Programming for Artificial Intelligence

Semester Project:

Weather-Trend-System-and-Statistical-Engine

Bahria University, Islamabad

Department of Computer Science

Group Members:	Instructor:
Hasnain Shah # 01-136242-013	Ma’am Samia Kiran
Muhammad Hammad # 01-136242-022	
Zuhad Ahmad # 01-136242-0490	

GitHub Repository Link: https://github.com/mhk61366-svg/Weather-Trend-System-and-Statistical-Engine.git

Submission date: 12/12/2025

Table of Contents

******

Introduction:

By exploring long-term global temperature data, this project addresses the domain of climate change analysis. The increasing frequency of extreme weather events makes understanding historical temperature variations a critically relevant topic. Our primary aim is to analyze and then visualize the monthly average temperature and uncertainty across different cities and countries over a broad time period.

The motivation behind this study is rooted in the urgent need to quantitatively assess and track the Earth's changing climate patterns. We aim to explore whether the fluctuations in temperature are following expected cyclical patterns, or if they exhibit a consistently directional trend, commonly known as global warming.

The selected dataset that records temperatures specifically by city offers us a crucial high-granularity view. This level of detail provides significant real-world value by moving beyond broad global averages to focus on localized climatic phenomena. This allows us to precisely identify:

- **Temperature Hot Spots:** Regions experiencing unusually high or rapidly increasing average temperatures.
- **Cold Areas:** Regions that remain consistently cool or have experienced minimal warming.
- **Climatic Stability or Volatility:** Assessed using the Average Temperature Uncertainty data, which reveals the confidence interval (or degree of variation) in the measurements across various geographic regions. High uncertainty can often correlate with regions experiencing extreme temperature swings (volatility).

By building a systematic analytical framework using Python's Object-Oriented capabilities, we are aiming to provide data-driven and clear insights that can support environmental modeling and inform regional planning regarding climate resilience.

Dataset Description:

The project uses the "GlobalLandTemperaturesByCity" dataset.

- **Dataset Name and Source:** GlobalLandTemperaturesByCity (Source: Kaggle, derived from Berkeley Earth
- **Number of Rows & Columns:** The original dataset has over 8.5 million rows and 7 columns.
- **Meaning of Key Variables:** dt is the timestamp, AverageTemperature is the core variable for analysis, and AverageTemperatureUncertainty provides a measure of data reliability.

- **Known Limitations:** The primary limitation is missing data (NaN) in the AverageTemperature and AverageTemperatureUncertainty columns, particularly in early historical records. This necessitates cleaning steps.
- **Purpose and Fit:** The datasets are used to calculate country-level statistics and map global temperature patterns, directly fitting the goal of identifying trends and volatility.

Methodology:

This project follows an Object-Oriented approach, where we're structuring the code into modular classes responsible for specific tasks.

4.1 Data Cleaning & Preprocessing:

The load_data_set class manages data acquisition and preparation:

- **Handling Missing Values:** The clean_data() method uses dropna() to remove all rows containing NaN values from the DataFrame.
- **Duplicate Removal:** The clean_data() method uses drop_duplicates() to ensure data integrity by removing redundant records.
- **Type Conversions:** The convert_date() method converts the dt column from an object (string) to a proper Pandas datetime object, enabling time-series analysis.
- **Feature Engineering:** The convert_date() method extracts and creates new features: Month, Year, and YearMonth (using dt.to_period("M")) for streamlined grouping and analysis.
- **Rationale:** Removing NaN values and converting the date column are critical to allow mathematical and time-series operations in the statistical and visualization engines.

4.2 Exploratory Data Analysis (EDA):

The Statistical_Engine class performs descriptive analysis through aggregation:

- **Descriptive Statistics:** The class methods calculate country-wide metrics: average_temp(), sd_temp() (Standard Deviation), and variance_temp().
- **Trends Discovered:** Analyzing the output of average_temp() immediately reveals which countries have the highest and lowest mean temperatures over the dataset's period, forming the basis for hottest/coldest country insights.
- **Category-wise Insights:** The use of df.groupby("Country") across all statistical methods focuses the analysis on country-level temperature characteristics.

4.3 Visualizations:

The Weather_Visualizer class contains methods to display the analytical results:

- **Bar Charts (Top 5 Hottest/Coldest):** These plots use plt.bar to visualize the results of the average_temp() aggregation, clearly highlighting the countries at the extremes of the global temperature scale.
- **Standard Deviation Bar Chart:** Visualizing the standard deviation per country helps reveal volatility or anomaly patterns, showing which regions have the most extreme seasonal or year-to-year temperature swings.
- **Global Hotspots Scatter Plot:** The scatter_plot_graph() method creates a scatter plot of Longitude vs. Latitude, coloring the points by the AverageTemperature . This visualization effectively maps temperature data onto a spatial dimension, making geographical patterns and hotspots immediately apparent.

4.4 System Logic / Implementation:

The system is built on an inheritance-based OOP hierarchy:

- **Functions/Classes Created:** Three main classes are defined:
    - ❖ load_data_set: Handles I/O, cleaning, and basic transformation.
    - ❖ Statistical_Engine (inherits from load_data_set): Calculates mean, standard deviation, and variance, utilizing the base class's data.
    - ❖ Weather_Visualizer: Handles all plotting (matplotlib.pyplot).
    - ❖ Weather_Report: Utilizes the Statistical_Engine to generate detailed textual reports (e.g., hottest month per country).
- **Control Structures:** The main_menu(), statistical_analysis_menu(), and graphical_analysis_menu() use infinite while True loops and conditional statements (if/elif/else) based on user input to navigate the program.
- **Code Flow Explanation (Pseudo-Diagram):** The main flow is sequential: Data Loading -> Data Cleaning -> Data Transformation (in the main_menu setup) -> User Interaction (Menu loops) -> Statistical Calculation/Visualization (triggered by user choice).

Results & Insights:

The analysis provided meaningful findings across the global dataset:

- **Patterns/Trends Identified:**

- ❖ The Global Temperature Trend Over Time line plot (Menu 2a), even with short-term fluctuations, reveals a noticeable long-term warming trend when aggregated globally.
- ❖ The Hotspots Map (Menu 2d) clearly shows that countries near the equator generally have the highest average temperatures, and those near the poles or in high mountain regions are the coldest.

- **Real-world Interpretations:** Countries with high Standard Deviation (Menu 2c) experience the most volatile climate, which means challenges for agricultural and infrastructural planning due to unpredictable temperature extremes.
- **Answers to Problem Statement:**
    - ❖ **Hottest Regions:** Identified by the Top 5 Hottest Average Temperatures bar chart.
    - ❖ **Coldest Regions:** Identified by the Top 5 Coldest Average Temperatures bar char.
    - ❖ **Temperature Stability:** Assessed through analyzing the Standard Deviation across countries.

Challenges & Limitations:

- **Data Issues:** The original dataset contained Null values (NaN), which required the wholesale removal of rows using dropna(). This process reduced the available data for early years particularly, potentially skewing the long-term trend analysis.
- **Limitations of Analysis:** The current prediction summary (in Weather_Report class) relies on a simple linear regression model (np.polyfit). This will project the next three months, and this simple model ignores seasonality and complex, non-linear climate factors which makes the predictions indicative but not strictly robust.
- **Tool Constraints:** The project uses Pandas for statistical processing primarily, which is efficient, though it does lack the specialized statistical modeling capabilities of dedicated libraries like SciPy or StatsModels.

Conclusion:

This project successfully achieved its goal of building an OOP-based system to load, clean, sort, analyze, and then visualize global temperature data. The use of Python classes and methods provided a modular and also reusable structure for statistical, graphical engines. The analysis was effective in identifying key geographical and temporal patterns which provided us with insights into global climate trends that prove valuable. Possible extensions include incorporating seasonal decomposition for better time-series analysis. We could also implement a more sophisticated predictive model (like ARIMA) and add a user interface (GUI) for non-terminal interaction.

References:

- [1] The Global Land Temperatures by City Dataset. (Source: Kaggle, link must be provided)
- [2] Pandas Development Team. (2024). *pandas: powerful Python data analysis and visualization*.
- [3] Matplotlib Development Team. (2024). *Matplotlib: a Python data visualization library*.

******