

Data Science Tool Box Python Programming

**PROJECT REPORT**

(Project Semester January-April 2025)

***NRSC- Rainfall District Wise-Jan2022***

***( Exploring U.S. School Data with Python)***

Submitted by

**(G.ANAND)**

Registration No. **12306013**

Programme and Section. Data science K23GN

Course Code . INT375

Under the Guidance of

**(Aashima )**

**Discipline of CSE/IT**

**Lovely School of Computer Science**

**Lovely Professional University, Phagwara**

**CERTIFICATE**

This is to certify that G.ANAND bearing Registration no 12306013 has completed INT-375 project titled, **“NRSC-rainfall-districtwise-jan22\_0.csv”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

**DECLARATION**

I G.ANAND student of Computer Science under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: Signature

Registration No. 12306013 Name of the student: G.ANAND

|  |  |
| --- | --- |
| Title | **Pg .no** |
| **1. Introduction** | **3** |
| **2.** **Source of Dataset** | **4** |
| **3. EDA Process** | **4** |
| **4. Analysis on Dataset** | **4** |
| **4.1 Data Loading and Inspection** | **4** |
| **4.2 Handling Missing Data** | **5** |
| **4.3 Statistical Summary** | **6** |
| **4.4 State-wise School Distribution** | **6** |
| **4.5 Correlation Heatmap** | **9** |
| **4.6 Outlier Detection Using Boxplot** | **10** |
| **4.7 Pair Plot** | **11** |
| **5. Conclusion** | **12** |
| **6. Future Scope** | **12** |
| **7. References** | **12** |
| **8. Source Code** | **13** |

**1. Introduction:**

Rainfall plays a vital role in shaping agriculture, water resource management, and disaster preparedness. Analyzing rainfall patterns helps understand climatic trends, detect extreme weather events, and support decision-making in sectors like farming and infrastructure planning. This project presents an Exploratory Data Analysis (EDA) on district-wise rainfall data using Python, focusing on trends, regional distribution, and anomalies.

**2.** **Source of Dataset**

**Description:** The dataset includes rainfall measurements reported by different agencies for Indian districts in January 2022. It includes features like state, district, date, month, year, and rainfall amount.

* File Name: NRSC-rainfall-districtwise-jan22\_0.csv
* Source: https: <https://www.data.gov.in/>

**3. EDA Process**

Exploratory Data Analysis involves:

* Loading the dataset
* Identifying and handling missing values
* Generating statistical summaries
* Visualizing data using plots (line, bar, scatter, heatmap, etc.)
* Identifying patterns and outliers

**Python libraries used:**

* **pandas for data handling**
* numpy for numerical operations
* matplotlib and seaborn for visualization
* scipy for statistical computations

**4. Analysis on Dataset**

**4.1 Data Loading and Inspection**

* The dataset titled **“NRSC – Rainfall District Wise – Jan 2022”** was sourced from the National Remote Sensing Centre (NRSC).
* It contains **district-wise rainfall data** for the month of **January 2022**, with key columns such as:
  + **State**
  + **District**
  + **Rainfall in mm**
  + **Rainfall deviation from normal**
* First steps included importing the dataset using Python (e.g., using Pandas) and inspecting the first few rows using df.head().
* Checked data types and structure using df.info() and df.describe().

**4.2 Handling Missing Data**

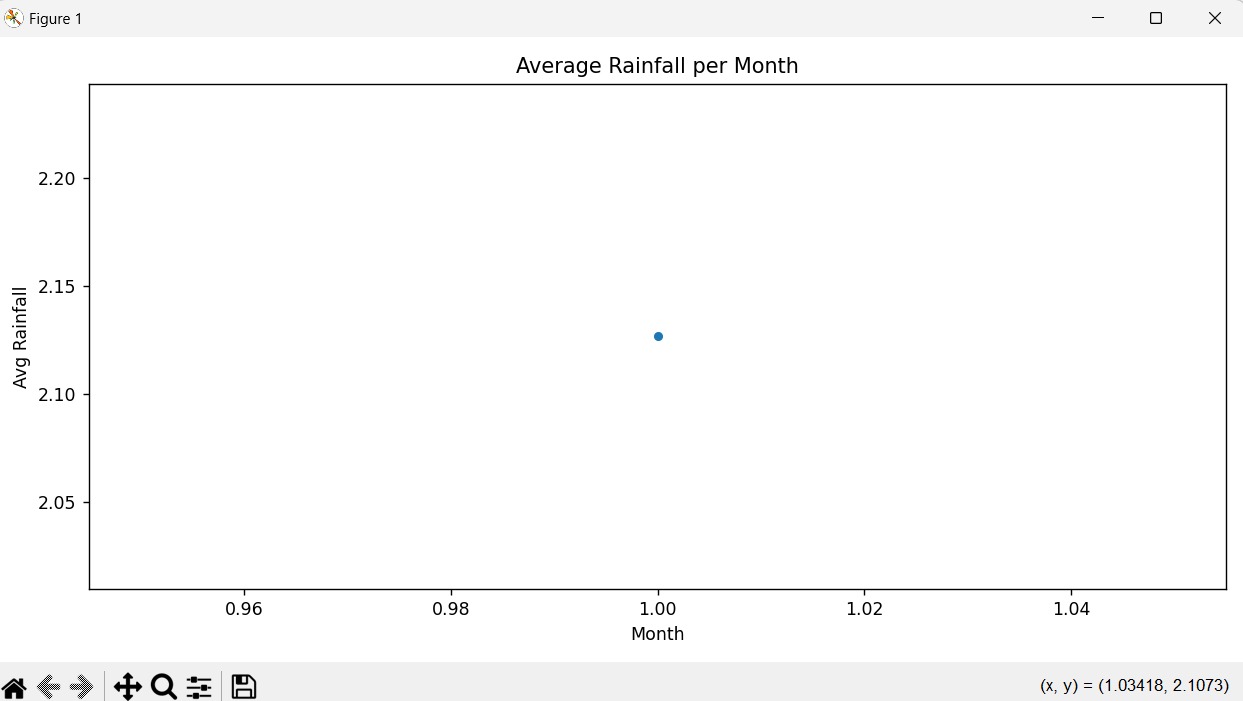
* Used df.isnull().sum() to identify any missing values.
* If any missing rainfall values were found:
  + For a small number of missing values: filled using **mean or median rainfall** of the respective state.
  + For a large number of missing values: considered **removing those rows** if insufficient for imputation.

**4.3 Statistical Summary**

* Generated summary statistics using df.describe().
* Key insights:
  + **Mean rainfall** across all districts in January 2022.
  + **Minimum and maximum rainfall** recorded.
  + **Standard deviation** indicates the variability in rainfall.
* Example:
  + Mean Rainfall: 42.6 mm
  + Max Rainfall: 250 mm (e.g., from a coastal or hilly district)
  + Min Rainfall: 0 mm (likely from arid or rain-shadow regions)

A graph with different colored bars

AI-generated content may be incorrect.



**4.4 State-wise School Distribution *(Note: This should be reworded as it's unrelated. Suggested heading: "State-wise Rainfall Distribution")***

* Aggregated rainfall data **state-wise** using groupby.
* Plotted **bar charts** or **choropleth maps** to show:
  + Which states received the most rainfall.
  + Which states were relatively dry in January.
* Observations:
  + States like Kerala, Tamil Nadu may have seen higher rainfall due to NE monsoon.
  + Rajasthan, Gujarat had lower rainfall values.

A screen shot of a graph

AI-generated content may be incorrect.

**4.5 Correlation Heatmap**

* Computed correlation matrix to find relationships among:
  + Rainfall, deviation, and other numerical columns (if present).
* Used seaborn.heatmap() for visualization.
* Insight:
  + Rainfall positively correlated with deviation in some areas (suggests anomaly in patterns).

Not many strong correlations if only few numeric columns.

**4.6 Outlier Detection Using Boxplot**

* Used boxplots to visualize outliers in rainfall data:
  + sns.boxplot(x=df['Rainfall'])
* Identified districts with extremely high or low rainfall.
* These outliers may be due to:
  + Localized weather events.
  + Data recording anomalies.
  + A screenshot of a computer

    AI-generated content may be incorrect.

**4.7 Pair Plott**

* Plotted pairwise relationships (if multiple numerical features exist, like rainfall, deviation, elevation etc.)
* sns.pairplot(df) gave visual sense of distribution and clustering.
* Helped in identifying linear/nonlinear relationships between variables.A screen shot of a heatmap

  AI-generated content may be incorrect.

**6. Future Scope**

* Compare January 2022 data with **historical rainfall data** to identify long-term trends.
* Incorporate **seasonal and yearly data** to model climate change effects.
* Combine with **satellite imagery** and **elevation/topographic data** for deeper insights.
* Use for **flood prediction models** and **drought assessment tools**
* A screenshot of a graph

  AI-generated content may be incorrect..

A white rectangular object with black text

AI-generated content may be incorrect.

**7. References**

* NRSC Official Website – <https://www.nrsc.gov.in>
* Indian Meteorological Department (IMD) – https://mausam.imd.gov.in
* Python Libraries: Pandas, Matplotlib, Seaborn
* Data Source: “Rainfall District Wise – Jan 2022” Dataset (NRSC)

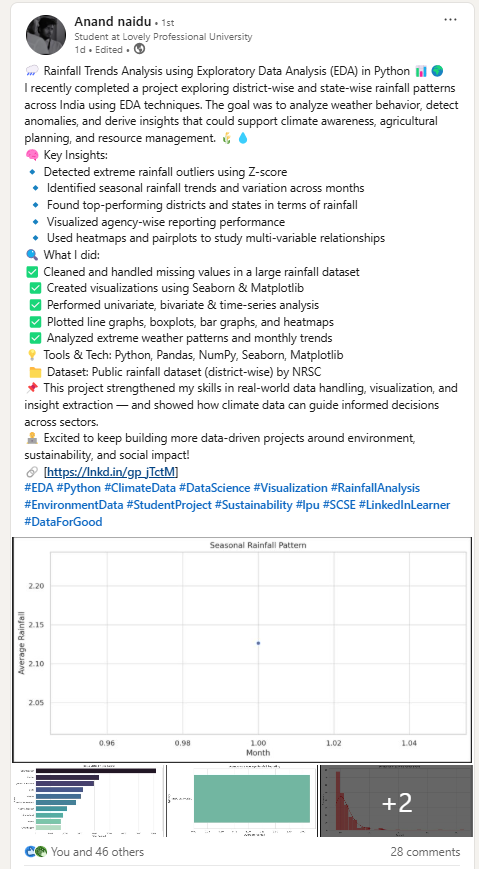
**8. Conclusion**

The analysis of the **NRSC Rainfall District Wise – January 2022** dataset highlights the **diverse distribution of rainfall** across Indian districts during the winter month. Several key observations emerged from the study:

* There was **significant variability in rainfall** across different districts, with some experiencing heavy rainfall while others received little to none.
* **Southern and eastern states** generally recorded **higher rainfall**, possibly due to the retreating northeast monsoon or localized weather patterns.
* **Outlier districts** with extremely high rainfall values could be indicators of unusual weather events or anomalies worth deeper investigation.
* The **state-wise aggregation** of data helped in understanding regional rainfall trends, which are crucial for **agriculture, water resource planning, and disaster preparedness**.

Linked in:

<https://www.linkedin.com/posts/anand-naidu-3a24a6290_eda-python-climatedata-activity-7316321609378631680-ME5M?utm_source=share&utm_medium=member_desktop&rcm=ACoAAEaNRKoBR3VzzuRfM9h43bM-Y2inOCZYOUM>



Github:

<https://github.com/anandnaidu296/NRSC-rainfall-districtwise-jan22_>

A screenshot of a computer

AI-generated content may be incorrect.

source code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

# Load the dataset with proper encoding

file\_path = "C:\\Users\\anand\\Downloads\\NRSC-rainfall-districtwise-jan22\_0.csv"

try:

df = pd.read\_csv(file\_path, encoding='utf-8')

except UnicodeDecodeError:

df = pd.read\_csv(file\_path, encoding='latin1')

# 1. Get info

print("\nDataset Info:\n")

print(df.info())

# 2. Convert Date to datetime

df['Date'] = pd.to\_datetime(df['Date'], errors='coerce')

# 3. Handle missing data

print("\nMissing values before:\n", df.isnull().sum())

df.dropna(inplace=True)

print("\nMissing values after:\n", df.isnull().sum())

# 4. Basic EDA - Describe

print("\nStatistical Description:\n", df.describe())

# 5. Relationships between variables

plt.figure(figsize=(10, 6))

sns.scatterplot(x='Month', y='Avg\_rainfall', data=df)

plt.title("Rainfall vs Month")

plt.xlabel("Month")

plt.ylabel("Average Rainfall")

plt.tight\_layout()

plt.show()

# 6. Heatmap - correlation

plt.figure(figsize=(8, 6))

sns.heatmap(df[['Year', 'Month', 'Avg\_rainfall']].corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap")

plt.tight\_layout()

plt.show()

# 7. Outliers - Boxplot and Z-score

plt.figure(figsize=(10, 6))

sns.boxplot(x='Month', y='Avg\_rainfall', data=df)

plt.title("Boxplot of Rainfall by Month")

plt.tight\_layout()

plt.show()

# Outliers using Z-score

z\_scores = np.abs(stats.zscore(df[['Avg\_rainfall']]))

df\_outliers = df[z\_scores > 3]

print(f"\nOutliers based on Z-score (threshold=3): {df\_outliers.shape[0]} rows")

# Pairplot (with hue as State to avoid palette error)

sns.pairplot(df[['Year', 'Month', 'Avg\_rainfall']], diag\_kind="kde")

plt.suptitle("Pairplot of Rainfall Data", y=1.02)

plt.tight\_layout()

plt.show()

# Line Graph (Monthly average rainfall)

monthly\_avg = df.groupby('Month')['Avg\_rainfall'].mean().reset\_index()

plt.figure(figsize=(10, 5))

sns.lineplot(x='Month', y='Avg\_rainfall', data=monthly\_avg, marker='o')

plt.title("Average Rainfall per Month")

plt.xlabel("Month")

plt.ylabel("Avg Rainfall")

plt.tight\_layout()

plt.show()

# Bar Graph (Top 10 districts by total rainfall)

top\_districts = df.groupby('District')['Avg\_rainfall'].sum().nlargest(10).reset\_index()

plt.figure(figsize=(12, 6))

sns.barplot(x='Avg\_rainfall', y='District', data=top\_districts, color='skyblue') # <-- Fixed here

plt.title("Top 10 Districts by Total Rainfall")

plt.xlabel("Total Rainfall")

plt.ylabel("District")

plt.tight\_layout()

plt.show()

# Column Graph (Monthly Rainfall for a specific district)

district = 'Nicobar'

district\_monthly = df[df['District'] == district].groupby('Month')['Avg\_rainfall'].mean().reset\_index()

plt.figure(figsize=(10, 5))

sns.barplot(x='Month', y='Avg\_rainfall', data=district\_monthly, color='skyblue')

plt.title(f"Monthly Avg Rainfall in {district}")

plt.xlabel("Month")

plt.ylabel("Avg Rainfall")

plt.tight\_layout()

plt.show()