Contents

9. PANDAS – DATAFRAME – METHODS

1. DataFrame Methods

- ✓ DataFrame is a predefined class.
- ✓ DataFrame having different methods.
- ✓ These methods perform an operation on DataFrame and returns result

1.1. head()

- ✓ head() is predefined method in DataFrame class.
- ✓ We can access head() method by using DataFrame object only.
- ✓ This method returns first five rows from the DataFrame

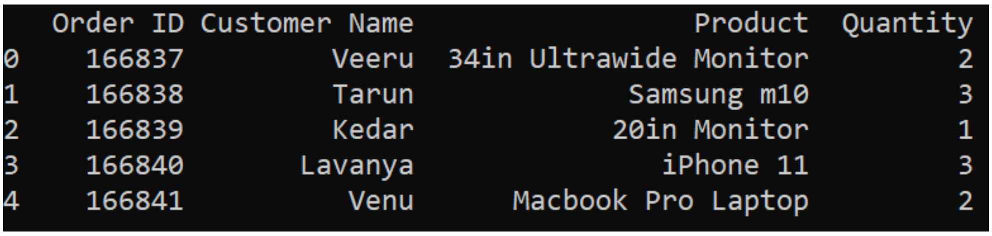
Program	Accessing first five rows from DataFrame
Name	demo1.py
Input file	sales1.csv

```
import pandas as pd

df1 = pd.read_csv("sales1.csv")
df2 = df1.head()

print(df2)
```

Output



```
   Order ID Customer Name                 Product  Quantity
0    166837         Veeru  34in Ultrawide Monitor         2
1    166838         Tarun             Samsung m10         3
2    166839         Kedar            20in Monitor         1
3    166840       Lavanya               iPhone 11         3
4    166841          Venu     Macbook Pro Laptop         2
```

1.2. tail()

- ✓ tail() is predefined method in DataFrame class.
- ✓ We can access tail() method by using DataFrame object only.
- ✓ This method returns last five rows from the DataFrame

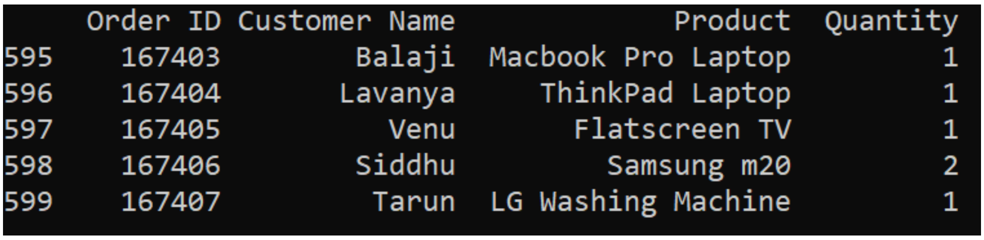
Program   Accessing last five rows from DataFrame
Name   demo2.py
Input file   sales1.csv

```
import pandas as pd

df1 = pd.read_csv("sales1.csv")
df2 = df1.tail()

print(df2)
```

Output



```
     Order ID Customer Name              Product  Quantity
595    167403        Balaji  Macbook Pro Laptop         1
596    167404       Lavanya     ThinkPad Laptop         1
597    167405          Venu       Flatscreen TV         1
598    167406        Siddhu         Samsung m20         2
599    167407         Tarun  LG Washing Machine         1
```

1.3. info()

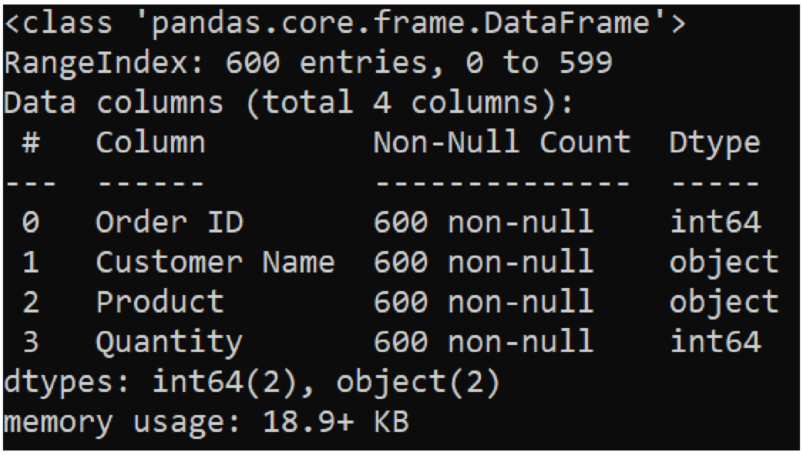
- ✓ info() is predefined method in DataFrame class.
- ✓ We can access info() method by using DataFrame object only.
- ✓ This method returns below information about DataFrame,
    - ○ Type of object
    - ○ Range of object
    - ○ Number of columns
    - ○ Number of rows
    - ○ The data type of each column
    - ○ Number of data types
    - ○ Total memory usage

Program Accessing info() method from DataFrame
Name demo3.py
Input file sales1.csv

```
import pandas as pd

df1 = pd.read_csv("sales1.csv")
df1.info()
```

Output



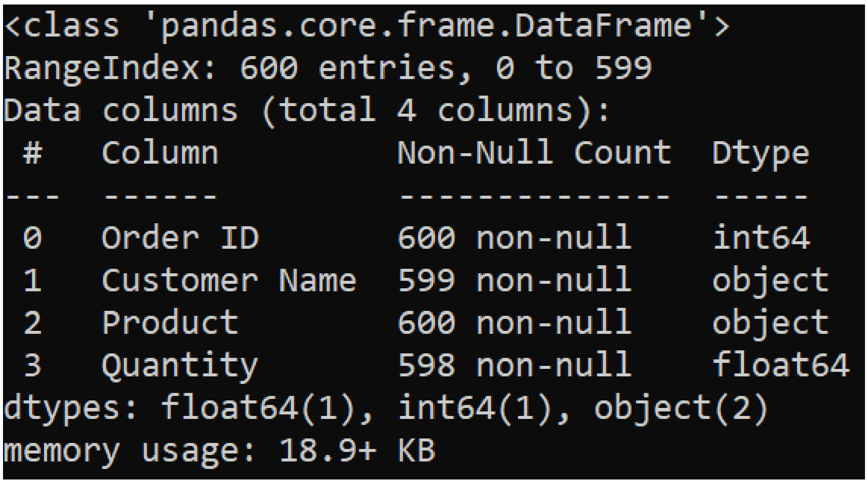
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order ID       600 non-null    int64
 1   Customer Name  600 non-null    object
 2   Product        600 non-null    object
 3   Quantity       600 non-null    int64
dtypes: int64(2), object(2)
memory usage: 18.9+ KB
```

Program	Accessing info() method from DataFrame
Name	demo4.py
Input file	sales1_with_nan.csv

```
import pandas as pd

df1 = pd.read_csv("sales1_with_nan.csv")
df1.info()
```

Output



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order ID       600 non-null    int64
 1   Customer Name  599 non-null    object
 2   Product        600 non-null    object
 3   Quantity       598 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 18.9+ KB
```

1.4. count()

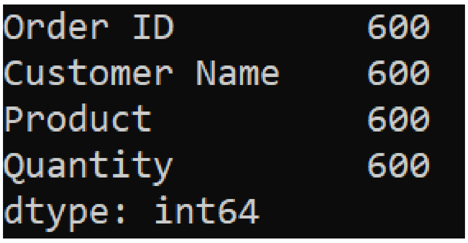
- ✓ count() is predefined method in DataFrame class.
- ✓ We can access count() method by using DataFrame object only.
- ✓ This method returns number of non-null values from each column.

Program Accessing count() method from DataFrame
Name demo5.py
Input file sales1.csv

```
import pandas as pd

df1 = pd.read_csv("sales1.csv")
c = df1.count()
print(c)
```

Output



```
Order ID         600
Customer Name    600
Product          600
Quantity         600
dtype: int64
```

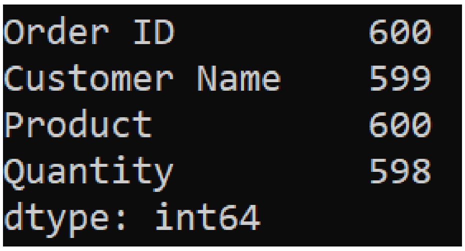
Program Accessing count() method from DataFrame
Name demo6.py
Input file sales1_with_nan.csv

```
import pandas as pd

df1 = pd.read_csv("sales1_with_nan.csv")
c = df1.count()

print(c)
```

Output



```
Order ID         600
Customer Name    599
Product          600
Quantity         598
dtype: int64
```

1.5. describe()

- ✓ describe() is predefined method in DataFrame class.
- ✓ We can access describe() method by using DataFrame object.
- ✓ This method returns the below values,
    - count
    - mean
    - std
    - min
    - 25%
    - 50%
    - 75%
    - max

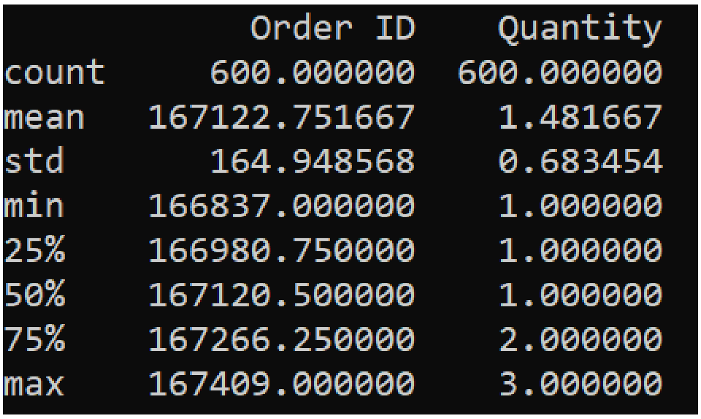
Program	Accessing describe() method from DataFrame
Name	demo7.py
Input file	sales1_with_nan.csv

```
import pandas as pd

df1 = pd.read_csv("sales1.csv")
dsc = df1.describe()

print(dsc)
```

Output



```
            Order ID    Quantity
count     600.000000  600.000000
mean   167122.751667    1.481667
std       164.948568    0.683454
min    166837.000000    1.000000
25%    166980.750000    1.000000
50%    167120.500000    1.000000
75%    167266.250000    2.000000
max    167409.000000    3.000000
```

1.6. nunique()

- ✓ nunique() is predefined method in DataFrame class.
- ✓ We can access nunique() method by using DataFrame object only.
- ✓ This method returns number of unique values from the DataFrame.

Program Name

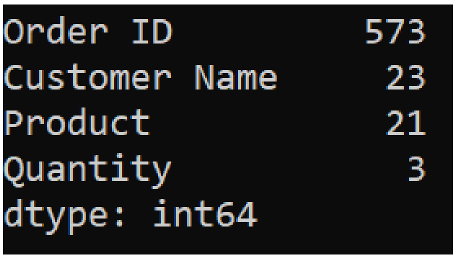
Get the number of nunique values from the DataFrame
demo8.py

```
import pandas as pd

df1 = pd.read_csv("sales1.csv")
nu = df1.nunique()

print(nu)
```

Output



```
Order ID         573
Customer Name     23
Product           21
Quantity           3
dtype: int64
```

2. Accessing single column From DataFrame

- ✓ We can access columns from the DataFrame,
    - If we access single column then it returns the Series
    - If we access two column then it returns the DataFrame with two columns

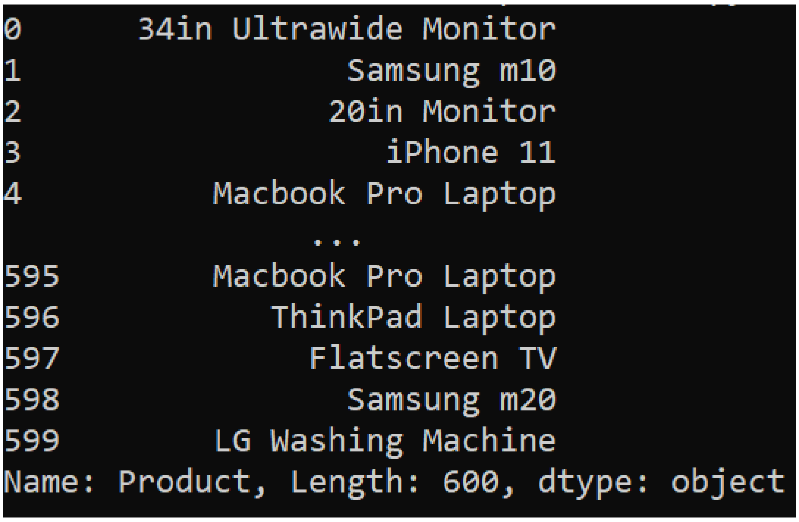
Program	Accessing single column from the DataFrame
Name	demo9.py
Input file	sales1.csv

```
import pandas as pd

df = pd.read_csv("sales1.csv")

print(df.Product)
```

Output



```
0      34in Ultrawide Monitor
1                Samsung m10
2               20in Monitor
3                  iPhone 11
4          Macbook Pro Laptop
                 ...
595        Macbook Pro Laptop
596           ThinkPad Laptop
597             Flatscreen TV
598               Samsung m20
599         LG Washing Machine
Name: Product, Length: 600, dtype: object
```

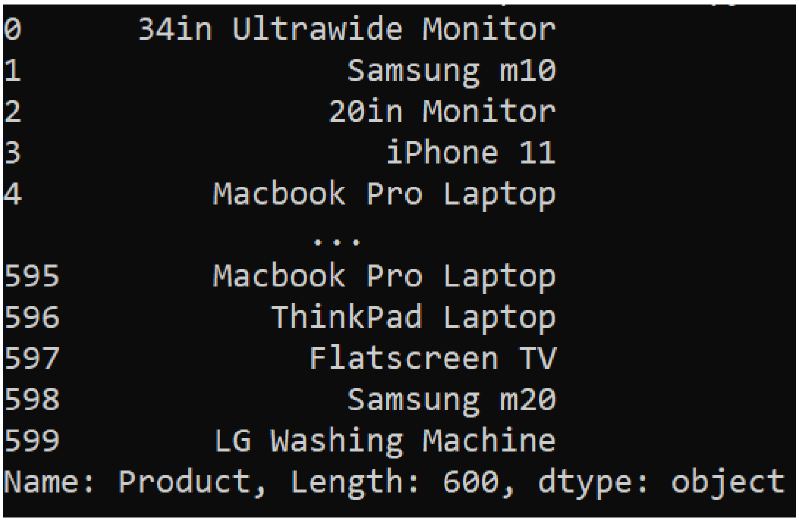
Program Accessing single column from the DataFrame
Name demo10.py
Input file sales1.csv

```
import pandas as pd

df = pd.read_csv("sales1.csv")

print(df["Product"])
```

Output



```
0      34in Ultrawide Monitor
1                Samsung m10
2             20in Monitor
3                  iPhone 11
4         Macbook Pro Laptop
                   ...
595       Macbook Pro Laptop
596          ThinkPad Laptop
597            Flatscreen TV
598              Samsung m20
599       LG Washing Machine
Name: Product, Length: 600, dtype: object
```

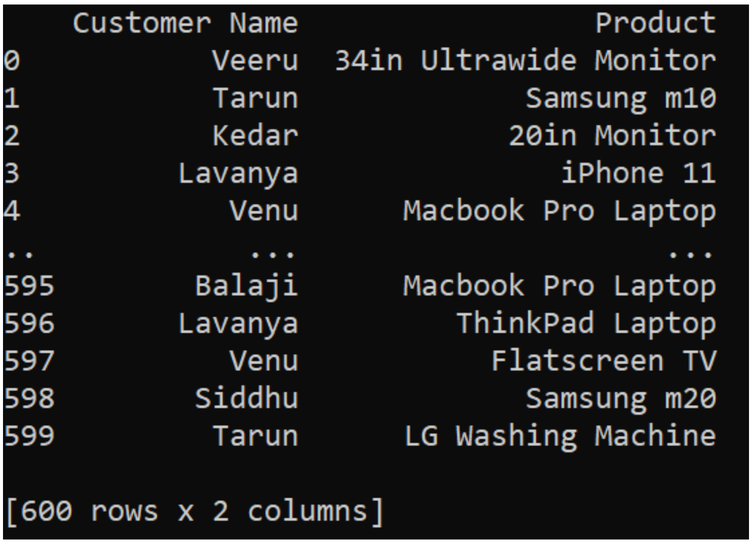
Program Accessing two column from the DataFrame
Name demo11.py
Input file sales1.csv

```
import pandas as pd

df = pd.read_csv("sales1.csv")

print(df[["Customer Name", "Product"]])
```

Output



```
    Customer Name                 Product
0           Veeru  34in Ultrawide Monitor
1           Tarun             Samsung m10
2           Kedar            20in Monitor
3         Lavanya               iPhone 11
4            Venu      Macbook Pro Laptop
..            ...                     ...
595        Balaji      Macbook Pro Laptop
596       Lavanya         ThinkPad Laptop
597          Venu           Flatscreen TV
598        Siddhu             Samsung m20
599         Tarun      LG Washing Machine

[600 rows x 2 columns]
```

Program Accessing single column from the DataFrame, applying sum
Name demo12.py
Input file sales1.csv

```
import pandas as pd

df = pd.read_csv("sales1.csv")
total = df['Quantity'].sum()

print(total)
```

Output

```
889
```

3. Rearranging columns in DataFrame

- ✓ We can rearrange columns in DataFrame
- ✓ We can customise the order of columns in DataFrame

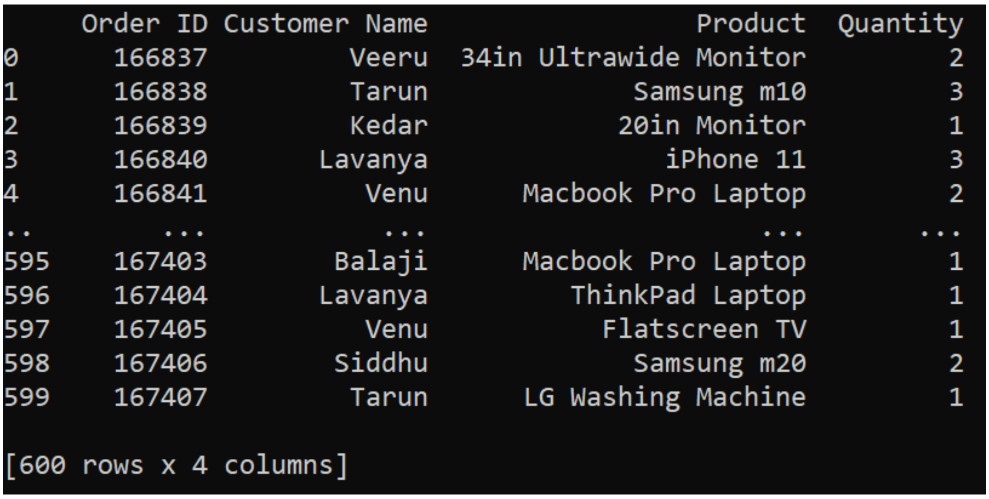
Program	Creating DataFrame by loading csv file
Name	demo13.py
Input file	sales1.csv

```
import pandas as pd

df = pd.read_csv("sales1.csv")

print(df)
```

Output



```
     Order ID Customer Name                 Product  Quantity
0      166837         Veeru  34in Ultrawide Monitor         2
1      166838         Tarun             Samsung m10         3
2      166839         Kedar            20in Monitor         1
3      166840       Lavanya               iPhone 11         3
4      166841          Venu      Macbook Pro Laptop         2
..        ...           ...                     ...       ...
595    167403        Balaji      Macbook Pro Laptop         1
596    167404       Lavanya         ThinkPad Laptop         1
597    167405          Venu           Flatscreen TV         1
598    167406        Siddhu             Samsung m20         2
599    167407         Tarun      LG Washing Machine         1

[600 rows x 4 columns]
```

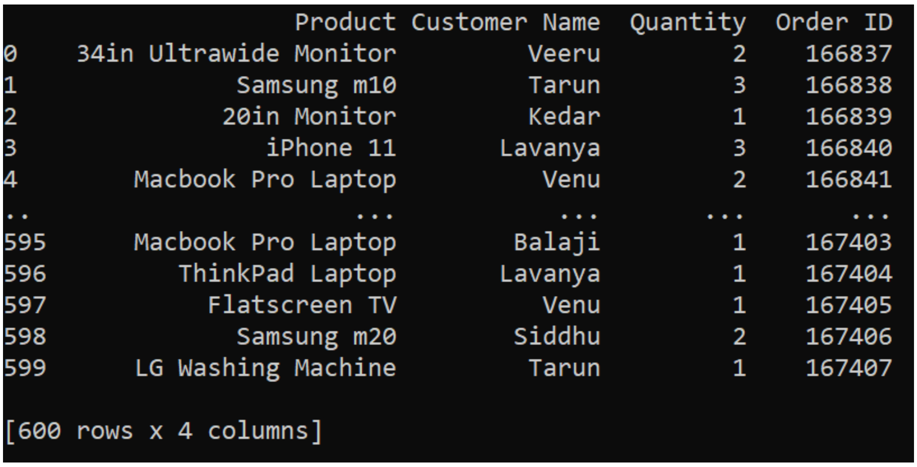
Program Name Rearranging columns in DataFrame
demo14.py
Input file sales1.csv

```
import pandas as pd

df = pd.read_csv("sales1.csv")
df = df[["Product", "Customer Name", "Quantity", "Order ID"]]

print(df)
```

Output



```
                 Product Customer Name  Quantity  Order ID
0    34in Ultrawide Monitor         Veeru         2    166837
1              Samsung m10         Tarun         3    166838
2             20in Monitor         Kedar         1    166839
3                iPhone 11       Lavanya         3    166840
4        Macbook Pro Laptop          Venu         2    166841
..                     ...           ...       ...       ...
595      Macbook Pro Laptop        Balaji         1    167403
596         ThinkPad Laptop       Lavanya         1    167404
597           Flatscreen TV          Venu         1    167405
598             Samsung m20        Siddhu         2    167406
599      LG Washing Machine         Tarun         1    167407

[600 rows x 4 columns]
```