```
import os
os.listdir('/kaggle/input/orders-and-details') # checking the dataset upload successful
```

```
['Details.csv', 'Orders.csv']
```

Begin Here

Introduction

Here, we analyze the sales data from a company which sells multiple products ranging across multiple categories across many different cities and states of India. The data although has almost all the required details, is limited in its approach since it is only for the year 2018.

Our goal is to find out which cities, states, categories and subcategories recorded most sales, which one recorded least. Where profit was maximum, where loss occured. We are also trying to find which mode of payment was used most.

Phase 1 -> Define the Objective

Here, we have total 5 objectives -

1. To find out which city made most sales and which one earned most profit
2. To find out which state made most sales and which one earned most profit
3. To find out which mode of payment was made to do most sales
4. To find out which category is most sold and which one is most profitable
5. To find out which subcategory is most sold and which one is most profitable

Phase 2 -> Gathering a view over structure of the data

Looking at the data, we see that most of our concern with the data is with two columns that should carry integer/float values. That would be sales which is represented as 'Amount' column and profit which is represented as 'Profit' column. Rest columns can be in string format.

One more concern with the privacy factor is that the dataset contais a column name - 'CustomerName'. To avoid distinguising unique characteristics, this column only contains the first name of the customer and no last name or any other identifier which can connect any datapoint from data to a person.

Data Credibility

Data_Credibility :\ The dataset has almost all the data that we might need to make the conclusions for the questions asked above.\ But the performance of a business is determined by many facotrs other than the factors mentioned above. Alse the data is dynamic, it will change every second and it is not necessarily true that all the assumptions and conclusions made on the data available will be true all the time.

```
# reading data and listing it in the format required for analysis

import pandas as pd

Orders = pd.read_csv('/kaggle/input/orders-and-details/Orders.csv')
print(Orders.head())

Details = pd.read_csv('/kaggle/input/orders-and-details/Details.csv')
print(Details.head())
```

```
    Order_ID  Order_Date CustomerName          State     City
0  B-26055  10-03-2018    Harivansh   Uttar Pradesh  Mathura
1  B-25993  03-02-2018       Madhav           Delhi    Delhi
2  B-25973  24-01-2018  Madan Mohan   Uttar Pradesh  Mathura
3  B-25923  27-12-2018        Gopal     Maharashtra   Mumbai
4  B-25757  21-08-2018     Vishakha  Madhya Pradesh   Indore
    Order_ID  Amount  Profit  Quantity     Category      Sub-Category  \
0  B-25681    1096     658         7  Electronics  Electronic Games
1  B-26055    5729      64        14    Furniture            Chairs
2  B-25955    2927     146         8    Furniture         Bookcases
3  B-26093    2847     712         8  Electronics          Printers
4  B-25602    2617    1151         4  Electronics            Phones

   PaymentMode
0          COD
1          EMI
2          EMI
3  Credit Card
4  Credit Card
```

```
# combining data from two separate work sheets into one combined pandas dataframe
combined_data = pd.merge(Orders,Details)
print(combined_data)
```

```
     Order_ID  Order_Date CustomerName             State     City  Amount  \
0     B-26055  10-03-2018    Harivansh     Uttar Pradesh  Mathura    5729
1     B-26055  10-03-2018    Harivansh     Uttar Pradesh  Mathura     671
2     B-26055  10-03-2018    Harivansh     Uttar Pradesh  Mathura     443
3     B-26055  10-03-2018    Harivansh     Uttar Pradesh  Mathura      57
4     B-26055  10-03-2018    Harivansh     Uttar Pradesh  Mathura     227
...       ...         ...          ...               ...      ...     ...
1495  B-25742  03-08-2018       Ashwin               Goa      Goa      11
1496  B-26088  26-03-2018       Bhavna            Sikkim  Gangtok      11
1497  B-25707  01-07-2018      Shivani       Maharashtra   Mumbai       8
1498  B-25758  22-08-2018      Shubham  Himachal Pradesh    Simla       8
1499  B-26095  28-03-2018      Monisha         Rajasthan   Jaipur       6

      Profit  Quantity     Category Sub-Category  PaymentMode
0         64        14    Furniture       Chairs          EMI
1        114         9  Electronics       Phones  Credit Card
2         11         1     Clothing        Saree          COD
3          7         2     Clothing        Shirt          UPI
4         48         5     Clothing        Stole          COD
...      ...       ...          ...          ...          ...
1495      -8         2     Clothing        Skirt          UPI
1496       5         2     Clothing  Hankerchief          UPI
1497      -6         1     Clothing        Stole          COD
1498      -2         1     Clothing        Stole          COD
1499       1         1     Clothing        Kurti          UPI

[1500 rows x 11 columns]
```

Data Collection

In the dataset above, we have all the columns we need in one single dataframe. Now we just need to confirm if the data that we have doesn't contain any non-numerical value in the two columns we need.

```
# Checking if the data we need, the columns Amount (which represents total sales of datapoint) and Profit (which represents total profit of datapoint)
combined_data.dtypes
```

```
Order_ID        object
Order_Date      object
CustomerName    object
State           object
City            object
Amount           int64
Profit           int64
Quantity         int64
Category        object
Sub-Category    object
PaymentMode     object
dtype: object
```

Now that we have all the data in the format for analysis, we will try to get the data plotted and analyzed.

```
# Finding out list of cities across the data
cities = [ city for city,something in combined_data.groupby('City')]
print(cities)
```

```
['Ahmedabad', 'Amritsar', 'Bangalore', 'Bhopal', 'Chandigarh', 'Chennai', 'Delhi', 'Gangtok', 'Goa', 'Hyderabad', 'Indore', 'Jaipur', 'Kashmir', '
```

```
# creating the grouped series for city and adding sales amount
cities_amount = combined_data.groupby('City').sum()['Amount']
print(cities_amount)
```

```
City
Ahmedabad     14543
Amritsar       4507
Bangalore     12520
Bhopal        23783
Chandigarh    21142
Chennai        6276
Delhi         22957
Gangtok        5276
Goa            6705
Hyderabad     13256
Indore        63680
Jaipur        11261
Kashmir       10829
Kohima        11993
Kolkata       14328
Lucknow        5726
Mathura       28747
Mumbai        58886
Patna         13417
Prayagraj      3889
```

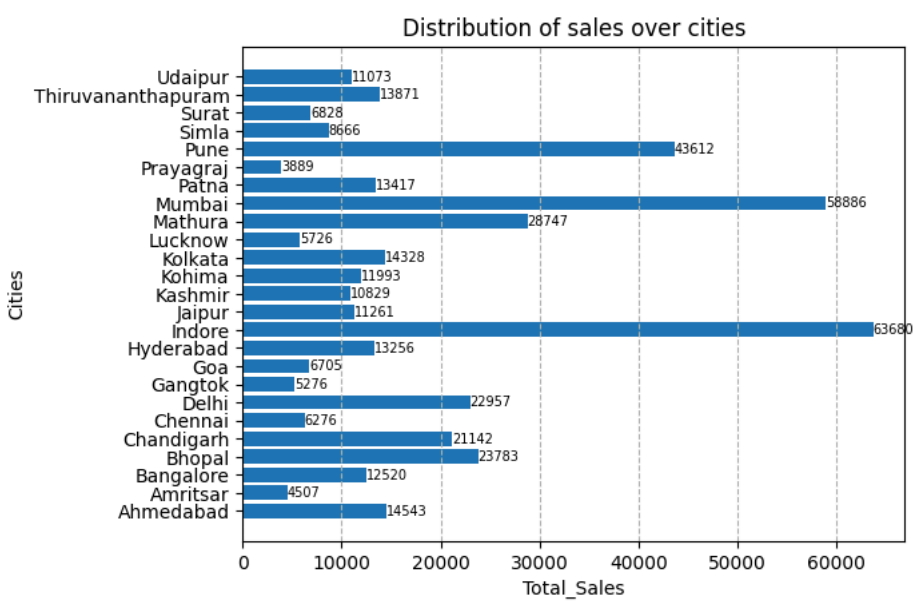
```
Pune                43612
Simla                8666
Surat                6828
Thiruvananthapuram  13871
Udaipur             11073
Name: Amount, dtype: int64
```

```
# Finding and collecting the data that we gathered into list as only lists are considered for plotting
cities_amount_list = []
for i in range(0,len(cities_amount)):
    cities_amount_list.append(cities_amount.values[i])
print(cities_amount_list)
```

```
[14543, 4507, 12520, 23783, 21142, 6276, 22957, 5276, 6705, 13256, 63680, 11261, 10829, 11993, 14328, 5726, 28747, 58886, 13417, 3889, 43612, 8666
```

```
import matplotlib.pyplot as plt

plt.barh(cities,cities_amount_list)
for index,value in enumerate(cities_amount_list):
    plt.text(value,index,str(value),size = 7,verticalalignment='center')
plt.grid(axis = 'x',linestyle = '--')
plt.title('Distribution of sales over cities')
plt.xlabel('Total_Sales')
plt.ylabel('Cities')
plt.show()
```



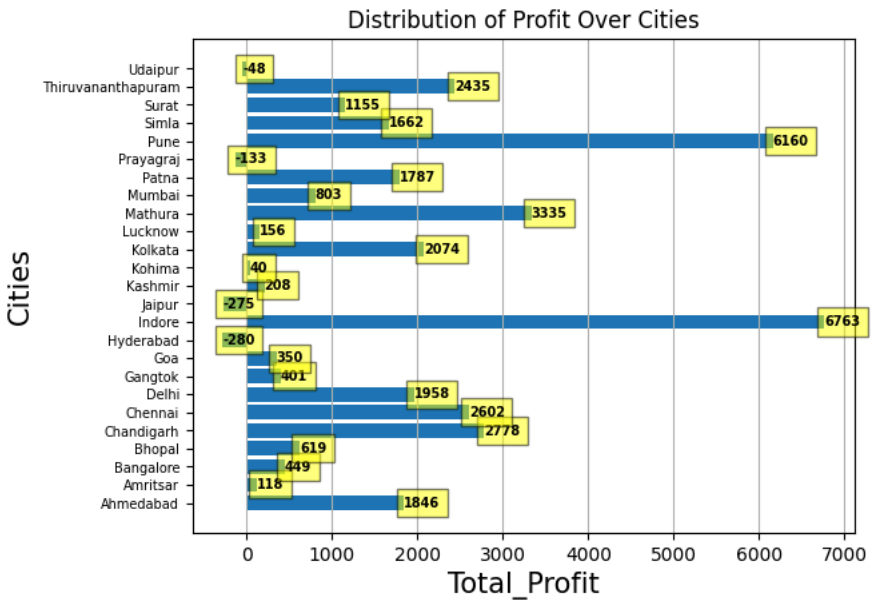

```
cities_profit = combined_data.groupby('City').sum()['Profit']
cities_profit
```

```
City
Ahmedabad            1846
Amritsar              118
Bangalore             449
Bhopal                619
Chandigarh           2778
Chennai              2602
Delhi                1958
Gangtok               401
Goa                   350
Hyderabad            -280
Indore               6763
Jaipur               -275
Kashmir               208
Kohima                 40
Kolkata              2074
Lucknow               156
Mathura              3335
Mumbai                803
Patna                1787
Prayagraj            -133
Pune                 6160
Simla                1662
Surat                1155
Thiruvananthapuram   2435
Udaipur               -48
Name: Profit, dtype: int64
```

```
cities_profit_list = []
for i in range(0,len(cities_profit)):
    cities_profit_list.append(cities_profit.values[i])
cities_profit_list
```

```
[1846,
 118,
 449,
 619,
 2778,
 2602,
 1958,
 401,
 350,
 -280,
 6763,
 -275,
 208,
 40,
 2074,
 156,
 3335,
 803,
 1787,
 -133,
 6160,
 1662,
 1155,
 2435,
 -48]
```

```
plt.barh(cities,cities_profit_list)
for index, value in enumerate(cities_profit_list):
    plt.text(value, index,str(value),size = 7,verticalalignment='center',weight = 'bold',bbox=dict(facecolor='yellow', alpha=0.5))
plt.title('Distribution of Profit Over Cities')
plt.yticks(fontsize = 7)
plt.xlabel('Total_Profit',fontsize = 15)
plt.ylabel('Cities',fontsize = 15)
plt.grid(axis = 'x')
plt.show()
```



Conclusion_1

Above plot helps us understand that Indore is the city with highes sales and it also generated most profit. Although a significant sales amount was recorded at shops in cities Jaipur and Hyderabad, these shops failed to record any profit. Apart from these two, loss was recorded at Prayagraj & Udaipur.

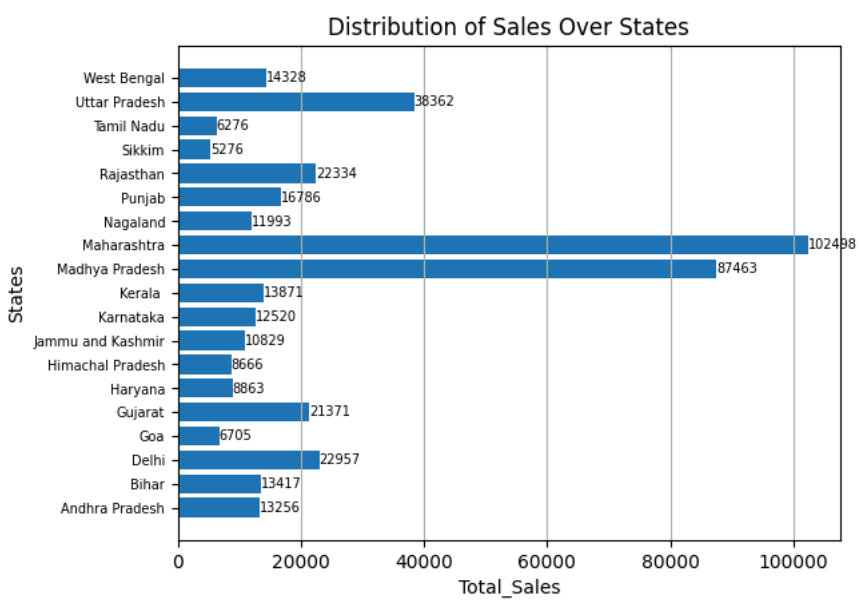
```
states = [state for state,something in combined_data.groupby('State')]
print(states)
```

```
['Andhra Pradesh', 'Bihar', 'Delhi', 'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir', 'Karnataka', 'Kerala ', 'Madhya Pradesh
```

```
state_sales = combined_data.groupby('State').sum()['Amount']
state_sales_list = []
for i in range(0,len(state_sales)):
    state_sales_list.append(state_sales.values[i])
print(state_sales_list)
```

```
[13256, 13417, 22957, 6705, 21371, 8863, 8666, 10829, 12520, 13871, 87463, 102498, 11993, 16786, 22334, 5276, 6276, 38362, 14328]
```

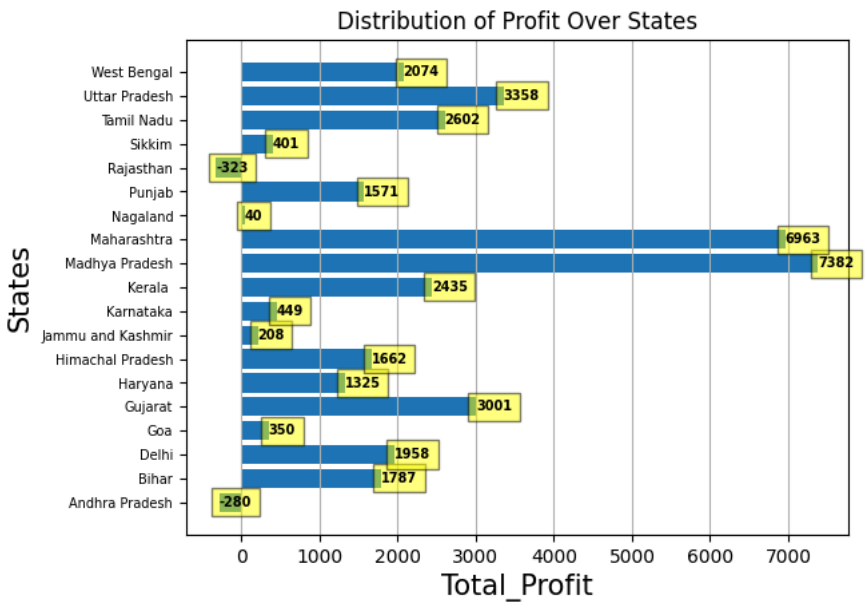
```
plt.barh(states,state_sales_list)
for index, value in enumerate(state_sales_list):
    plt.text(value, index,str(value),size = 7,verticalalignment='center')
plt.title('Distribution of Sales Over States')
plt.yticks(fontsize = 7)
plt.xlabel('Total_Sales',fontsize = 10)
plt.ylabel('States',fontsize = 10)
plt.grid(axis = 'x')
plt.show()
```




```
state_profit = combined_data.groupby('State').sum()['Profit']
state_profit_list = []
for i in range(0,len(state_profit)):
    state_profit_list.append(state_profit.values[i])
print(state_profit_list)
```

```
[-280, 1787, 1958, 350, 3001, 1325, 1662, 208, 449, 2435, 7382, 6963, 40, 1571, -323, 401, 2602, 3358, 2074]
```

```
plt.barh(states,state_profit_list)
for index, value in enumerate(state_profit_list):
    plt.text(value, index,str(value),size = 7,verticalalignment='center',weight = 'bold',bbox=dict(facecolor='yellow', alpha=0.5))
plt.title('Distribution of Profit Over States')
plt.yticks(fontsize = 7)
plt.xlabel('Total_Profit',fontsize = 15)
plt.ylabel('States',fontsize = 15)
plt.grid(axis = 'x')
plt.show()
```



Conclusion_2

Above plots indicate that Maharashtra is the most state with most sales, but the most profit was earned by the state of Madhya pradesh. Shops in the staet of Rajsthan and Andhra Pradesh failed to earn any profit.

```
mode_of_payment = [mop for mop,something in combined_data.groupby('PaymentMode')]
mode_of_payment
```

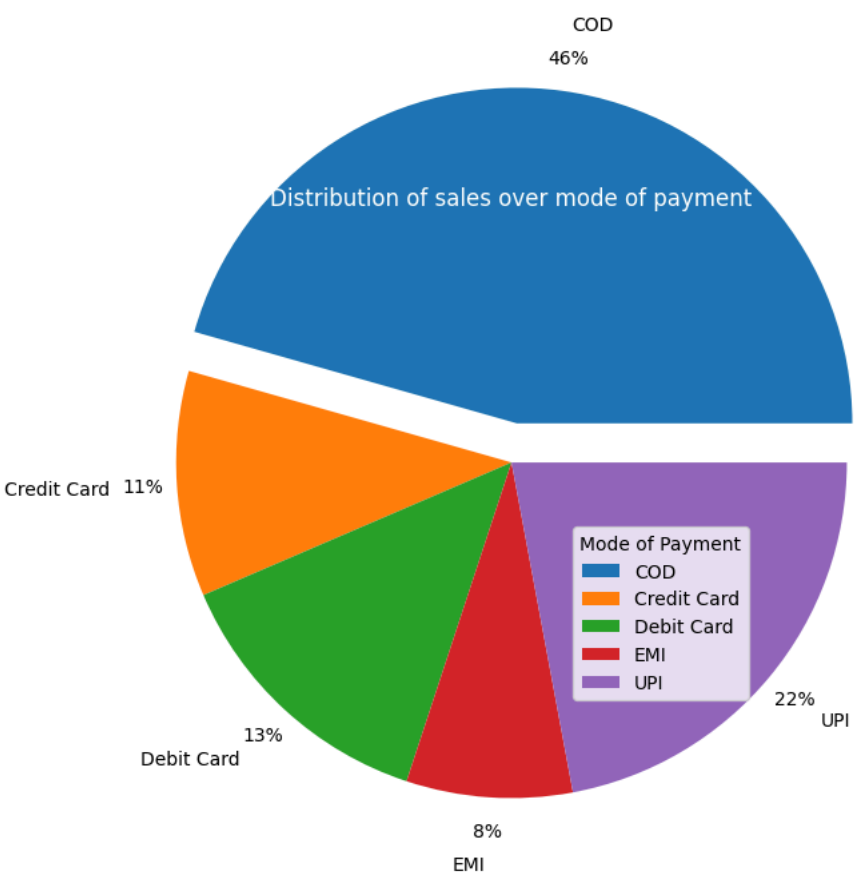
```
['COD', 'Credit Card', 'Debit Card', 'EMI', 'UPI']
```

```
payment_mode = combined_data.groupby('PaymentMode').count()['Amount']
payment_mode_list = []
for i in range(0,len(payment_mode)):
    payment_mode_list.append(payment_mode.values[i])
payment_mode_list
```

```
[684, 163, 202, 120, 331]
```

```
myexplode_max = payment_mode_list.index(max(payment_mode_list))
myexplode = []
for i in range(0,len(payment_mode_list)):
    if i == myexplode_max:
        myexplode.append(0.2)
    else:
        myexplode.append(0)

plt.pie(payment_mode_list,labels = mode_of_payment,autopct='%1.0f%%',pctdistance=1.1, labeldistance=1.2,explode = myexplode,radius = 1.7)
plt.legend(mode_of_payment,title = 'Mode of Payment',loc = 'lower right')
plt.title('Distribution of sales over mode of payment',color = 'White')
plt.show()
```



Conclusion 3

Above plot help us understand that altghough all the methods of payment were available, most people still preferred the cash on delivery option. UPI, being the second most prevalent mode of payment.

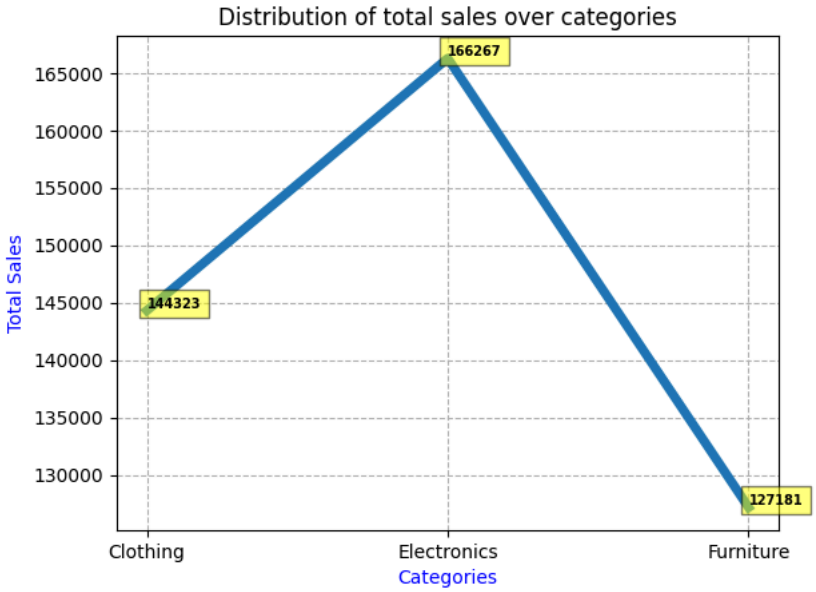
```
category = [category for category,something in combined_data.groupby('Category')]
category
```

```
['Clothing', 'Electronics', 'Furniture']
```

```
cat_sales = combined_data.groupby('Category').sum()['Amount']
cat_sales_list = []
for i in range(0,len(cat_sales)):
    cat_sales_list.append(cat_sales.values[i])
cat_sales_list
```

```
[144323, 166267, 127181]
```

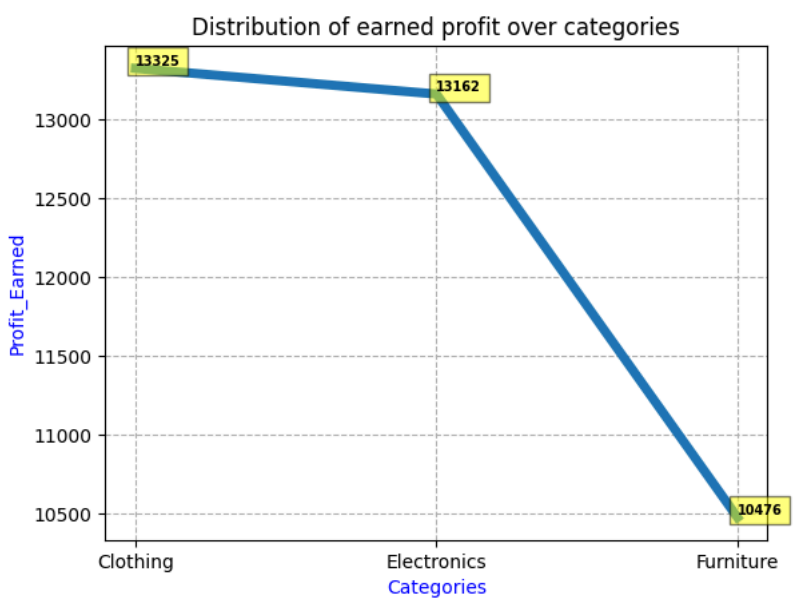
```
plt.plot(category,cat_sales_list,linewidth = 5)
for index,value in enumerate(cat_sales_list):
    plt.text(index,value,str(value),size = 7,verticalalignment='bottom',weight = 'bold',bbox=dict(facecolor='yellow', alpha=0.5))
plt.grid(axis = 'both',linestyle = '--')
plt.title('Distribution of total sales over categories')
plt.xlabel('Categories',color = 'blue')
plt.ylabel('Total Sales',color = 'blue')
plt.show()
```




```
cat_profit = combined_data.groupby('Category').sum()['Profit']
cat_profit_list = []
for i in range(0,len(cat_profit)):
    cat_profit_list.append(cat_profit.values[i])
cat_profit_list
```

```
[13325, 13162, 10476]
```

```
plt.plot(category,cat_profit_list,linewidth = 5)
for index,value in enumerate(cat_profit_list):
    plt.text(index,value,str(value),size = 7,verticalalignment='bottom',weight = 'bold',bbox=dict(facecolor='yellow', alpha=0.5))
plt.grid(axis = 'both',linestyle = '--')
plt.title('Distribution of earned profit over categories')
plt.xlabel('Categories',color = 'blue')
plt.ylabel('Profit_Earned',color = 'blue')
plt.show()
```



Conclusion 4

Above plots help us understand that the most profitable category was clothing while the category of electronics recorded the most sales.

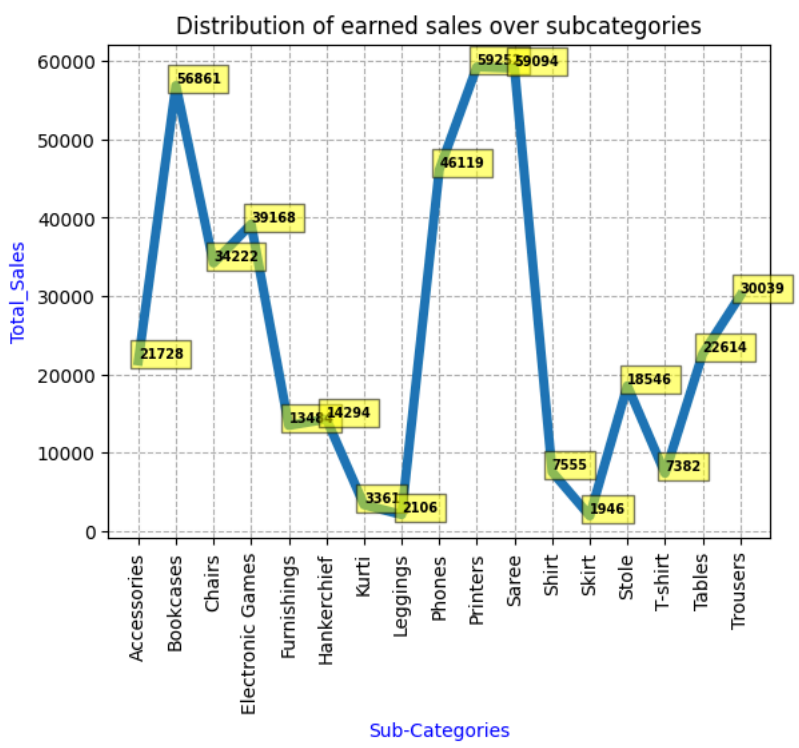
```
subcategory = [subcategory for subcategory,something in combined_data.groupby('Sub-Category')]
subcategory
```

```
['Accessories',
 'Bookcases',
 'Chairs',
 'Electronic Games',
 'Furnishings',
 'Hankerchief',
 'Kurti',
 'Leggings',
 'Phones',
 'Printers',
 'Saree',
 'Shirt',
 'Skirt',
 'Stole',
 'T-shirt',
 'Tables',
 'Trousers']
```

```
subcat_sales = combined_data.groupby('Sub-Category').sum()['Amount']
subcat_sales_list = []
for i in range(0,len(subcat_sales)):
    subcat_sales_list.append(subcat_sales.values[i])
subcat_sales_list
```

```
[21728,
 56861,
 34222,
 39168,
 13484,
 14294,
 3361,
 2106,
 46119,
 59252,
 59094,
 7555,
 1946,
 18546,
 7382,
 22614,
 30039]
```

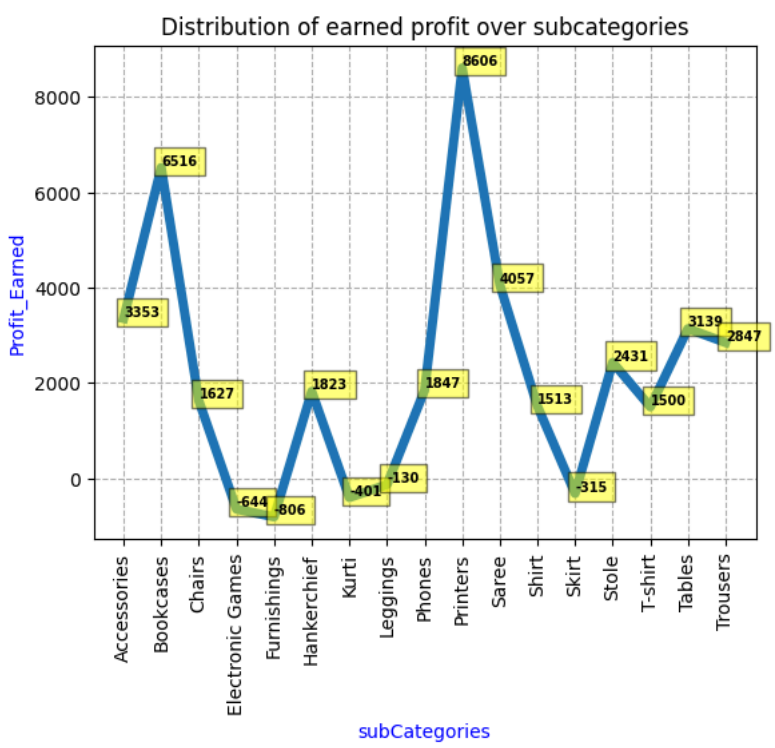
```
plt.plot(subcategory,subcat_sales_list,linewidth = 5)
for index,value in enumerate(subcat_sales_list):
    plt.text(index,value,str(value),size = 7,verticalalignment='bottom',weight = 'bold',bbox=dict(facecolor='yellow', alpha=0.5))
plt.grid(axis = 'both',linestyle = '--')
plt.xticks(rotation = 'vertical')
plt.title('Distribution of earned sales over subcategories')
plt.xlabel('Sub-Categories',color = 'blue')
plt.ylabel('Total_Sales',color = 'blue')
plt.show()
```




```
subcat_profit = combined_data.groupby('Sub-Category').sum()['Profit']
subcat_profit_list = []
for i in range(0,len(subcat_profit)):
    subcat_profit_list.append(subcat_profit.values[i])
subcat_profit_list
```

```
[3353,
 6516,
 1627,
 -644,
 -806,
 1823,
 -401,
 -130,
 1847,
 8606,
 4057,
 1513,
 -315,
 2431,
 1500,
 3139,
 2847]
```

```
plt.plot(subcategory,subcat_profit_list,linewidth = 5)
for index,value in enumerate(subcat_profit_list):
    plt.text(index,value,str(value),size = 7,verticalalignment='bottom',weight = 'bold',bbox=dict(facecolor='yellow', alpha=0.5))
plt.grid(axis = 'both',linestyle = '--')
plt.title('Distribution of earned profit over subcategories')
plt.xticks(rotation = 'vertical')
plt.xlabel('subCategories',color = 'blue')
plt.ylabel('Profit_Earned',color = 'blue')
plt.show()
```



Conclusion 5

Here we can see that the most sales were recorded for three subcategories - sarees, printers and bookcases. However, the most profitable subcategory was printers.

Below is some additional exploration I have done in the data.

```
combined_data['Month'] = combined_data['Order_Date'].str[3:5].astype('int64')
combined_data
```

	Order_ID	Order_Date	CustomerName	State	City	Amount	Profit	Quantity	Categ
0	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	5729	64	14	Furnit
1	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	671	114	9	Electroı

```
combined_data['Quarter'] = ((combined_data['Month']-1)//3)+1
combined_data
```

	Order_ID	Order_Date	CustomerName	State	City	Amount	Profit	Quantity	Categ
0	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	5729	64	14	Furnit
1	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	671	114	9	Electroı
2	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	443	11	1	Cloth
3	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	57	7	2	Cloth
4	B-26055	10-03-2018	Harivansh	Uttar Pradesh	Mathura	227	48	5	Cloth
...	...	...	...	...	...	...	...	...	
1495	B-25742	03-08-2018	Ashwin	Goa	Goa	11	-8	2	Cloth
1496	B-26088	26-03-2018	Bhavna	Sikkim	Gangtok	11	5	2	Cloth

```
quarter_list = [quarter for quarter,something in combined_data.groupby('Quarter')]
quarter_list
```

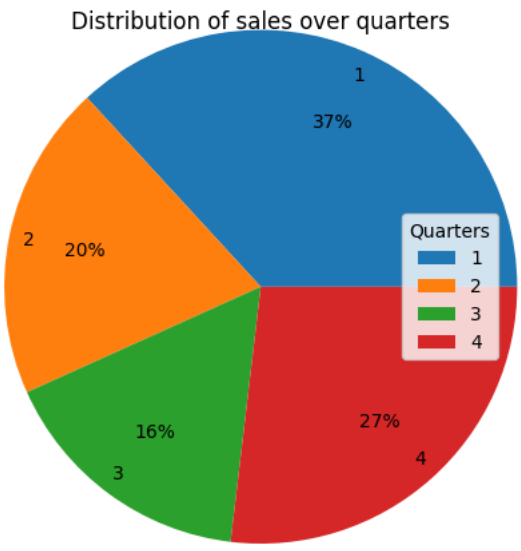
```
[1, 2, 3, 4]
```

```
quarter_group = combined_data.groupby('Quarter').sum()['Amount']
quarter_group_list = []
for i in range(0,len(quarter_group)):
    quarter_group_list.append(quarter_group.values[i])

print(quarter_group_list)
```

```
[161288, 87081, 71741, 117661]
```

```
plt.pie(quarter_group_list,labels = quarter_list,autopct='%1.0f%%',pctdistance=0.7, labeldistance=0.9,radius = 1.3)
plt.legend(quarter_list,title = 'Quarters',loc = 'right')
plt.title('Distribution of sales over quarters',color = 'Black')
plt.show()
```



As we can see from the above plot, the highes sales occured in the first quarter of the year.