Shopsales

April 29, 2025

[1]:
```
import pandas as pd
```

[2]:
```
df=pd.read_csv("storedataset.csv")
```

[3]:
```
df.head()
```

[3]:
```
   Row ID        Order ID  Order Date   Ship Date       Ship Mode Customer ID  \
0       1  CA-2017-152156  08/11/2017  11/11/2017    Second Class    CG-12520
1       2  CA-2017-152156  08/11/2017  11/11/2017    Second Class    CG-12520
2       3  CA-2017-138688  12/06/2017  16/06/2017    Second Class    DV-13045
3       4  US-2016-108966  11/10/2016  18/10/2016  Standard Class    SO-20335
4       5  US-2016-108966  11/10/2016  18/10/2016  Standard Class    SO-20335

     Customer Name    Segment        Country             City       State  \
0      Claire Gute   Consumer  United States        Henderson    Kentucky
1      Claire Gute   Consumer  United States        Henderson    Kentucky
2  Darrin Van Huff  Corporate  United States      Los Angeles  California
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale     Florida
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale     Florida

   Postal Code Region       Product ID         Category Sub-Category  \
0      42420.0  South  FUR-BO-10001798        Furniture    Bookcases
1      42420.0  South  FUR-CH-10000454        Furniture       Chairs
2      90036.0   West  OFF-LA-10000240  Office Supplies       Labels
3      33311.0  South  FUR-TA-10000577        Furniture       Tables
4      33311.0  South  OFF-ST-10000760  Office Supplies      Storage

                                        Product Name     Sales
0                  Bush Somerset Collection Bookcase  261.9600
1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400
2  Self-Adhesive Address Labels for Typewriters b…   14.6200
3      Bretford CR4500 Series Slim Rectangular Table  957.5775
4                     Eldon Fold 'N Roll Cart System   22.3680
```

[4]:
```
df.shape
```

[4]:
```
(9800, 18)
```

```
[ ]: Find out all the null value in the data
```

```
[5]: df.isnull().sum()
```

```
[5]: Row ID             0
     Order ID           0
     Order Date         0
     Ship Date          0
     Ship Mode          0
     Customer ID        0
     Customer Name      0
     Segment            0
     Country            0
     City               0
     State              0
     Postal Code       11
     Region             0
     Product ID         0
     Category           0
     Sub-Category       0
     Product Name       0
     Sales              0
     dtype: int64
```

```
[24]: df['Postal Code'].fillna('no exist', inplace=True)
```

```
[7]: df.columns
```

```
[7]: Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
            'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
            'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
            'Product Name', 'Sales'],
           dtype='object')
```

1.Time analysis: Sales trends over time: How have sales changed monthly, weekly, or annually?

```
[29]: df['Order Date'].unique
```

```
[29]: <bound method Series.unique of 0       2017-11-08
      1       2017-11-08
      2       2017-06-12
      3       2016-10-11
      4       2016-10-11
                 …
      9795    2017-05-21
      9796    2016-01-12
      9797    2016-01-12
      9798    2016-01-12
```

```
9799   2016-01-12
Name: Order Date, Length: 9800, dtype: datetime64[ns]>
```

pd.to_datetime():This function automatically detects different date formats and converts them to datetime type.

```
[9]: df['Order Date'] = pd.to_datetime(df['Order Date'], dayfirst=True)
```

```
[15]: df['Year'] = df['Order Date'].dt.year
      df['Month'] = df['Order Date'].dt.month
      df['YearMonth'] = df['Order Date'].dt.to_period('M')
```

```
[16]: df.head(2)
```

```
[16]:    Row ID        Order ID Order Date   Ship Date     Ship Mode Customer ID  \
      0       1  CA-2017-152156 2017-11-08  11/11/2017  Second Class    CG-12520
      1       2  CA-2017-152156 2017-11-08  11/11/2017  Second Class    CG-12520

        Customer Name   Segment        Country       City  … Postal Code  Region  \
      0   Claire Gute  Consumer  United States  Henderson  …     42420.0   South
      1   Claire Gute  Consumer  United States  Henderson  …     42420.0   South

              Product ID   Category Sub-Category  \
      0  FUR-BO-10001798  Furniture    Bookcases
      1  FUR-CH-10000454  Furniture       Chairs

                                              Product Name   Sales  Year  Month  \
      0                  Bush Somerset Collection Bookcase  261.96  2017     11
      1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.94  2017     11

        YearMonth
      0   2017-11
      1   2017-11

      [2 rows x 21 columns]
```

```
[17]: monthly_sales = df.groupby('YearMonth')['Sales'].sum()
```

```
[18]: monthly_sales
```

```
[18]: YearMonth
      2015-01     14205.7070
      2015-02      4519.8920
      2015-03     55205.7970
      2015-04     27906.8550
      2015-05     23644.3030
      2015-06     34322.9356
      2015-07     33781.5430
```

```
2015-08     27117.5365
2015-09     81623.5268
2015-10     31453.3930
2015-11     77907.6607
2015-12     68167.0585
2016-01     18066.9576
2016-02     11951.4110
2016-03     32339.3184
2016-04     34154.4685
2016-05     29959.5305
2016-06     23599.3740
2016-07     28608.2590
2016-08     36818.3422
2016-09     63133.6060
2016-10     31011.7375
2016-11     75249.3995
2016-12     74543.6012
2017-01     18542.4910
2017-02     22978.8150
2017-03     51165.0590
2017-04     38679.7670
2017-05     56656.9080
2017-06     39724.4860
2017-07     38320.7830
2017-08     30542.2003
2017-09     69193.3909
2017-10     59583.0330
2017-11     79066.4958
2017-12     95739.1210
2018-01     43476.4740
2018-02     19920.9974
2018-03     58863.4128
2018-04     35541.9101
2018-05     43825.9822
2018-06     48190.7277
2018-07     44825.1040
2018-08     62837.8480
2018-09     86152.8880
2018-10     77448.1312
2018-11    117938.1550
2018-12     83030.3888
Freq: M, Name: Sales, dtype: float64
```

[19]:
```
yearly_sales = df.groupby('Year')['Sales'].sum()
```

[20]:
```
yearly_sales
```

[20]:
```
Year
2015    479856.2081
2016    459436.0054
2017    600192.5500
2018    722052.0192
Name: Sales, dtype: float64
```

[21]:
```
monthly_avg = df.groupby('Month')['Sales'].mean()
```

[22]:
```
monthly_avg
```

[22]:
```
Month
1     257.627403
2     199.902745
3     290.549393
4     207.432269
5     212.533412
6     211.052856
7     208.802997
8     227.007110
9     221.642106
10    246.596162
11    241.657496
12    232.619515
Name: Sales, dtype: float64
```