

An analysis on weather parameters of different Indian cities

Jaykant Nayak, Aditya N Singh, Tapas Mandal, Dhruv Shetty

Introduction

Understanding weather patterns is essential for various fields, including agriculture, disaster management, public health, and urban planning. In a country as diverse as India, with its wide range of climatic zones, accurate weather predictions can help improve decision-making and preparedness for a range of weather-dependent activities. This project aims to analyze weather data across various Indian cities, providing insights into daily and seasonal patterns through both historical and live data. By leveraging this information, we can make data-driven predictions that support proactive weather-related planning and response.

Goals

The primary goal of this project is to develop a robust weather analysis and prediction tool for Indian cities. Using both historical data and live data scraped from city-specific URLs, this project will assess and visualize weather trends, offering predictions to help anticipate temperature and precipitation levels. We aim to deliver insights through a user-friendly Shiny app that can serve as a valuable resource for understanding and predicting weather variations.

Objectives

To achieve these goals, the project is structured around the following objectives:

1. Data Collection and Preprocessing

- To systematically collect live and historical weather data for selected Indian cities.
- To preprocess and clean the data, ensuring accuracy and consistency for analysis.

2. Visualization of Weather Patterns

- To create interactive visualizations that illustrate weather trends, such as daily temperature and precipitation levels, across different time periods.
- To provide users with tools to explore weather variations and gain insights into city-specific climate characteristics.

3. User-Friendly Shiny Application

- To create an interactive Shiny app that allows users to access real-time weather data, analyze historical trends, and view weather predictions.
- To ensure the app is intuitive, enabling users from diverse backgrounds to easily interpret weather insights and make informed decisions.

Through this comprehensive analysis, we aim to contribute to the growing need for reliable and accessible weather data tools in India. The final Shiny app will serve as a platform for analyzing weather data and providing future temperature predictions, ultimately supporting individuals and organizations in weather-sensitive planning.

Data Collection

The weather data used in this analysis was sourced from several platforms, including publicly available datasets and weather APIs. Specifically, the primary data for this project was obtained from:

- **EaseWeatherAPI:** A free weather API that provides daily weather data for various cities. The API was used to scrape historical weather data, including parameters like temperature, precipitation, wind speed, and air pressure.
- **India Meteorological Department (IMD):** Historical weather data for Indian cities was accessed through datasets shared by the IMD, providing detailed records of weather parameters over the past few years.
- **OpenWeatherMap API:** For real-time weather data, this API was used to fetch current temperature, precipitation, and other atmospheric conditions for cities across India. This data was crucial for generating current weather reports and short-term forecasts.

The decision to use these data sources was based on their comprehensive coverage of weather parameters, the availability of historical data, and the ease of access through API endpoints. These datasets provided information for cities across India, with a focus on regions that show significant variation in weather patterns.

Data Selection and Relevance

The weather data chosen for this report includes the following parameters: - **Temperature:** Including average, maximum, and minimum temperature for each day. Temperature is a key

factor in understanding climate trends and is important for both daily planning and long-term climate analysis. - **Precipitation:** This includes daily rainfall and snow data, which is essential for analyzing seasonal variations and identifying the wettest or driest periods of the year. - **Wind Speed and Direction:** Wind data is useful for studying storm patterns and wind-chill effects, important for both meteorological predictions and environmental studies. - **Air Pressure:** Changes in air pressure can provide insights into weather systems, such as high-pressure zones (clear weather) and low-pressure zones (stormy weather).

These parameters were selected because they are commonly used in meteorological studies and provide a well-rounded understanding of weather patterns. By including a combination of temperature, precipitation, wind speed, and air pressure, we can explore multiple facets of the climate and identify significant trends.

Data Cleaning Process

The data cleaning process was essential to ensure consistency and accuracy in the datasets. Here are the specific steps taken to clean the data:

1. **Handling Missing Data:** Any missing values for precipitation were replaced with zeroes, assuming that a lack of reported precipitation indicates no rainfall or snow on that day. Other missing values were handled through imputation or exclusion, depending on the parameter and context.
2. **Removing Irrelevant Columns:** Several columns that were not useful for this analysis (e.g., snow depth, wind gust speed) were removed from the dataset to simplify the data and focus on the core weather parameters.
3. **Merging Data:** The weather data for each city was split across multiple files for different years. These datasets were merged into a single dataset per city for a cohesive time series, allowing for easier analysis and comparison across years.
4. **Date Formatting:** The date columns were standardized across all datasets to ensure that all records aligned properly for time-based analysis.
5. **Standardizing Units:** Some datasets had temperature values in Fahrenheit, while others were in Celsius. All temperature values were converted to Celsius to maintain uniformity across the datasets.

Explanation of some important functions in R code:

Reading Data (`read.csv`)

We start by reading CSV files for different years (2022, 2023, and 2024) into separate data frames. The `read.csv()` function is used here to import data from the CSV files into R.

Combining Data (`bind_rows`)

Once the individual datasets are loaded, we combine them into a single dataset using `bind_rows()`. This function is preferred over `merge()` because it is more efficient and easier to use when simply concatenating rows from multiple data frames. The `bind_rows()` function automatically handles the alignment of columns across the datasets. This approach is ideal for merging data with the same columns but possibly in different files.

Why not `merge()`?

The `merge()` function is more commonly used when we need to combine datasets based on a common column (such as an ID column). Since we are combining data based on the same structure and order of rows, `bind_rows()` is the more straightforward option.

Selecting Relevant Columns (`select`)

After merging, we use the `select()` function from the `dplyr` package to retain only the relevant columns needed for our analysis. In this case, we remove columns like `snow`, `wpgt`, and `tsun`, which are not essential for weather analysis in this context.

Handling Missing Data (`mutate with replace_na`)

Missing values in the dataset (such as missing precipitation data) are handled using the `mutate()` function from `dplyr`. The `replace_na(prcp, 0)` command replaces any NA values in the `prcp` (precipitation) column with 0, assuming that if no precipitation is reported, there was no rain or snow on that day. This method ensures that the dataset is consistent and complete for further analysis.

Renaming Columns (`rename`)

Finally, the `rename()` function is used to give more meaningful names to the columns. This improves readability and makes the dataset easier to work with. For example: - `tavg` is renamed to `Average_Temperature`. - `tmin` is renamed to `Minimum_Temperature`. - `tmax` is renamed to `Maximum_Temperature`.

Biases in the data

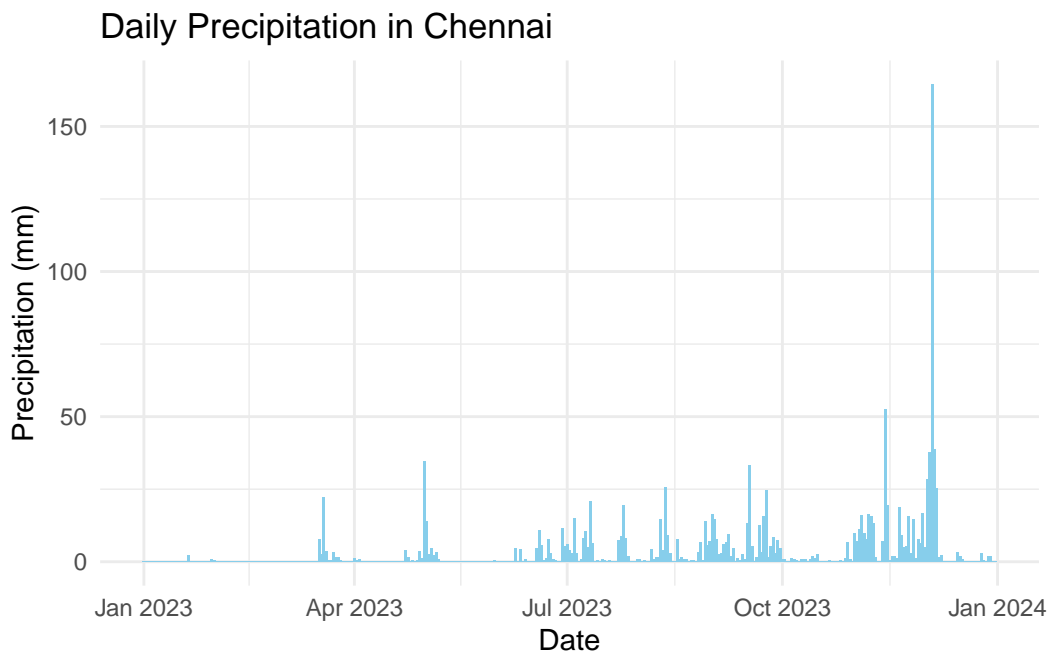
- Each platform may use different weather station networks, satellite data, and algorithms to generate forecasts and reports. This could lead to discrepancies when comparing data across platforms.

- Weather data providers often use smoothing or interpolation to estimate conditions for locations without direct weather stations. This can result in averaged values that may not capture extreme events or local anomalies accurately.
- The period of 2022-2024 might reflect ongoing climate changes, with possible warming trends or unusual weather patterns that differ from historical averages. This could affect predictive modeling if assumptions are based on past “normal” conditions.
- Data collection intervals (e.g., hourly, daily) may differ, meaning some sources could miss brief but significant events (e.g., short rain showers, sudden temperature drops) depending on the granularity of their measurements.

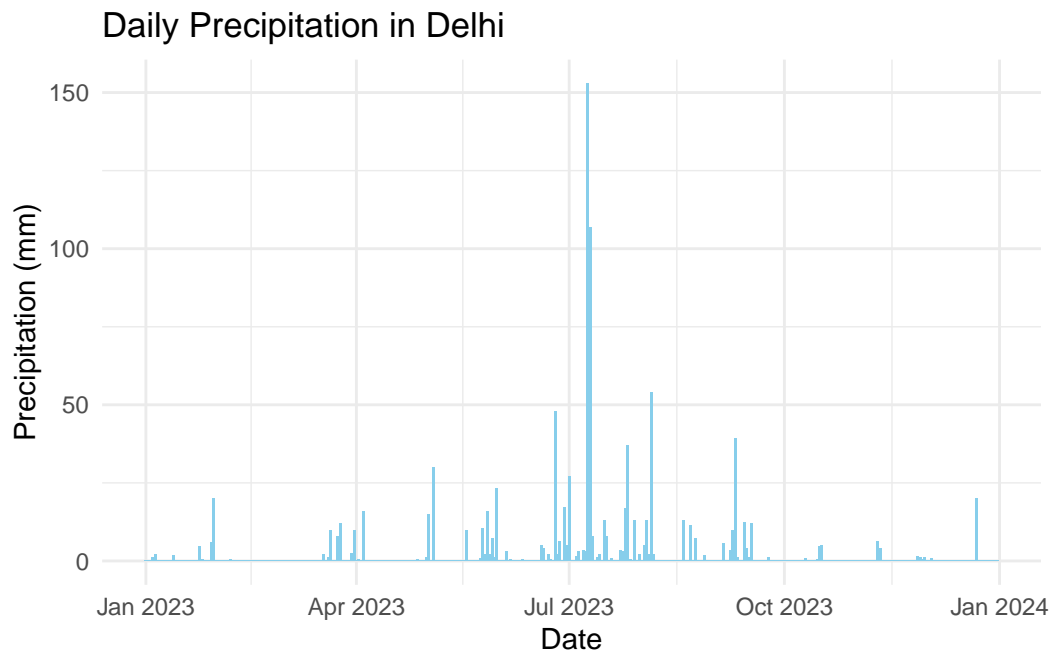
Intriguing questions

- What are the average temperature and precipitation trends across different seasons in each city?
- Has there been a noticeable increase in average temperatures across cities over the last few years (climate change)?
- How do temperature and precipitation levels compare between northern and southern cities of India?

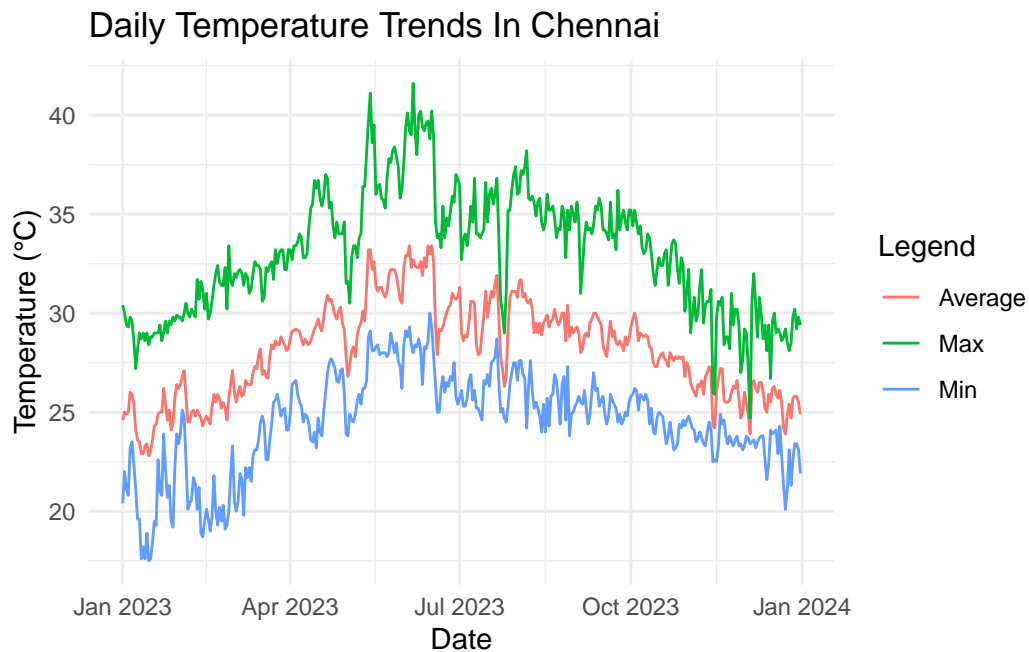
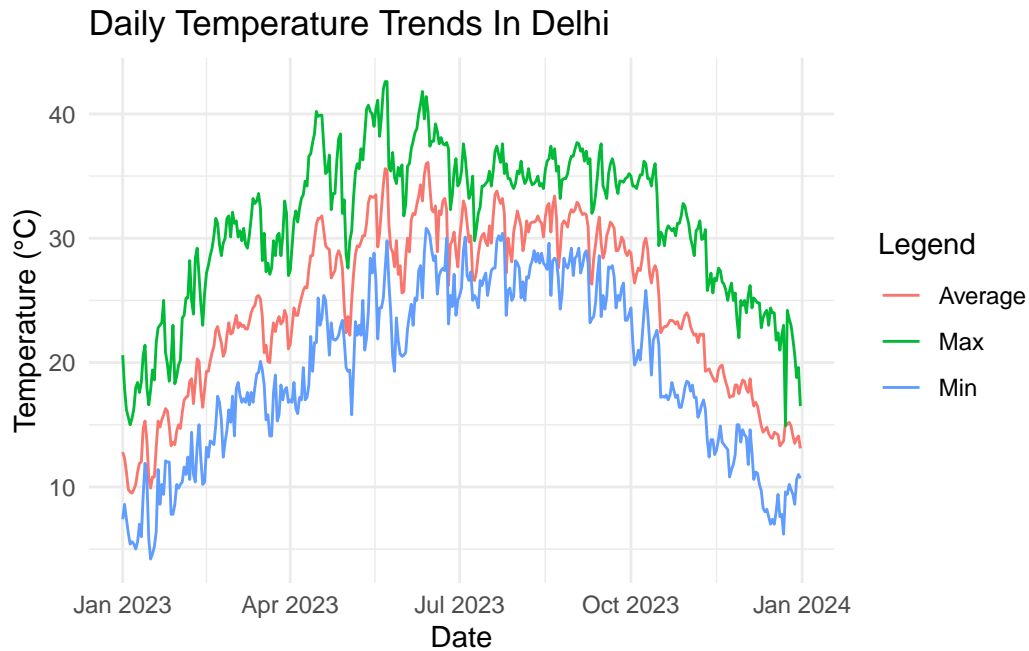
Visualisations



Precipitation in Chennai occurs mainly in mid-October to December-end.



Whereas, in Delhi it occurs in June to October. This clearly shows that monsoon hits early in Delhi whereas in Chennai its the returning monsoon due to which the rain occurs at the end of year.



From both the plots we can see that the average temperature of Chennai is quite higher than Delhi. Similiar inferences can be made for different cities as well.

Conclusions

- The app provides clear insights into the seasonal temperature and precipitation trends for each city, showing how weather conditions fluctuate throughout the year. We can conclude that certain cities experience more extreme variations, while others have more stable weather.
- The app can provide a clear comparison between cities in the north (such as Delhi) and the south (such as Chennai), showing differences in temperature, humidity, and precipitation. This could be used to conclude that weather in coastal cities is influenced by factors like proximity to the sea, which leads to higher humidity and milder temperatures compared to inland cities.
- The app reveals that rainfall patterns have shifted, with some cities experiencing heavier rainfall during shorter periods, or reduced rainfall, which could have implications for agriculture, water management, and disaster preparedness.

References

1. [EaseWeatherAPI](#)
2. [India Meteorological Department \(IMD\)](#)
3. [OpenWeatherMap API](#)