Pandas is a crucial library in Data Science (DS) for data manipulation and analysis. It provides flexible data structures, primarily Series and DataFrame, that allow efficient data handling.

Key Features of Pandas in Data Science

1. Data Structures
   - Series: A one-dimensional labeled array.
   - DataFrame: A two-dimensional table with labeled rows and columns.
2. Data Loading
   - Read and write from various formats: CSV, Excel, JSON, SQL, Parquet, etc.

Creating a Pandas Series

A **Series** is a one-dimensional labeled array.

```
import pandas as pd
pd.Series(data, index=index)
```

- **data**: Can be a list, NumPy array, dictionary, or scalar value.
- **index**: Optional, used to label elements.

```
import pandas as pd

# Creating a Series from a list
s1 = pd.Series([10, 20, 30, 40])
print(s1)

# Creating a Series with custom index
s2 = pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])
```

```
print(s2)
# Creating a Series from a dictionary
s3 = pd.Series({'a': 1, 'b': 2, 'c': 3})
print(s3)

# Creating a Series from a scalar value
s4 = pd.Series(5, index=['x', 'y', 'z'])
print(s4)
```

2️⃣Creating a Pandas DataFrame

A **DataFrame** is a two-dimensional table with labeled rows and columns.

```
pd.DataFrame(data, index=index, columns=columns)
```

- **data**: Can be a dictionary, list of lists, NumPy array, or another DataFrame.
- **index**: Optional, specifies row labels.
- **columns**: Optional, specifies column labels.

```
import pandas as pd

# Creating DataFrame from a dictionary
data = {'A': [1, 2, 3], 'B': [4, 5, 6]}
df1 = pd.DataFrame(data)
print(df1)

# Creating DataFrame from a list of lists
```

```
data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
df2 = pd.DataFrame(data, columns=['X', 'Y', 'Z'])
print(df2)

# Creating DataFrame from a NumPy array
import numpy as np
arr = np.array([[10, 20], [30, 40], [50, 60]])
df3 = pd.DataFrame(arr, columns=['Col1', 'Col2'])
print(df3)

# Creating DataFrame with custom index
df4 = pd.DataFrame({'Name': ['Alice', 'Bob'], 'Age': [25, 30]}, index=['ID1', 'ID2'])
print(df4)

import pandas as pd
df = pd.read_csv("data.csv")  # Read CSV file
```

Pandas Properties and Methods

1. Pandas Series

A **Series** is a one-dimensional labeled array.

Properties of Pandas Series

Property	Description
s.index	Returns index labels
s.values	Returns values as a NumPy array
s.dtype	Data type of elements
s.shape	Returns shape of the Series

s.size	Total number of elements
s.empty	Checks if Series is empty
s.name	Name of the Series
s.nbytes	Memory usage in bytes
s.is_unique	Checks if values are unique

Methods of Pandas Series

1. Creation & Conversion

```
import pandas as pd
import numpy as np
s = pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])
```

Method	Description
pd.Series(data, index)	Create Series
s.astype(dtype)	Change data type
s.to_list()	Convert to list
s.to_numpy()	Convert to NumPy array
s.to_dict()	Convert to dictionary

2. Data Selection & Indexing

Method	Description
s.loc[label]	Select by label
s.iloc[pos]	Select by position
s.get(key, default)	Get value with default

3. Data Manipulation

Method	Description
s.add(n), s.sub(n), s.mul(n), s.div(n)	Element-wise arithmetic
s.apply(func)	Apply function
s.replace(old, new)	Replace values
s.drop(index)	Drop element

4. Statistical & Aggregation

Method	Description
s.count()	Count non-null values
s.sum()	Sum of values
s.mean()	Mean
s.median()	Median
s.std()	Standard deviation
s.min(), s.max()	Min & Max
s.idxmin(), s.idxmax()	Index of min/max
s.unique()	Unique values
s.nunique()	Number of unique values
s.value_counts()	Count occurrences of values

2. Pandas DataFrame

A **DataFrame** is a two-dimensional table with labeled rows and columns.

Properties of Pandas DataFrame

Property	Description

df.index	Returns index labels
df.columns	Returns column labels
df.dtypes	Data types of columns
df.shape	Returns shape (rows, columns)
df.size	Total number of elements
df.empty	Checks if DataFrame is empty
df.ndim	Returns dimensions (2 for DataFrame)
df.T	Transposes the DataFrame

Methods of Pandas DataFrame

1. Creation & Conversion

```
data = {'A': [1, 2, 3], 'B': [4, 5, 6]}
df = pd.DataFrame(data)
```

Method	Description
pd.DataFrame(data)	Create DataFrame
df.to_numpy()	Convert to NumPy array
df.to_dict()	Convert to dictionary
df.to_csv(filename)	Save as CSV
df.to_excel(filename)	Save as Excel

2. Data Selection & Indexing

Method	Description
df['col']	Select column

df.loc[row, col]	Select by label
df.iloc[row, col]	Select by position
df.at[row, col]	Fast label access
df.iat[row, col]	Fast position access

3. Data Cleaning & Manipulation

Method	Description
df.drop(columns='A')	Drop column
df.drop(index=0)	Drop row
df.fillna(value)	Fill missing values
df.replace(old, new)	Replace values

4. Statistical & Aggregation

Method	Description
df.describe()	Summary statistics
df.count()	Count non-null values
df.mean()	Mean of columns
df.median()	Median
df.std()	Standard deviation
df.min(), df.max()	Min & Max
df.value_counts()	Count occurrences

5. Merging & Joining

Method	Description
pd.concat([df1, df2])	Concatenate DataFrames
pd.merge(df1, df2, on='key')	SQL-like join

6. Data Visualization

Method	Description
df.plot(kind='bar')	Bar chart
df.plot(kind='line')	Line chart