

# PANDAS

Pandas is a popular open-source Python library used for data manipulation and analysis.

## Key Features of Pandas:

- Data Cleaning: Pandas provides powerful tools to clean and transform data, like handling missing values, duplicate data, and outliers.
- Data Manipulation: It supports a wide range of operations like filtering, sorting, grouping, merging, and reshaping data.
- Time Series Analysis: It includes functionality for working with dates, times, and time-indexed data.
- Input/Output: Pandas allows reading from and writing to various file formats, including CSV, Excel, SQL databases & JSON.

```
In [1]: import pandas as pd  
import numpy as np
```

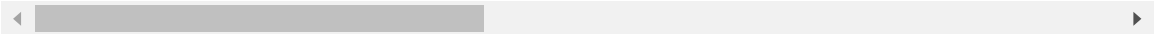
```
In [2]: store = pd.read_csv(r'C:\Users\Shriniwas\Desktop\Data Analyst Course\11. 14_Nov_
```

```
In [3]: store
```

Out[3]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns



In [4]:

```
id(store) # Represents the memory address of the DataFrame store
```

Out[4]: 1742743200752

In [5]: `len(store)` # To Check the total no. of the Rows of the DataFrame store

Out[5]: 10194

In [6]: `store.columns` # Returns the column Labels of the DataFrame store

Out[6]: Index(['Category', 'City', 'Country/Region', 'Customer Name', 'Manufacturer', 'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region', 'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category', 'Discount', 'Profit', 'Quantity', 'Sales'], dtype='object')

In [7]: `len(store.columns)` # Returns the Total number of columns in the store DataFrame

Out[7]: 19

In [8]: `store.shape` # Returns the Total number of Rows & Columns in the store DataFrame

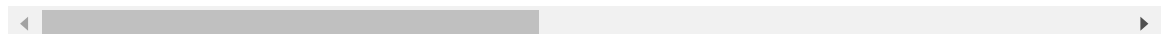
Out[8]: (10194, 19)

In [9]: `store.isnull()` # Used to detect missing or NaN (Not a Number) values in a DataFrame

Out[9]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Postal Code
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
10189	False	False	False	False	False	False	False	False
10190	False	False	False	False	False	False	False	False
10191	False	False	False	False	False	False	False	False
10192	False	False	False	False	False	False	False	False
10193	False	False	False	False	False	False	False	False

10194 rows × 19 columns

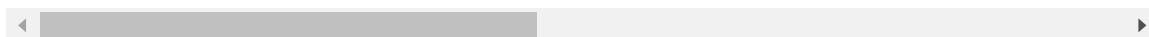


In [10]: `store.isna()` # Used to detect missing or NaN (Not a Number) values in a DataFrame

Out[10]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Postal Code
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
10189	False	False	False	False	False	False	False	False
10190	False	False	False	False	False	False	False	False
10191	False	False	False	False	False	False	False	False
10192	False	False	False	False	False	False	False	False
10193	False	False	False	False	False	False	False	False

10194 rows × 9 columns



In [11]: `store.isnull().sum()` *# Helps you quickly count the number of missing values (Na*

Out[11]:

Category	0
City	0
Country/Region	0
Customer Name	0
Manufacturer	0
Order Date	0
Order ID	0
Postal Code	0
Product Name	0
Region	0
Segment	0
Ship Date	0
Ship Mode	0
State/Province	0
Sub-Category	0
Discount	0
Profit	0
Quantity	0
Sales	0

dtype: int64

In [12]: `store.dtypes` *# will return a Series that shows the data type of each column in t*

```
Out[12]: Category      object
City                  object
Country/Region        object
Customer Name          object
Manufacturer            object
Order Date             object
Order ID              object
Postal Code            object
Product Name           object
Region                 object
Segment                object
Ship Date              object
Ship Mode              object
State/Province         object
Sub-Category           object
Discount               float64
Profit                 float64
Quantity               int64
Sales                  float64
dtype: object
```

```
In [13]: store.info() # for checking the data types, null values, and overall size.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10194 entries, 0 to 10193
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Category              10194 non-null  object
 1   City                  10194 non-null  object
 2   Country/Region        10194 non-null  object
 3   Customer Name          10194 non-null  object
 4   Manufacturer            10194 non-null  object
 5   Order Date             10194 non-null  object
 6   Order ID              10194 non-null  object
 7   Postal Code            10194 non-null  object
 8   Product Name           10194 non-null  object
 9   Region                 10194 non-null  object
10   Segment                10194 non-null  object
11   Ship Date              10194 non-null  object
12   Ship Mode              10194 non-null  object
13   State/Province         10194 non-null  object
14   Sub-Category           10194 non-null  object
15   Discount               10194 non-null  float64
16   Profit                 10194 non-null  float64
17   Quantity               10194 non-null  int64
18   Sales                  10194 non-null  float64
dtypes: float64(3), int64(1), object(15)
memory usage: 1.5+ MB
```

```
In [14]: pd.__version__ # will display the version of Pandas
```

```
Out[14]: '2.2.2'
```

```
In [15]: store.head() # Used to display the first 5 rows of a DataFrame (by default)
```

Out[15]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Post Code
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800	77058
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	US-2020-112326	60540
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	US-2020-112326	60540
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	US-2020-112326	60540
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	US-2020-141817	19104

In [16]: `store.head(3)` # Used to display the first 3 rows of the DataFrame

Out[16]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Post Code
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800	77058
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	US-2020-112326	60540
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	US-2020-112326	60540

In [17]: `store.tail()` # Used to display the last 5 rows of a DataFrame (by default)

Out[17]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

In [18]: store.tail(3) # Used to display the last 3 rows of the DataFrame

Out[18]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

In [19]: store # To display all the records

Out[19]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns



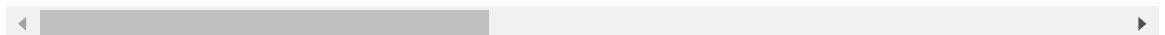
In [20]: `store[:]` # To display all the records



Out[20]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order Number
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns



# Slice indexing

- 0:50:10: This represents a slice with the following parameters:
- 0: Start at the 0th row (inclusive).
- 50: Stop at the 50th row (exclusive).
- 10: Step by 10 rows.

```
In [21]: store[0:50:10]
```

Out[21]:

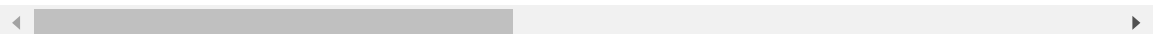
	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Po C
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800	77
10	Office Supplies	Henderson	United States	Maria Etezadi	Southworth	06-01-2020	US-2020-167199	42
20	Furniture	Dover	United States	Seth Vernon	DAX	11-01-2020	US-2020-130092	19
30	Office Supplies	San Francisco	United States	Brian Dahlen	Tennsco	13-01-2020	US-2020-157147	94
40	Office Supplies	Scottsdale	United States	Toby Swindell	GBC	19-01-2020	US-2020-146591	85



```
In [22]: store.head(1) # Used to display the first row of the DataFrame
```

Out[22]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Postal Code
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800	77095



```
In [23]: store['Category'] # Used to access a specific column in the DataFrame
```

```
Out[23]: 0      Office Supplies
1      Office Supplies
2      Office Supplies
3      Office Supplies
4      Office Supplies
...
10189   Office Supplies
10190   Office Supplies
10191   Office Supplies
10192      Technology
10193   Office Supplies
Name: Category, Length: 10194, dtype: object
```

```
In [24]: store[['Customer Name', 'Category', 'City']] # # Used to access no. of column in t
```

```
Out[24]:
```

	Customer Name	Category	City
0	Darren Powers	Office Supplies	Houston
1	Phillina Ober	Office Supplies	Naperville
2	Phillina Ober	Office Supplies	Naperville
3	Phillina Ober	Office Supplies	Naperville
4	Mick Brown	Office Supplies	Philadelphia
...	...	...	...
10189	Patrick O'Donnell	Office Supplies	New York City
10190	Erica Bern	Office Supplies	Fairfield
10191	Jill Matthias	Office Supplies	Loveland
10192	Patrick O'Donnell	Technology	New York City
10193	Harry Olson	Office Supplies	Charlottetown

10194 rows × 3 columns

```
In [25]: store.columns # Used to get the column names of a DataFrame
```

```
Out[25]: Index(['Category', 'City', 'Country/Region', 'Customer Name', 'Manufacturer',
               'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',
               'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category',
               'Discount', 'Profit', 'Quantity', 'Sales'],
              dtype='object')
```

```
In [26]: store.dtypes # sed to get the data types of each column in a DataFrame
```

```
Out[26]: Category      object
City                  object
Country/Region        object
Customer Name          object
Manufacturer            object
Order Date             object
Order ID               object
Postal Code            object
Product Name           object
Region                 object
Segment                object
Ship Date              object
Ship Mode              object
State/Province         object
Sub-Category           object
Discount               float64
Profit                 float64
Quantity               int64
Sales                  float64
dtype: object
```

## To Split Numerical Data Set

- Will use New Dataframe to store Numerial Data From Exitisg DataFrame

```
In [27]: store.columns
```

```
Out[27]: Index(['Category', 'City', 'Country/Region', 'Customer Name', 'Manufacturer',
               'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',
               'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category',
               'Discount', 'Profit', 'Quantity', 'Sales'],
              dtype='object')
```

```
In [28]: store_num = store[['Discount', 'Profit', 'Quantity', 'Sales']]
store_num # Used to select specific columns from a Pandas DataFrame (store) and
```

Out[28]:

	Discount	Profit	Quantity	Sales
0	0.2	5.5512	2	16.448
1	0.8	-5.4870	2	3.540
2	0.2	4.2717	3	11.784
3	0.2	-64.7748	3	272.736
4	0.2	4.8840	3	19.536
...	...	...	...	...
10189	0.2	19.7910	3	52.776
10190	0.2	6.4750	2	20.720
10191	0.2	-0.6048	3	3.024
10192	0.0	2.7279	7	90.930
10193	0.2	-0.6048	3	3.024

10194 rows × 4 columns

In [29]:

```
store_cate = store[['Category', 'City', 'Country/Region', 'Customer Name', 'Manu
                    'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',
                    'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-
store_cate
```

Out[29]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 15 columns



In [30]: store\_cate.dtypes # to display the data types of each column in the store\_cate D

```
Out[30]: Category      object
City      object
Country/Region  object
Customer Name  object
Manufacturer  object
Order Date    object
Order ID      object
Postal Code   object
Product Name  object
Region        object
Segment       object
Ship Date     object
Ship Mode     object
State/Province  object
Sub-Category  object
dtype: object
```

```
In [31]: store_num.dtypes # to display the data types of each column in the store_cate Da
```

```
Out[31]: Discount      float64
Profit      float64
Quantity      int64
Sales      float64
dtype: object
```

```
In [32]: store['Profit'].mean() # Average
```

```
Out[32]: 28.673417166960963
```

```
In [33]: store['Profit'].median() # Middle value after assending the value
```

```
Out[33]: 8.69
```

```
In [34]: store['Profit'].mode() # the most frequent value or values
```

```
Out[34]: 0      0.0
Name: Profit, dtype: float64
```

```
In [35]: store['Profit'].var() # Variance
```

```
Out[35]: 54040.02971828826
```

```
In [36]: store['Profit'].std() # o calculate the standard deviation
```

```
Out[36]: 232.46511505662147
```