Summer of Code

Artificial Intelligence

(Machine Learning & Deep Learning)

Instructor
Wajahat Ullah
- *Research Assistant* (DIP Lab)

Duration
03 Months
(September – November)

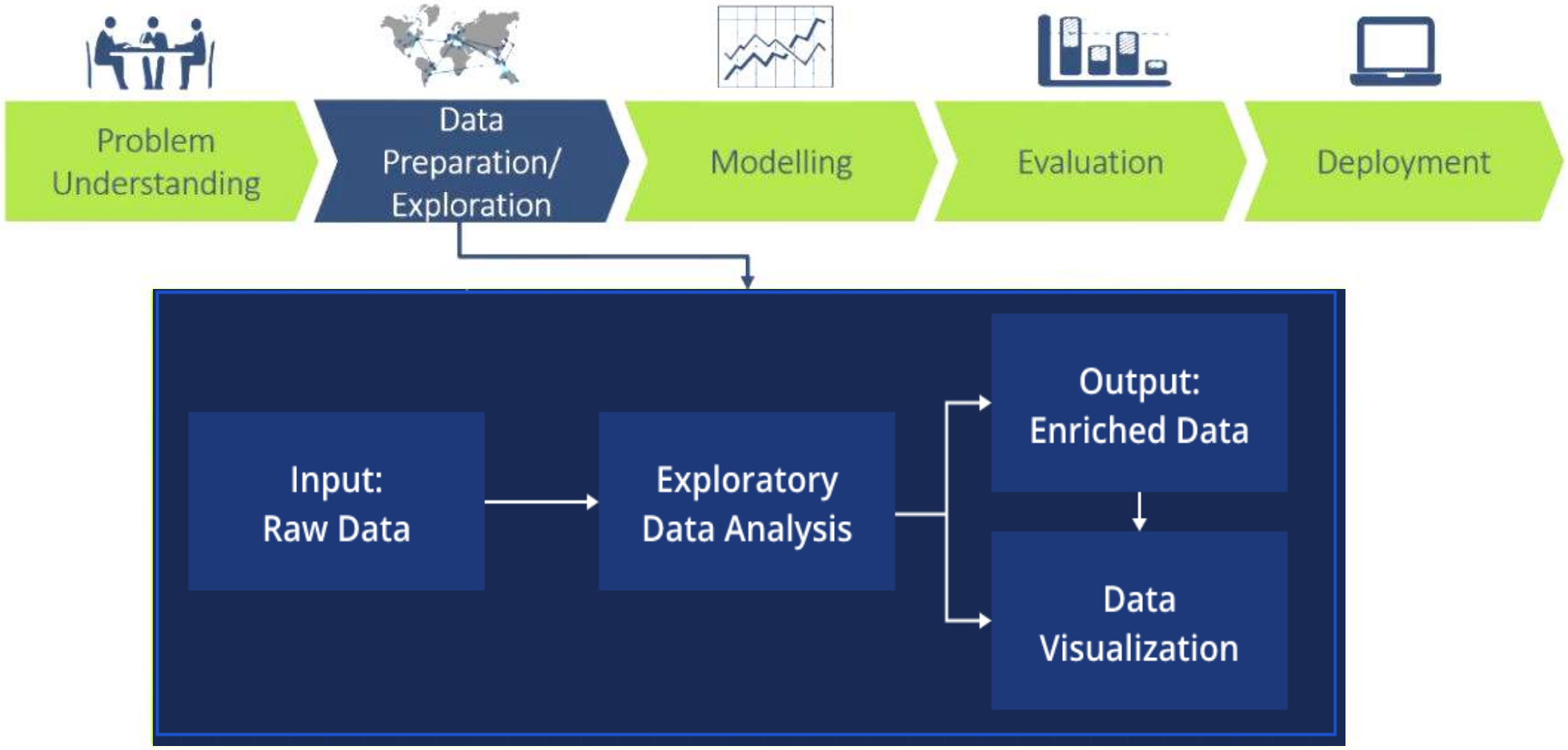
Day 01 – Exploratory Data Analysis
(Introduction to Pandas)

Objectives:

- Introduction to Pandas
- Series and DataFrame
- Operations on DataFrame
- Handling Missing Data

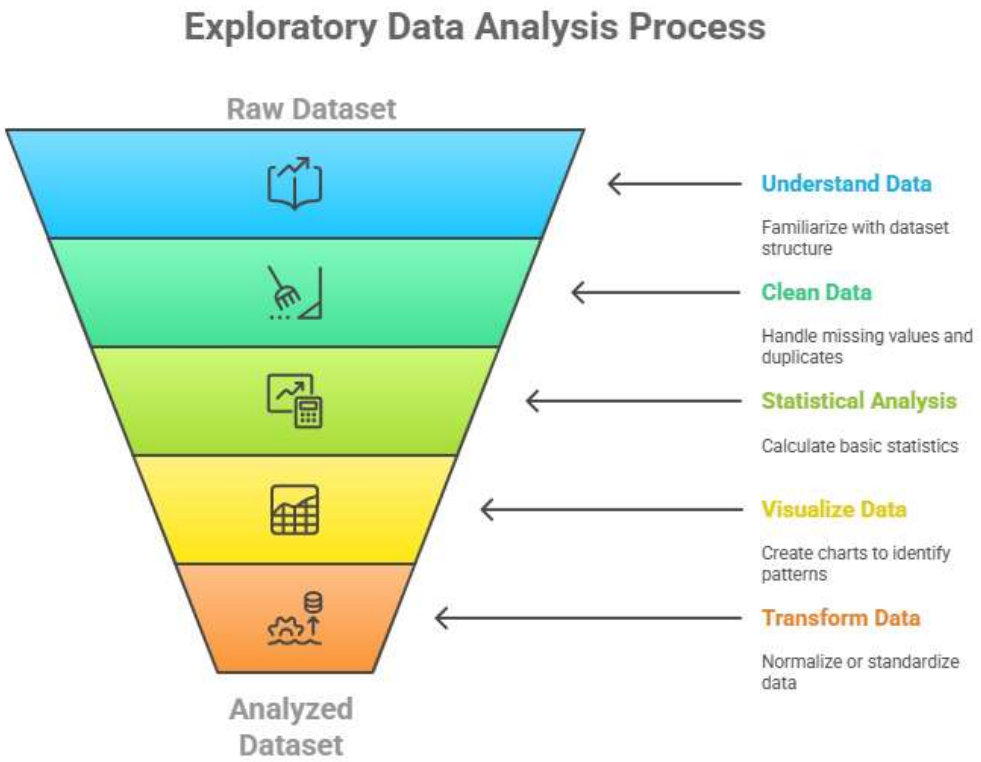
Exploratory Data Analysis

The process of examining datasets to summarize their main characteristics. It is a crucial step in the data analysis workflow to gain a deep understanding of the dataset before modeling.



Key Steps in EDA

- **Understanding the Data:** Get familiar with the dataset, check number of rows, columns, and data types.
- **Data Cleaning:** Handle missing values, duplicates, and inconsistencies.
- **Statistical Analysis:** Use basic statistics (mean, median, standard deviation) to summarize each variable.
- **Data Visualization:** Use charts to uncover patterns, trends and outliers.
- **Data Transformation** (if needed)**:** Normalize or standardize values, or convert data into a better format for further analysis or modeling.



Python Libraries for EDA

- **`NumPy:`** Essential for numerical operations in Python, it provides support for multi-dimensional arrays, along with mathematical functions on these arrays.
- **`Pandas:`** Library for data manipulation and analysis. It makes it easy to clean, transform, and aggregate data.
- **`Matplotlib:`** A versatile plotting library used to create static, interactive, and animated visualizations in Python.
- **`sklearn:`** Primarily a machine learning library but includes many tools useful for data preprocessing and feature selection, which are key parts of EDA.

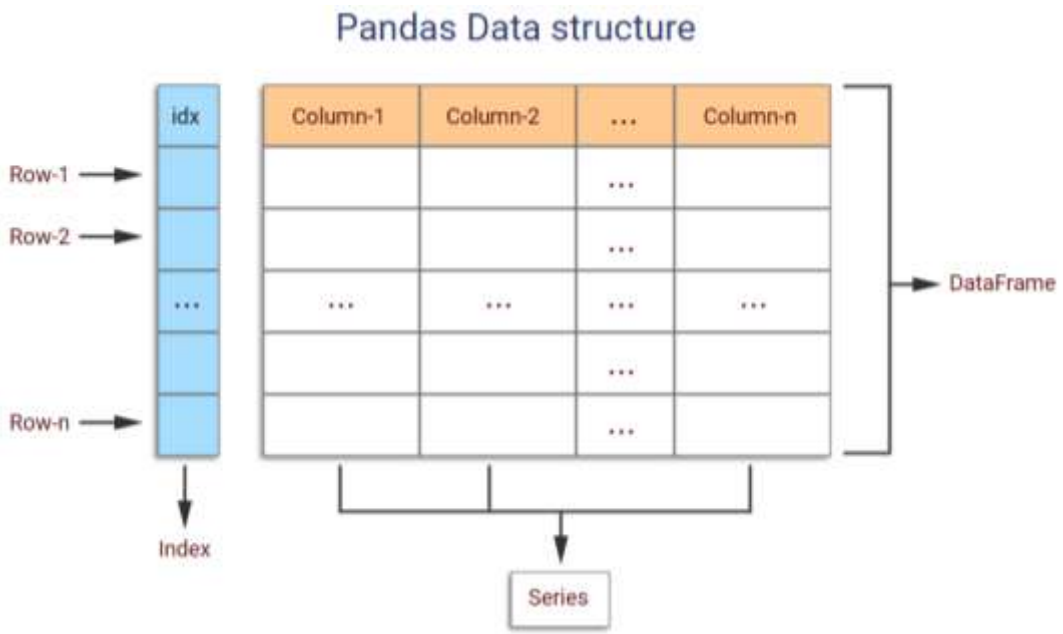
Introduction to Pandas

Pandas: The foundational library for data analysis in Python.

- **Initial release:** 2009
- Built on top of NumPy library
- **Core Data Structure**: Series and DataFrame, which are basically NumPy arrays with additional functionalities for data analysis.

Why Pandas?

- Handles structured data efficiently.
- Simplifies tasks like data cleaning, transformation, and visualization.
- Offers powerful tools for working with tabular and time-series data.




```
import pandas as pd
print(pd.__version__)
```

Pandas Data Structures

Series:

- A 1-dimensional, array-like structure with labeled indices.
- Used for storing and manipulating a single column or list of data.

DataFrame:

- A 2-dimensional tabular structure with rows and columns.
- Can be created from dictionaries, lists, or NumPy arrays.

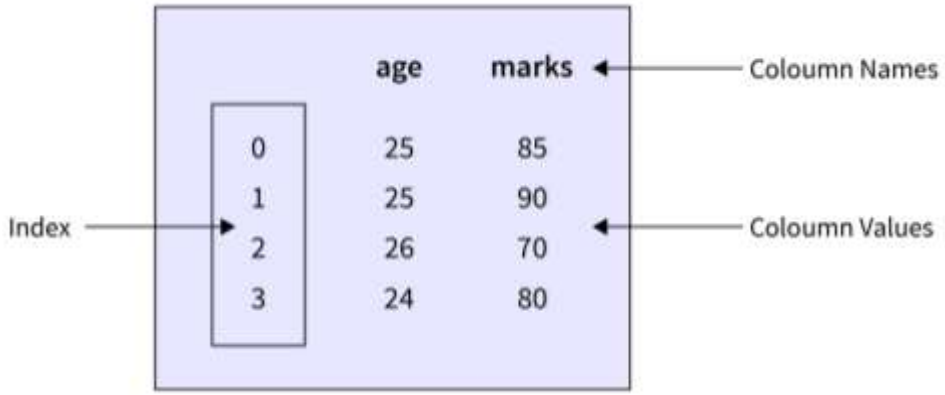
Key Features:

- Supports heterogeneous data types.
- Easy data manipulation and aggregation.
- Offers methods to filter, group, and transform data efficiently.

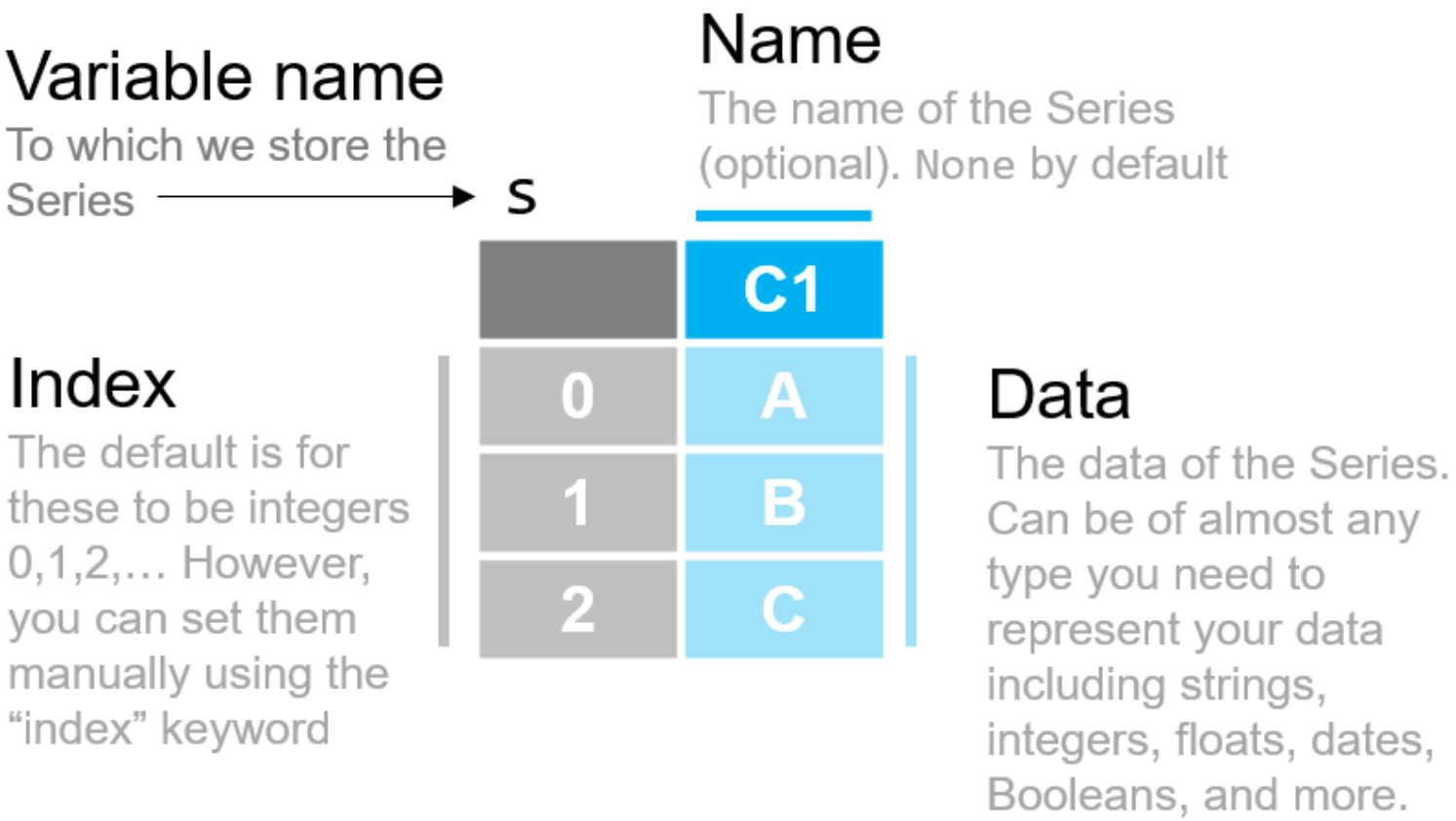
Series

Index	Data
0	Mark
1	Justin
2	John
3	Vicky

DataFrame

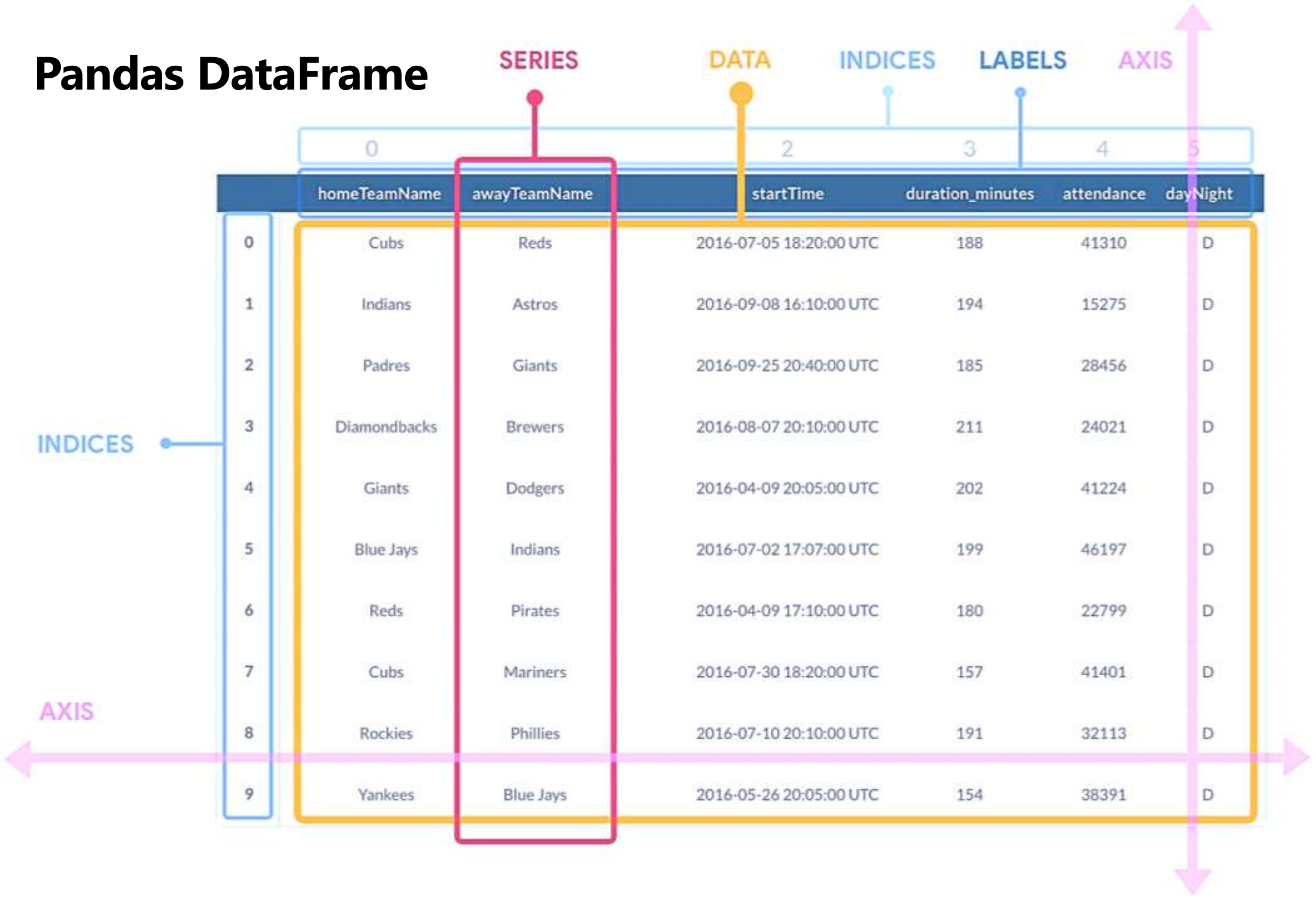


Pandas Series




```
s = pd.Series(data=["A","B","C"],
              name="C1")
```

Pandas DataFrame



	homeTeamName	awayTeamName	startTime	duration_minutes	attendance	dayNight
0	Cubs	Reds	2016-07-05 18:20:00 UTC	188	41310	D
1	Indians	Astros	2016-09-08 16:10:00 UTC	194	15275	D
2	Padres	Giants	2016-09-25 20:40:00 UTC	185	28456	D
3	Diamondbacks	Brewers	2016-08-07 20:10:00 UTC	211	24021	D
4	Giants	Dodgers	2016-04-09 20:05:00 UTC	202	41224	D
5	Blue Jays	Indians	2016-07-02 17:07:00 UTC	199	46197	D
6	Reds	Pirates	2016-04-09 17:10:00 UTC	180	22799	D
7	Cubs	Mariners	2016-07-30 18:20:00 UTC	157	41401	D
8	Rockies	Phillies	2016-07-10 20:10:00 UTC	191	32113	D
9	Yankees	Blue Jays	2016-05-26 20:05:00 UTC	154	38391	D

Indexing and Selection

.iloc selections - position based selection

data.iloc[<row selection], <column selection>]

Integer list of rows: [0,1,2]
Slice of rows: [4:7]
Single values: 1

Integer list of columns: [0,1,2]
Slice of columns: [4:7]
Single column selections: 1

---

loc selections - position based selection

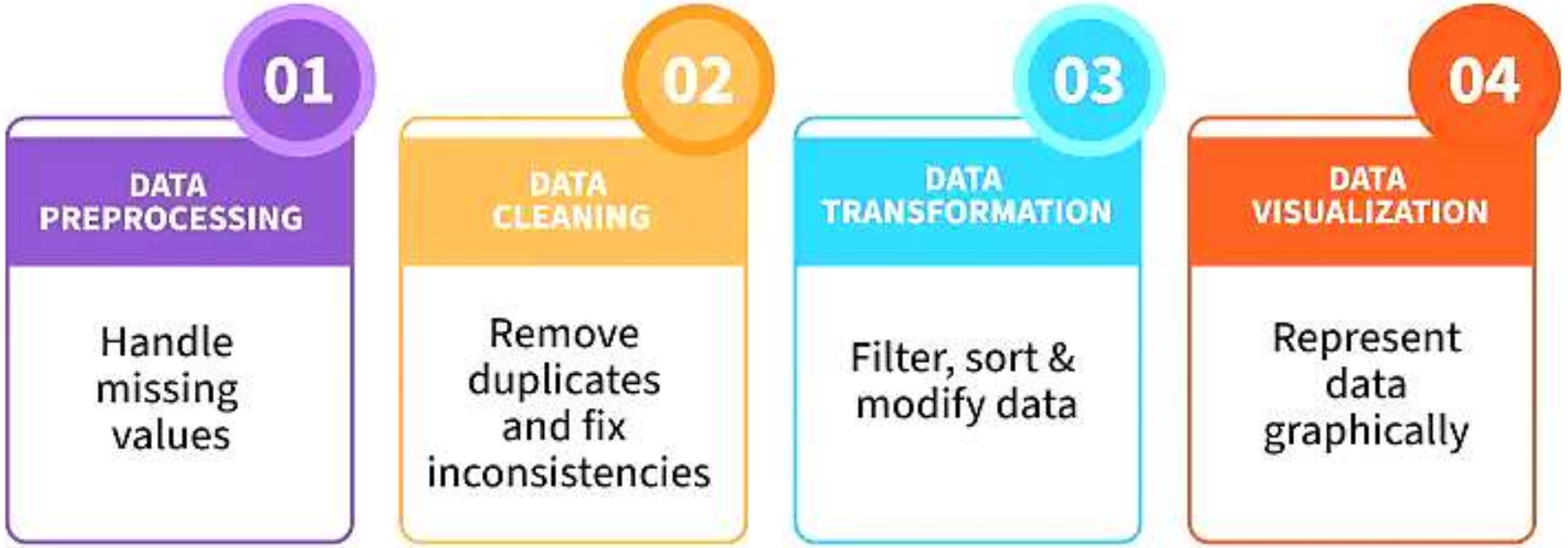
data.loc[<row selection], <column selection>]

Index/Label value: 'john'
List of labels: ['john', 'sarah']
Logical/Boolean index: data['age'] == 10

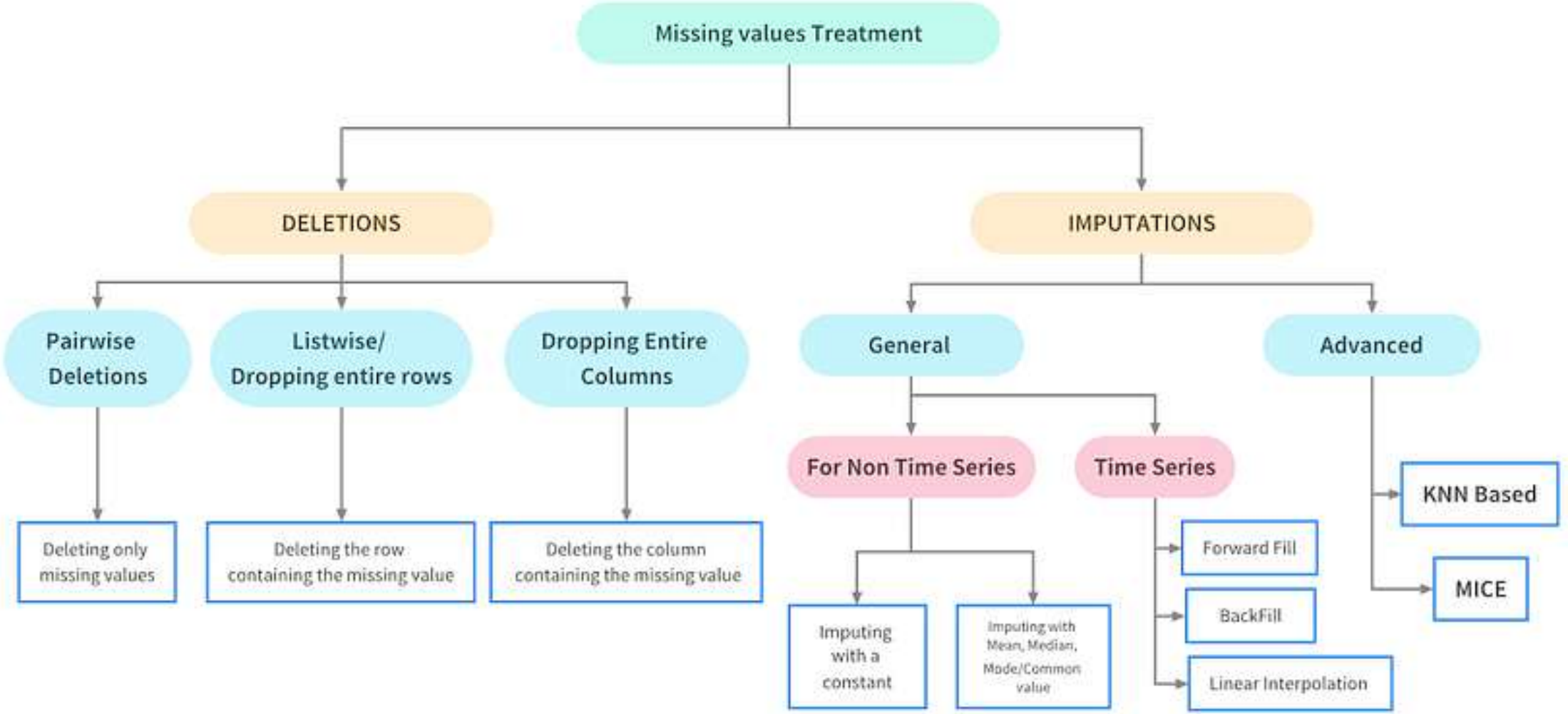
Named column: 'first_name'
List of column names: ['first_name', 'age']
Slice of columns: 'first_name':'address'

Pandas Basic Operations

Pandas simplifies data handling by enabling efficient preprocessing, cleaning, transformation, and visualization.



Handling Missing Values



Thank You