Data Analysis on a Sample Dataset (CSV)

Requirements:

1. Python
2. Libraries: `pandas`, `matplotlib` (Install them with `pip install pandas matplotlib`)

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset (replace the path with the location of your CSV file)
data = pd.read_csv('path_to_your_data.csv')

# Display the first few rows of the data to understand its structure
print("First 5 rows of the dataset:")
print(data.head())

# Step 1: Data Cleaning - Handling Missing Values
# Check for missing values
missing_data = data.isnull().sum()
print("\nMissing data in each column:")
print(missing_data)

# For simplicity, fill missing values with the column's mean
data.fillna(data.mean(), inplace=True)

# Step 2: Basic Data Analysis
# Descriptive statistics (mean, median, std deviation, etc.)
print("\nDescriptive Statistics of the data:")
print(data.describe())

# Correlation analysis between numerical columns
print("\nCorrelation Matrix:")
print(data.corr())

# Step 3: Data Visualization
# Create a histogram for a specific column (e.g., 'Age' column if present)
plt.figure(figsize=(8, 6))
plt.hist(data['Age'], bins=20, edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

# Create a scatter plot to show the relationship between two variables (e.g.,
'Age' and 'Salary')
plt.figure(figsize=(8, 6))
plt.scatter(data['Age'], data['Salary'], color='blue')
plt.title('Age vs. Salary')
plt.xlabel('Age')
plt.ylabel('Salary')
plt.show()
```

Explanation:

1. **Loading Data**:
   - The dataset is loaded using `pd.read_csv()`. Make sure to replace `'path_to_your_data.csv'` with the actual path to your CSV file.
2. **Data Cleaning**:
   - The code checks for missing data using `data.isnull ().sum ()`. Then, it fills missing values in numerical columns with the mean of that column (`data.fillna (data. Mean (), in place=True)`).
3. **Basic Analysis**:
   - `data.describe()` generates descriptive statistics such as mean, median, and standard deviation for each column.
   - `data.corr()` shows the correlation matrix, helping you understand how the columns relate to each other.
4. **Visualization**:
   - **Histogram**: A histogram of the 'Age' column is created using `plt.hist()`.
   - **Scatter Plot**: A scatter plot between 'Age' and 'Salary' is created using `plt.scatter()` to explore any relationship between these two variables.

Steps to Run the Code:

1. **Install the Required Libraries**: