

```
In [10]: import pandas as pd
```

```
In [11]: dataset=pd.read_csv("RetailDataset.csv")
dataset
```

Out[11]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount	Co
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17	1153
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29	499
2	NaN	Electronics	Cust_205	UK	22-10-2024	8	1470	0.22	1040
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08	753
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01	637
...	...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08	82
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13	318
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21	623
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17	479
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17	1091

5000 rows × 10 columns

```
In [12]: dataset.head(10)
```

Out[12]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount	Cost
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17	1153.
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29	499.
2	Nan	Electronics	Cust_205	UK	22-10-2024	8	1470	0.22	1040.
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08	753.
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01	637.
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25	633.
6	Tablet	Accessories	Cust_457	USA	20-12-2024	6	118	0.26	241.
7	Mouse	Office	Cust_193	Germany	15-06-2022	2	1703	0.27	877.
8	Phone	Electronics	Cust_194	UK	21-06-2022	4	932	0.15	1298.
9	Headphones	Electronics	Cust_464	Germany	16-12-2024	2	248	0.10	242.

In [13]: `dataset.tail(10)`

Out[13]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
4990	Headphones	Electronics	Cust_299	Germany	05-05-2022	8	1271	0.14
4991	Phone	Office	Cust_122	USA	27-01-2023	1	1052	0.08
4992	Laptop	Accessories	Cust_291	Germany	29-05-2022	6	446	0.10 1
4993	Printer	Electronics	Cust_423	Canada	20-04-2022	3	1080	0.26
4994	Phone	Accessories	Cust_241	Australia	24-05-2023	7	1221	0.25
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 1

In [14]: `dataset.describe(include='all')`

Out[14]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Cost	Sales
<b>count</b>	4904	5000	5000	5000	5000	5000.000000	5000.000000	4900	5000
<b>unique</b>	8	3	500	6	1085	NaN	NaN	NaN	NaN
<b>top</b>	Printer	Accessories	Cust_480	Germany	27-01-2023	NaN	NaN	NaN	NaN
<b>freq</b>	644	1692	19	878	15	NaN	NaN	NaN	NaN
<b>mean</b>	NaN	NaN	NaN	NaN	NaN	4.954800	1025.086200	0	0
<b>std</b>	NaN	NaN	NaN	NaN	NaN	2.554068	560.293526	0	0
<b>min</b>	NaN	NaN	NaN	NaN	NaN	1.000000	50.000000	0	0
<b>25%</b>	NaN	NaN	NaN	NaN	NaN	3.000000	535.000000	0	0
<b>50%</b>	NaN	NaN	NaN	NaN	NaN	5.000000	1014.000000	0	0
<b>75%</b>	NaN	NaN	NaN	NaN	NaN	7.000000	1510.250000	0	0
<b>max</b>	NaN	NaN	NaN	NaN	NaN	9.000000	1999.000000	0	0

In [15]: `dataset.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Product     4904 non-null   object 
 1   Category    5000 non-null   object 
 2   Customer    5000 non-null   object 
 3   Region      5000 non-null   object 
 4   Date        5000 non-null   object 
 5   Quantity    5000 non-null   int64  
 6   Unit Price  5000 non-null   int64  
 7   Discount    4900 non-null   float64
 8   Cost        4898 non-null   float64
 9   Sales       5000 non-null   float64
dtypes: float64(3), int64(2), object(5)
memory usage: 390.8+ KB
```

In [16]: `dataset.shape`

Out[16]: (5000, 10)

In [17]: `dataset.isnull().sum()`

```
Out[17]: Product      96  
Category        0  
Customer        0  
Region          0  
Date            0  
Quantity         0  
Unit Price      0  
Discount        100  
Cost            102  
Sales           0  
dtype: int64
```

```
In [18]: dataset=dataset.dropna(subset=['Product','Cost'])  
dataset
```

Out[18]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...								
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4803 rows × 10 columns

In [19]: `dataset.isnull().sum()`

```
Out[19]: Product      0  
Category      0  
Customer      0  
Region        0  
Date          0  
Quantity       0  
Unit Price    0  
Discount      95  
Cost          0  
Sales          0  
dtype: int64
```

```
In [20]: # Fill Discount null with 0  
dataset['Discount'] = dataset['Discount'].fillna(0)  
dataset['Discount']
```

C:\Users\Tirupati Kundagir\AppData\Local\Temp\ipykernel\_3912\1472741552.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
dataset['Discount'] = dataset['Discount'].fillna(0)

```
Out[20]: 0      0.17  
1      0.29  
3      0.08  
4      0.01  
5      0.25  
...  
4995    0.08  
4996    0.13  
4997    0.21  
4998    0.17  
4999    0.17  
Name: Discount, Length: 4803, dtype: float64
```

```
In [21]: dataset.isnull().sum()
```

```
Out[21]: Product      0  
Category      0  
Customer      0  
Region        0  
Date          0  
Quantity       0  
Unit Price    0  
Discount      0  
Cost          0  
Sales          0  
dtype: int64
```

```
In [ ]:
```

```
In [22]: dataset.duplicated().sum()
```

```
Out[22]: np.int64(90)
```

```
In [23]: # it actually shows the duplicate values  
dataset[dataset.duplicated()]
```

Out[23]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
478	Headphones	Electronics	Cust_232	Canada	30-10-2024	3	1578	0.09
612	Keyboard	Office	Cust_9	Australia	09-01-2023	8	1679	0.05 1
843	Mouse	Office	Cust_393	Canada	03-01-2024	9	926	0.12 .
927	Printer	Electronics	Cust_140	Australia	31-05-2023	6	823	0.30 1
986	Keyboard	Electronics	Cust_366	UK	24-07-2023	3	365	0.27
...	...	...	...	...	...	...	...	...
4890	Headphones	Office	Cust_452	India	09-04-2023	4	301	0.13 1.
4896	Printer	Accessories	Cust_395	UK	14-07-2024	4	440	0.28 1
4925	Tablet	Electronics	Cust_249	India	05-12-2022	8	864	0.08 .
4935	Keyboard	Office	Cust_287	Canada	26-12-2024	7	943	0.04 .
4985	Headphones	Office	Cust_154	Germany	19-01-2022	1	1518	0.07 1

90 rows × 10 columns



In [24]:

```
dataset = dataset.drop_duplicates()
dataset
```

Out[24]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...								
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 10 columns



In [ ]:

In [25]: `dataset.shape`

Out[25]: (4713, 10)

In [26]: `dataset['Customer'].value_counts()`

```
Out[26]: Customer
Cust_281    18
Cust_271    17
Cust_235    17
Cust_480    17
Cust_77     16
...
Cust_124    4
Cust_117    3
Cust_97     3
Cust_36     3
Cust_14     2
Name: count, Length: 500, dtype: int64
```

```
In [27]: dataset
```

Out[27]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 10 columns

In [28]:

```
dataset['Profit']=dataset['Sales']- dataset['Cost']
dataset['Profit']
```

C:\Users\Tirupati Kundagir\AppData\Local\Temp\ipykernel\_3912\767533865.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
dataset['Profit']=dataset['Sales']- dataset['Cost']
```

```
Out[28]: 0      3714.12
         1      5175.32
         3      231.40
         4      206.48
         5     8559.75
         ...
        4995    4779.28
        4996    4005.90
        4997    4439.32
        4998    6447.35
        4999    5217.00
Name: Profit, Length: 4713, dtype: float64
```

```
In [29]: dataset
```

Out[29]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 11 columns

In [30]: `dataset["Profit Margin"]=(dataset["Profit"]/ dataset["Sales"])*100  
dataset["Profit Margin"]`

C:\Users\Tirupati Kundagir\AppData\Local\Temp\ipykernel\_3912\582804610.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

`dataset["Profit Margin"]=(dataset["Profit"]/ dataset["Sales"])*100`

```
Out[30]: 0      76.310426
1      91.205995
3      23.506705
4      24.479537
5      93.114139
...
4995   98.313201
4996   92.645528
4997   87.693390
4998   93.084381
4999   82.704502
Name: Profit Margin, Length: 4713, dtype: float64
```

```
In [31]: dataset
```

Out[31]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 12 columns

In [32]:

```
dataset["Total_Discount"] = dataset["Quantity"] * dataset["Unit Price"] * dataset["Discount"]
dataset["Total_Discount"]
```

C:\Users\Tirupati Kundagir\AppData\Local\Temp\ipykernel\_3912\2863222932.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
dataset["Total\_Discount"] = dataset["Quantity"] \* dataset["Unit Price"] \* dataset["Discount"]

```
Out[32]: 0      996.88
         1     2317.68
         3      85.60
         4      8.52
         5    3064.25
         ...
        4995   422.72
        4996   646.10
        4997  1345.68
        4998  1418.65
        4999  1292.00
Name: Total_Discount, Length: 4713, dtype: float64
```

```
In [33]: dataset
```

Out[33]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 13 columns

In [34]:

```
# Average revenue(Sales) of the each product
avg_revenue = dataset.groupby('Product')['Sales'].mean()
avg_revenue
```

```
Out[34]: Product
Headphones    4532.157053
Keyboard      4640.283678
Laptop         5037.781142
Monitor        4832.701466
Mouse          5114.606324
Phone          4717.600565
Printer        4332.108685
Tablet         4750.433643
Name: Sales, dtype: float64
```

```
In [35]: dataset
```

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount	
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17	1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29	4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08	7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01	6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25	6%
...	...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08	8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13	3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21	6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17	4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17	10%

4713 rows × 13 columns

```
In [36]: # Total revenue of the each product  
Total_Revenue=dataset.groupby("Product")["Sales"].sum()  
Total_Revenue
```

```
Out[36]: Product  
Headphones      2583329.52  
Keyboard        2737767.37  
Laptop           3088159.84  
Monitor          2967278.70  
Mouse            2838606.51  
Phone             2755078.73  
Printer          2668578.95  
Tablet            2712497.61  
Name: Sales, dtype: float64
```

```
In [37]: # Total revenue and profit of the each Region  
Region_wise=dataset.groupby("Region")[["Sales","Profit"]].sum()  
Region_wise
```

```
Out[37]:
```

	Sales	Profit
Region		
<b>Australia</b>	3650380.00	3054088.00
<b>Canada</b>	3555734.70	2926495.70
<b>Germany</b>	4030712.96	3397016.96
<b>India</b>	3834840.28	3257565.28
<b>UK</b>	3547774.58	2980757.58
<b>USA</b>	3731854.71	3125402.71

```
In [38]: dataset
```

Out[38]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...								
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 13 columns



In [39]:

```
category_qty = dataset.groupby('Category')['Quantity'].sum()
category_qty
```

Out[39]:

```
Category
Accessories    7884
Electronics    7789
Office         7639
Name: Quantity, dtype: int64
```

In [ ]:

```
In [40]: high_margin = dataset[dataset['Profit Margin'] > 20]
high_margin
low_margin = dataset[dataset['Profit Margin'] < 10]
low_margin
```

Out[40]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount	Cost
15	Tablet	Office	Cust_29	UK	14-09-2023	2	63	0.08	938.1
16	Laptop	Electronics	Cust_289	USA	11-03-2023	2	674	0.29	1463.1
18	Phone	Accessories	Cust_382	Australia	16-05-2022	7	178	0.01	1151.1
30	Phone	Office	Cust_175	Canada	12-12-2022	3	157	0.24	539.1
48	Mouse	Electronics	Cust_369	UK	25-05-2023	1	54	0.08	235.1
...	...	...	...	...	...	...	...	...	...
4960	Laptop	Office	Cust_10	Germany	22-08-2023	3	60	0.05	1022.1
4971	Laptop	Electronics	Cust_279	UK	23-10-2022	2	406	0.11	654.1
4972	Tablet	Accessories	Cust_189	UK	13-12-2023	3	310	0.08	863.1
4973	Laptop	Accessories	Cust_16	Canada	09-01-2022	2	647	0.21	1304.1
4988	Monitor	Accessories	Cust_393	USA	06-08-2022	6	317	0.21	1388.1

663 rows × 13 columns



```
In [41]: dataset
```

Out[41]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

4713 rows × 13 columns

In [42]:

```
from sqlalchemy import create_engine
engine= create_engine(
    "mssql+pyodbc://@TIRUPATI\\SQLEXPRESS/Retail?driver=ODBC+Driver+18+for+SQL+SQL")
dataset.to_sql("Retail_Data", engine, if_exists="replace", index=False)
print("successfully data Loaded")
```

successfully data Loaded

In [43]:

dataset

Out[43]:

	Product	Category	Customer	Region	Date	Quantity	Unit Price	Discount
0	Printer	Electronics	Cust_440	Canada	18-03-2023	8	733	0.17 1%
1	Printer	Office	Cust_480	Canada	01-08-2024	4	1998	0.29 4%
3	Monitor	Electronics	Cust_336	UK	21-08-2023	5	214	0.08 7%
4	Monitor	Accessories	Cust_440	Canada	17-10-2022	4	213	0.01 6%
5	Headphones	Accessories	Cust_138	Canada	10-01-2023	7	1751	0.25 6%
...	...	...	...	...	...	...	...	...
4995	Keyboard	Accessories	Cust_336	Canada	23-10-2024	4	1321	0.08 8%
4996	Mouse	Electronics	Cust_420	USA	12-01-2022	5	994	0.13 3%
4997	Tablet	Office	Cust_265	Australia	01-05-2024	9	712	0.21 6%
4998	Keyboard	Electronics	Cust_84	Australia	18-02-2023	5	1669	0.17 4%
4999	Monitor	Electronics	Cust_51	UK	03-03-2024	8	950	0.17 10%

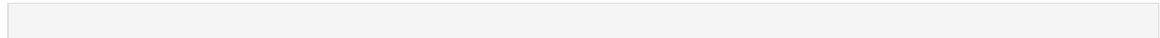
4713 rows × 13 columns



In [44]: `dataset.shape`

Out[44]: (4713, 13)

In [ ]:



In [ ]:

