



Working with Pandas and NumPy: Data Analysis Fundamentals

Welcome to Week 2! We'll explore essential Python libraries for data analysis: Pandas and NumPy. These powerful tools will transform how you handle data.

Introduction to Pandas and NumPy

Pandas

A powerful data manipulation library with dataframes as its core structure.

- Fast data analysis
- Flexible data structures
- Built-in visualization

NumPy

The foundation for scientific computing in Python.

- Multi-dimensional arrays
- Mathematical functions
- Linear algebra operations

```
≡ =hhportiets -=Trrer[, tnt;, lngucrt]]"] +  
≡ -excel  
≡ import igi="--[-Tfron ]eate],,  
    nportieats,[li]=t[] ,l[-, ]}  
≡ -  
≡ json  
≡ -  
≡ json  
≡ -
```

Loading Datasets



CSV Files

```
import pandas as pd  
df = pd.read_csv('data.csv')
```

Perfect for tabular data. Simple structure with comma-separated values.



Excel Files

```
df = pd.read_excel('data.xlsx',  
                  sheet_name='Sheet1')
```

Great for multiple sheets. Preserves formatting from spreadsheets.



JSON Files

```
df = pd.read_json('data.json')
```

Ideal for nested data. Common format for web APIs.

Exploring Datasets

1 Quick Overview

```
df.head() # First 5 rows  
df.tail() # Last 5 rows  
df.shape # (rows, columns)
```

2 Data Structure

```
df.info() # Data types, non-null values  
df.dtypes # Column data types
```

3 Statistical Summary

```
df.describe() # Summary statistics  
df.columns # Column names
```

These functions help you understand your data quickly. Always explore before analyzing.

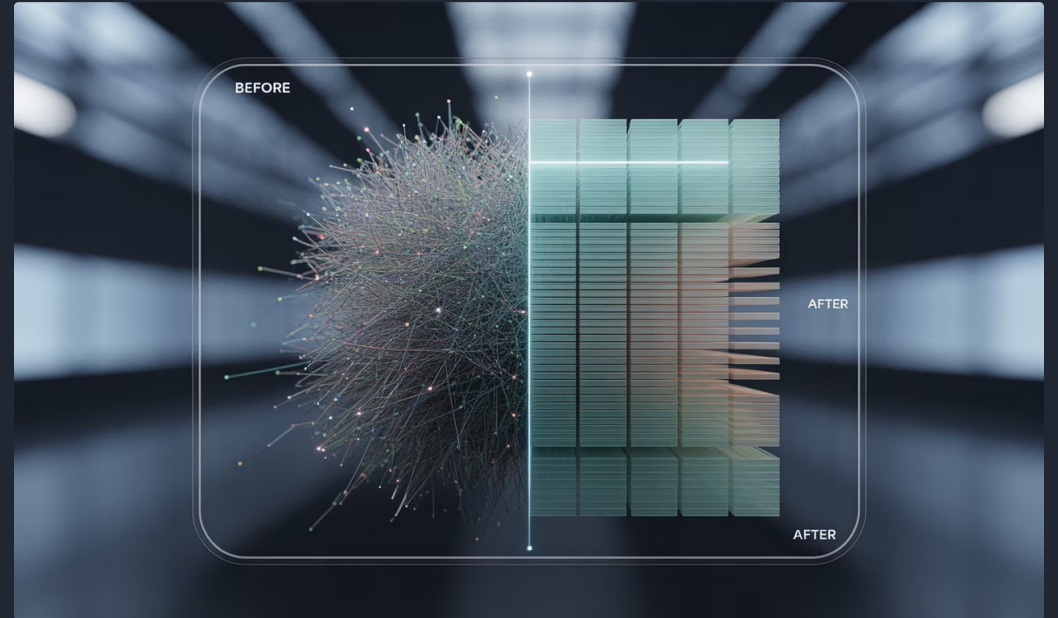


Data Cleaning Basics

Handling Missing Values

```
df.isna().sum()    # Count nulls  
df.dropna()        # Remove rows  
df.fillna(value)    # Replace nulls
```

Missing data can skew analysis. Learn to identify and address it properly.



Replacing Values

```
df.replace('old', 'new')  
df['col'] = df['col'].str.strip()
```

Indexing and Slicing in Pandas

Basic Selection

```
df['column_name'] #  
Single column  
df[['col1', 'col2']] # Multiple  
columns
```

Row Selection

```
df.iloc[0] # First row  
df.iloc[0:5] # First 5 rows  
df.loc['label'] # By label
```

Filtering

```
df[df['age'] > 30] # Condition  
df.query('age > 30') # Query method
```

Think of dataframes as spreadsheets. You can select rows, columns, or specific cells.



Hands-On Lab: Data Cleaning

In today's lab, you'll clean and format a real-world dataset using Pandas.

Load the Dataset

Import the provided "sales_data.csv" file using Pandas.

Explore and Understand

Use methods like `head()`, `info()`, and `describe()` to examine the data.

Clean Missing Values

Identify and handle null values using appropriate strategies.

Format and Standardize

Fix data types and standardize text fields for consistency.



Next Steps and Resources

Coming Up Next Week

- Data visualization with Matplotlib
- Statistical analysis techniques
- Group operations in Pandas

Complete this week's lab assignment by Friday at 11:59 PM.

Additional Resources

- Pandas documentation: pandas.pydata.org
- NumPy user guide: numpy.org/doc
- Office hours: Tuesday 2-4 PM, Room 302