import libraries

In [22]:
```
import pandas as pd
```

import the global super store sales Dataset

In [24]:
```
df = pd.read_excel('Global_Superstore_dataset.xlsx')
df
```

Out[24]:

	Row ID	Order_ID	order_date	Ship_date	Ship_ode	Customer_ID	Customer_name	Segment	City	State	...	P
0	32298	CA-2012-124891	31-07-2012	31-07-2012	Same Day	RH-19495	Rick Hansen	Consumer	New York City	New York	...	
1	26341	IN-2013-77878	2013-05-02 00:00:00	2013-07-02 00:00:00	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New South Wales	...	
2	25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter	Consumer	Brisbane	Queensland	...	
3	13524	ES-2013-1579342	28-01-2013	30-01-2013	First Class	KM-16375	Katherine Murray	Home Office	Berlin	Berlin	...	
4	47221	SG-2013-4320	2013-05-11 00:00:00	2013-06-11 00:00:00	Same Day	RH-9495	Rick Hansen	Consumer	Dakar	Dakar	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
51285	29002	IN-2014-62366	19-06-2014	19-06-2014	Same Day	KE-16420	Katrina Edelman	Corporate	Kure	Hiroshima	...	
51286	35398	US-2014-102288	20-06-2014	24-06-2014	Standard Class	ZC-21910	Zuschuss Carroll	Consumer	Houston	Texas	...	
51287	40470	US-2013-155768	2013-02-12 00:00:00	2013-02-12 00:00:00	Same Day	LB-16795	Laurel Beltran	Home Office	Oxnard	California	...	
51288	9596	MX-2012-140767	18-02-2012	22-02-2012	Standard Class	RB-19795	Ross Baird	Home Office	Valinhos	São Paulo	...	
51289	6147	MX-2012-134460	22-05-2012	26-05-2012	Second Class	MC-18100	Mick Crebagga	Consumer	Tipitapa	Managua	...	

51290 rows × 23 columns

display top 5 rows

In [21]:
```
df.head()
```

Out[21]:

	Row ID	Order_ID	order_date	Ship_date	Ship_ode	Customer_ID	Customer_name	Segment	City	State	...	Produ
0	32298	CA-2012-124891	31-07-2012	31-07-2012	Same Day	RH-19495	Rick Hansen	Consumer	New York City	New York	...	TE 100
1	26341	IN-2013-77878	2013-05-02 00:00:00	2013-07-02 00:00:00	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New South Wales	...	FU 100
2	25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter	Consumer	Brisbane	Queensland	...	TE 100
3	13524	ES-2013-1579342	28-01-2013	30-01-2013	First Class	KM-16375	Katherine Murray	Home Office	Berlin	Berlin	...	TE 100
4	47221	SG-2013-4320	2013-05-11 00:00:00	2013-06-11 00:00:00	Same Day	RH-9495	Rick Hansen	Consumer	Dakar	Dakar	...	TEC 100

5 rows × 23 columns

display last 5 rows

In [25]:
```
df.tail()
```

Out[25]:

	Row ID	Order_ID	order_date	Ship_date	Ship_ode	Customer_ID	Customer_name	Segment	City	State	...	Produ
51285	29002	IN-2014-62366	19-06-2014	19-06-2014	Same Day	KE-16420	Katrina Edelman	Corporate	Kure	Hiroshima	...	OF 100
51286	35398	US-2014-102288	20-06-2014	24-06-2014	Standard Class	ZC-21910	Zuschuss Carroll	Consumer	Houston	Texas	...	OF 100
51287	40470	US-2013-155768	2013-02-12 00:00:00	2013-02-12 00:00:00	Same Day	LB-16795	Laurel Beltran	Home Office	Oxnard	California	...	OF 100
51288	9596	MX-2012-140767	18-02-2012	22-02-2012	Standard Class	RB-19795	Ross Baird	Home Office	Valinhos	São Paulo	...	OF 100
51289	6147	MX-2012-134460	22-05-2012	26-05-2012	Second Class	MC-18100	Mick Crebagga	Consumer	Tipitapa	Managua	...	OF 100

5 rows × 23 columns

find how many rows and columns are there

In [26]:
```
df.shape
```

Out[26]:
```
(51290, 23)
```

check the data info

In [27]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 23 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          51290 non-null  int64
 1   Order_ID        51290 non-null  object
 2   order_date      51290 non-null  object
 3   Ship_date       51290 non-null  object
 4   Ship_ode        51290 non-null  object
 5   Customer_ID     51290 non-null  object
 6   Customer_name   51290 non-null  object
 7   Segment         51290 non-null  object
 8   City            51290 non-null  object
 9   State           51290 non-null  object
 10  Country         51290 non-null  object
 11  Market          51290 non-null  object
 12  Region          51290 non-null  object
 13  Product_ID      51290 non-null  object
 14  Category        51290 non-null  object
 15  Sub_Category    51290 non-null  object
 16  Product_name    51290 non-null  object
 17  Sales           51290 non-null  float64
 18  Quantity        51290 non-null  int64
 19  Discount        51290 non-null  float64
 20  Profit          51290 non-null  float64
 21  Shipping_cost   51290 non-null  float64
 22  Order_priority  51290 non-null  object
dtypes: float64(4), int64(2), object(17)
memory usage: 9.0+ MB
```

we covert the order date , ship date in to date format

In [29]:
```
df['order_date'] = pd.to_datetime(df['order_date'], dayfirst=True)
df['Ship_date'] = pd.to_datetime(df['Ship_date'], dayfirst=True)
```

I found in this dataset one oulier which is present in profit column because it have contain negative values so i removed it

In [42]:
```
df = df[df['Profit'] >= 0]
df.reset_index(drop=True, inplace=True)
```

In [43]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38746 entries, 0 to 38745
Data columns (total 23 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          38746 non-null  int64
 1   Order_ID        38746 non-null  object
 2   order_date      38746 non-null  datetime64[ns]
 3   Ship_date       38746 non-null  datetime64[ns]
 4   Ship_ode        38746 non-null  object
 5   Customer_ID     38746 non-null  object
 6   Customer_name   38746 non-null  object
 7   Segment         38746 non-null  object
 8   City            38746 non-null  object
 9   State           38746 non-null  object
 10  Country         38746 non-null  object
 11  Market          38746 non-null  object
 12  Region          38746 non-null  object
 13  Product_ID      38746 non-null  object
 14  Category        38746 non-null  object
 15  Sub_Category    38746 non-null  object
 16  Product_name    38746 non-null  object
 17  Sales           38746 non-null  float64
 18  Quantity        38746 non-null  int64
 19  Discount        38746 non-null  float64
 20  Profit          38746 non-null  float64
 21  Shipping_cost   38746 non-null  float64
 22  Order_priority  38746 non-null  object
dtypes: datetime64[ns](2), float64(4), int64(2), object(15)
memory usage: 6.8+ MB
```

find how many unique,count,top values in each column

In [44]:
```
df.describe(include = 'object').T
```

Out[44]:

	count	unique	top	freq
Order_ID	38746	20319	CA-2014-100111	14
Ship_ode	38746	4	Standard Class	23241
Customer_ID	38746	1581	BE-11335	84
Customer_name	38746	795	Bill Eplett	88
Segment	38746	3	Consumer	19997
City	38746	3258	New York City	875
State	38746	926	California	1896
Country	38746	136	United States	8123
Market	38746	7	US	8123
Region	38746	13	Central	8335
Product_ID	38746	9386	OFF-AR-10003651	31
Category	38746	3	Office Supplies	24270
Sub_Category	38746	17	Binders	4624
Product_name	38746	3729	Staples	213
Order_priority	38746	4	Medium	22195

checking for null values

In [45]:
```
df.isna().sum()
```

Out[45]:
```
Row ID           0
Order_ID         0
order_date       0
Ship_date        0
Ship_ode         0
Customer_ID      0
Customer_name    0
Segment          0
City             0
State            0
Country          0
Market           0
Region           0
Product_ID       0
Category         0
Sub_Category     0
Product_name     0
Sales            0
Quantity         0
Discount         0
Profit           0
Shipping_cost    0
Order_priority   0
dtype: int64
```

remove the null and missing values

In [46]:
```
df.dropna(inplace = True)
```

check for any duplicate values are present or not

In [47]:
```
df.duplicated()
```

Out[47]:
```
0        False
1        False
2        False
3        False
4        False
         ...
38741    False
38742    False
38743    False
38744    False
38745    False
Length: 38746, dtype: bool
```

in this dataset now no null values,missing values,no duplicated values and no outlier

now we download this cleaned dataset for visualisations

In [ ]:
```
df.to_csv('cleaned_dataset.csv', index=False)
```