

## Practical No.2

**Aim:-** Data Frames and Basic Data Processing.

- Read data from CSV and JSON files into a data frame.
- Perform basic data pre-processing tasks such as handling missing values and outliers.
- Manipulate and transform data using functions like filtering, sorting, and grouping.

Create a CSV file given below:

Roll no	Name	Marks	Grade
1	Harshal	80	O
2	Rahul	75	A+
3	Priya	55	B+
4	Roshan	67	A+
5	Om	78	A+
6	Soham	90	O+

Read Data from .CSV File

```
[1]: import pandas as pd

[2]: data = pd.read_csv("Student.csv")

#Display first five rows of file
print("Reading data from CSV file:")
data.head()
```

Reading data from CSV file:

```
[2]:
```

	Roll No	Name	Marks	Grade
0	1	Harshal	80	O
1	2	Rahul	75	A+
2	3	Priya	55	B+
3	4	Roshan	67	A+
4	5	Om	78	A+

Create a JSON file as given below: (animals1)

```
[
  {
    "name": "Kitty",
    "species" : "cat",
    "foods": {
      "likes": ["fresh food"],
      "dislikes": ["stale food"]
    }
  },
  {
    "name": "Pupster",
    "species" : "dog",
    "foods": {
      "likes": ["tomatoes", "peas"],
      "dislikes": ["bread"]
    }
  },
  {
    "name": "Tux",
    "species" : "cat",
    "foods": {
      "likes": ["fancy dishes"],
      "dislikes": ["basic cat food"]
    }
  }
]
```

## Read Data from JSON File

```
[9]: data_json = pd.read_json("animals1.json")
      print("Reading data from JSON file:")
      data_json.head()
```

```
[9]:
```

	name	species	foods
0	Kitty	cat	{'likes': ['fresh food'], 'dislikes': ['stale ...
1	Pupster	dog	{'likes': ['tomatoes', 'peas'], 'dislikes': [...
2	Tux	cat	{'likes': ['fancy dishes'], 'dislikes': ['basi...

## Create JSON data manually

```
[10]: import json
```

```
[12]: input = '''[
      {"studid": "001", "name": "Nandita", "Age": "25"},
      {"studid": "002", "name": "Rinki", "Age": "19"},
      {"studid": "009", "name": "Kajal", "Age": "15"}
      ]'''
      info = json.loads(input)
      print("User Count :", len(info))
      for item in info:
          print("\nStudent ID:", item["studid"])
          print("Name:", item["name"])
          print("Age:", item["Age"])
```

User Count : 3

Student ID: 001

Name: Nandita

Age: 25

Student ID: 002

Name: Rinki

Age: 19

Student ID: 009

Name: Kajal

Age: 15

---

Handling Missing Values

```
[2]: df_car = pd.read_csv("Car Inventory Details.csv")  
df_car
```

```
[2]:
```

	Make	Model	Color	Mileage	Sell Price	Buy Price
0	Honda	Accord	Red	63,512	4000.0	3000
1	Honda	Accord	Blue	95,135	2500.0	2000
2	Toyota	Camry	Black	75,006	45000.0	44000
3	Nissan	Altima	Green	69,847	3826.0	3000
4	Toyota	Corolla	Black	87,278	2224.0	2100
5	Honda	Civic	White	1,38,789	2723.0	1900
6	Ford	F-150	Black	89,073	3950.0	3000
7	Chevrolet	Silverado	Green	1,09,231	4959.0	4500
8	Chevrolet	Impala	Silver	87,675	3791.0	3500
9	Dodge	Charger	Silver	34,853	4349.0	3500
10	Dodge	Charger	Silver	58,173	4252.0	4000
11	Dodge	Charger	NaN	60,000	NaN	4000

```
[3]: #Drop rows with missing values
df_car_clean = df_car.dropna(axis=0,how='any')
df_car_clean
```

```
[3]:
```

	Make	Model	Color	Mileage	Sell Price	Buy Price
0	Honda	Accord	Red	63,512	4000.0	3000
1	Honda	Accord	Blue	95,135	2500.0	2000
2	Toyota	Camry	Black	75,006	45000.0	44000
3	Nissan	Altima	Green	69,847	3826.0	3000
4	Toyota	Corolla	Black	87,278	2224.0	2100
5	Honda	Civic	White	1,38,789	2723.0	1900
6	Ford	F-150	Black	89,073	3950.0	3000
7	Chevrolet	Silverado	Green	1,09,231	4959.0	4500
8	Chevrolet	Impala	Silver	87,675	3791.0	3500
9	Dodge	Charger	Silver	34,853	4349.0	3500
10	Dodge	Charger	Silver	58,173	4252.0	4000

```
[4]: #Filling missing values with a specific value.
df_car_filled = df_car_clean.fillna(0)
df_car_filled
```

```
[4]:
```

	Make	Model	Color	Mileage	Sell Price	Buy Price
0	Honda	Accord	Red	63,512	4000.0	3000
1	Honda	Accord	Blue	95,135	2500.0	2000
2	Toyota	Camry	Black	75,006	45000.0	44000
3	Nissan	Altima	Green	69,847	3826.0	3000
4	Toyota	Corolla	Black	87,278	2224.0	2100
5	Honda	Civic	White	1,38,789	2723.0	1900
6	Ford	F-150	Black	89,073	3950.0	3000
7	Chevrolet	Silverado	Green	1,09,231	4959.0	4500
8	Chevrolet	Impala	Silver	87,675	3791.0	3500
9	Dodge	Charger	Silver	34,853	4349.0	3500
10	Dodge	Charger	Silver	58,173	4252.0	4000

## Handling Outliers

```
[5]: #Assume 'Sell Price' is the coulumn with outliers eg. Sell Price with value 45000.
#Replace outliers with the median value.

median_value = df_car_filled['Sell Price'].median()
upper_threshold = df_car_filled['Sell Price'].mean() + 2 * df_car_filled['Sell Price'].std()
lower_threshold = df_car_filled['Sell Price'].mean() - 2 * df_car_filled['Sell Price'].std()
df_car_filled['Sell Price'] = df_car_filled['Sell Price'].apply(lambda x: median_value if x > upper_threshold or x < lower_threshold else x)
```

```
[6]: df_car_filled
```

```
[6]:
```

	Make	Model	Color	Mileage	Sell Price	Buy Price
0	Honda	Accord	Red	63,512	4000.0	3000
1	Honda	Accord	Blue	95,135	2500.0	2000
2	Toyota	Camry	Black	75,006	3950.0	44000
3	Nissan	Altima	Green	69,847	3826.0	3000
4	Toyota	Corolla	Black	87,278	2224.0	2100
5	Honda	Civic	White	1,38,789	2723.0	1900
6	Ford	F-150	Black	89,073	3950.0	3000
7	Chevrolet	Silverado	Green	1,09,231	4959.0	4500
8	Chevrolet	Impala	Silver	87,675	3791.0	3500
9	Dodge	Charger	Silver	34,853	4349.0	3500
10	Dodge	Charger	Silver	58,173	4252.0	4000

```
[7]: #filtering
filtered_data = df_car_filled[df_car_filled['Sell Price'] > 3000]
filtered_data
```

```
[7]:
```

	Make	Model	Color	Mileage	Sell Price	Buy Price
0	Honda	Accord	Red	63,512	4000.0	3000
2	Toyota	Camry	Black	75,006	3950.0	44000
3	Nissan	Altima	Green	69,847	3826.0	3000
6	Ford	F-150	Black	89,073	3950.0	3000
7	Chevrolet	Silverado	Green	1,09,231	4959.0	4500
8	Chevrolet	Impala	Silver	87,675	3791.0	3500
9	Dodge	Charger	Silver	34,853	4349.0	3500
10	Dodge	Charger	Silver	58,173	4252.0	4000



```
[8]: #sorting
sorted_data = df_car_filled.sort_values(by='Sell Price', ascending=False)
sorted_data
```

```
[8]:
```

	Make	Model	Color	Mileage	Sell Price	Buy Price
7	Chevrolet	Silverado	Green	1,09,231	4959.0	4500
9	Dodge	Charger	Silver	34,853	4349.0	3500
10	Dodge	Charger	Silver	58,173	4252.0	4000
0	Honda	Accord	Red	63,512	4000.0	3000
2	Toyota	Camry	Black	75,006	3950.0	44000
6	Ford	F-150	Black	89,073	3950.0	3000
3	Nissan	Altima	Green	69,847	3826.0	3000
8	Chevrolet	Impala	Silver	87,675	3791.0	3500
5	Honda	Civic	White	1,38,789	2723.0	1900
1	Honda	Accord	Blue	95,135	2500.0	2000
4	Toyota	Corolla	Black	87,278	2224.0	2100

```
[9]: #Grouping and calculating mean for numeric columns
numeric_columns= ['Sell Price', 'Buy Price']
grouped_data = df_car_filled.groupby('Make')[numeric_columns].mean()
grouped_data
```

```
[9]:
```

	Sell Price	Buy Price
<b>Make</b>		
<b>Chevrolet</b>	4375.000000	4000.0
<b>Dodge</b>	4300.500000	3750.0
<b>Ford</b>	3950.000000	3000.0
<b>Honda</b>	3074.333333	2300.0
<b>Nissan</b>	3826.000000	3000.0
<b>Toyota</b>	3087.000000	23050.0