Data Manipulation with Pandas

Explore powerful techniques for structured data analysis in Python.

by Aravind K

Introduction to Pandas

Power Tool

Python library for data manipulation and analysis.

Structured Data

Ideal for spreadsheet-like data handling.

Efficiency

Streamlines complex data operations.

DataFrames and Series

DataFrame

2D labeled data structure, like a spreadsheet.

Series

1D labeled array, similar to a single column.

Flexibility

Mix data types within structures.

Indexing

Efficient data access and manipulation.

Creating a DataFrame

1

Import Pandas

Start by importing the pandas library.

2

Define Data

Create a dictionary with column names and values.

3

Construct DataFrame

Use pd.DataFrame() to create the structure.

4

Display Result

Print the DataFrame to view the output.

```
1       igbas bally .intto semttam);
2       igntllan(seeriak))
3       tgnddlas (ance[ ".sky"l,/Datfframe);
4       ignddld -otariae);
6       igntocaces(raspiv, fark:a);
5       fandatae(;, Portine)
6       igndalas -Prmay Dataframme. mistorace: perinndnall()
9       [try.);
.0  )
```

Data Selection Techniques

1 **loc**

Label-based indexing for intuitive data access.

2 **iloc**

Integer-based indexing for positional data selection.

3 **Boolean Indexing**

Filter data based on conditions.

4 **Multi-level Indexing**

Access hierarchical data structures efficiently.

Data Cleaning Strategies

Handle Missing Values

Use fillna() or dropna() for data completeness.

Remove Duplicates

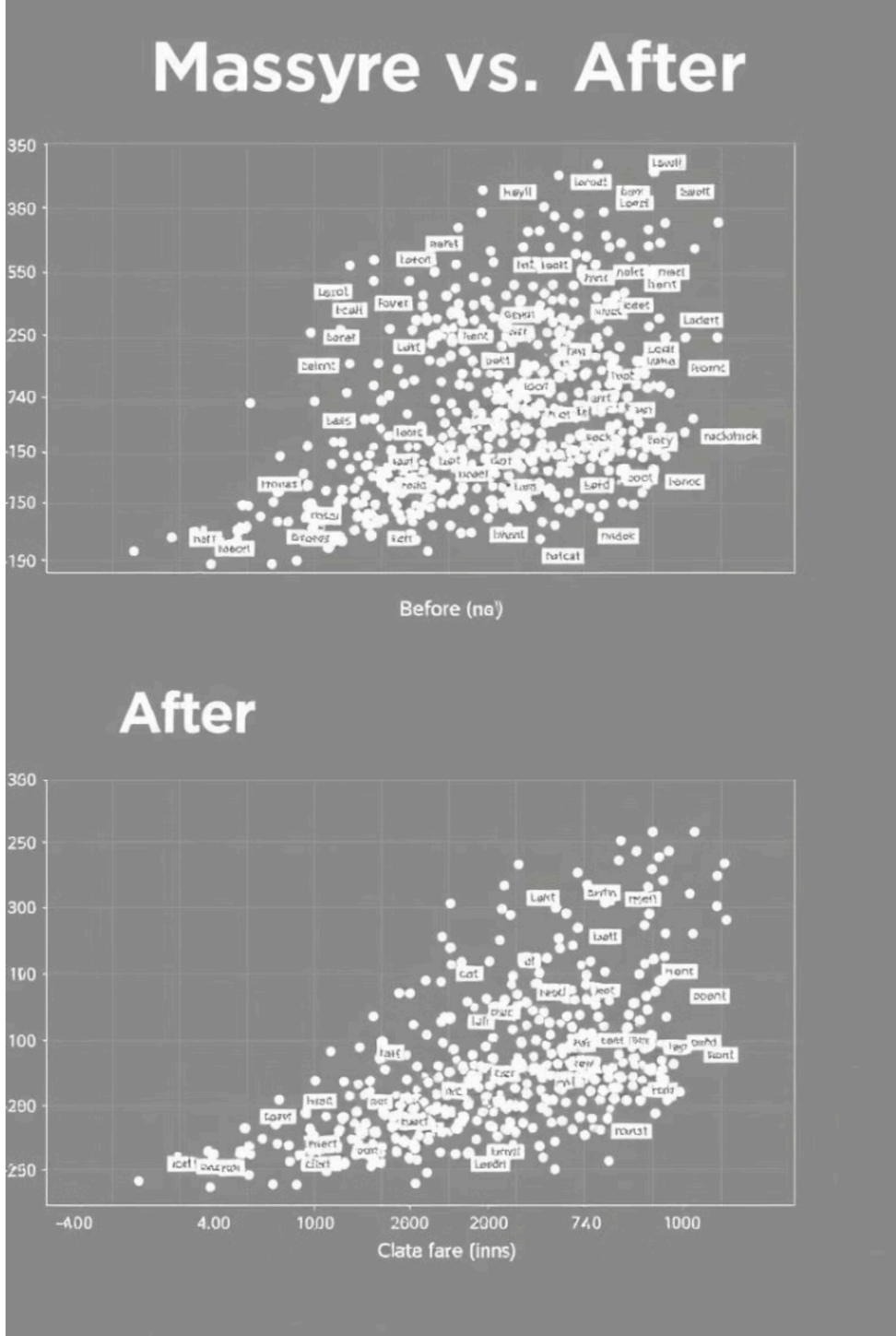
Ensure data integrity with drop_duplicates().

Data Type Conversion

Transform data types for consistency.

String Manipulation

Clean and standardize text data.



Data Aggregation Techniques

1

Group Data

Use groupby() to segment data.

2

Apply Function

Utilize aggregate functions like mean(), sum().

3

Combine Results

Merge aggregated data with original DataFrame.

4

Analyze Insights

Interpret aggregated results for decision-making.

Merging and Joining DataFrames

Method	Use Case	Key Feature
merge()	Combining on keys	SQL-style joins
concat()	Appending data	Axis-based combination
join()	Index-based joining	Flexible index alignment