Visualizing Global Temperature Time Series

Soumabha Bhim

M.Sc. in Statistics and Computing, Banaras Hindu University

25th August - 19th September 2025

Submitted to:

IDEAS – Institute of Data Engineering, Analytics and Science Foundation,
Indian Statistical Institute, Kolkata

Abstract

This project focuses on the visualization and analysis of global temperature data using time series methods. The dataset contains historical records of global temperatures measured from different sources. The primary aim is to understand the trends, variability, and possible climate change signals embedded in the data. Using Python's data science stack (pandas, matplotlib, seaborn, statsmodels), the dataset was cleaned, transformed, and visualized. Key insights such as long-term warming trends and short-term fluctuations were identified. The analysis demonstrates the effectiveness of simple statistical and visualization techniques in interpreting complex environmental data. The project serves as an introductory application of time series exploration to real-world climate-related datasets, highlighting how statistical tools can aid in climate studies. This work provides a baseline for more advanced forecasting and modeling in future projects.

1 Introduction

Global warming and climate change are among the most significant challenges faced today. To study these phenomena, temperature data plays a vital role. Time series analysis allows researchers to observe changes over time, detect patterns, and make inferences about long-term environmental processes.

In this project, a global temperature dataset is analyzed to uncover trends and visualize changes across different time spans. The dataset includes dates and corresponding temperature values recorded by multiple sources. Since temperature is a continuous variable and depends on multiple factors, visualizing it through time provides an effective means of understanding climate behavior.

The project uses Python programming language and libraries such as:

- **pandas** for data manipulation
- **matplotlib** and **seaborn** for visualization
- **statsmodels** for seasonal decomposition

This analysis is relevant because it shows how simple statistical methods can reveal meaningful information from climate datasets.

Topics covered during the training phase:

- Introduction to Time Series
- Exploratory Data Analysis (EDA)

- Data cleaning and preprocessing
- Plotting and visualization
- Basic forecasting and decomposition techniques

2 Project Objectives

1. To explore and visualize the global temperature dataset.
2. To detect long-term warming trends in the data.
3. To study seasonal patterns and short-term fluctuations.
4. To demonstrate preprocessing and visualization techniques in Python.
5. To summarize findings and provide a baseline for advanced time series modeling.

3 Methodology

The project was carried out in the following steps:

1. Dataset loading using pandas.
2. Conversion of the Date column to datetime format.
3. Extraction of Year and other features from Date.
4. Data cleaning and handling missing values.
5. Exploratory Data Analysis (EDA):
   - Plotted temperature trends over time.
   - Aggregated data by year.
   - Compared different data sources.
6. Seasonal decomposition using `statsmodels`.
7. Visualization through line plots, histograms, and decomposition charts.
8. Interpretation of the results in the context of climate change.

Tools and Libraries: Python 3.12, pandas, matplotlib, seaborn, statsmodels.

4 Data Analysis and Results

- **Trend Analysis:** The temperature data shows a clear upward trend over time.
- **Yearly Aggregation:** Averaged yearly data confirms consistent warming.
- **Source Comparison:** Multiple sources agree on rising temperatures.
- **Seasonal Decomposition:** Strong upward trend with seasonal variation and random residuals.
- **Rolling Averages:** Help smooth short-term noise and highlight long-term climate signals.

Visualizations included in this report:

1. Line plots of global temperature vs. time.
2. Comparison across sources.
3. Seasonal decomposition plots.

Data Analysis and Results

The following cells contain the code and outputs from the original notebook. They perform data loading, cleaning, visualization, and analysis on the global temperature time series dataset.

Run the cells sequentially to reproduce all figures and tables.

In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
from datetime import datetime
```

Loading Our Dataset

Now that our setup is ready, it’s time to bring in the data we’ll be working with.
In the next coding cell, we’ll load the COVID-19 dataset directly from csv file.

Once the file is loaded, we’ll **print the first few rows** so that we can get a quick look at what the dataset contains before moving into analysis and visualization.

In [6]:
```
# Replace with your Google Drive file ID which has public view access
file_id = "1kRogzVjnT_2qcjLd0x8jHFmWsq1Ek9NB"

# Construct the download URL
url = f"https://drive.google.com/uc?export=download&id={file_id}"

try:
    # Read CSV directly into pandas
    df_temp = pd.read_csv('data/monthly_csv.csv')
    print(df_temp.head())
except Exception as e:
    print("Error loading data:", e)
```

```
    Source        Date    Mean
0     GCAG  09-10-2017  0.6895
1  GISTEMP  09-10-2017  0.6700
2     GCAG  06-11-2017  0.7851
3  GISTEMP  06-11-2017  0.8500
4     GCAG  06-10-2016  0.7872
```

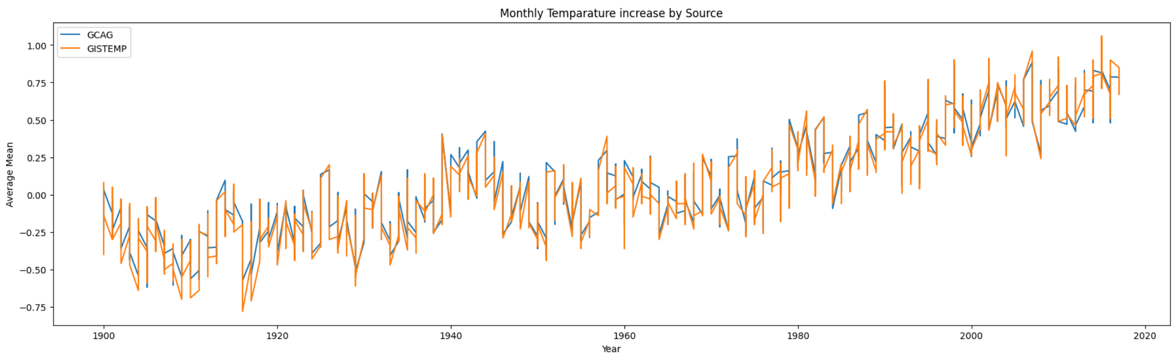
EDA including visualization

Plot 1:

Monthly Revenue Over Time

Visualizing monthly avg temparature obtaining from 2 different sources, which will help helps us understand trends across the years.
Are there peaks during certain months (like due to any particular season)? Let's find out.

In [9]:
```
import matplotlib.pyplot as plt

# Convert Date column to datetime
df_temp['Date'] = pd.to_datetime(df_temp['Date'],  errors = 'coerce')
df_temp['Year'] = df_temp['Date'].dt.year


# Plot line chart
plt.figure(figsize=(22,6))
for source in df_temp['Source'].unique():
    subset = df_temp[df_temp['Source'] == source]
    plt.plot(subset['Year'], subset['Mean'], label=source)

plt.title("Monthly Temparature increase by Source")
plt.xlabel("Year")
plt.ylabel("Average Mean")
plt.legend()
plt.grid(True, linestyle="--", alpha=0)
plt.show()
```



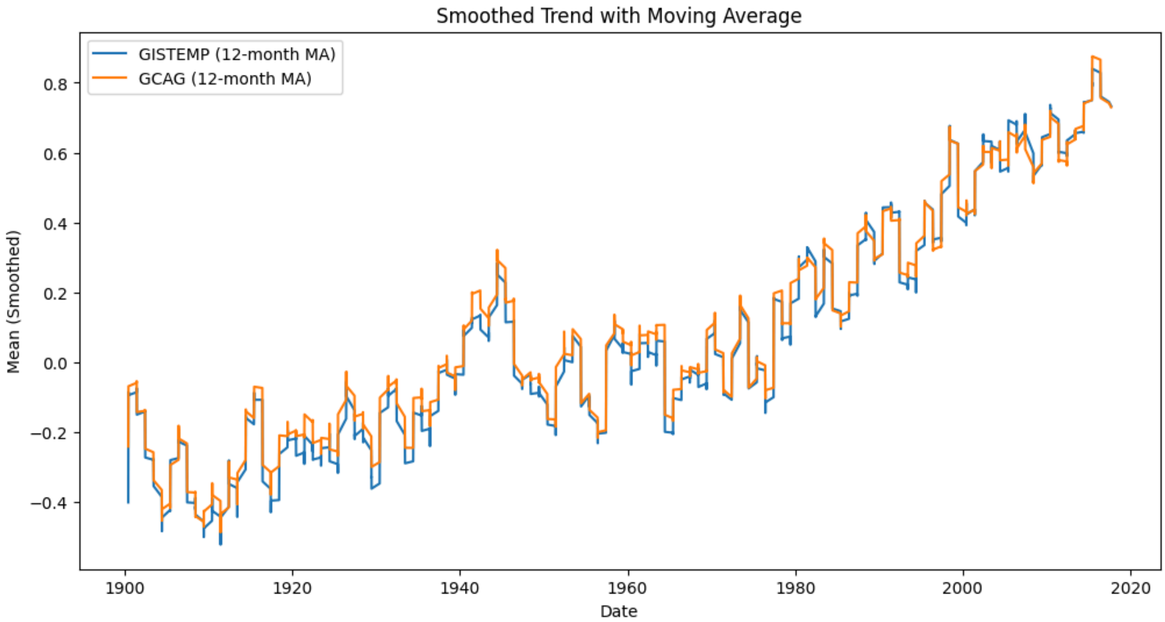
Plot 2:

12 Months Moving Average

Smoothing out seasonal fluctuations and short-term variations to reveal the underlying long-term temperature trends. The 12-month moving average provides a clearer view of climate patterns by averaging each month with the 11 months surrounding it, making it easier to identify gradual warming or cooling trends over the years.

In [10]:
```
# Sort data by Date
df_temp = df_temp.sort_values(by="Date")

# Calculate moving average (e.g., 12-month window)
df_temp['Moving_Avg'] = df_temp.groupby('Source')['Mean'].transform(lambda x: x.

# Plot moving average
plt.figure(figsize=(12,6))
```

```python
for source in df_temp['Source'].unique():
    subset = df_temp[df_temp['Source'] == source]
    plt.plot(subset['Date'], subset['Moving_Avg'], label=f"{source} (12-month MA

plt.title("Smoothed Trend with Moving Average")
plt.xlabel("Date")
plt.ylabel("Mean (Smoothed)")
plt.legend()
plt.show()
```



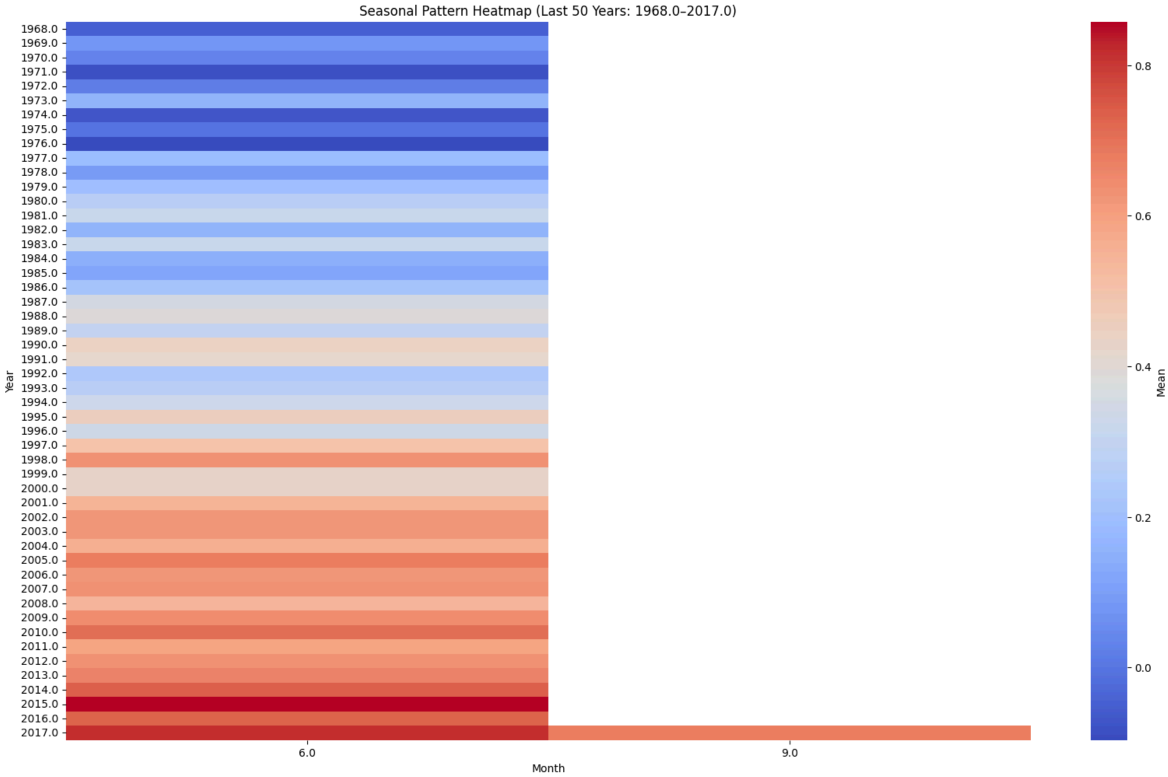
Plot 3

Seasonal Temperature Heatmap (Last 50 Years)

A comprehensive heatmap displaying temperature variations across months and years over the past five decades. This visualization reveals seasonal patterns, climate shifts, and anomalies by showing how temperatures have changed month-by-month across different years. Warmer periods appear as lighter/warmer colors while cooler periods show as darker/cooler colors, making it easy to spot seasonal consistency, unusual weather events, and long-term climate trends.

In [11]:
```python
# Extract Year and Month
df_temp['Year'] = df_temp['Date'].dt.year
df_temp['Month'] = df_temp['Date'].dt.month

# Filter last 50 years
latest_year = df_temp['Year'].max()
df_last50 = df_temp[df_temp['Year'] >= latest_year - 49]

# Group by Year and Month (average across sources if needed)
seasonal_data = df_last50.groupby(['Year','Month'])['Mean'].mean().reset_index()

# Pivot for heatmap
heatmap_data = seasonal_data.pivot(index='Year', columns='Month', values='Mean')

# Plot heatmap
```

```
plt.figure(figsize=(20,12))
sns.heatmap(heatmap_data, cmap="coolwarm", annot=False, cbar_kws={'label': 'Mean
plt.title(f"Seasonal Pattern Heatmap (Last 50 Years: {latest_year-49}–{latest_ye
plt.xlabel("Month")
plt.ylabel("Year")
plt.show()
```



5 Conclusion

The analysis of the global temperature dataset reveals a clear long-term upward trend, consistent with global warming. Short-term fluctuations exist, but the long-term rise is evident. This project shows how statistical tools and simple visualization techniques can effectively analyze environmental data.
Now we can do further study :

- Apply hypothesis testing to measure significance of observed trends.
- Implement forecasting models such as ARIMA or LSTM.
- Expand the dataset with regional studies for deeper insights.

Appendices

References

- IPCC Climate Reports
- NOAA Global Climate Data: `https://www.ncdc.noaa.gov/`
- Documentation: pandas, matplotlib, seaborn, statsmodels