```
import numpy as np
import pandas as pd
```

DATASET 1:Titanic-Like Passenger Dataset

Create Dataset using NumPy

```
np.random.seed(1)
rows = 200

df = pd.DataFrame({
    "PassengerId": np.arange(1, rows + 1),
        "Name": np.random.choice(["Mr. John", "Mrs. Anna", "Miss Emma", "Dr. Smith"], rows),
            "Age": np.random.choice(np.append(np.random.randint(1, 80, 180), [np.nan]*20), r
                "Sex": np.random.choice(["male", "female"], rows),
                    "Pclass": np.random.choice([1, 2, 3], rows),
                        "Fare": np.round(np.random.uniform(10, 500, rows), 2)
                        })
```

view Data

```
df.head()
df.tail()
df.sample(3)
```

```
     PassengerId       Name   Age     Sex  Pclass    Fare
127          128   Mr. John  61.0  female       1  246.30
152          153   Mr. John  35.0  female       1  419.86
109          110  Dr. Smith  43.0    male       3  181.72
```

Inspect Structure

```
df.shape
df.columns
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  200 non-null    int64  
 1   Name         200 non-null    object 
 2   Age          200 non-null    float64
 3   Sex          200 non-null    object 
 4   Pclass       200 non-null    int64  
 5   Fare         200 non-null    float64
dtypes: float64(2), int64(2), object(2)
memory usage: 9.5+ KB
```

	PassengerId	Age	Pclass	Fare
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	43.318681	2.035000	251.249700
std	57.879185	21.055981	0.804369	146.847353
min	1.000000	1.000000	1.000000	11.250000
25%	50.750000	27.000000	1.000000	115.550000
50%	100.500000	43.318681	2.000000	265.430000
75%	150.250000	62.000000	3.000000	371.432500
max	200.000000	79.000000	3.000000	493.680000

Handle Missing Data

```
df.isnull().sum()
df["Age"].fillna(df["Age"].mean(), inplace=True)
```

```
/tmp/ipython-input-2336408632.py:2: FutureWarning: A value is trying to be set on a copy of a DataFram
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate c

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inpl


  df["Age"].fillna(df["Age"].mean(), inplace=True)
```

Filtering Rows

```
df[df["Age"] > 30]
df[df["Sex"] == "female"]
```

```
     PassengerId        Name        Age     Sex  Pclass    Fare
  3            4     Mr. John  63.000000  female       2   92.36
  4            5    Dr. Smith  39.000000  female       1  346.02
  5            6    Mrs. Anna  26.000000  female       2  304.88
  8            9    Dr. Smith  29.000000  female       3   44.05
 10           11     Mr. John  77.000000  female       3  247.52
...          ...          ...        ...     ...     ...     ...
186          187     Mr. John  43.318681  female       1  148.20
190          191   Miss Emma  68.000000  female       1  367.68
191          192   Miss Emma  78.000000  female       3   12.94
193          194    Dr. Smith  43.318681  female       1   25.29
196          197    Dr. Smith  33.000000  female       1  300.03

99 rows × 6 columns
```

Grouping and Aggregation

```
df.groupby("Pclass")["Fare"].mean()
```

```
               Fare
Pclass
     1   237.222623
     2   281.545775
     3   232.200147

dtype: float64
```