In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```python
df = pd.read_excel("SuperStoreCanada (2).xlsx")
df
```

Out[2]:

	RefID	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustID	CustomerSegment	CustName	PostalCode	...	ProdCategory	ProdSubCat	ProdName	ProdContainer	Quantity	SalePrice	CostPrice	Discount
0	100001.0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	NaN	Home Office	Matt Collister	NaN	...	Office Supplies	Storage & Organization	Safco Industrial Wire Shelving	Large Box	10.0	95.99	57.5940	0.08
1	100002.0	2009-04-01	10002.0	High	2009-04-03	Regular Air	NaN	Small Business	Jessica Myrick	NaN	...	Office Supplies	Storage & Organization	Perma STOR-ALL™ Hanging File Box, 13 1/8"W x 1...	Small Box	36.0	5.98	1.9136	0.10
2	100003.0	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	NaN	Consumer	David Philippe	NaN	...	Office Supplies	Binders and Binder Accessories	Avery Trapezoid Ring Binder, 3" Capacity, Blac...	Small Box	4.0	40.98	26.2272	0.06
3	100004.0	2009-04-02	10004.0	Low	2009-04-04	Delivery Truck	NaN	Corporate	Alan Schoenberger	NaN	...	Furniture	Chairs & Chairmats	Hon 4070 Series Pagoda™ Armless Upholstered St...	Jumbo Drum	3.0	291.73	128.3612	0.00
4	100005.0	2009-04-02	10004.0	Low	2009-04-04	Delivery Truck	NaN	Corporate	Alan Schoenberger	NaN	...	Furniture	Chairs & Chairmats	Hon Valutask™ Swivel Chairs	Jumbo Drum	29.0	100.98	31.3038	0.07
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8394	108395.0	2013-03-30	15494.0	Not Specified	2013-04-01	Regular Air	NaN	Small Business	Jim Epp	NaN	...	Furniture	Office Furnishings	DAX Wood Document Frame.	Wrap Bag	85.0	13.73	6.5904	0.08
8395	108396.0	2013-03-30	15495.0	Low	2013-04-01	Regular Air	NaN	Home Office	Maribeth Yedwab	NaN	...	Technology	Computer Peripherals	Belkin 105-Key Black Keyboard	Small Box	68.0	19.98	9.5904	0.00
8396	108397.0	2013-03-30	15495.0	Low	2013-04-01	Regular Air	NaN	Home Office	Maribeth Yedwab	NaN	...	Office Supplies	Scissors, Rulers and Trimmers	Martin Yale Chadless Opener Electric Letter Op...	Medium Box	1.0	832.81	199.8744	0.09
8397	108398.0	2013-03-30	15496.0	Not Specified	2013-04-01	Delivery Truck	NaN	Corporate	Tony Molinari	NaN	...	Furniture	Chairs & Chairmats	Novimex Fabric Task Chair	Jumbo Drum	8.0	60.98	26.2214	0.06
8398	108399.0	2013-03-30	15496.0	Not Specified	2013-04-01	Express Air	NaN	Corporate	Tony Molinari	NaN	...	Office Supplies	Storage & Organization	Tenex Personal Project File with Scoop Front D...	Small Box	44.0	13.48	5.7964	0.10

8399 rows × 26 columns

Exploratory Data Analysis - EDA

In [3]:
```python
df.size
```

Out[3]: 218374

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

Out[4]: (8399, 26)

In [5]:
```python
df.columns
```

Out[5]:
```
Index(['RefID', 'OrderDate', 'OrderID', 'Priority', 'ShipDate', 'ShipMode',
       'CustID', 'CustomerSegment', 'CustName', 'PostalCode', 'City', 'State',
       'Country', 'Region', 'Market', 'ProdID', 'ProdCategory', 'ProdSubCat',
       'ProdName', 'ProdContainer', 'Quantity', 'SalePrice', 'CostPrice',
       'Discount', 'ShipCost', 'Status'],
      dtype='object')
```

In [6]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8399 entries, 0 to 8398
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RefID            8399 non-null   float64
 1   OrderDate        8399 non-null   datetime64[ns]
 2   OrderID          8399 non-null   float64
 3   Priority         8399 non-null   object
 4   ShipDate         8399 non-null   datetime64[ns]
 5   ShipMode         8399 non-null   object
 6   CustID           0 non-null      float64
 7   CustomerSegment  8399 non-null   object
 8   CustName         8399 non-null   object
 9   PostalCode       0 non-null      float64
 10  City             0 non-null      float64
 11  State            8399 non-null   object
 12  Country          0 non-null      float64
 13  Region           8399 non-null   object
 14  Market           0 non-null      float64
 15  ProdID           0 non-null      float64
 16  ProdCategory     8399 non-null   object
 17  ProdSubCat       8399 non-null   object
 18  ProdName         8399 non-null   object
 19  ProdContainer    8399 non-null   object
 20  Quantity         8399 non-null   float64
 21  SalePrice        8399 non-null   float64
 22  CostPrice        8399 non-null   float64
 23  Discount         8399 non-null   float64
 24  ShipCost         8399 non-null   float64
 25  Status           8399 non-null   object
dtypes: datetime64[ns](2), float64(13), object(11)
memory usage: 1.7+ MB
```

In [7]:
```python
df.describe()
```

Out[7]:

	RefID	OrderID	CustID	PostalCode	City	Country	Market	ProdID	Quantity	SalePrice	CostPrice	Discount	ShipCost
count	8399.000000	8399.000000	0.0	0.0	0.0	0.0	0.0	0.0	8399.000000	8399.000000	8399.000000	8399.000000	8399.000000
mean	104200.000000	12735.332778	NaN	NaN	NaN	NaN	NaN	NaN	35.258959	89.346259	43.356775	0.049671	12.838557
std	2424.726789	1588.813787	NaN	NaN	NaN	NaN	NaN	NaN	23.146174	290.354383	140.709955	0.031823	17.264052
min	100001.000000	10001.000000	NaN	NaN	NaN	NaN	NaN	NaN	1.000000	0.990000	0.182400	0.000000	0.490000
25%	102100.500000	11359.500000	NaN	NaN	NaN	NaN	NaN	NaN	16.000000	6.480000	3.168000	0.020000	3.300000
50%	104200.000000	12726.000000	NaN	NaN	NaN	NaN	NaN	NaN	32.000000	20.990000	10.146000	0.050000	6.070000
75%	106299.500000	14105.500000	NaN	NaN	NaN	NaN	NaN	NaN	51.000000	85.990000	40.839200	0.080000	13.990000
max	108399.000000	15496.000000	NaN	NaN	NaN	NaN	NaN	NaN	111.000000	6783.020000	4137.642200	0.250000	164.730000

In [8]:
```python
df.isnull().sum()
```

Out[8]:
```
RefID                 0
OrderDate             0
OrderID               0
Priority              0
ShipDate              0
ShipMode              0
CustID             8399
CustomerSegment       0
CustName              0
PostalCode         8399
City               8399
State                 0
Country            8399
Region                0
Market             8399
ProdID             8399
ProdCategory          0
ProdSubCat            0
ProdName              0
ProdContainer         0
Quantity              0
SalePrice             0
CostPrice             0
Discount              0
ShipCost              0
Status                0
dtype: int64
```

In [9]:
```python
df1=df.drop(['CustID','PostalCode','City', 'Country','Market','ProdID', 'RefID', 'CustName'], axis=1)
df1
```

Out[9]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	ProdName	ProdContainer	Quantity	SalePrice	CostPrice	Discount	ShipCost	Status
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	Safco Industrial Wire Shelving	Large Box	10.0	95.99	57.5940	0.08	35.00	OK
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	Perma STOR-ALL™ Hanging File Box, 13 1/8"W x 1...	Small Box	36.0	5.98	1.9136	0.10	4.69	OK
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	Avery Trapezoid Ring Binder, 3" Capacity, Blac...	Small Box	4.0	40.98	26.2272	0.06	2.99	OK
3	2009-04-02	10004.0	Low	2009-04-04	Delivery Truck	Corporate	British Columbia	West	Furniture	Chairs & Chairmats	Hon 4070 Series Pagoda™ Armless Upholstered St...	Jumbo Drum	3.0	291.73	128.3612	0.00	48.80	OK
4	2009-04-02	10004.0	Low	2009-04-04	Delivery Truck	Corporate	British Columbia	West	Furniture	Chairs & Chairmats	Hon Valutask™ Swivel Chairs	Jumbo Drum	29.0	100.98	31.3038	0.07	45.00	OK
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8394	2013-03-30	15494.0	Not Specified	2013-04-01	Regular Air	Small Business	New Brunswick	Atlantic	Furniture	Office Furnishings	DAX Wood Document Frame.	Wrap Bag	85.0	13.73	6.5904	0.08	6.85	OK
8395	2013-03-30	15495.0	Low	2013-04-01	Regular Air	Home Office	Saskachewan	Prarie	Technology	Computer Peripherals	Belkin 105-Key Black Keyboard	Small Box	68.0	19.98	9.5904	0.00	4.00	OK
8396	2013-03-30	15495.0	Low	2013-04-01	Regular Air	Home Office	Saskachewan	Prarie	Office Supplies	Scissors, Rulers and Trimmers	Martin Yale Chadless Opener Electric Letter Op...	Medium Box	1.0	832.81	199.8744	0.09	24.49	OK
8397	2013-03-30	15496.0	Not Specified	2013-04-01	Delivery Truck	Corporate	British Columbia	West	Furniture	Chairs & Chairmats	Novimex Fabric Task Chair	Jumbo Drum	8.0	60.98	26.2214	0.06	30.00	OK
8398	2013-03-30	15496.0	Not Specified	2013-04-01	Express Air	Corporate	British Columbia	West	Office Supplies	Storage & Organization	Tenex Personal Project File with Scoop Front D...	Small Box	44.0	13.48	5.7964	0.10	4.51	OK

8399 rows × 18 columns

In [10]:
```python
df1.columns
```

Out[10]:
```
Index(['OrderDate', 'OrderID', 'Priority', 'ShipDate', 'ShipMode',
       'CustomerSegment', 'State', 'Region', 'ProdCategory', 'ProdSubCat',
       'ProdName', 'ProdContainer', 'Quantity', 'SalePrice', 'CostPrice',
       'Discount', 'ShipCost', 'Status'],
      dtype='object')
```

In [11]:
```python
#Create a new column total sales
df1['TotalSales']= df1['SalePrice']*df1['Quantity']
df1.head(3)
```

Out[11]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	ProdName	ProdContainer	Quantity	SalePrice	CostPrice	Discount	ShipCost	Status	TotalSales
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	Safco Industrial Wire Shelving	Large Box	10.0	95.99	57.5940	0.08	35.00	OK	959.90
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	Perma STOR-ALL™ Hanging File Box, 13 1/8"W x 1...	Small Box	36.0	5.98	1.9136	0.10	4.69	OK	215.28
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	Avery Trapezoid Ring Binder, 3" Capacity, Blac...	Small Box	4.0	40.98	26.2272	0.06	2.99	OK	163.92

In [12]:
```python
#Create a new column total cost
df1['TotalCost']= df1['CostPrice']*df1['Quantity']
df1.head(3)
```

Out[12]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	ProdName	ProdContainer	Quantity	SalePrice	CostPrice	Discount	ShipCost	Status	TotalSales	TotalCo
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	Safco Industrial Wire Shelving	Large Box	10.0	95.99	57.5940	0.08	35.00	OK	959.90	575.940
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	Perma STOR-ALL™ Hanging File Box, 13 1/8"W x 1...	Small Box	36.0	5.98	1.9136	0.10	4.69	OK	215.28	68.889
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	Avery Trapezoid Ring Binder, 3" Capacity, Blac...	Small Box	4.0	40.98	26.2272	0.06	2.99	OK	163.92	104.908

In [13]:
```python
df1['Discount'].unique()
```

Out[13]:
```
array([0.08, 0.1 , 0.06, 0.  , 0.07, 0.05, 0.09, 0.03, 0.04, 0.01, 0.02,
       0.21, 0.17, 0.11, 0.16, 0.25])
```

In [14]:
```python
#Calculating discount amount
df1['discount_perc'] = df1['Discount']*100
df1.head(3)
```

Out[14]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	ProdContainer	Quantity	SalePrice	CostPrice	Discount	ShipCost	Status	TotalSales	TotalCost
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	Large Box	10.0	95.99	57.5940	0.08	35.00	OK	959.90	575.9400
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	Small Box	36.0	5.98	1.9136	0.10	4.69	OK	215.28	68.8896
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	...	Small Box	4.0	40.98	26.2272	0.06	2.99	OK	163.92	104.9088

3 rows × 21 columns

In [15]:
```python
#Calculating the discount amount
df1['Discount_amount'] = df1['discount_perc']*df1['TotalSales']/100
df1.head(2)
```

Out[15]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	Quantity	SalePrice	CostPrice	Discount	ShipCost	Status	TotalSales	TotalCost	discount_perc
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	10.0	95.99	57.5940	0.08	35.00	OK	959.90	575.9400	8.0
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	36.0	5.98	1.9136	0.10	4.69	OK	215.28	68.8896	10.0

2 rows × 22 columns

In [16]:
```python
#Sales after Discount
df1['GrossSales']= df1['TotalSales']- df1['Discount_amount']
df1.head(2)
```

Out[16]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	SalePrice	CostPrice	Discount	ShipCost	Status	TotalSales	TotalCost	discount_perc	Discount_amou
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	95.99	57.5940	0.08	35.00	OK	959.90	575.9400	8.0	76.7
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	5.98	1.9136	0.10	4.69	OK	215.28	68.8896	10.0	21.5

2 rows × 23 columns

In [17]:
```python
#Calculating Cost Price
df1['GrossCostPrice'] = df1['TotalCost']+df1['ShipCost']
df1.head(2)
```

Out[17]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	CostPrice	Discount	ShipCost	Status	TotalSales	TotalCost	discount_perc	Discount_amount	GrossS
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	57.5940	0.08	35.00	OK	959.90	575.9400	8.0	76.792	883
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	1.9136	0.10	4.69	OK	215.28	68.8896	10.0	21.528	193

2 rows × 24 columns

In [18]:
```python
#Create a column for Profit
df1['Profit']= df1['GrossSales'] - df1['GrossCostPrice']
df1.head(3)
```

Out[18]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	Discount	ShipCost	Status	TotalSales	TotalCost	discount_perc	Discount_amount	GrossSales
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	0.08	35.00	OK	959.90	575.9400	8.0	76.7920	883.1080
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	0.10	4.69	OK	215.28	68.8896	10.0	21.5280	193.7520
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	...	0.06	2.99	OK	163.92	104.9088	6.0	9.8352	154.0848

3 rows × 25 columns

In [19]:
```python
df1['Status'].value_counts()
```

Out[19]:
```
OK     7560
Ret     839
Name: Status, dtype: int64
```

In [20]:
```python
df1['ProdCategory'].unique()
```

Out[20]:
```
array(['Office Supplies', 'Furniture', 'Technology'], dtype=object)
```

In [21]:
```python
df1['ProdSubCat'].unique()
```

Out[21]:
```
array(['Storage & Organization', 'Binders and Binder Accessories',
       'Chairs & Chairmats', 'Paper', 'Pens & Art Supplies',
       'Computer Peripherals', 'Office Machines', 'Office Furnishings',
       'Telephones and Communication', 'Tables', 'Bookcases',
       'Copiers and Fax', 'Labels', 'Appliances', 'Rubber Bands',
       'Envelopes', 'Scissors, Rulers and Trimmers'], dtype=object)
```

In [22]:
```python
df1['State'].unique()
```

Out[22]:
```
array(['Quebec', 'Ontario', 'Nova Scotia', 'British Columbia', 'Alberta',
       'Saskachewan', 'New Brunswick', 'Manitoba',
       'Northwest Territories', 'Yukon', 'Newfoundland', 'Nunavut',
       'Prince Edward Island'], dtype=object)
```

In [ ]:

In [23]:
```python
# getting month year from the dataset
df['month_year']= df['OrderDate'].apply(lambda x: x.strftime('%Y-%m'))
```

In [24]:
```python
# Grouping month year
df_trend = df.groupby('month_year').sum()['SalePrice'].reset_index()
df_trend
```

Out[24]:

	month_year	SalePrice
0	2009-04	26773.31
1	2009-05	16709.89
2	2009-06	33065.77
3	2009-07	20633.36
4	2009-08	10390.12
5	2009-09	12314.05
6	2009-10	27939.49
7	2009-11	18902.12
8	2009-12	13180.40
9	2010-01	22177.93
10	2010-02	10824.46
11	2010-03	19862.73
12	2010-04	15289.40
13	2010-05	11143.31
14	2010-06	8541.87
15	2010-07	13632.20
16	2010-08	13241.38
17	2010-09	11165.28
18	2010-10	10402.49
19	2010-11	9505.02
20	2010-12	22018.77
21	2011-01	18660.71
22	2011-02	12834.54
23	2011-03	15839.01
24	2011-04	13623.05
25	2011-05	16961.66
26	2011-06	10767.98
27	2011-07	10476.70
28	2011-08	9911.10
29	2011-09	9225.00
30	2011-10	16086.60
31	2011-11	10478.26
32	2011-12	12885.97
33	2012-01	14492.75
34	2012-02	21920.60
35	2012-03	12955.45
36	2012-04	14482.02
37	2012-05	13216.99
38	2012-06	18830.65
39	2012-07	16910.24
40	2012-08	22241.78
41	2012-09	11976.42
42	2012-10	11758.84
43	2012-11	15305.05
44	2012-12	13935.23
45	2013-01	20853.47
46	2013-02	16239.49
47	2013-03	19836.32

In [25]:
```python
# grouping product name column
prod_sales=pd.DataFrame(df1.groupby('ProdSubCat').sum()['SalePrice'])
prod_sales
```

Out[25]:

ProdSubCat	SalePrice
Appliances	31525.68
Binders and Binder Accessories	52706.18
Bookcases	32665.80
Chairs & Chairmats	70691.13
Computer Peripherals	32218.23
Copiers and Fax	64899.13
Envelopes	6445.84
Labels	1585.98
Office Furnishings	26263.81
Office Machines	190647.47
Paper	17651.99
Pens & Art Supplies	6087.97
Rubber Bands	538.95
Scissors, Rulers and Trimmers	6101.49
Storage & Organization	43624.31
Tables	79921.10
Telephones and Communication	86844.17

In [26]:
```python
#sorting product sales column
prod_sales = prod_sales.sort_values('SalePrice', ascending=False)
prod_sales[:10]
```

Out[26]:

ProdSubCat	SalePrice
Office Machines	190647.47
Telephones and Communication	86844.17
Tables	79921.10
Chairs & Chairmats	70691.13
Copiers and Fax	64899.13
Binders and Binder Accessories	52706.18
Storage & Organization	43624.31
Bookcases	32665.80
Computer Peripherals	32218.23
Appliances	31525.68

Q1 What is the % of Sales of Appliances in the Year 2012?

In [27]:
```python
#extracting year from order date
df1['Year'] = pd.DatetimeIndex(df1['OrderDate']).year
df1.head(3)
```

Out[27]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	ShipCost	Status	TotalSales	TotalCost	discount_perc	Discount_amount	GrossSales	GrossCostPrice
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	35.00	OK	959.90	575.9400	8.0	76.7920	883.1080	610.9400
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	4.69	OK	215.28	68.8896	10.0	21.5280	193.7520	73.5796
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	...	2.99	OK	163.92	104.9088	6.0	9.8352	154.0848	107.8988

3 rows × 26 columns

In [28]:
```python
year_wise=pd.DataFrame(df1.groupby('Year').sum()['TotalSales'])
year_wise
```

Out[28]:

Year	TotalSales
2009	3285611.47
2010	4389688.09
2011	5157065.76
2012	6644711.11
2013	1701178.20

In [29]:
```python
App_2012 = df1[(df1.ProdSubCat=='Appliances') & (df1.Year==2012) & (df1.Status=='OK')]
```

In [30]:
```python
App_sales = App_2012['TotalSales'].sum()
App_sales
```

Out[30]: 205898.95000000004

In [31]:
```python
Sales_2012 = df1[(df1.Year==2012) & (df1.Status=='OK')]
Sales_year_2012 = Sales_2012['TotalSales'].sum()
Sales_year_2012
```

Out[31]: 6045294.17

In [32]:
```python
print("% of application on sales in 2012 (in %):", (App_sales*100)/Sales_year_2012)
```

```
% of application on sales in 2012 (in %): 3.4059376468688876
```

Q2 What is the % of Sales of “Home Office” in the Year 2011?

In [33]:
```python
HO_2011 = df1[(df1.CustomerSegment=='Home Office') & (df1.Year==2011) & (df1.Status=='OK')]
```

In [34]:
```python
HO_Sales = HO_2011['TotalSales'].sum()
HO_Sales
```

Out[34]: 1166331.63

In [35]:
```python
Sales_2011 = df1[(df1.Year==2011) & (df1.Status=='OK')]
Sales_year_2011 = Sales_2011['TotalSales'].sum()
Sales_year_2011
```

Out[35]: 4612236.3

In [36]:
```python
print("% of Home Office on sales in 2011 (in %):", (HO_Sales*100)/Sales_year_2011)
```

```
% of Home Office on sales in 2011 (in %): 25.28776832184422
```

Q3 Find the Top 10 product names by sales in each State. Which product is highest in 3 States?

In [37]:
```python
# grouping product name column
prod_sale=pd.DataFrame(df1.groupby(['State', 'ProdSubCat']).sum()['TotalSales'])
prod_sale
```

Out[37]:

State	ProdSubCat	TotalSales
Alberta	Appliances	88290.27
	Binders and Binder Accessories	128678.00
	Bookcases	148724.52
	Chairs & Chairmats	282909.57
	Computer Peripherals	101343.43
...	...	...
Yukon	Rubber Bands	661.95
	Scissors, Rulers and Trimmers	2240.46
	Storage & Organization	115210.59
	Tables	239489.54
	Telephones and Communication	204726.32

218 rows × 1 columns

In [38]:
```python
#sorting product sales column
prod_salez = prod_sale.sort_values('TotalSales', ascending=False)
prod_salez[:10]
```

Out[38]:

State	ProdSubCat	TotalSales
Ontario	Telephones and Communication	669315.91
	Chairs & Chairmats	600253.32
	Tables	575672.83
Alberta	Office Machines	494097.69
Saskachewan	Office Machines	477437.09
Ontario	Office Machines	413267.90
Quebec	Tables	411898.90
British Columbia	Telephones and Communication	405619.74
Ontario	Binders and Binder Accessories	396898.42
Alberta	Telephones and Communication	363585.55

Q4 Find the 5 States with Lowest overall profit?

In [39]:
```python
# grouping state by profi
Prof =pd.DataFrame(df1.groupby('State').sum()['Profit'])
profit = Prof.sort_values('Profit')
profit[:5]
```

Out[39]:

State	Profit
Newfoundland	61753.2681
Nunavut	78930.6194
Prince Edward Island	285645.2046
New Brunswick	414413.3226
Northwest Territories	508157.4097

Q5 Show FinYear wise strategic data with Months (Apr to Dec) in Rows and FinYear as columns.

In [40]:
```python
#extracting month from order date
df1['Month'] = pd.DatetimeIndex(df1['OrderDate']).month
df1.head(3)
```

Out[40]:

	OrderDate	OrderID	Priority	ShipDate	ShipMode	CustomerSegment	State	Region	ProdCategory	ProdSubCat	...	Status	TotalSales	TotalCost	discount_perc	Discount_amount	GrossSales	GrossCostPrice	Profit
0	2009-04-01	10001.0	Not Specified	2009-04-03	Express Air	Home Office	Quebec	Quebec	Office Supplies	Storage & Organization	...	OK	959.90	575.9400	8.0	76.7920	883.1080	610.9400	272.1680
1	2009-04-01	10002.0	High	2009-04-03	Regular Air	Small Business	Ontario	Ontario	Office Supplies	Storage & Organization	...	OK	215.28	68.8896	10.0	21.5280	193.7520	73.5796	120.1724
2	2009-04-02	10003.0	Critical	2009-04-04	Regular Air	Consumer	Nova Scotia	Atlantic	Office Supplies	Binders and Binder Accessories	...	OK	163.92	104.9088	6.0	9.8352	154.0848	107.8988	46.1860

3 rows × 27 columns

In [41]:
```python
fin =pd.DataFrame(df1.groupby('Month').sum()['TotalSales'])
fin
```

Out[41]:

Month	TotalSales
1	1934748.39
2	1691451.39
3	2100722.59
4	1975476.86
5	1629787.33
6	1776715.80
7	1690254.02
8	1774493.61
9	1476423.18
10	1600204.57
11	1583437.04
12	1944539.85

In [42]:
```python
df_new = df1[['Month','TotalSales','Year']]
df_new
```

Out[42]:

	Month	TotalSales	Year
0	4	959.90	2009
1	4	215.28	2009
2	4	163.92	2009
3	4	875.19	2009
4	4	2928.42	2009
...	...	...	...
8394	3	1167.05	2013
8395	3	1358.64	2013
8396	3	832.81	2013
8397	3	487.84	2013
8398	3	593.12	2013

8399 rows × 3 columns

In [43]:
```python
fin_year = df1.groupby(['Month', 'Year'])['TotalSales'].sum().unstack(fill_value=0)
fin_year
```

Out[43]:

Month \ Year	2009	2010	2011	2012	2013
1	0.00	374325.25	463533.68	488339.16	608550.30
2	0.00	269855.34	382743.55	586993.12	451859.38
3	0.00	437669.73	504930.37	517353.97	640768.52
4	563591.21	418606.47	385093.83	608185.35	0.00
5	334401.58	331757.55	463144.86	500483.34	0.00
6	429746.66	272721.12	455402.62	618845.40	0.00
7	390364.30	359664.95	450894.97	489329.80	0.00
8	243430.47	377719.20	425081.23	728262.71	0.00
9	244942.11	397861.87	325758.52	507860.68	0.00
10	371969.73	316874.64	455277.82	456082.38	0.00
11	358333.44	258282.99	415761.96	551058.65	0.00
12	348831.97	574348.98	429442.35	591916.55	0.00

In [44]:
```python
#deleting first three rows
fin_years = fin_year.iloc[3:]
fin_years
```

Out[44]:

Month \ Year	2009	2010	2011	2012	2013
4	563591.21	418606.47	385093.83	608185.35	0.0
5	334401.58	331757.55	463144.86	500483.34	0.0
6	429746.66	272721.12	455402.62	618845.40	0.0
7	390364.30	359664.95	450894.97	489329.80	0.0
8	243430.47	377719.20	425081.23	728262.71	0.0
9	244942.11	397861.87	325758.52	507860.68	0.0
10	371969.73	316874.64	455277.82	456082.38	0.0
11	358333.44	258282.99	415761.96	551058.65	0.0
12	348831.97	574348.98	429442.35	591916.55	0.0

Q7 Show all Invoice data with Order Date wise Totals

In [45]:
```python
df1.columns
```

Out[45]:
```
Index(['OrderDate', 'OrderID', 'Priority', 'ShipDate', 'ShipMode',
       'CustomerSegment', 'State', 'Region', 'ProdCategory', 'ProdSubCat',
       'ProdName', 'ProdContainer', 'Quantity', 'SalePrice', 'CostPrice',
       'Discount', 'ShipCost', 'Status', 'TotalSales', 'TotalCost',
       'discount_perc', 'Discount_amount', 'GrossSales', 'GrossCostPrice',
       'Profit', 'Year', 'Month'],
      dtype='object')
```

In [46]:
```python
#extracting columns for invoice
df_new = df1.iloc[:, [0,1,8,9,10,12,13,18]]
df_new.head()
```

Out[46]:

	OrderDate	OrderID	ProdCategory	ProdSubCat	ProdName	Quantity	SalePrice	TotalSales
0	2009-04-01	10001.0	Office Supplies	Storage & Organization	Safco Industrial Wire Shelving	10.0	95.99	959.90
1	2009-04-01	10002.0	Office Supplies	Storage & Organization	Perma STOR-ALL™ Hanging File Box, 13 1/8"W x 1...	36.0	5.98	215.28
2	2009-04-02	10003.0	Office Supplies	Binders and Binder Accessories	Avery Trapezoid Ring Binder, 3" Capacity, Blac...	4.0	40.98	163.92
3	2009-04-02	10004.0	Furniture	Chairs & Chairmats	Hon 4070 Series Pagoda™ Armless Upholstered St...	3.0	291.73	875.19
4	2009-04-02	10004.0	Furniture	Chairs & Chairmats	Hon Valutask™ Swivel Chairs	29.0	100.98	2928.42

In [47]:
```python
df_new['TotalSales'].sum()
```

Out[47]: 21178254.63

In [ ]: