

import the data

```
In [8]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

In [9]: df=pd.read_excel("C:/Users/PC/Documents/Course/Adida/Adida US Sales Datasets.xlsx")
```

```
In [10]: df.head()
```

	Unnamed: 0	Unnamed: 1	Unnamed: 2		Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6		Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
0	NaN	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price per Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
4	NaN	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	600000	300000	0.5	In-store

```
In [11]: df.drop(["Unnamed: 0"],axis=1,inplace=True)

In [12]: df.head()
```

	Unnamed: 1	Unnamed: 2		Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6		Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
0	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price per Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	600000	300000	0.5	In-store

```
In [13]: df.columns = [
    "Retailer",
    "Retailer ID",
    "Invoice Date",
    "Region",
    "State",
    "City",
    "Product",
    "Price_per_Unit",
    "Units Sold",
    "Total Sales",
    "Operating Profit",
    "Operating Margin",
    "Sales Method"
]

In [14]: df.head()
```

	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price_per_Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
0	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price per Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	600000	300000	0.5	In-store

```
In [18]: df.drop([0,1,2,3],inplace=True)

In [19]: df.head()
```

	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price_per_Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	600000	300000	0.5	In-store
5	Foot Locker	1185732		2020-01-02 00:00:00	Northeast	New York	New York		Men's Athletic Footwear	50	1000	500000	150000	0.3	In-store
6	Foot Locker	1185732		2020-01-03 00:00:00	Northeast	New York	New York		Women's Street Footwear	40	1000	400000	140000	0.35	In-store
7	Foot Locker	1185732		2020-01-04 00:00:00	Northeast	New York	New York		Women's Athletic Footwear	45	850	382500	133875	0.35	In-store
8	Foot Locker	1185732		2020-01-05 00:00:00	Northeast	New York	New York		Men's Apparel	60	900	540000	162000	0.3	In-store

```
In [10]: df["Operating margin"]*df["Operating Margin"]*100

In [11]: df.drop(["Operating Profit","Total Sales","Operating Margin"],axis=1,inplace=True)

In [12]: df.head()
```

	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price_per_Unit	Units Sold	Sales Method	%Operating margin
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	In-store	50.0
5	Foot Locker	1185732		2020-01-02 00:00:00	Northeast	New York	New York		Men's Athletic Footwear	50	1000	In-store	30.0
6	Foot Locker	1185732		2020-01-03 00:00:00	Northeast	New York	New York		Women's Street Footwear	40	1000	In-store	35.0
7	Foot Locker	1185732		2020-01-04 00:00:00	Northeast	New York	New York		Women's Athletic Footwear	45	850	In-store	35.0
8	Foot Locker	1185732		2020-01-05 00:00:00	Northeast	New York	New York		Men's Apparel	60	900	In-store	30.0

```
In [13]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9648 entries, 4 to 9651
Data columns (total 11 columns):
# Column Non-Null Count Dtype
---
0 Retailer 9648 non-null object
1 Retailer ID 9648 non-null object
2 Invoice Date 9648 non-null object
3 Region 9648 non-null object
4 State 9648 non-null object
5 City 9648 non-null object
6 Product 9648 non-null object
7 Price_per_Unit 9648 non-null int64
8 Units Sold 9648 non-null int64
9 Sales Method 9648 non-null object
10 %Operating margin 9648 non-null object
dtypes: object(11)
memory usage: 829.3+ KB
```

```
In [14]: df = df.astype({
    "%Operating margin":"int64",
    "Units Sold":"int64",
    "Price_per_Unit":"int64"
})

In [15]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9648 entries, 4 to 9651
Data columns (total 11 columns):
# Column Non-Null Count Dtype
---
0 Retailer 9648 non-null object
1 Retailer ID 9648 non-null object
2 Invoice Date 9648 non-null object
3 Region 9648 non-null object
4 State 9648 non-null object
5 City 9648 non-null object
6 Product 9648 non-null object
7 Price_per_Unit 9648 non-null int64
8 Units Sold 9648 non-null int64
9 Sales Method 9648 non-null object
10 %Operating margin 9648 non-null int64
dtypes: int64(11)
memory usage: 829.3+ KB
```

```
In [16]: df["Total Sale"]*df["Units Sold"]*df["Price_per_Unit"]

In [17]: df["Operating profit"] = (df["%Operating margin"] / 100) * df["Total Sale"]

In [18]: df[["Year", "Month", "Day"]] = df["Invoice Date"].astype(str).str.split("-", expand=True)

In [19]: df.head()
```

	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price_per_Unit	Units Sold	Sales Method	%Operating margin	Total Sale	Operating profit	Year	Month	Day
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	In-store	50	60000	30000.0	2020	01	01 00:00:00
5	Foot Locker	1185732		2020-01-02 00:00:00	Northeast	New York	New York		Men's Athletic Footwear	50	1000	In-store	30	50000	15000.0	2020	01	02 00:00:00
6	Foot Locker	1185732		2020-01-03 00:00:00	Northeast	New York	New York		Women's Street Footwear	40	1000	In-store	35	40000	14000.0	2020	01	03 00:00:00
7	Foot Locker	1185732		2020-01-04 00:00:00	Northeast	New York	New York		Women's Athletic Footwear	45	850	In-store	35	38250	13387.5	2020	01	04 00:00:00
8	Foot Locker	1185732		2020-01-05 00:00:00	Northeast	New York	New York		Men's Apparel	60	900	In-store	30	54000	16200.0	2020	01	05 00:00:00

```
In [20]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9648 entries, 4 to 9651
Data columns (total 16 columns):
# Column Non-Null Count Dtype
---
0 Retailer 9648 non-null object
1 Retailer ID 9648 non-null object
2 Invoice Date 9648 non-null object
3 Region 9648 non-null object
4 State 9648 non-null object
5 City 9648 non-null object
6 Product 9648 non-null object
7 Price_per_Unit 9648 non-null int64
8 Units Sold 9648 non-null int64
9 Sales Method 9648 non-null object
10 %Operating margin 9648 non-null int64
11 Total Sale 9648 non-null int64
12 Operating profit 9648 non-null float64
13 Year 9648 non-null object
14 Month 9648 non-null object
15 Day 9648 non-null object
dtypes: float64(1), int64(8), object(11)
memory usage: 1.2+ MB
```

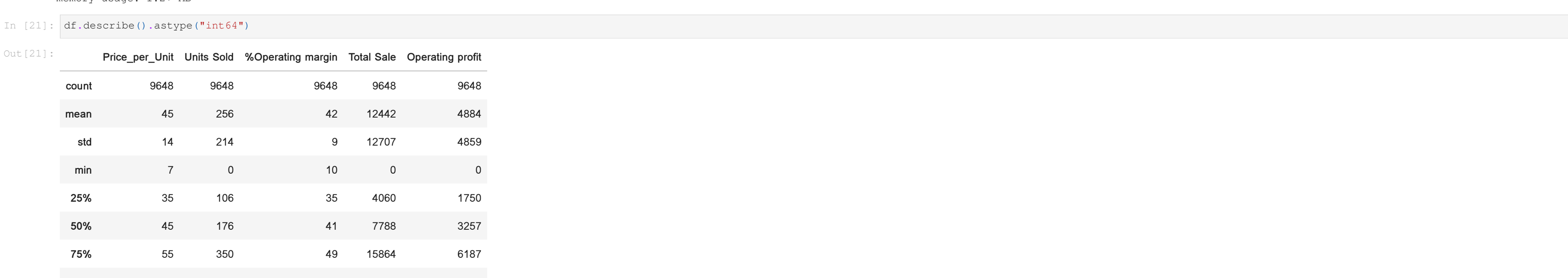
```
In [21]: df.describe().astype("int64")

Out[21]:
```

	Price_per_Unit	Units Sold	%Operating margin	Total Sale	Operating profit
count	9648	9648	9648	9648	9648
mean	45	256	42	12442	4884
std	14	214	9	12707	4859
min	7	0	10	0	0
25%	35	106	35	4060	1750
50%	45	176	41	7788	3257
75%	55	350	49	15864	6187
max	110	1275	80	82500	39000

```
In [22]: num_data=df[["Total Sale","Operating profit","Operating margin"]]
num_data.plot(kind="box")

Out[22]: <Axes: >
```



```
In [31]: df.head()
```

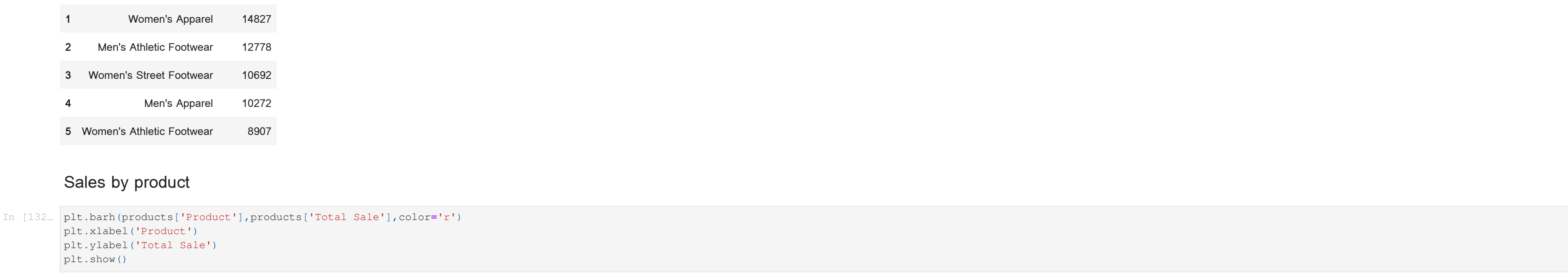
	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price_per_Unit	Units Sold	Sales Method	%Operating margin	Total Sale	Operating profit	Year	Month	Day
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	In-store	50	60000	30000.0	2020	01	01 00:00:00
5	Foot Locker	1185732		2020-01-02 00:00:00	Northeast	New York	New York		Men's Athletic Footwear	50	1000	In-store	30	50000	15000.0	2020	01	02 00:00:00
6	Foot Locker	1185732		2020-01-03 00:00:00	Northeast	New York	New York		Women's Street Footwear	40	1000	In-store	35	40000	14000.0	2020	01	03 00:00:00
7	Foot Locker	1185732		2020-01-04 00:00:00	Northeast	New York	New York		Women's Athletic Footwear	45	850	In-store	35	38250	13387.5	2020	01	04 00:00:00
8	Foot Locker	1185732		2020-01-05 00:00:00	Northeast	New York	New York		Men's Apparel	60	900	In-store	30	54000	16200.0	2020	01	05 00:00:00

```
In [30]: product=df.groupby("Product")[["Total Sale"].mean().sort_values(ascending=False).astype("int64").to_frame().reset_index()
products
```

	Product	Total Sale
0	Men's Street Footwear	17196
1	Women's Apparel	14827
2	Men's Athletic Footwear	12778
3	Women's Street Footwear	10692
4	Men's Apparel	10272
5	Women's Athletic Footwear	8907

Sales by product

```
In [32]: plt.bar(products["Product"],products["Total Sale"],color="r")
plt.xlabel("Product")
plt.ylabel("Total Sale")
plt.show()
```



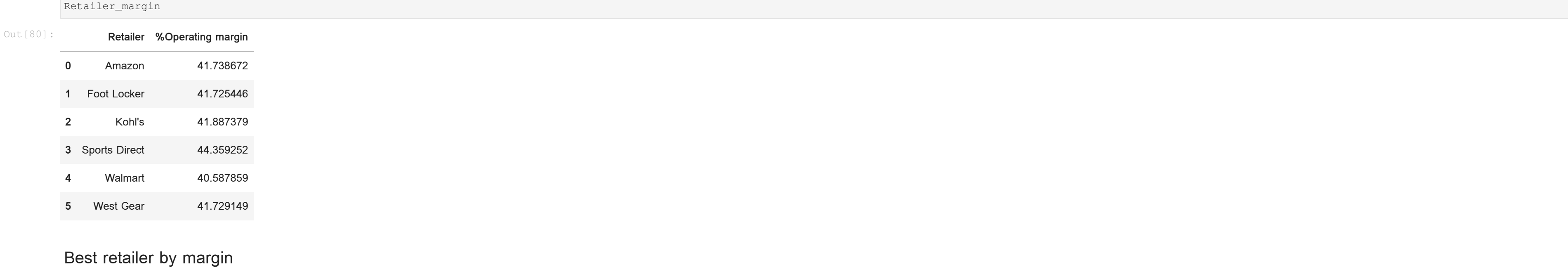
```
In [ ]:
```

```
In [80]: Retailer_margin=df.groupby("Retailer")[["%Operating margin"].mean().to_frame().reset_index()
Retailer_margin
```

	Retailer	%Operating margin
0	Amazon	41.738672
1	Foot Locker	41.725445
2	Kohl's	41.887379
3	Sports Direct	44.369252
4	Walmart	40.587859
5	West Gear	41.729149

Best retailer by margin

```
In [33]: plt.pie(Retailer_margin["Operating margin"],label=Retailer_margin["Retailer"],autopct="%1.1f%%")
plt.show()
```

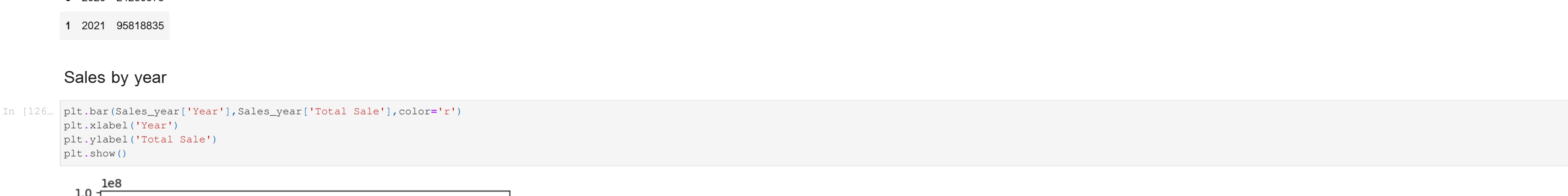


```
In [119]: Sales_year=df.groupby("Year")[["Total Sale"].sum().to_frame().reset_index()
Sales_year
```

	Year	Total Sale
0	2020	24230975
1	2021	95818835

Sales by year

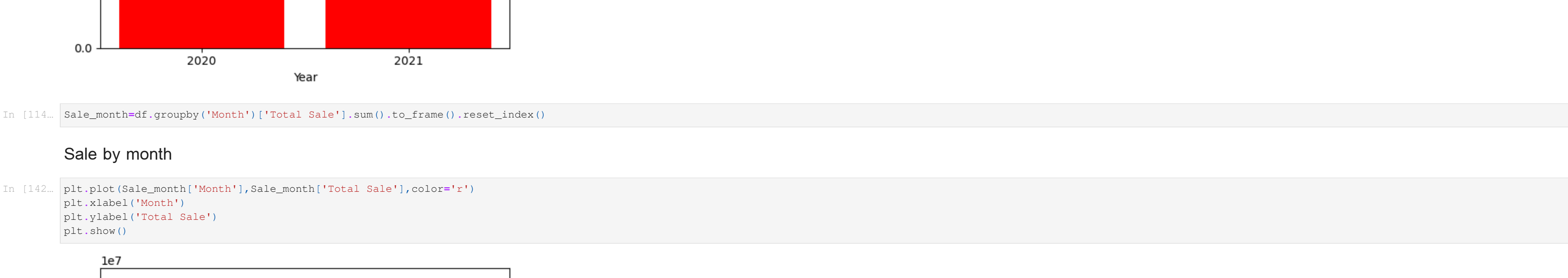
```
In [126]: plt.bar(Sales_year["Year"],Sales_year["Total Sale"],color="r")
plt.xlabel("Year")
plt.ylabel("Total Sale")
plt.show()
```



```
In [114]: Sale_month=df.groupby("Month")[["Total Sale"].sum().to_frame().reset_index()
```

Sale by month

```
In [112]: plt.plot(Sale_month["Month"],Sale_month["Total Sale"],color="r")
plt.xlabel("Month")
plt.ylabel("Total Sale")
plt.show()
```



```
In [133]: df.head()
```

	Retailer	Retailer ID		Invoice Date	Region	State	City		Product	Price_per_Unit	Units Sold	Sales Method	%Operating margin	Total Sale	Operating profit	Year	Month	Day
4	Foot Locker	1185732		2020-01-01 00:00:00	Northeast	New York	New York		Men's Street Footwear	50	1200	In-store	50	60000	30000.0	2020	01	01 00:00:00
5	Foot Locker	1185732		2020-01-02 00:00:00	Northeast	New York	New York		Men's Athletic Footwear	50	1000	In-store	30	50000	15000.0	2020	01	02 00:00:00
6	Foot Locker	1185732		2020-01-03 00:00:00	Northeast	New York	New York		Women's Street Footwear	40	1000	In-store	35	40000	14000.0	2020	01	03 00:00:00
7	Foot Locker	1185732		2020-01-04 00:00:00	Northeast	New York	New York		Women's Athletic Footwear	45	850	In-store	35	38250	13387.5	2020	01	04 00:00:00
8	Foot Locker	1185732		2020-01-05 00:00:00	Northeast	New York	New York		Men's Apparel	60	900	In-store	30	54000	16200.0	2020	01	05 00:00:00

```
In [136]: Sale_method=df.groupby("Sales Method")[["Operating profit"].mean().astype("int64").to_frame().reset_index()
Sale_method
```

	Sales Method	Operating profit
0	In-store	7319
1	Online	3992
2	Outlet	4925

profit by Sale_method

```
In [140]: plt.pie(Sale_method["Operating profit"],label=Sale_method["Sales Method"],autopct="%1.1f%%",startangle=90)
plt.show()
```



```
In [149]: num=df[["Operating profit","%Operating margin","Total Sale"]]
correlation=num.corr()
```

Correlation analysis

```
In [154]: sns.heatmap(correlation,annot=True,cmap="Blues")

Out[154]: <Axes: >
```

