🔷 Part 1: Loan Dataset Analysis (`loan_data.csv`)

🔹 Step 1: Import and Load the Dataset

python

CopyEdit

```
import pandas as pd

data = pd.read_csv("loan_data.csv")
```

- Loads the CSV file into a pandas DataFrame for processing.

🔹 Step 2: Basic Exploration

python

CopyEdit

```
data.head()        # Shows first 5 rows

data.tail()        # Shows last 5 rows

data.info()        # Shows column names, datatypes, non-null counts

data.describe()    # Shows mean, std, min, 25%, 50%, 75%, max

data.isnull().sum()# Displays total missing values per column
```

- Helps understand the structure, data types, and whether any values are missing.

🔹 Step 3: Summary Statistics for Entire Dataset

python

CopyEdit

```
mean = data.mean(numeric_only=True)

median = data.median(numeric_only=True)

minimum = data.min(numeric_only=True)

maximum = data.max(numeric_only=True)

std = data.std(numeric_only=True)
```

- **Calculates core statistical measures (mean, median, min, max, std) for all numeric columns.**

🔹 Step 4: Individual Column Statistics

python

CopyEdit

```
data['LoanAmount'].mean()

data['Loan_Amount_Term'].mean()

data['Age'].median()

data['Age'].std()
```

- **Retrieves statistics specifically for important columns.**

🔹 Step 5: Grouped Summary by Categorical Column (`Loan_Status`)

python

CopyEdit

```
grouped_data = data.groupby('Loan_Status').agg({
    'Age': ['mean', 'median', 'min', 'max', 'std'],
    'ApplicantIncome': ['mean', 'median', 'min', 'max', 'std'],
    'CoapplicantIncome': ['mean', 'median', 'min', 'max', 'std'],
    'LoanAmount': ['mean', 'median', 'min', 'max', 'std']
})
```

- **Groups the data by `Loan_Status` and computes statistical measures for each numeric column.**
- **Useful to analyze patterns like whether loan approval is affected by age or income.**

🔷 Part 2: Iris Dataset Analysis (`iris.csv`)

🔹 Step 1: Load Dataset

python

CopyEdit

```
import numpy as np
data = pd.read_csv("iris.csv")
```

- **Loads the Iris flower dataset into memory for analysis.**

🔹 Step 2: Grouped Statistics

python

CopyEdit

```
data.groupby('Species').count()

data.groupby('Species').mean()
```

- Calculates the count and average for each numeric column grouped by `Species` (Setosa, Versicolor, Virginica).

🔹 Step 3: Specific Stats

python

CopyEdit

```
data.Species.mode()          # Most common species

data.SepalWidthCm.std()      # Standard deviation for sepal width

data.SepalLengthCm.std()     # Standard deviation for sepal length
```

- Displays mode (most frequent), and variation in specific features.

🔹 Step 4: Violin Plot

python

CopyEdit

```
import seaborn as sns

sns.violinplot(x="SepalWidthCm", y="Species", data=data)
```

- Creates a violin plot to show the distribution and density of `SepalWidthCm` for each flower species.

🔹 Step 5: Correlation Matrix

python

CopyEdit

```
numeric_df = data.select_dtypes(include=[np.number])

correlation_matrix = numeric_df.corr(method='pearson')
```

- Selects only numeric columns.
- Computes Pearson correlation between features, helping to understand relationships (e.g., between Sepal length and Petal length).

🔷 Part 2: Iris Dataset Analysis (`iris.csv`)

📌 Objective:

To perform grouped statistical analysis on the Iris flower dataset and visualize feature distribution using violin plots and correlation matrix.

🔹 Step-by-Step Explanation

1. Load the Dataset

python

CopyEdit

```
import numpy as np

import pandas as pd

data = pd.read_csv("iris.csv")
```

- Reads the Iris dataset CSV file using pandas.
- The dataset typically includes:
    - SepalLengthCm
    - SepalWidthCm
    - PetalLengthCm
    - PetalWidthCm
    - Species (categorical: Setosa, Versicolor, Virginica)

---

2. View Column Names

python

CopyEdit

```
print(data.columns)
```

- Displays all column names to verify correct loading of data.

---

3. Grouped Statistics by Species

python

CopyEdit

```
data.groupby('Species').count()
data.groupby('Species').mean()
```

- `.groupby('Species')` groups the dataset by flower type.
- `.count()` shows how many samples per species.
- `.mean()` calculates average sepal and petal sizes per species.

---

4. Summary Statistics

python

CopyEdit

```
data.Species.mode()           # Most frequent species
data.SepalWidthCm.std()       # Standard deviation for Sepal Width
data.SepalLengthCm.std()      # Standard deviation for Sepal Length
```

- These lines compute:
  - Mode: the most common species
  - Standard deviation: how much Sepal sizes vary

---

5. Violin Plot

python

CopyEdit

```
import seaborn as sns
```

```
sns.violinplot(x="SepalWidthCm", y="Species", data=data)
```

- This violin plot visualizes the distribution of Sepal Width for each species.
- Combines boxplot and KDE (Kernel Density Estimation) to show data spread and frequency.

---

6. Correlation Matrix

python

CopyEdit

```
numeric_df = data.select_dtypes(include=[np.number])  # Select only numeric columns

correlation_matrix = numeric_df.corr(method='pearson')
```

- Selects only numeric features (excluding species).
- `.corr()` generates a Pearson correlation matrix, showing linear relationships between features.
  - A value near +1 = strong positive correlation
  - Near 0 = no correlation
  - Near -1 = strong negative correlation