

```
In [3]: import numpy as np  
import pandas as pd
```

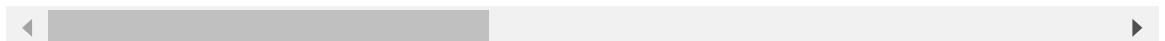
```
In [5]: store=pd.read_csv(r"C:\Users\Admin\Downloads\Sample - Superstore_Orders.csv") #i
```

```
In [6]: store
```

Out[6]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns



In [9]: store.columns #columns present in the data

```
Out[9]: Index(['Category', 'City', 'Country/Region', 'Customer Name', 'Manufacturer',  
            'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',  
            'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category',  
            'Discount', 'Profit', 'Quantity', 'Sales'],  
            dtype='object')
```

```
In [11]: id(store)
```

```
Out[11]: 2107214670704
```

```
In [13]: len(store) #length of store
```

```
Out[13]: 10194
```

```
In [15]: store.shape #shape of store
```

```
Out[15]: (10194, 19)
```

```
In [17]: len(store.columns) #len of store columns
```

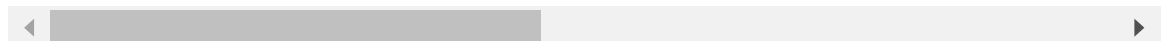
```
Out[17]: 19
```

```
In [19]: store.isnull() #this function tells about the null values present in the data
```

```
Out[19]:
```

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Postal Code
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
10189	False	False	False	False	False	False	False	False
10190	False	False	False	False	False	False	False	False
10191	False	False	False	False	False	False	False	False
10192	False	False	False	False	False	False	False	False
10193	False	False	False	False	False	False	False	False

10194 rows × 19 columns

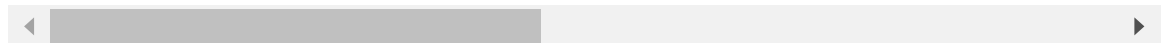


```
In [21]: store.isna() #null
```

Out[21]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Postal Code
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
10189	False	False	False	False	False	False	False	False
10190	False	False	False	False	False	False	False	False
10191	False	False	False	False	False	False	False	False
10192	False	False	False	False	False	False	False	False
10193	False	False	False	False	False	False	False	False

10194 rows × 19 columns

In [23]: `store.isnull().sum()` *#sum of all null datas*

```
Out[23]: Category      0
City      0
Country/Region  0
Customer Name  0
Manufacturer  0
Order Date  0
Order ID  0
Postal Code  0
Product Name  0
Region  0
Segment  0
Ship Date  0
Ship Mode  0
State/Province  0
Sub-Category  0
Discount  0
Profit  0
Quantity  0
Sales  0
dtype: int64
```

In [25]: `store.dtypes` *#datatypes of the data*

```
Out[25]: Category      object
City                  object
Country/Region        object
Customer Name          object
Manufacturer            object
Order Date             object
Order ID               object
Postal Code            object
Product Name           object
Region                 object
Segment               object
Ship Date              object
Ship Mode              object
State/Province         object
Sub-Category           object
Discount               float64
Profit                 float64
Quantity               int64
Sales                  float64
dtype: object
```

```
In [27]: store.info() #this function gives the all info about the data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10194 entries, 0 to 10193
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Category              10194 non-null  object
 1   City                  10194 non-null  object
 2   Country/Region        10194 non-null  object
 3   Customer Name          10194 non-null  object
 4   Manufacturer           10194 non-null  object
 5   Order Date             10194 non-null  object
 6   Order ID               10194 non-null  object
 7   Postal Code            10194 non-null  object
 8   Product Name           10194 non-null  object
 9   Region                 10194 non-null  object
10   Segment               10194 non-null  object
11   Ship Date              10194 non-null  object
12   Ship Mode              10194 non-null  object
13   State/Province         10194 non-null  object
14   Sub-Category           10194 non-null  object
15   Discount               10194 non-null  float64
16   Profit                 10194 non-null  float64
17   Quantity               10194 non-null  int64
18   Sales                  10194 non-null  float64
dtypes: float64(3), int64(1), object(15)
memory usage: 1.5+ MB
```

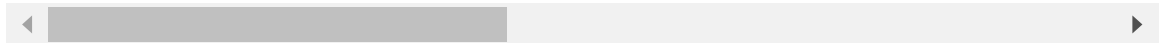
```
In [28]: pd.__version__ #version of pandas
```

```
Out[28]: '2.2.3'
```

```
In [31]: store.head() #this function prints the head values of the data by default it pri
```

Out[31]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Po C
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800	77
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	US-2020-112326	60
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	US-2020-112326	60
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	US-2020-112326	60
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	US-2020-141817	19

In [33]: `store.tail()` *#this function prints the tail values of the data by default it pri*

Out[33]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

In [35]: `store.tail(3)` *#it prints the last three tail values of the data*

Out[35]:

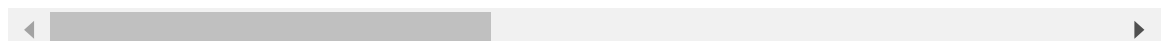
	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

In [36]: `store`

Out[36]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns



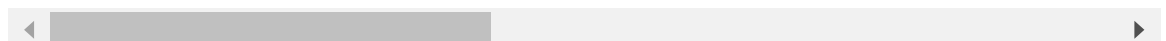
In [39]: store[:] #data can be sliced



Out[39]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns



In [41]: store[0:50:3]

Out[41]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	US-2020-112326
6	Office Supplies	Henderson	United States	Maria Etezadi	Rogers	06-01-2020	US-2020-167199
9	Office Supplies	Henderson	United States	Maria Etezadi	Alliance	06-01-2020	US-2020-167199
12	Technology	Henderson	United States	Maria Etezadi	Other	06-01-2020	US-2020-167199
15	Office Supplies	Huntsville	United States	Vivek Sundaresam	Acco	07-01-2020	US-2020-105417
18	Furniture	Springfield	United States	Anthony Jacobs	Howard Miller	10-01-2020	US-2020-149020
21	Furniture	San Francisco	United States	Brian Dahlen	O'Sullivan	13-01-2020	US-2020-157147
24	Office Supplies	Newark	United States	Michael Moore	Avery	13-01-2020	US-2020-118192
27	Office Supplies	Bossier City	United States	Chris Selesnick	Staple envelope	13-01-2020	US-2020-162775
30	Office Supplies	San Francisco	United States	Brian Dahlen	Tennsco	13-01-2020	US-2020-157147
33	Technology	Roswell	United States	Erica Hackney	Logitech	15-01-2020	US-2020-103366

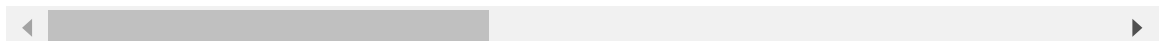
	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
36	Office Supplies	Philadelphia	United States	Delfina Latchford	Other	16-01-2020	US-2020-115791
39	Furniture	Scottsdale	United States	Toby Swindell	O'Sullivan	19-01-2020	US-2020-146591
42	Office Supplies	Scottsdale	United States	Toby Swindell	TOPS	19-01-2020	US-2020-146591
45	Furniture	Westland	United States	Xylona Preis	Eldon	20-01-2020	US-2020-167927
48	Office Supplies	Westland	United States	Xylona Preis	Holmes	20-01-2020	US-2020-167927

```
In [43]: store[:, -1]
```

Out[43]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
...	...	...	...	...	...	...	...
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103

10194 rows × 19 columns



In [45]: store.columns

```
Out[45]: Index(['Category', 'City', 'Country/Region', 'Customer Name', 'Manufacturer',  
              'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',  
              'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category',  
              'Discount', 'Profit', 'Quantity', 'Sales'],  
             dtype='object')
```

```
In [47]: store_cate=store[['Category', 'City', 'Country/Region', 'Customer Name', 'Manufa  
              'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',  
              'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category',  
              'Discount', 'Profit', 'Quantity', 'Sales']]
```

```
In [49]: store_cate.dtypes #categories data types
```

```
Out[49]: Category          object  
City                      object  
Country/Region           object  
Customer Name            object  
Manufacturer              object  
Order Date               object  
Order ID                 object  
Postal Code              object  
Product Name             object  
Region                  object  
Segment                 object  
Ship Date               object  
Ship Mode               object  
State/Province          object  
Sub-Category            object  
Discount                float64  
Profit                  float64  
Quantity                int64  
Sales                   float64  
dtype: object
```

```
In [51]: store_num=store[['Discount', 'Profit', 'Quantity', 'Sales']] #we can also print
```

```
In [53]: store_num
```

Out[53]:

	Discount	Profit	Quantity	Sales
0	0.2	5.5512	2	16.448
1	0.8	-5.4870	2	3.540
2	0.2	4.2717	3	11.784
3	0.2	-64.7748	3	272.736
4	0.2	4.8840	3	19.536
...	...	...	...	...
10189	0.2	19.7910	3	52.776
10190	0.2	6.4750	2	20.720
10191	0.2	-0.6048	3	3.024
10192	0.0	2.7279	7	90.930
10193	0.2	-0.6048	3	3.024

10194 rows × 4 columns

In [55]: `store_num.mean` *#mean of the given values*

Out[55]: <bound method DataFrame.mean of

	Discount	Profit	Quantity	Sales
0	0.2	5.5512	2	16.448
1	0.8	-5.4870	2	3.540
2	0.2	4.2717	3	11.784
3	0.2	-64.7748	3	272.736
4	0.2	4.8840	3	19.536
...	...	...	...	...
10189	0.2	19.7910	3	52.776
10190	0.2	6.4750	2	20.720
10191	0.2	-0.6048	3	3.024
10192	0.0	2.7279	7	90.930
10193	0.2	-0.6048	3	3.024

[10194 rows x 4 columns]>

In [57]: `store['Profit'].mean`

Out[57]: <bound method Series.mean of 0

1	-5.4870
2	4.2717
3	-64.7748
4	4.8840
...	...
10189	19.7910
10190	6.4750
10191	-0.6048
10192	2.7279
10193	-0.6048

Name: Profit, Length: 10194, dtype: float64>

In [59]: `store['Profit'].median`

```
Out[59]: <bound method Series.median of 0          5.5512
1          -5.4870
2           4.2717
3         -64.7748
4           4.8840
...
10189      19.7910
10190       6.4750
10191      -0.6048
10192       2.7279
10193      -0.6048
Name: Profit, Length: 10194, dtype: float64>
```

```
In [61]: store['Profit'].mode
```

```
Out[61]: <bound method Series.mode of 0          5.5512
1          -5.4870
2           4.2717
3         -64.7748
4           4.8840
...
10189      19.7910
10190       6.4750
10191      -0.6048
10192       2.7279
10193      -0.6048
Name: Profit, Length: 10194, dtype: float64>
```

```
In [63]: store['Profit'].var
```

```
Out[63]: <bound method Series.var of 0          5.5512
1          -5.4870
2           4.2717
3         -64.7748
4           4.8840
...
10189      19.7910
10190       6.4750
10191      -0.6048
10192       2.7279
10193      -0.6048
Name: Profit, Length: 10194, dtype: float64>
```

```
In [65]: store['Profit'].std
```

```
Out[65]: <bound method Series.std of 0          5.5512
1          -5.4870
2           4.2717
3         -64.7748
4           4.8840
...
10189      19.7910
10190       6.4750
10191      -0.6048
10192       2.7279
10193      -0.6048
Name: Profit, Length: 10194, dtype: float64>
```

```
In [67]: store['Discount'].mean
```

```
Out[67]: <bound method Series.mean of 0      0.2
1      0.8
2      0.2
3      0.2
4      0.2
...
10189  0.2
10190  0.2
10191  0.2
10192  0.0
10193  0.2
Name: Discount, Length: 10194, dtype: float64>
```

```
In [69]: store['Discount'].median
```

```
Out[69]: <bound method Series.median of 0      0.2
1      0.8
2      0.2
3      0.2
4      0.2
...
10189  0.2
10190  0.2
10191  0.2
10192  0.0
10193  0.2
Name: Discount, Length: 10194, dtype: float64>
```

```
In [71]: store['Discount'].mode
```

```
Out[71]: <bound method Series.mode of 0      0.2
1      0.8
2      0.2
3      0.2
4      0.2
...
10189  0.2
10190  0.2
10191  0.2
10192  0.0
10193  0.2
Name: Discount, Length: 10194, dtype: float64>
```

```
In [73]: store['Discount'].var
```

```
Out[73]: <bound method Series.var of 0      0.2
1      0.8
2      0.2
3      0.2
4      0.2
...
10189  0.2
10190  0.2
10191  0.2
10192  0.0
10193  0.2
Name: Discount, Length: 10194, dtype: float64>
```

```
In [75]: store['Discount'].std
```



```
Out[75]: <bound method Series.std of 0      0.2
1      0.8
2      0.2
3      0.2
4      0.2
...
10189  0.2
10190  0.2
10191  0.2
10192  0.0
10193  0.2
Name: Discount, Length: 10194, dtype: float64>
```

```
In [77]: store['Quantity'].mean
```

```
Out[77]: <bound method Series.mean of 0      2
1      2
2      3
3      3
4      3
..
10189  3
10190  2
10191  3
10192  7
10193  3
Name: Quantity, Length: 10194, dtype: int64>
```

```
In [79]: store['Quantity'].median
```

```
Out[79]: <bound method Series.median of 0      2
1      2
2      3
3      3
4      3
..
10189  3
10190  2
10191  3
10192  7
10193  3
Name: Quantity, Length: 10194, dtype: int64>
```

```
In [81]: store['Quantity'].mode
```

```
Out[81]: <bound method Series.mode of 0      2
1      2
2      3
3      3
4      3
..
10189  3
10190  2
10191  3
10192  7
10193  3
Name: Quantity, Length: 10194, dtype: int64>
```

```
In [83]: store['Quantity'].var
```

```
Out[83]: <bound method Series.var of 0          2
1          2
2          3
3          3
4          3
..
10189      3
10190      2
10191      3
10192      7
10193      3
Name: Quantity, Length: 10194, dtype: int64>
```

```
In [85]: store['Quantity'].std
```

```
Out[85]: <bound method Series.std of 0          2
1          2
2          3
3          3
4          3
..
10189      3
10190      2
10191      3
10192      7
10193      3
Name: Quantity, Length: 10194, dtype: int64>
```

```
In [87]: store['Sales'].mean
```

```
Out[87]: <bound method Series.mean of 0          16.448
1          3.540
2          11.784
3          272.736
4          19.536
...
10189      52.776
10190      20.720
10191      3.024
10192      90.930
10193      3.024
Name: Sales, Length: 10194, dtype: float64>
```

```
In [89]: store['Sales'].median
```

```
Out[89]: <bound method Series.median of 0          16.448
1          3.540
2          11.784
3          272.736
4          19.536
...
10189      52.776
10190      20.720
10191      3.024
10192      90.930
10193      3.024
Name: Sales, Length: 10194, dtype: float64>
```

```
In [91]: store['Sales'].mode
```

```
Out[91]: <bound method Series.mode of 0          16.448
1          3.540
2          11.784
3          272.736
4          19.536
...
10189      52.776
10190      20.720
10191       3.024
10192      90.930
10193       3.024
Name: Sales, Length: 10194, dtype: float64>
```

```
In [93]: store['Sales'].var
```

```
Out[93]: <bound method Series.var of 0          16.448
1          3.540
2          11.784
3          272.736
4          19.536
...
10189      52.776
10190      20.720
10191       3.024
10192      90.930
10193       3.024
Name: Sales, Length: 10194, dtype: float64>
```

```
In [95]: store['Sales'].std
```

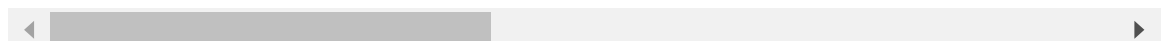
```
Out[95]: <bound method Series.std of 0          16.448
1          3.540
2          11.784
3          272.736
4          19.536
...
10189      52.776
10190      20.720
10191       3.024
10192      90.930
10193       3.024
Name: Sales, Length: 10194, dtype: float64>
```

```
In [97]: store
```

Out[97]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	20103
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	20112
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	20112
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	20112
4	Office Supplies	Philadelphia	United States	Mick Brown	Avery	05-01-2020	20141
...	...	...	...	...	...	...	...
10189	Office Supplies	New York City	United States	Patrick O'Donnell	Wilson Jones	30-12-2023	20143
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

10194 rows × 19 columns

In [99]: `store_intial=store.head(4) #we have printed first four values in the data`

store\_intial

Out[99]:

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID	Post Coc
0	Office Supplies	Houston	United States	Darren Powers	Message Book	03-01-2020	US-2020-103800	7705
1	Office Supplies	Naperville	United States	Phillina Ober	GBC	04-01-2020	US-2020-112326	6054
2	Office Supplies	Naperville	United States	Phillina Ober	Avery	04-01-2020	US-2020-112326	6054
3	Office Supplies	Naperville	United States	Phillina Ober	SAFCO	04-01-2020	US-2020-112326	6054

In [101]...

store\_intial.columns

Out[101]...

```
Index(['Category', 'City', 'Country/Region', 'Customer Name', 'Manufacturer',
      'Order Date', 'Order ID', 'Postal Code', 'Product Name', 'Region',
      'Segment', 'Ship Date', 'Ship Mode', 'State/Province', 'Sub-Category',
      'Discount', 'Profit', 'Quantity', 'Sales'],
      dtype='object')
```

In [103]...

store\_intial['Profit'].mean *#we can get the profits mean of the particular data*

Out[103]...

```
<bound method Series.mean of 0      5.5512
1      -5.4870
2       4.2717
3     -64.7748
Name: Profit, dtype: float64>
```

In [105]...

```
store_middle=store[100:105] #we have sliced the values from 100 to 104
store_middle
```

Out[105...

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
100	Furniture	Columbia	United States	Guy Armstrong	Global	11-02-2020	US-2020-168368
101	Furniture	Columbia	United States	Guy Armstrong	Other	11-02-2020	US-2020-168368
102	Furniture	Chesapeake	United States	Natalie Fritzler	Hon	11-02-2020	US-2020-127614
103	Office Supplies	Chesapeake	United States	Natalie Fritzler	Wilson Jones	11-02-2020	US-2020-127614
104	Office Supplies	Columbia	United States	Guy Armstrong	Other	11-02-2020	US-2020-168368

In [107...

```
store_middle['Profit'].mean #checked the profit mean of 100 to 104th data
```

Out[107...

```
<bound method Series.mean of 100    15.2225
101    53.2704
102    75.3732
103     8.2062
104    24.3930
Name: Profit, dtype: float64>
```

In [109...

```
store_tail=store.tail(4) #printed the last four values of the data
store_tail
```

Out[109...

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order
10190	Office Supplies	Fairfield	United States	Erica Bern	GBC	30-12-2023	20115
10191	Office Supplies	Loveland	United States	Jill Matthias	Other	30-12-2023	20156
10192	Technology	New York City	United States	Patrick O'Donnell	Other	30-12-2023	20143
10193	Office Supplies	Charlottetown	Canada	Harry Olson	Wilson Jones	30-12-2023	20143

In [111... `store_tail['Profit'].mean` *#checked the profits mean of last four values in data*

Out[111... `<bound method Series.mean of 10190`    6.4750  
10191    -0.6048  
10192    2.7279  
10193    -0.6048  
Name: Profit, dtype: float64>

In [113... *#by analysing data we can say that at intial stage and at ending stage of store #and the profit goes high in the middle*

In [115... `store_high=store[550:555]`  
`store_high`

Out[115...

	Category	City	Country/Region	Customer Name	Manufacturer	Order Date	Order ID
550	Office Supplies	Richmond	United States	Ellis Ballard	Ampad	30-05-2020	US-2020-146885
551	Office Supplies	New York City	United States	Laurel Workman	Xerox	30-05-2020	US-2020-103429
552	Technology	Chicago	United States	Mark Cousins	Brother	30-05-2020	US-2020-140473
553	Technology	New York City	United States	Laurel Workman	Other	30-05-2020	US-2020-103429
554	Technology	Jackson	United States	Jim Karlsson	Grandstream	31-05-2020	US-2020-166051

In [117...

```
store_high['Profit'].mean
```

Out[117...

```
<bound method Series.mean of 550      6.1290
551      17.9820
552     134.9955
553     134.5600
554      32.9817
Name: Profit, dtype: float64>
```

In [119...

```
#at the middle most values of the data the profit is high
```