**🧾 Research Paper: Data Cleansing and Aggregation using Python (Pandas)**

**Author: Manmeet Kaur**

**Date: October 2025**

**Abstract**

Raw data is rarely clean or ready for analysis. This research paper presents a practical mini project on **data cleansing and aggregation using Python’s Pandas library**. The project demonstrates how missing values, string inconsistencies, and unstructured data can be identified and corrected efficiently. It also illustrates how datasets can be merged and summarized using aggregation functions such as groupby(). A simple dataset of employees and departments was used, followed by visualization of average salary by department using Matplotlib. The results highlight the crucial role of preprocessing in improving data quality and making analysis more reliable.

**1. Introduction**

Data has become one of the most valuable assets in the digital era. However, the data collected from multiple sources often contains **missing values**, **inconsistencies**, and **duplicates**, which can lead to incorrect analysis.  
This project focuses on two major preprocessing steps:

1. **Data Cleansing** – identifying and fixing missing or invalid data.
2. **Data Aggregation** – combining and summarizing data for analysis.

Using the **Pandas** and **Matplotlib** libraries in Python, this project demonstrates the process of cleaning and merging small sample datasets to prepare them for meaningful visualization and interpretation.

**2. Objectives**

The main objectives of this research were to:

* Detect and handle missing data using Pandas functions.
* Apply string operations to standardize data.
* Merge multiple datasets using different join operations.
* Perform aggregation using the groupby() function.
* Visualize results to interpret the cleaned data effectively.

**3. Methodology**

The project was implemented in **Jupyter Notebook (Anaconda environment)** using Python. The process included the following steps:

**Step 1: Import Libraries**

import pandas as pd

**Step 2: Create Sample Datasets**

Two datasets were created:

* **Dataset 1:** Employee details (Emp\_ID, Name, Department, Salary)
* **Dataset 2:** Project details (Emp\_ID, Project)

These were converted into Pandas DataFrames.

**Step 3: Check for Missing Data**

Used df.isnull().sum() to identify missing entries in columns like *Name*, *Department*, and *Salary*.

**Step 4: Fill Missing Data**

* Missing names were filled with "Unknown".
* Department missing values were replaced with the most frequent value using mode().
* Missing salary values were replaced with the column’s mean value.

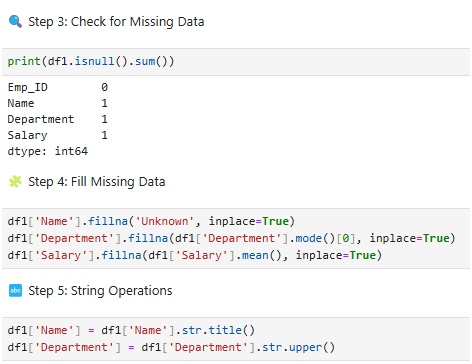
**Step 5: String Operations**

Used Pandas string functions to clean text:

*df['Name'] = df['Name'].str.title()*

*df['Department'] = df['Department'].str.upper()*

This ensured consistent capitalization across data.



*Figure 1: Data cleansing steps — checking for missing values, filling data, and standardizing strings using Pandas.*

**Step 6: Merging Data**

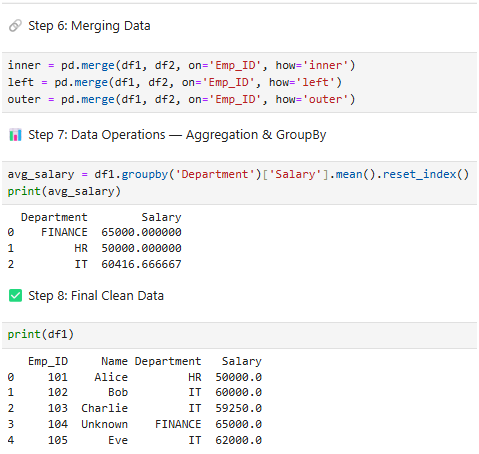
The datasets were merged using different join types (inner, left, outer) on the *Emp\_ID* column:

*inner = pd.merge(df1, df2, on='Emp\_ID', how='inner')*

**Step 7: Aggregation and GroupBy**

Average salary per department was calculated using:

*avg\_salary = df1.groupby('Department')['Salary'].mean().reset\_index()*

**

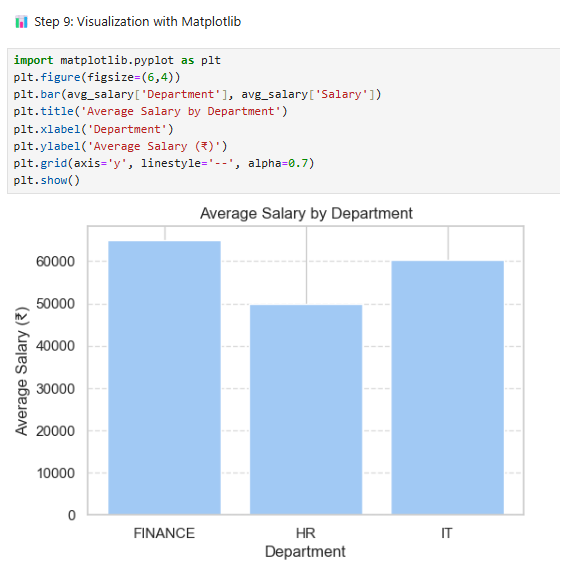
*Figure 2: Merging employee and project datasets and aggregating salary by department.*

**Step 8: Visualization with Matplotlib**

A bar chart was created to show the **average salary by department**:

*plt.bar(avg\_salary['Department'], avg\_salary['Salary'])*

This visualization clearly showed salary differences across departments (Finance, HR, IT).



*Figure 3: Visualization of average salary by department using Matplotlib.*

**4. Results and Discussion**

After cleansing and aggregation, the final dataset was completely clean and consistent.  
**Key results included:**

* Missing data successfully filled with meaningful values.
* Department names standardized for uniformity.
* Average salaries computed accurately for each department.
* Visualization displayed clear insights, showing Finance with the highest average salary.

This demonstrates that even small datasets require proper cleaning and preprocessing to avoid misleading interpretations.

**5. Conclusion**

This project illustrates how **Pandas** simplifies the process of cleaning and preparing data for analysis. The experiment showed that:

* Handling missing and inconsistent data improves dataset reliability.
* Aggregation provides meaningful summaries.
* Visualization enhances interpretability of results.

Therefore, data preprocessing is a fundamental step in any data science or machine learning workflow.

**6. Future Scope**

* Applying the same process to large real-world datasets.
* Automating data cleaning scripts.
* Extending analysis with advanced visualizations (Seaborn, Plotly).
* Performing predictive analysis on the cleaned data.

**7. References**

1. Wes McKinney, *Python for Data Analysis*, O’Reilly Media.
2. Pandas Official Documentation — https://pandas.pydata.org/
3. Matplotlib Documentation — https://matplotlib.org/