**AIM:**

To Read Dataset using Pandas , Show the Descriptive statistics Apply preprocessing methods, Cleaning Feature engineering and Outlier analysis, Find Std Dev Mean and Error .Report using Line plot, Scatter plot, Histogram, Boxplot etc..

**DESCRIPTION:**

**Descriptive Statistics:**

* + Provides summary statistics like mean, standard deviation
  + Useful for understanding the central tendency and variability of the data.

**Preprocessing and Cleaning:**

* + Removed duplicates and handled missing values.
  + Essential to ensure data quality before analysis.

**Feature Engineering:**

* + Created a new feature, 'New\_Feature,' as a combination of existing features.
  + Removed unwanted features and produce clean dataset

**Outlier Analysis:**

* + Helps identify extreme values that might impact analysis

**Visualization:**

* + Used line plot to show the trend of 'Feature\_A.'
  + Employed a scatter plot for visualizing the relationship between two features
  + Created a histogram to illustrate the distribution of 'Feature\_A.'

**CODE**:

**READ THE DATASET**

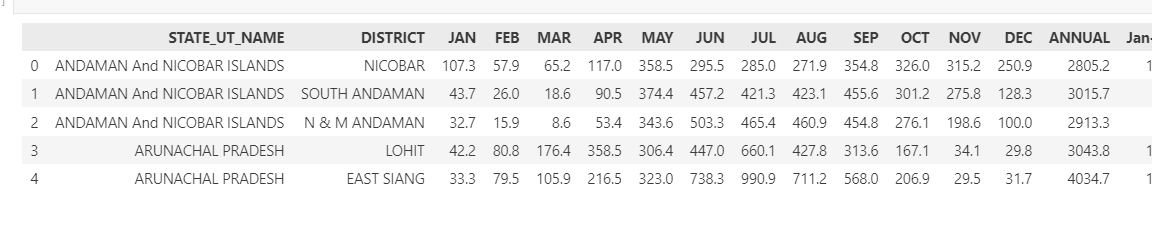
#READ THE DATASET USING PANDAS LIBRARY

import pandas as pd

df=pd.read\_csv("./district wise rainfall normal.csv")

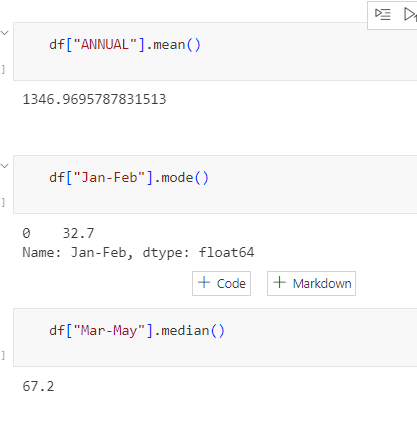
df

df.head()

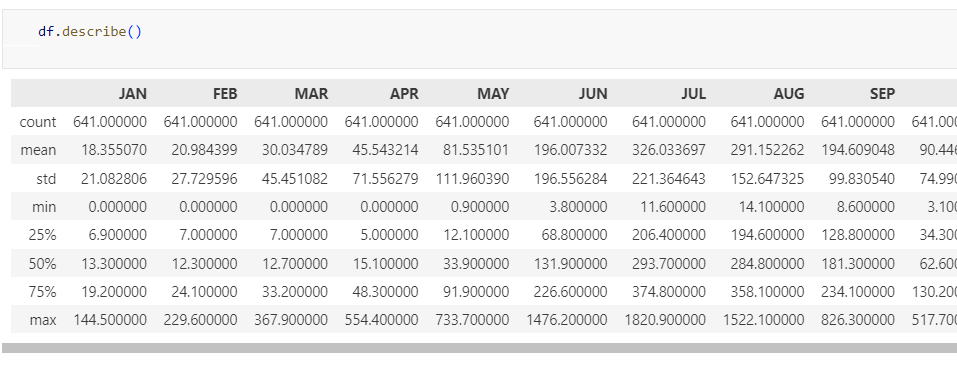


**DESCRIPTIVE STATISTICS**

**MEASURES OF CENTRAL TENDENCY MEAN, MEDIAN , MODE**

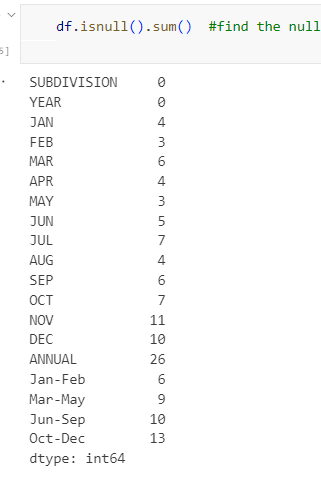


**MEASURES OF VARIABILITY(STANDARD DEVIATION, VARIANCE,PERCENTILES)**



DATA PRE-PR0CESSING AND FEATURE ENGINEERING

**DATA PRE-PR0CESSING AND FEATURE ENGINEERING**



from sklearn.impute import SimpleImputer

# Specify the column to exclude (string type)

column\_to\_exclude = 'SUBDIVISION'

# Create a list of numerical columns

numerical\_columns = [col for col in df.columns if col != column\_to\_exclude and df[col].dtype != object]

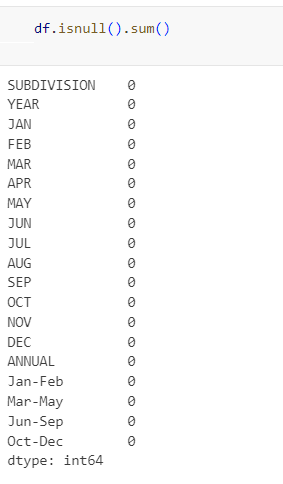
# Create the imputer, excluding the string column

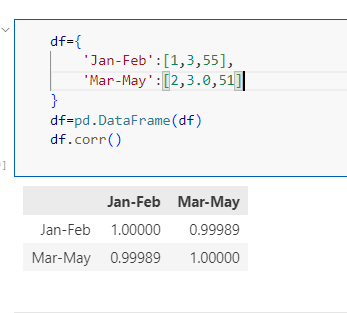
imputer = SimpleImputer(strategy='mean', copy=False)  # avoid unnecessary data copy

imputer.fit(df[numerical\_columns])

# Transform the data, excluding the string column

df[numerical\_columns] = imputer.transform(df[numerical\_columns])





**OUTLIER ANALYSIS AND VISUALIZATION USING BOX PLOT AND SCATTER PLOT AND DATA VISUALIZATION**

import seaborn as sns

import matplotlib.pyplot as plt

sns.boxplot(df["Jan-Feb"])

plt.xlabel("Rainfall for January to February")

plt.show()

# Create a scatter plot using Matplotlib

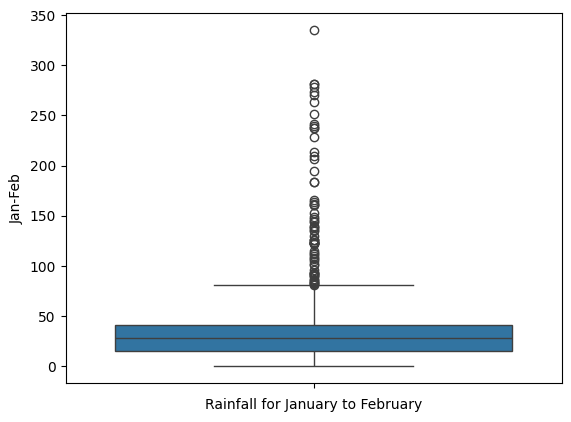
fig, ax = plt.subplots(figsize=(6, 4))

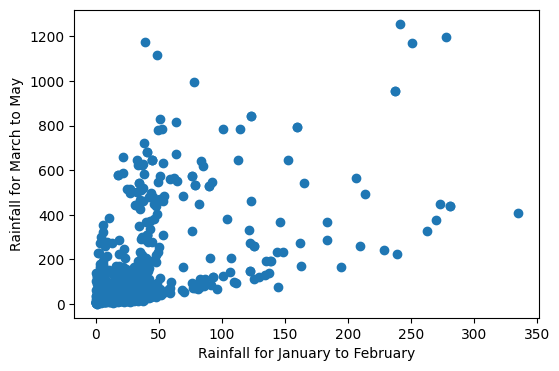
ax.scatter(df['Jan-Feb'], df['Mar-May'])

ax.set\_xlabel("Rainfall for January to February")

ax.set\_ylabel("Rainfall for March to May")

plt.show()





**REMOVE OUTLIERS**

import seaborn as sns

import matplotlib.pyplot as plt

# Function to remove outliers using IQR method

def remove\_outliers(df, column):

    Q1 = df[column].quantile(0.25)

    Q3 = df[column].quantile(0.75)

    IQR = Q3 - Q1

    lower\_bound = Q1 - 1.5 \* IQR

    upper\_bound = Q3 + 1.5 \* IQR

    df\_filtered = df[(df[column] >= lower\_bound) & (df[column] <= upper\_bound)]

    return df\_filtered

# Remove outliers from "Jan-Feb" column

df\_filtered = remove\_outliers(df, "Jan-Feb")

# Create a boxplot using Seaborn after removing outliers

sns.boxplot(df\_filtered["Jan-Feb"])

plt.xlabel("Rainfall for January to February")

plt.show()

# Create a scatter plot using Matplotlib after removing outliers

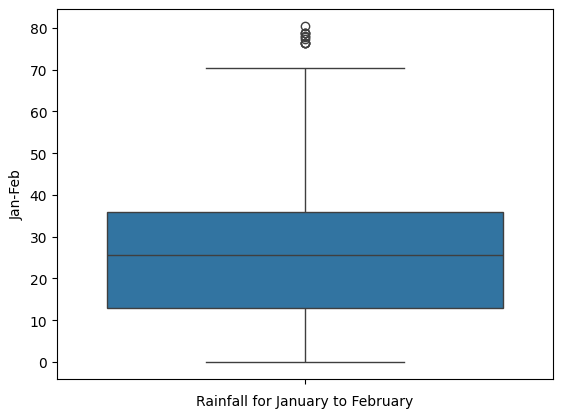
fig, ax = plt.subplots(figsize=(6, 4))

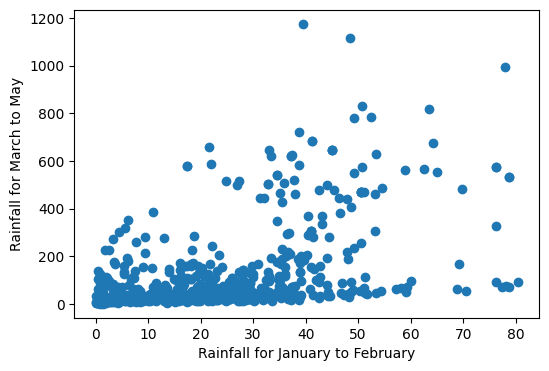
ax.scatter(df\_filtered['Jan-Feb'], df\_filtered['Mar-May'])

ax.set\_xlabel("Rainfall for January to February")

ax.set\_ylabel("Rainfall for March to May")

plt.show()





**VISUALIZATION**

annual\_data = df.groupby('STATE\_UT\_NAME')['ANNUAL'].mean()

# Plotting

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

annual\_data.plot(kind='bar')

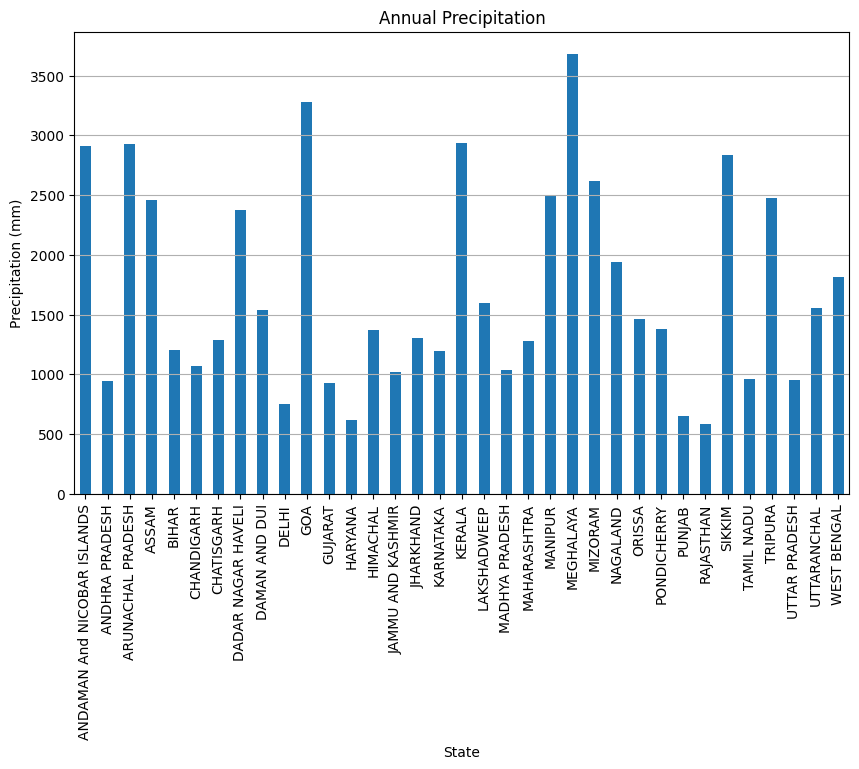
plt.title('Annual Precipitation')

plt.xlabel('State')

plt.ylabel('Precipitation (mm)')

plt.grid(axis='y')

plt.show()

****

**It is inferred that Meghalaya has received the highest annual rainfall.**

data=np.array(df['ANNUAL'])

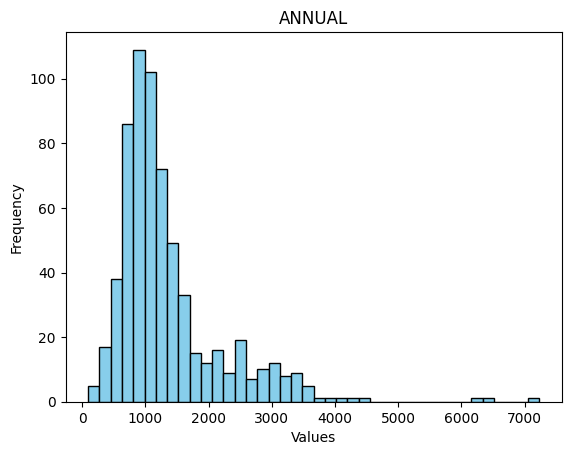
plt.hist(data,bins=40,color='skyblue',edgecolor='black')

plt.xlabel("Values")

plt.ylabel("Frequency")

plt.title("ANNUAL RAIN FALL DISTRIBUTION")

plt.show()

****

**It shows the right skewness in the frequency distribution of annual rainfall**

rainfall=pd.read\_csv("../RAIN DATASET/rainfall in india 1901-2015.csv")

# Assuming your DataFrame is named df

annual\_data = rainfall.groupby('YEAR')['ANNUAL'].mean()

# Plotting

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

plt.plot(annual\_data.index, annual\_data.values, marker='o', linestyle='-')

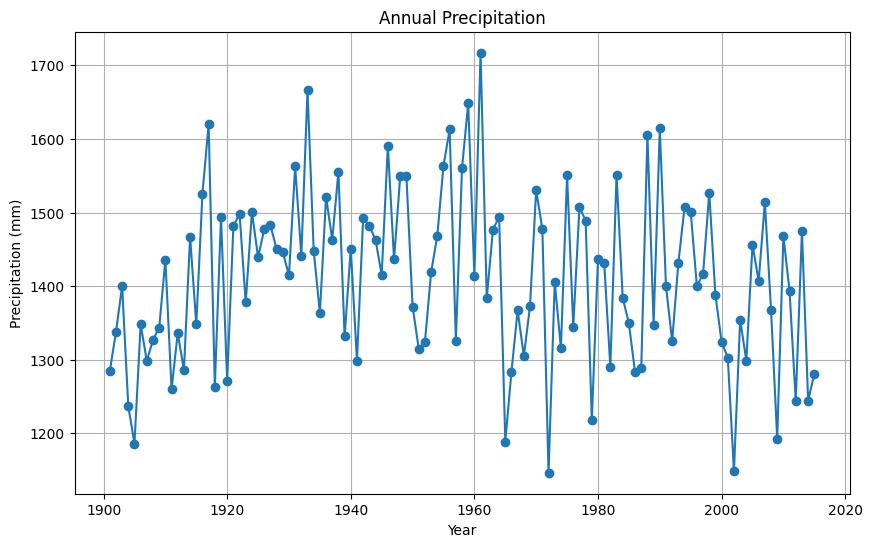
plt.title('Annual Precipitation')

plt.xlabel('Year')

plt.ylabel('Precipitation (mm)')

plt.grid(True)

plt.show()

****

**It is inferred that years between 1920 to 1965 have highest average rainfalls.**