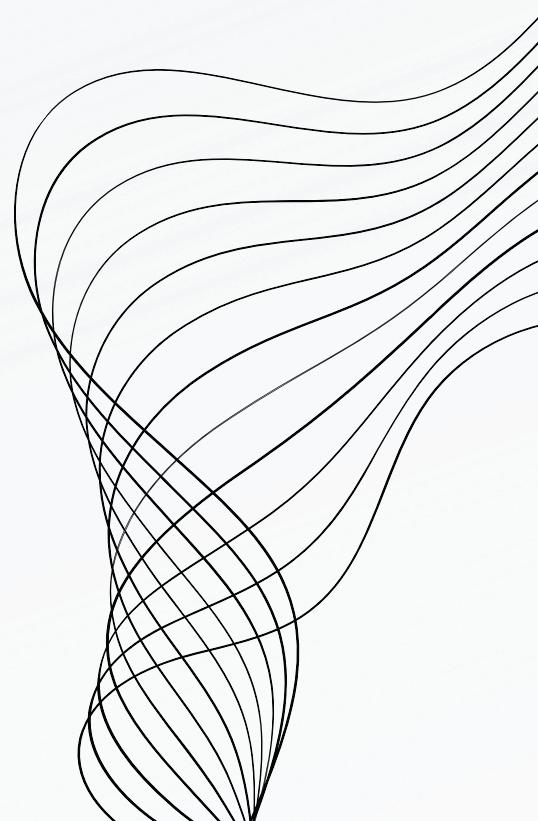


# DATA SCIENCE PROJECT

**HANDLING AND ANALYZING MULTIPLE DATASETS USING PYTHON:  
PANDAS, NUMPY, AND SEABORN**

A large, abstract graphic on the right side of the page consists of numerous thin, black, wavy lines that curve and twist across the frame, mirroring the design on the left.

# TABLE OF CONTENTS

- 01** INTRODUCTION
- 02** LIBRARIES USED
- 03** DATASET 1: IMPORT, INSIGHTS, AND LINE PLOT
- 04** DATASET 2: IMPORT, INSIGHTS, AND MERGING WITH DATASET 1
- 05** MERGED DATASET: INSIGHTS, HANDLING MISSING VALUES & OUTLIERS
- 06** DATASET 3: IMPORT, CONCATENATION, AND FINAL ANALYSIS
- 07** CONCLUSION

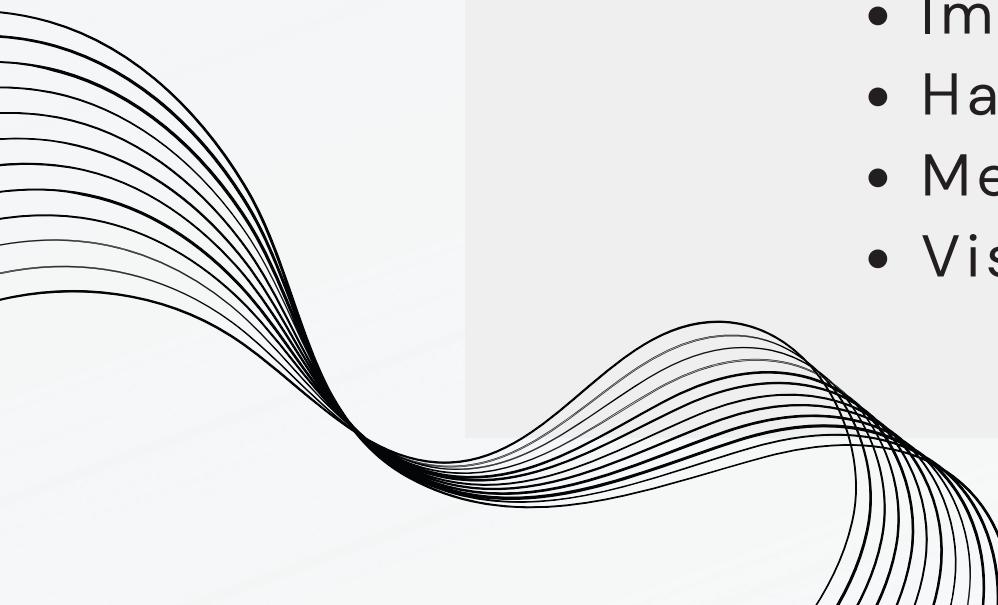
# INTRODUCTION

## Objective:

- Handle three datasets effectively using Python libraries.
- Perform cleaning, merging, and concatenation to extract insights.
- Visualize data to uncover patterns.

## Key Tasks:

- Import datasets and check insights.
- Handle missing values and outliers.
- Merge and concatenate datasets.
- Visualize data using Seaborn.



# LIBRARIES USED

## Pandas

Data manipulation and analysis.

## NumPy

Handling arrays and numerical computations.

## Seaborn

Data visualization and statistical graphics.



# DATASET 1 - IMPORT & INITIAL ANALYSIS

- Import Dataset 1:
- Load using pd.read\_csv().
- Insights:
  - Shape: Check with df1.shape.
  - Columns: List columns using df1.columns.
  - Data Types: Use df1.dtypes.
  - Null Values: Check with df1.isnull().sum().
  - Summary Statistics: Use df1.describe().
- Visualization:
  - Create a line plot using Seaborn's lineplot()

# DATASET 2 - IMPORT & ANALYSIS

- Import Dataset 2:
- Load using `pd.read_csv()`.
- Insights:
  - Shape: Use `df2.shape`.
  - Columns: Use `df2.columns`.
  - Null Values: Check with `df2.isnull().sum()`.
  - Summary Statistics: Use `df2.describe()`.

# MERGING DATASET 1 & DATASET 2

## Merge Datasets:

- Use the `merge()` method with a common column:
- csharp
- Copy code
- `merged_df = pd.merge(df1, df2, on='common_column', how='inner')`
- Insights from Merged Dataset:
- Shape: `merged_df.shape`
- Null Values: `merged_df.isnull().sum()`
- Handle Outliers:
- Use the 3-sigma rule to remove outliers:
- arduino
- Copy code
- `merged_df = merged_df[(np.abs(merged_df - merged_df.mean()) <= 3 * merged_df.std())]`

# DATASET 3 - IMPORT & CONCATENATION

1. Import Dataset 3:

- Load using pd.read\_csv().

2. Concatenate with Merged Dataset:

- Use the concat() method:
- css
- Copy code
- final\_df = pd.concat([merged\_df, df3], axis=0, ignore\_index=True)

3. Insights from Final Dataset:

- Shape: final\_df.shape
- Null Values: final\_df.isnull().sum()
- Summary Statistics: final\_df.describe()

# HANDLING MISSING VALUES AND OUTLIERS

1. Handling Missing Values and Outliers
  2. Fill NaN values using the column mean:
  3. graphql
  4. `final_df.fillna(final_df.mean(), inplace=True)`
  5. Handle Outliers:
  6. Use the 3-sigma rule again to handle outliers:
  7. Copy code
  8. `final_df = final_df[(np.abs(final_df - final_df.mean())  
<= 3 * final_df.std())]`

# DATA VISUALIZATION

- histplot
- barplot
- lineplot
- boxplot

# CONCLUSION

1. Successfully imported, analyzed, merged, and concatenated three datasets.
2. Handled missing values and outliers effectively.
3. Visualized the data to gain insights.