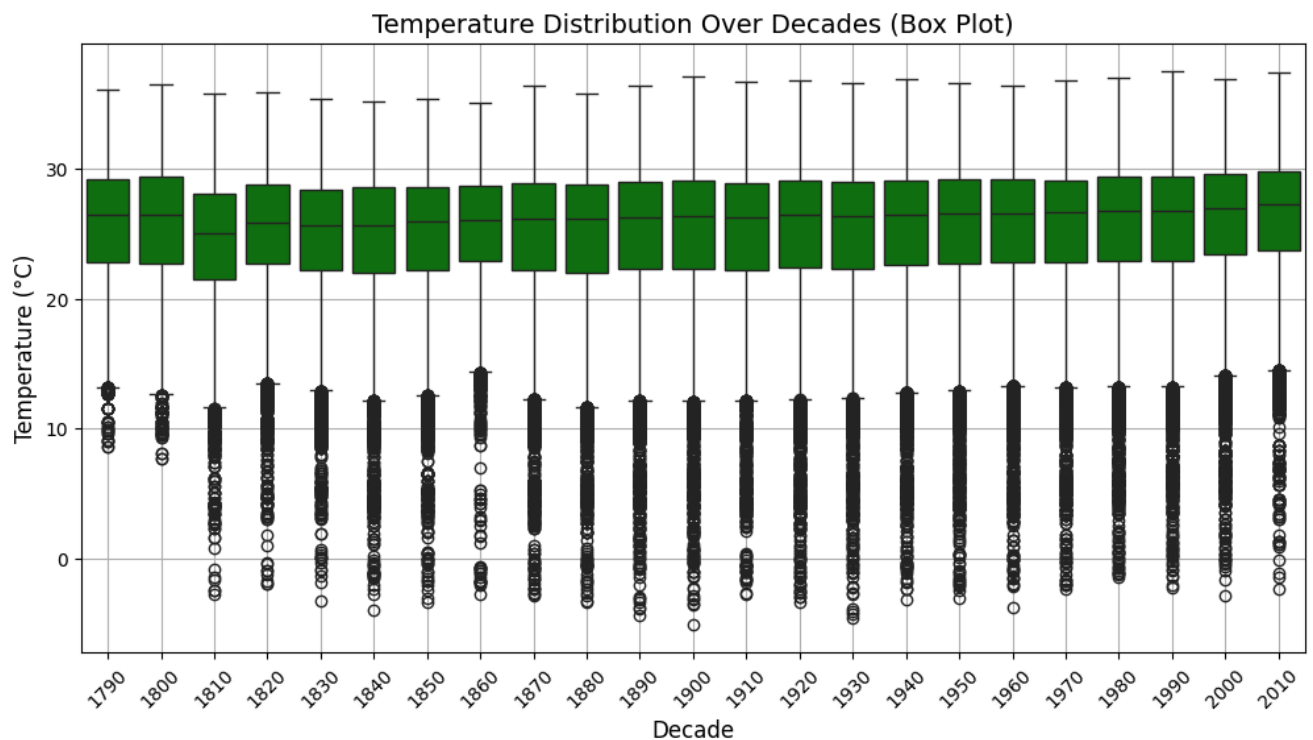
# Extract year
df_country["Year"] = df_country["dt"].dt.year

# Create decade groups
df_country["Decade"] = (df_country["Year"] // 10) * 10 # Grouping into decades

# 🌟 **Box Plot (Grouped by Decades)**
plt.figure(figsize=(12, 6))
sns.boxplot(x=df_country["Decade"], y=df_country["AverageTemperature"], color="green")

plt.xlabel("Decade", fontsize=12)
plt.ylabel("Temperature (°C)", fontsize=12)
plt.title("Temperature Distribution Over Decades (Box Plot)", fontsize=14)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

```



```

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Convert 'dt' to datetime if not already done
df["dt"] = pd.to_datetime(df["dt"])

# Filter for India
df_country = df[df["Country"] == "India"].copy()

# Extract year and month
df_country["Year"] = df_country["dt"].dt.year

```

```
df_country["Month"] = df_country["dt"].dt.month

# Group into **decades** for better visualization
df_country["Decade"] = (df_country["Year"] // 10) * 10

# Pivot table: Rows → Months, Columns → Decades, Values → Avg Temp
df_pivot = df_country.pivot_table(values="AverageTemperature", index="Month", columns="Decade", aggfunc="mean")

# 🌡️ **Heatmap: Monthly Temperature Trends Over Decades**
plt.figure(figsize=(12, 6))
sns.heatmap(df_pivot, cmap="coolwarm", linewidths=0.5, annot=True, fmt=".1f")

plt.xlabel("Decade", fontsize=12)
plt.ylabel("Month", fontsize=12)
plt.title("Monthly Temperature Trends Over Decades (Heatmap)", fontsize=14)
plt.yticks(ticks=np.arange(12), labels=["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"])
plt.show()
```

⚠️ <ipython-input-34-6b879034a887>:19: FutureWarning: The provided callable <function mean at 0x7b5...>
df_pivot = df_country.pivot_table(values="AverageTemperature", index="Month", columns="Decade", aggfunc="mean")

