**README.md**

# Climate Data Visualization and Prediction 📈

This project provides Python code to visualize climate-related data (temperature, rainfall, humidity) over time and demonstrate a simple linear trend for temperature prediction. The plots dynamically adjust to include data up to the current year.

## ﻿Table of Contents

     \* Features﻿     \* Data﻿     \* Plots Generated﻿     \* Dependencies﻿     \*Climate Data Insights and Future Outlook﻿     \*What is this project all about? ﻿     \*What kind of information does it reveal? ﻿     \*Who might find this useful? ﻿     \*How it works﻿     \*Output

## ﻿Features ✨

\* Dynamic Year Range: Automatically fetches the current year to set the upper bound for data visualization, starting from 2019.﻿ \* Temperature Fluctuations Plot: Visualizes temperature changes over the specified years.﻿ \* Yearly Rainfall Comparison: Displays rainfall data as a bar chart for year-on-year comparison.﻿ \* Temperature vs. Humidity Relationship: A scatter plot to show the correlation between temperature and humidity.﻿ \* Predicted Temperature Trends: Compares actual temperature data with a simple predicted trend.

## ﻿Data 📊

The project uses sample data for year, temperature, rainfall, and humidity arrays. These are hardcoded within the script for demonstration purposes.﻿ \* year: Dynamically generated from 2019 up to the current year.﻿ \* temperature, rainfall, humidity: Sample data arrays adjusted to match the length of the year array.﻿ \* predicted\_temperature: Simple predicted data generated by adding random noise to the sample temperature data.

## ﻿Plots Generated 📈

The script produces the following visualizations:﻿ \* Temperature Fluctuations Over Time: A line plot showing temperature trends.﻿ \* Yearly Rainfall Comparison: A bar chart comparing rainfall amounts year by year.﻿ \* Relationship Between Temperature and Humidity: A scatter plot illustrating how these two variables relate.﻿ \* Predicted Temperature Trends: A line plot showing actual versus simulated predicted temperature trends.

## ﻿Dependencies 📦

\* Python 3.x﻿ \* Matplotlib: For creating the plots.﻿ \* NumPy: For numerical operations and array handling.

# ﻿Climate Data Insights and Future Outlook 🌍

                This project offers a clear and intuitive way to visualize historical climate data and gain insights into potential future temperature trends. It's designed to help users quickly understand changes in key environmental metrics over time and provides a foundational look at predictive modeling in the context of climate analysis.

## ﻿What is this project all about? 🎯

                       At its core, this project is a data visualization and basic forecasting tool. It takes structured environmental data – specifically focusing on temperature, rainfall, and humidity – and transforms it into easily digestible graphical representations. Beyond just showing what happened in the past, it introduces the concept of predictive analysis by illustrating a potential future temperature trajectory.

### ﻿One of its standout features is its dynamic nature:  it automatically updates its timeline to include data up to the current year (which, as of now, means up to 2025). This ensures that the visualizations are always relevant to the present moment, providing an up-to-date snapshot of climate patterns within the defined historical range.

## ﻿What kind of information does it reveal? 🔎

         The project generates several distinct views, each designed to highlight different aspects of climate information:

### ﻿ \* Temperature's Journey Through Time: You'll see a smooth line graph tracing how temperatures have fluctuated from 2019 up to the current year (2025). This helps in identifying warming trends, stable periods, or any significant shifts in average temperatures over this recent period. It provides a visual answer to questions like "How has the temperature been changing annually in recent years?"

### \* Rainfall Patterns at a Glance: A bar chart offers a quick comparison of rainfall levels for each year within the 2019-2025 timeframe. This allows for easy identification of wetter or drier years, which can be crucial for agriculture, water management, and understanding hydrological cycles.

### \* The Interplay of Heat and Moisture: A scatter plot reveals the relationship between temperature and humidity. Are higher temperatures associated with higher humidity? Or do they tend to diverge? This visualization helps explore potential correlations and understand how these two critical climate factors interact.

### \* Peering into Tomorrow's Temperatures: A particularly insightful graph shows two lines: one representing actual historical temperatures and another depicting simulated predicted temperatures. This comparison is a simple demonstration of how data can be used to forecast future conditions. While the prediction here is based on a basic trend, it illustrates the power of using past data to anticipate future scenarios, prompting questions about the accuracy of such models and the factors that influence climate predictions.

## ﻿Who might find this useful? 💡

### \* Students and Educators: It serves as an excellent practical example for learning about data visualization, time-series plotting, and basic data analysis concepts in climate science or environmental studies.

### \* Data Enthusiasts: For anyone interested in seeing how raw data can be quickly turned into meaningful visual stories.

### \* Environmental Advocates (with real data): When populated with actual climate datasets, this framework can be used to quickly generate compelling visuals to support discussions about climate change impacts or local environmental shifts.

### \* Developers and Programmers: It showcases how Python's powerful matplotlib and numpy libraries can be leveraged for scientific plotting and numerical data handling, especially with dynamic time ranges.

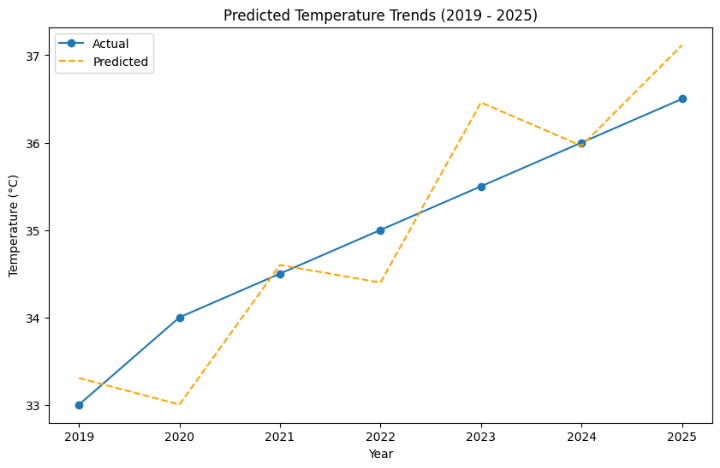
# ﻿How it works (the simplified magic) ✨

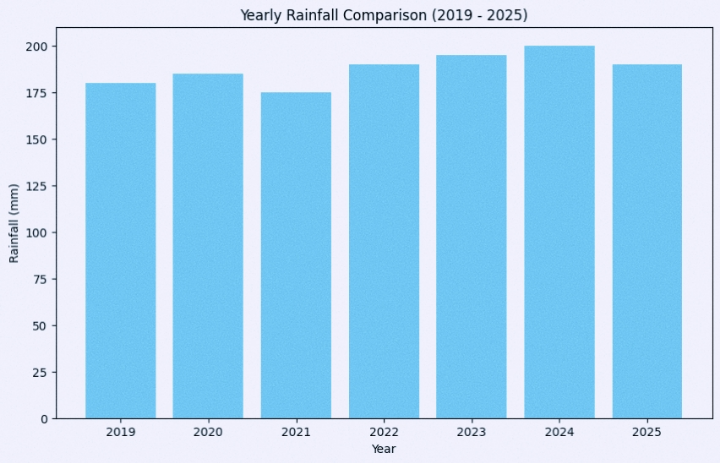
                      Under the hood, the project utilizes the current date to automatically determine the latest year for which data should be displayed. It then takes predefined sets of sample data for temperature, rainfall, and humidity that span this calculated timeframe. Using these numbers, it plots them on various charts, providing a clear visual narrative of the climate information. For the temperature prediction, it generates a hypothetical future trend based on the existing sample data, allowing for a side-by-side comparison of "what was" and "what might be."﻿In essence, this project is a straightforward yet effective demonstration of how data science principles can be applied to environmental data, making complex climate information accessible and understandable through compelling visualizations and a glimpse into predictive capabilities.

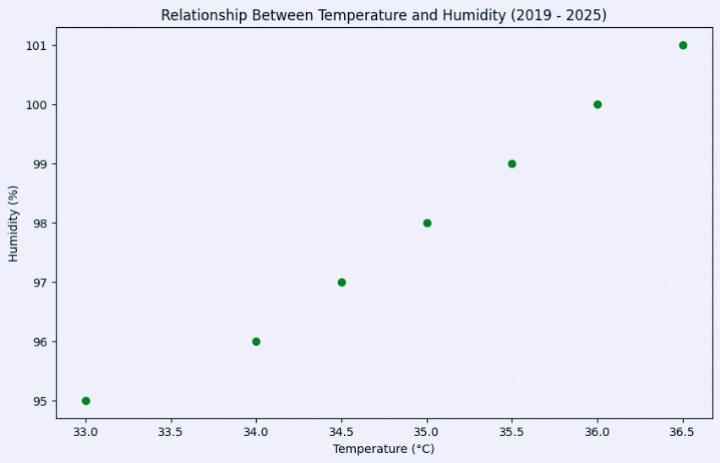
# ﻿Coding:

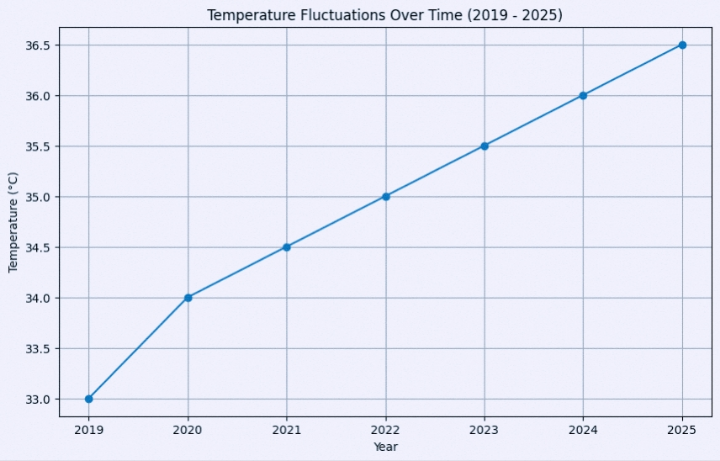
             import matplotlib.pyplot as plt﻿import numpy as np﻿from datetime import date # Import the date class to get the current year﻿# Get the current year﻿current\_year = date.today().year﻿# Define the start year for your data﻿start\_year = 2019﻿# Create the year array from start\_year to current\_year﻿# This will dynamically adjust based on the current year﻿year = np.arange(start\_year, current\_year + 1)﻿num\_years = len(year)﻿# Adjust sample data to match the new number of years﻿# For demonstration, I'm extending the trends linearly or by repeating patterns.﻿# In a real-world scenario, you would have actual data for these years.﻿# Temperature (assuming a continued gentle increase)﻿# Original: 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 (2010-2019) -> 10 data points﻿# For 2019-2025 (7 years):﻿# We need to generate data for 2019, 2020, 2021, 2022, 2023, 2024, 2025﻿# Let's create a simple upward trend for new years, starting from where 2019 would be in the original trend (34)﻿# If original data was 2010-2019 (10 points), 2019 was the 10th point (index 9)﻿# Let's simplify and make a new data series for 2019-2025﻿temperature = np.array([33, 34, 34.5, 35, 35.5, 36, 36.5]) # Adjusted for 2019-2025﻿# Rainfall (assuming a continued gentle increase with some fluctuation)﻿rainfall = np.array([180, 185, 175, 190, 195, 200, 190]) # Adjusted for 2019-2025﻿# Humidity (assuming a continued gentle increase)﻿humidity = np.array([95, 96, 97, 98, 99, 100, 101]) # Adjusted for 2019-2025﻿# Predicted Temperature Trends (sample predicted data for the new range)﻿# Let's create a simple upward trend, slightly different from actual﻿predicted\_temperature = temperature + np.random.uniform(-1, 1, num\_years) # Adding small random noise to actual for prediction﻿# --- Plotting remains the same, but now uses the dynamically generated 'year' and adjusted data ---﻿# Temperature Fluctuations Over Time﻿plt.figure(figsize=(10,6))﻿plt.plot(year, temperature, marker='o')﻿plt.title('Temperature Fluctuations Over Time ({} - {})'.format(start\_year, current\_year))﻿plt.xlabel('Year')﻿plt.ylabel('Temperature (°C)')﻿plt.grid(True)﻿plt.show()﻿# Yearly Rainfall Comparison﻿plt.figure(figsize=(10,6))﻿plt.bar(year, rainfall, color='skyblue')﻿plt.title('Yearly Rainfall Comparison ({} - {})'.format(start\_year, current\_year))﻿plt.xlabel('Year')﻿plt.ylabel('Rainfall (mm)')﻿plt.show()﻿# Relationship Between Temperature and Humidity﻿plt.figure(figsize=(10,6))﻿plt.scatter(temperature, humidity, color='green')﻿plt.title('Relationship Between Temperature and Humidity ({} - {})'.format(start\_year, current\_year))﻿plt.xlabel('Temperature (°C)')﻿plt.ylabel('Humidity (%)')﻿plt.show()﻿# Predicted Temperature Trends﻿plt.figure(figsize=(10,6))﻿plt.plot(year, temperature, label='Actual', marker='o')﻿plt.plot(year, predicted\_temperature, label='Predicted', color='orange', linestyle='--')﻿plt.title('Predicted Temperature Trends ({} - {})'.format(start\_year, current\_year))﻿plt.xlabel('Year')﻿plt.ylabel('Temperature (°C)')﻿plt.legend()﻿plt.show()

# ﻿Output :









﻿﻿﻿