A white robot hand with red and blue stripes

Description automatically generated

**Module-5**

**AI & ML**



**LAB MANUAL**

**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

**Statistical Analysis of Wheat and Rice Production Across Indian States Using Mean, Median, and Mode with NumPy and Pandas.**

**Objective:**

* Collect and analyze wheat and rice production data from different Indian states.
* Compute **Mean, Median, and Mode** to understand production distribution.
* Use **NumPy and Pandas** for statistical calculations.
* Visualize the data using **Matplotlib** to identify trends and variations.

**Duration** 2 Hrs

**Problem Statement:**

Agricultural production varies across different states in India, affecting the economy and food security. To analyze this variation, statistical measures such as **Mean, Median, and Mode** can be used to summarize and interpret wheat and rice production data across different Indian states. This analysis helps in understanding production trends, identifying outliers, and making informed policy decisions.

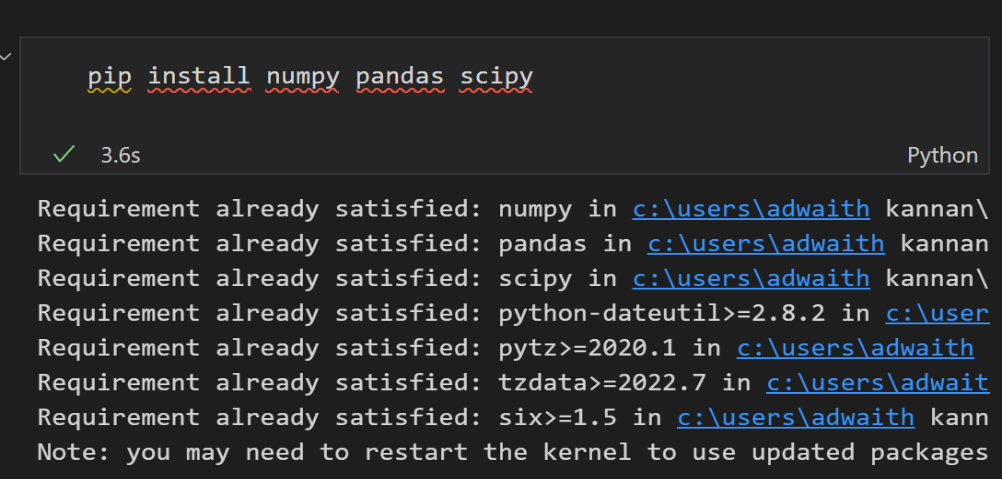
**Requirement:**

* **Python 3.x**
* **NumPy** (for numerical computations)
* **Pandas** (for handling tabular data)
* **SciPy** (for statistical calculations like mode)
* **Matplotlib** (for visualization)

**Dataset:** <https://www.kaggle.com/datasets/pyatakov/india-agriculture-crop-production>

**Procedure:**

**Step 1: Install Required Libraries**

****

**Step 2: Import Necessary Libraries**

* numpy is used for calculating **mean and median**.
* pandas is used to **store and process tabular data**.
* scipy.stats is used to **compute the mode**.
* matplotlib.pyplot is used for **visualizing the data**.

A screenshot of a computer

AI-generated content may be incorrect.

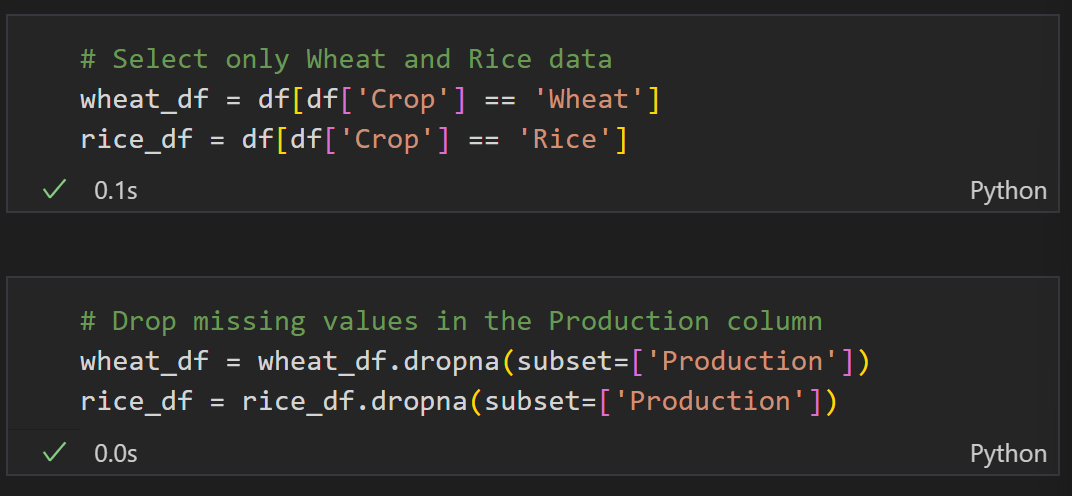
**Step 3: Load the Dataset**

**A screenshot of a computer

AI-generated content may be incorrect.**

* Print the dataset.

**Step 4: Select only wheat and rice data from the dataset and remove the missing values.**



**Step 5: Compute Mean, Median, and Mode**

* **np.mean()** → Computes the **average** production.
* **np.median()** → Finds the **middle value** when the data is sorted.
* **stats.mode()** → Identifies the **most frequently occurring value** in the dataset.

**Calculate Statistics for Wheat Production**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**Explanation:**

* df['Wheat\_Production'] extracts the **wheat production** values from the DataFrame (df).
* np.mean() is a **NumPy function** that calculates the **mean (average)** of the given data.

**Formula for Mean:**

**A mathematical equation with black letters

AI-generated content may be incorrect.**

Where:

* X = individual wheat production values
* N = total number of states

**Example Calculation:**

**A math problem with numbers

AI-generated content may be incorrect.**

So, **mean\_wheat = 10.48 million tons**.

**median\_wheat = np.median(df['Wheat\_Production'])**

**Explanation:**

* np.median() is a **NumPy function** that calculates the **median** of the data.
* The **median** is the **middle value** in a sorted list.
  + If **N is odd**, the median is the middle value.
  + If **N is even**, the median is the average of the two middle values.

**Step-by-Step Calculation:**

Sorted wheat production data: [1.1, 1.2, 3.5, 6.3, 8.7, 12.5, 16.2, 34.3]

A math equation with black text

AI-generated content may be incorrect.

So, **median\_wheat = 7.50 million tons**.

**mode\_wheat = stats.mode(df['Wheat\_Production'], keepdims=True)[0][0]**

**Explanation:**

* stats.mode() is a function from the **SciPy** library that finds the **mode** of the data.
* The **mode** is the value that appears **most frequently** in the dataset.
* keepdims=True ensures that the function returns a NumPy array (for compatibility).
* [0][0] is used to extract the actual mode value from the returned array.

**Step-by-Step Calculation:**

Given the wheat production data:  
[16.2, 34.3, 12.5, 1.2, 8.7, 6.3, 3.5, 1.1]

* Here, **all values appear only once**, so technically, there is **no mode**.
* In such cases, stats.mode() **returns the smallest value** as the default mode.

Thus, **mode\_wheat = 1.10 million tons**.

**Calculate Statistics for Rice Production**

A screen shot of a computer screen

AI-generated content may be incorrect.

**mean\_rice = np.mean(df['Rice\_Production'])**

**Explanation:**

* df['Rice\_Production'] extracts the **rice production values** from the DataFrame (df).
* np.mean() is a **NumPy function** that calculates the **mean (average)** of the given data.

**Formula for Mean:**

Where:

* X= individual rice production values
* N = total number of states

**Example Calculation:**

**A mathematical equation with black text

AI-generated content may be incorrect.**

A math problem with numbers

AI-generated content may be incorrect.

So, **mean\_rice = 9.52 million tons**.

**median\_rice = np.median(df['Rice\_Production'])**

**Explanation:**

* np.median() is a **NumPy function** that calculates the **median** of the data.
* The **median** is the **middle value** in a sorted list.
  + If **N is odd**, the median is the middle value.
  + If **N is even**, the median is the **average of the two middle values**.

**Example Calculation:**

**A math problem with numbers and equations

AI-generated content may be incorrect.**

So, **median\_rice = 7.60 million tons**.

**mode\_rice = stats.mode(df['Rice\_Production'], keepdims=True)[0][0]**

**Explanation:**

* stats.mode() is a **SciPy function** that calculates the **mode** of the dataset.
* The **mode** is the most **frequently occurring value**.
* keepdims=True ensures that the function returns an **array format**.
* [0][0] extracts the **actual mode value** from the returned array.

**Step-by-Step Calculation:**

Given the rice production data:  
[11.0, 15.5, 4.8, 16.0, 7.4, 7.8, 7.2, 6.5]

* Since **all values occur only once**, there is **no true mode**.
* In such cases, stats.mode() returns the **smallest value** as the default mode.

Thus, **mode\_rice = 7.20 million tons**.

**Step 6: Display the Results**

* We **format the numbers to 2 decimal places (.2f)**.
* **Print the Mean, Median, and Mode for both Wheat and Rice Production**.

A screen shot of a computer

AI-generated content may be incorrect.

**Conclusion**

The analysis of wheat and rice production across Indian states using Mean, Median, and Mode provides key insights into the distribution of crop production. The mean gives an overall average production level, while the median helps understand the central tendency, reducing the impact of outliers. The mode highlights the most frequently occurring production values. This statistical analysis helps policymakers and agricultural planners assess crop trends and make data-driven decisions for improving agricultural output.