The Spark Foundation

Author : Prakash Kumar Gupta

TASK:- "Exploratory Data Analysis" - Retail

EDA refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and check assumptions with the help of summary statistics and graphical representations.

DataSet

To perform EDA on the given dataset I am going to perform certain steps to explore data and will find out the weak areas where a business manager can work to make more profit and to know all the problems occuring in the business.

The steps are:

01 - Data Exploration

02 - Data Cleaning

03 - Data Grouping

04 - Data Visualization

In [5]:

```
#importing useful libraries
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

1. Data Exploration

Data exploration is an approach similar to initial data analysis, whereby a data analyst uses visual exploration to understand what is in a dataset and the characteristics of the data, rather than through traditional data management systems.

In [16]:

```
df = pd.read_csv("F:/SampleSuperstore.csv")
df.head()
```

Out[16]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage

Dropping unnecessary columns

In [18]:

```
df.drop(['Postal Code' , 'Quantity' , 'Discount' , 'Ship Mode'], axis='columns', inplace=True)
df.head()
```

Out[18]:

	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Pro
0	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	41.91
1	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.9400	219.58
2	Corporate	United States	Los Angeles	California	West	Office Supplies	Labels	14.6200	6.87
3	Consumer	United States	Fort Lauderdale	Florida	South	Furniture	Tables	957.5775	-383.03
4	Consumer	United States	Fort Lauderdale	Florida	South	Office Supplies	Storage	22.3680	2.51

Checking data structure

In [19]:

```
df.shape
```

Out[19]:

```
(9994, 9)
```

In [20]:

```
#printing columns name
df.columns
```

Out[20]:

```
Index(['Segment', 'Country', 'City', 'State', 'Region', 'Category',
       'Sub-Category', 'Sales', 'Profit'],
      dtype='object')
```

Summary of Data

In [21]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Segment       9994 non-null   object 
 1   Country       9994 non-null   object 
 2   City          9994 non-null   object 
 3   State         9994 non-null   object 
 4   Region        9994 non-null   object 
 5   Category      9994 non-null   object 
 6   Sub-Category  9994 non-null   object 
 7   Sales         9994 non-null   float64
 8   Profit        9994 non-null   float64
dtypes: float64(2), object(7)
memory usage: 702.8+ KB
```

In [22]:

```
df.dtypes
```

Out[22]:

```
Segment          object
Country          object
City             object
State            object
Region           object
Category         object
Sub-Category     object
Sales           float64
Profit          float64
dtype: object
```

In [23]:

```
#Describing statistical information on the dataset
df.describe()
```

Out[23]:

	Sales	Profit
count	9994.000000	9994.000000
mean	229.858001	28.656896
std	623.245101	234.260108
min	0.444000	-6599.978000
25%	17.280000	1.728750
50%	54.490000	8.666500
75%	209.940000	29.364000
max	22638.480000	8399.976000

In [24]:

```
corr = df.corr()
corr
```

Out[24]:

	Sales	Profit
Sales	1.000000	0.479064
Profit	0.479064	1.000000

2. Data Cleanning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

Checking for invalid values

In [25]:

```
df.isnull()
```

Out[25]:

	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Profit
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
9989	False	False	False	False	False	False	False	False	False
9990	False	False	False	False	False	False	False	False	False
9991	False	False	False	False	False	False	False	False	False
9992	False	False	False	False	False	False	False	False	False
9993	False	False	False	False	False	False	False	False	False

9994 rows × 9 columns

False

shows our dataset has no invalid value in it. If our dataset would have invalid values, we would use df.dropna(inplace=True) Since, we don't have such value our data is already clean.

Print clean dataset

In [26]:

```
df.head(3)
```

Out[26]:

	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Profit
0	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.96	41.9136
1	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.94	219.5820
2	Corporate	United States	Los Angeles	California	West	Office Supplies	Labels	14.62	6.8714

Adding additional column to dataset

In [27]:

```
# Adding column Revenue to dataset
df['Revenue'] = df['Sales'] - df['Profit']
df.head(3)
```

Out[27]:

	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Profit
0	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.96	41.9136
1	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.94	219.5820
2	Corporate	United States	Los Angeles	California	West	Office Supplies	Labels	14.62	6.8714

3. Data Grouping

Data grouping is done by groupby() function, which is used to split the data into groups based on some criteria. The abstract definition of grouping is to provide a mapping of labels to group names. It is done to make sense of variables using combinations.

Making combinations, grouping variables to know the weak area.

In [31]:

```
df[['Sales','State','Profit']].groupby(['State'],as_index=True).sum().sort_values(by='P
rofit',ascending=True)
```

Out[31]:

State	Sales	Profit
Texas	170188.0458	-25729.3563
Ohio	78258.1360	-16971.3766
Pennsylvania	116511.9140	-15559.9603
Illinois	80166.1010	-12607.8870
North Carolina	55603.1640	-7490.9122
Colorado	32108.1180	-6527.8579
Tennessee	30661.8730	-5341.6936
Arizona	35282.0010	-3427.9246
Florida	89473.7080	-3399.3017
Oregon	17431.1500	-1190.4705
Wyoming	1603.1360	100.1960
West Virginia	1209.8240	185.9216
North Dakota	919.9100	230.1497
South Dakota	1315.5600	394.8283
Maine	1270.5300	454.4862
Idaho	4382.4860	826.7231
Kansas	2914.3100	836.4435
District of Columbia	2865.0200	1059.5893
New Mexico	4783.5220	1157.1161
Iowa	4579.7600	1183.8119
New Hampshire	7292.5240	1706.5028
South Carolina	8481.7100	1769.0566
Montana	5589.3520	1833.3285
Nebraska	7464.9300	2037.0942
Louisiana	9217.0300	2196.1023
Vermont	8929.3700	2244.9783
Utah	11220.0560	2546.5335
Mississippi	10771.3400	3172.9762
Nevada	16729.1020	3316.7659
Connecticut	13384.3570	3511.4918
Arkansas	11678.1300	4008.6871
Oklahoma	19683.3900	4853.9560
Alabama	19510.6400	5786.8253
Missouri	22205.1500	6436.2105
Massachusetts	28634.4340	6785.5016
Maryland	23705.5230	7031.1788

State	Sales	Profit
Rhode Island	22627.9560	7285.6293
Wisconsin	32114.6100	8401.8004
New Jersey	35764.3120	9772.9138
Delaware	27451.0690	9977.3748
Minnesota	29863.1500	10823.1874
Kentucky	36591.7500	11199.6966
Georgia	49095.8400	16250.0433
Indiana	53555.3600	18382.9363
Virginia	70636.7200	18597.9504
Michigan	76269.6140	24463.1876
Washington	138641.2700	33402.6517
New York	310876.2710	74038.5486
California	457687.6315	76381.3871

By grouping sales and profit with states gives us a clear picture of areas or states manager should work on. Top starting 10 states shows a loss with Good to average sales.

In [32]:

```
df[['Category','Profit']].groupby(['Category'],as_index=True).sum().sort_values(by='Profit',ascending=True)
```

Out[32]:

Category	Profit
Furniture	18451.2728
Office Supplies	122490.8008
Technology	145454.9481

Here is another group of profit with category. It shows category Furniture gives lowest profit.

In [33]:

```
df[['Sub-Category','Profit']].groupby(['Sub-Category'],as_index=True).sum().sort_values
(by='Profit',ascending=True)
```

Out[33]:

Sub-Category	Profit
Tables	-17725.4811
Bookcases	-3472.5560
Supplies	-1189.0995
Fasteners	949.5182
Machines	3384.7569
Labels	5546.2540
Art	6527.7870
Envelopes	6964.1767
Furnishings	13059.1436
Appliances	18138.0054
Storage	21278.8264
Chairs	26590.1663
Binders	30221.7633
Paper	34053.5693
Accessories	41936.6357
Phones	44515.7306
Copiers	55617.8249

This combination is made to know the products which gives poor profit to business. We see tables, bookcases, supplies gives negative profit. By such groupings one thing is coming clear that business needs to work on category furniture to earn more profit.

In [34]:

```
df.groupby(["Category", "Sub-Category"], as_index=False)["Sales"].count()
```

Out[34]:

	Category	Sub-Category	Sales
0	Furniture	Bookcases	228
1	Furniture	Chairs	617
2	Furniture	Furnishings	957
3	Furniture	Tables	319
4	Office Supplies	Appliances	466
5	Office Supplies	Art	796
6	Office Supplies	Binders	1523
7	Office Supplies	Envelopes	254
8	Office Supplies	Fasteners	217
9	Office Supplies	Labels	364
10	Office Supplies	Paper	1370
11	Office Supplies	Storage	846
12	Office Supplies	Supplies	190
13	Technology	Accessories	775
14	Technology	Copiers	68
15	Technology	Machines	115
16	Technology	Phones	889

This combination is made to know how much sales impact on profit from these category and subcategory.

4. Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In [35]:

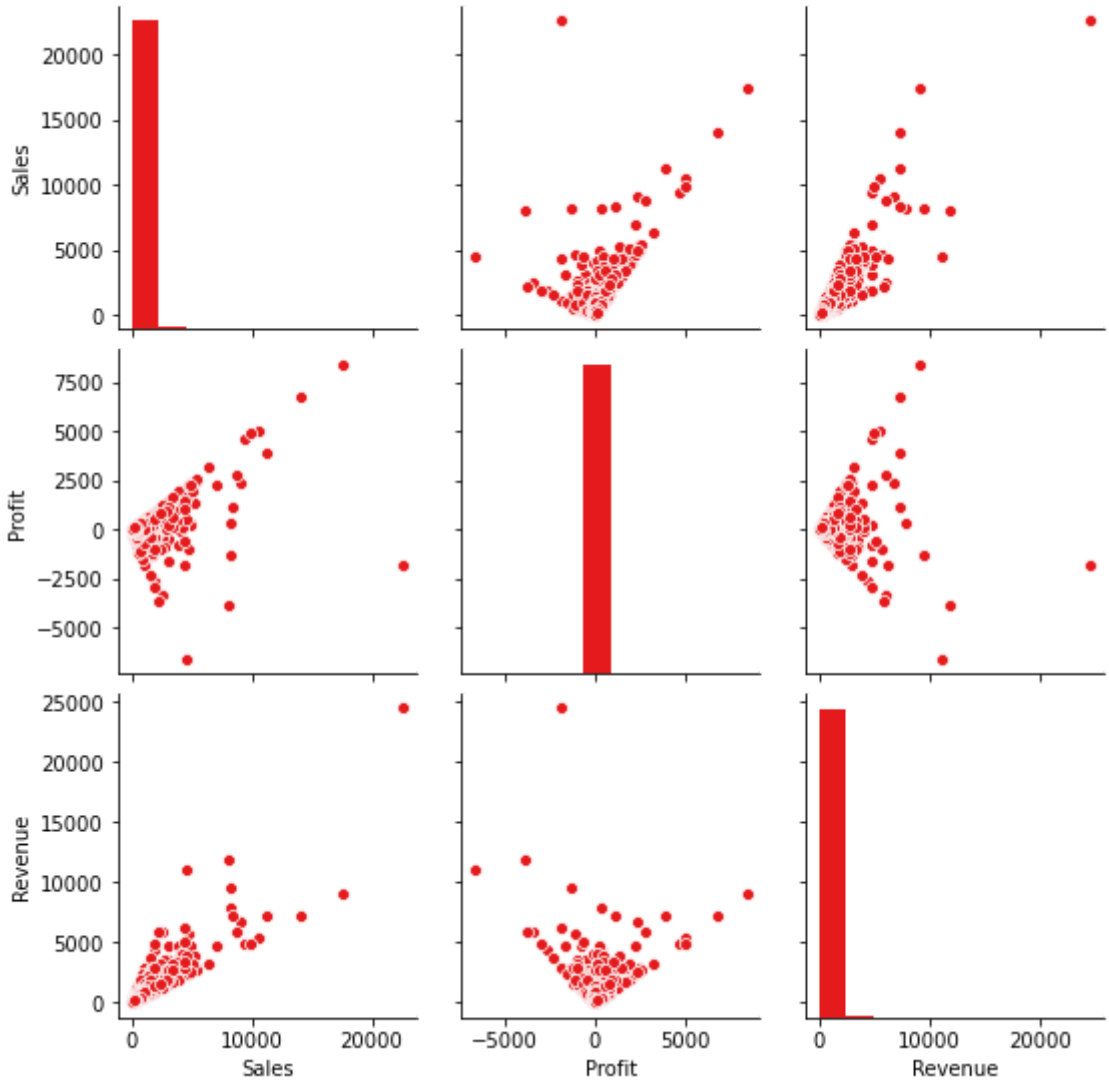
```
# Importing visualization libraries
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [36]:

```
sns.set_palette("Set1")
sns.pairplot(df)
```

Out[36]:

```
<seaborn.axisgrid.PairGrid at 0x17bba7887c0>
```

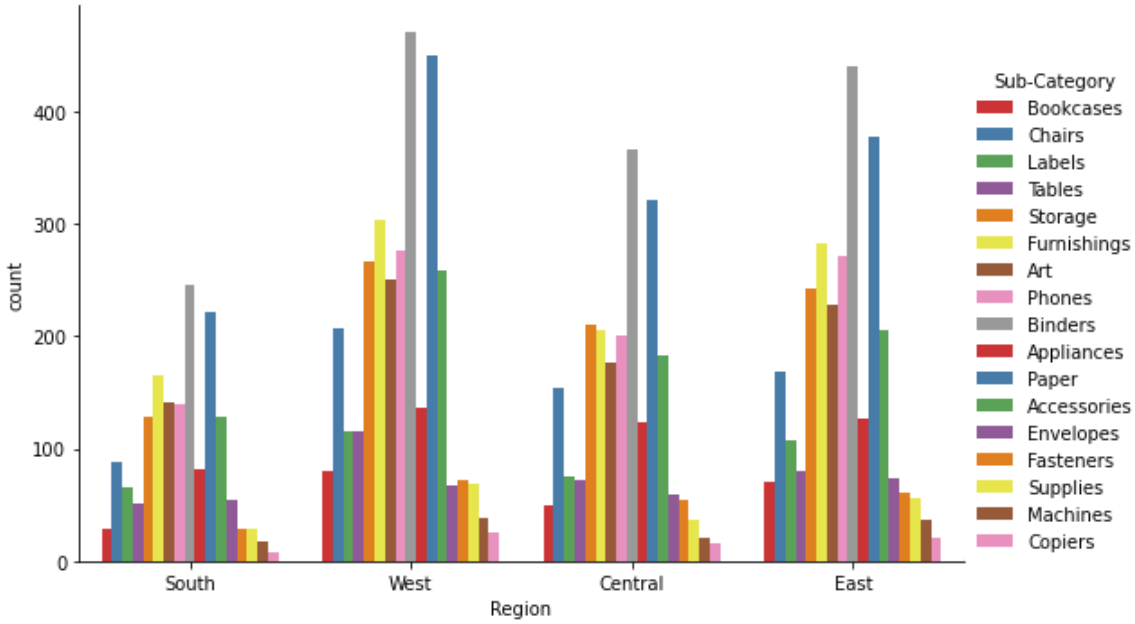


In [37]:

```
sns.catplot("Region", hue="Sub-Category", data=df, kind="count", aspect=1.5, palette="Set1")
```

Out[37]:

```
<seaborn.axisgrid.FacetGrid at 0x17bbdf3df70>
```



In [38]:

```
sns.heatmap(corr,annot=True)
```

Out[38]:

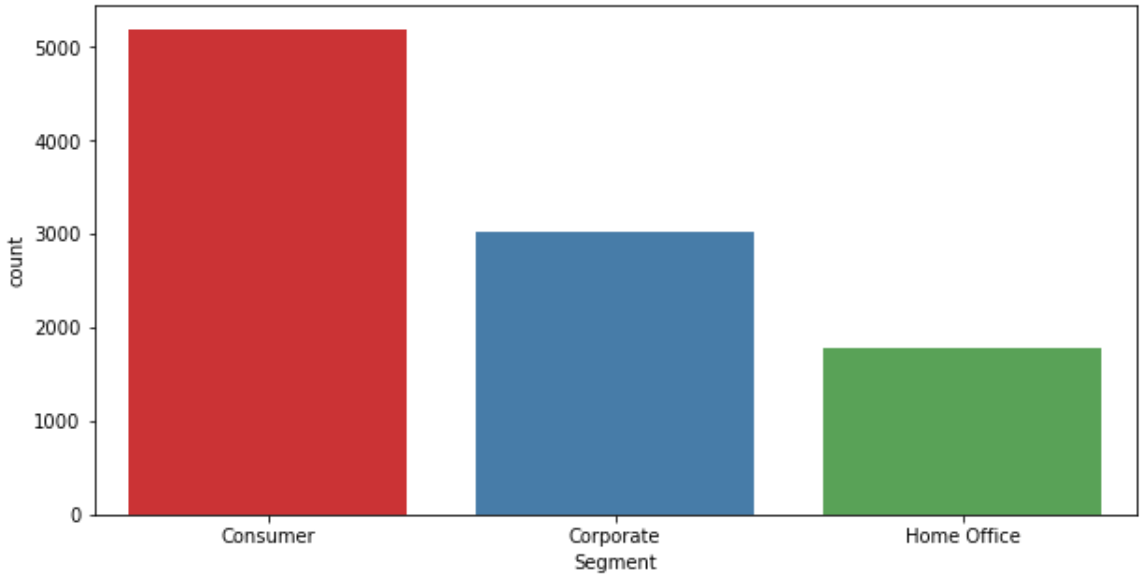
```
<matplotlib.axes._subplots.AxesSubplot at 0x17bbe2db370>
```

In [39]:

```
plt.figure(figsize=(10,5))
sns.countplot(x=df['Segment'])
```

Out[39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x17bbe449610>
```

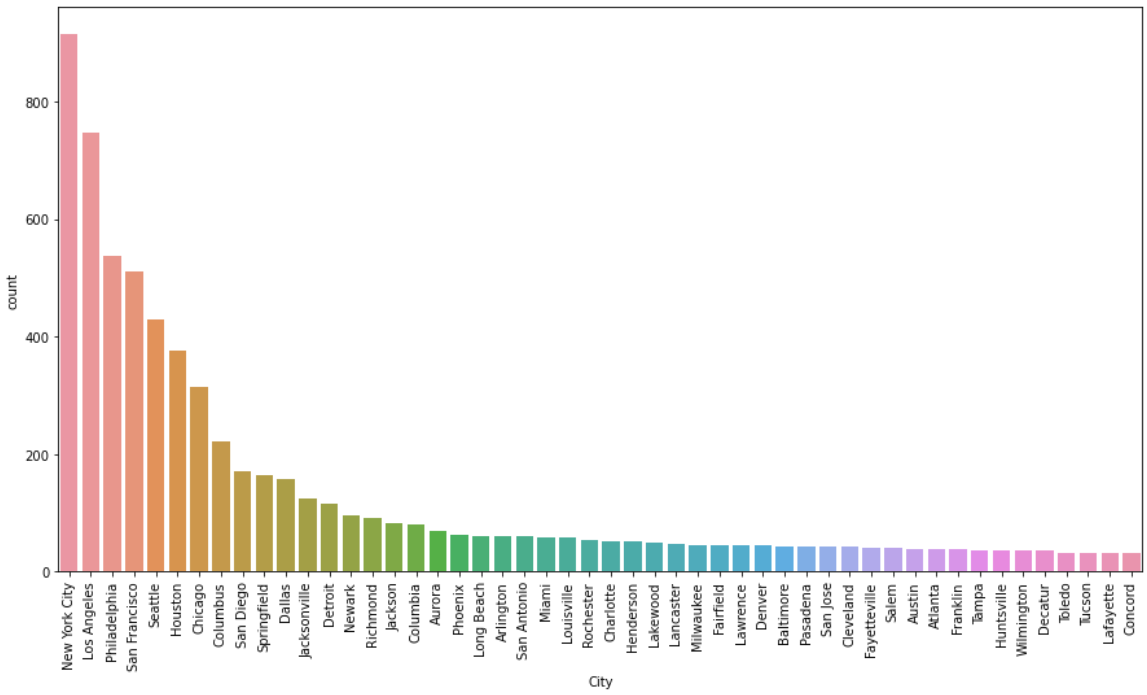


In [40]:

```
plt.figure(figsize=(15,8))
sns.countplot(x=df['City'], order=(df['City'].value_counts().head(50)).index)
plt.xticks(rotation=90)
```

Out[40]:

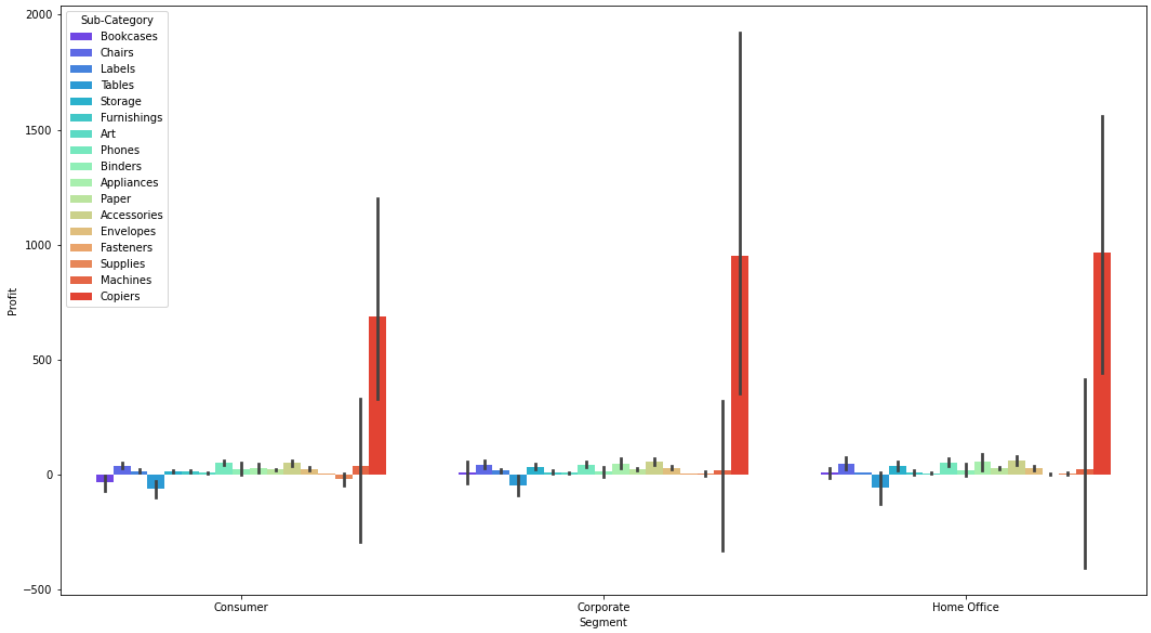
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 1
6,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 3
3,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]),
 <a list of 50 Text major ticklabel objects>)
```



Graph is representing all cities where retail business takes place from highest count to lowest.

In [41]:

```
plt.figure(figsize=[18,10])
ax = sns.barplot(x="Segment", y="Profit", hue="Sub-Category", data=df, palette="rainbo
w")
```



This visualization is done to check which product provides the highest/lowest profit in each sement.

In [43]:

```
plt.figure(figsize=(12,4))
sns.barplot(x=df['Category'], y=df['Profit'])
plt.xticks(rotation=90)
```

Out[43]:

```
(array([0, 1, 2]), <a list of 3 Text major ticklabel objects>)
```

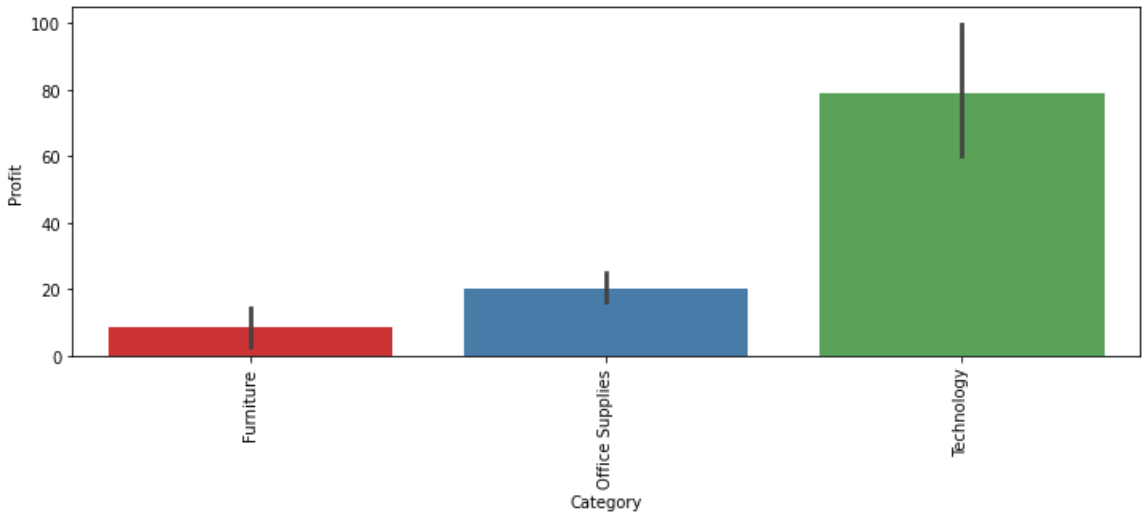
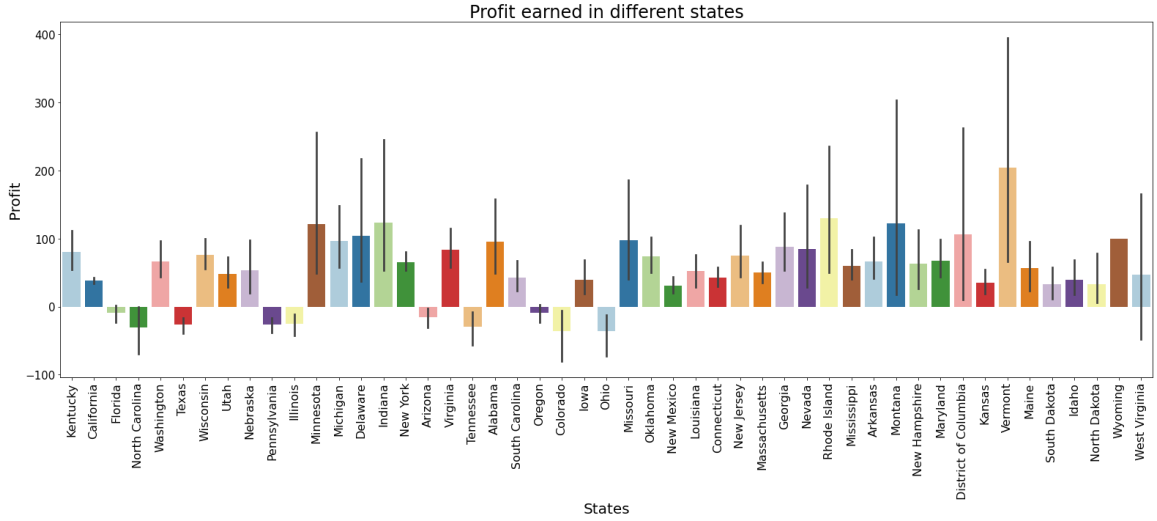


Figure shows how much profit is earned with each category. Maximum profit is earned by Technology and least by Furniture.

In [44]:

```
plt.figure(figsize=[22,10])
ax = sns.barplot(x="State", y="Profit", data=df, palette="Paired")
plt.xticks(rotation=90, fontsize=16)
plt.yticks(fontsize=15)
plt.title("Profit earned in different states",fontsize=24)
plt.xlabel("States",fontsize=20)
plt.ylabel("Profit",fontsize=20)
plt.tight_layout()
```



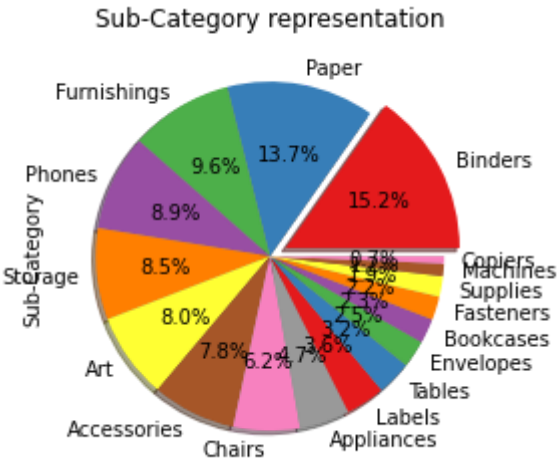
Profit/Loss earned in different states. Business manager should work on the states which gives loss and should look for the reason behind the loss.

In [45]:

```
S_Category=df["Sub-Category"].value_counts()
plt.get_cmap("hsv")
S_Category.plot.pie(autopct="%1.1f%%", shadow=True, explode=(0.1,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0))
plt.title("Sub-Category representation", fontsize=12)
```

Out[45]:

```
Text(0.5, 1.0, 'Sub-Category representation')
```



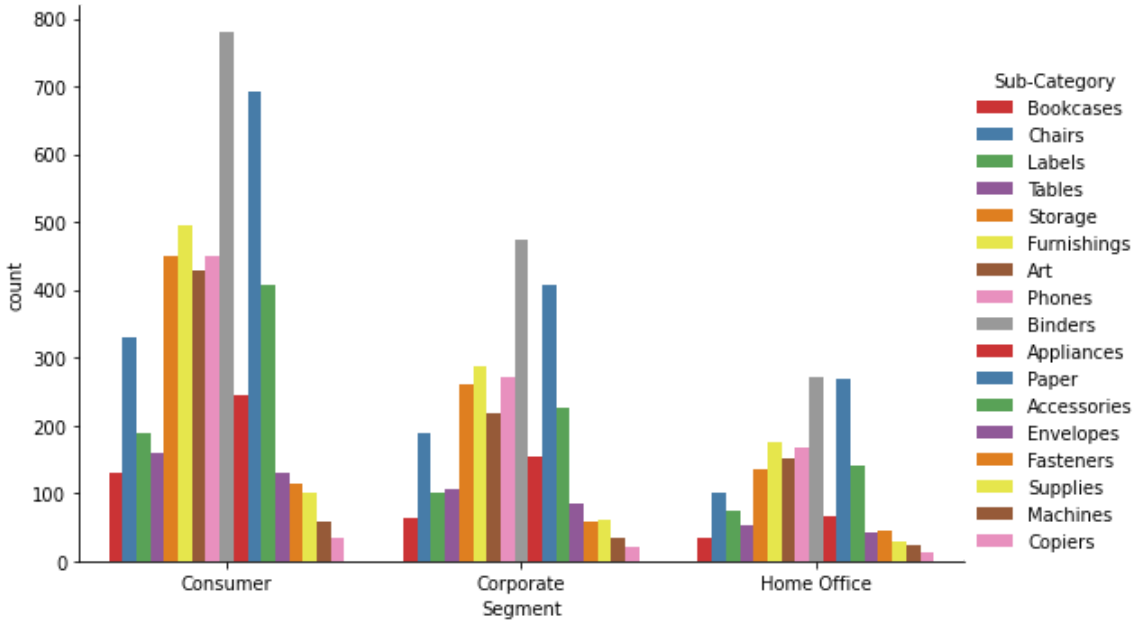
Representation of product(sub-category) sold in retail business.

In [46]:

```
sns.catplot("Segment", hue="Sub-Category", data=df, kind="count", aspect=1.5, palette=
"Set1")
```

Out[46]:

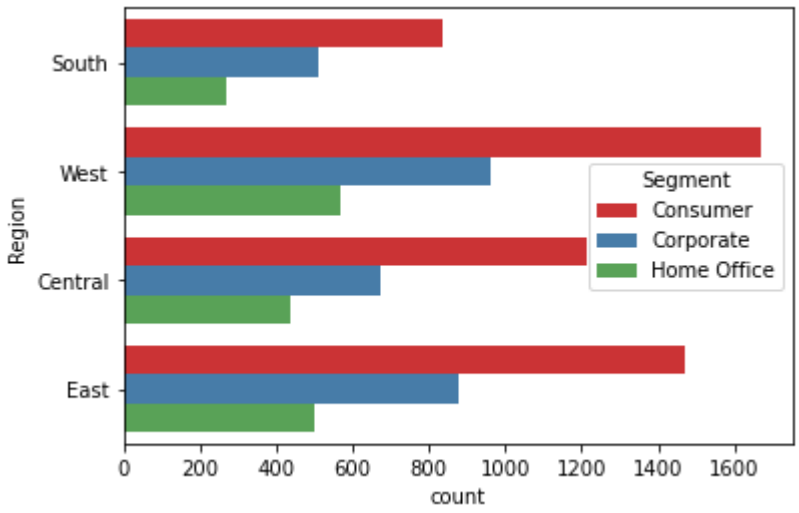
```
<seaborn.axisgrid.FacetGrid at 0x17bbe0fcbb0>
```



The figure shows subcategory count in each segment. By this representation manager will look into the areas where he needs to focus to improve sales and profit.

In [47]:

```
sns.countplot(y="Region", hue="Segment", data=df);
```

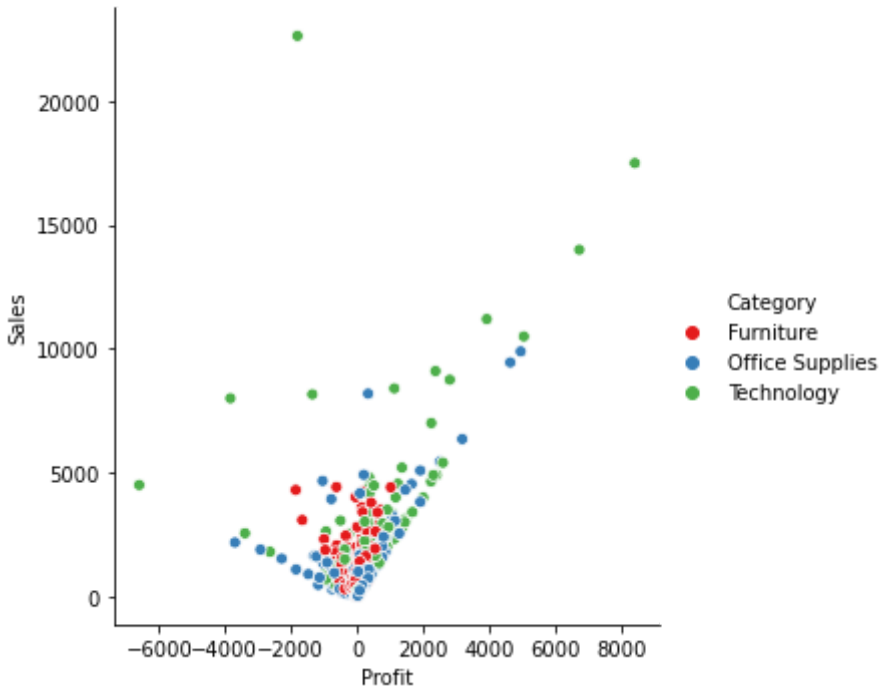


In [48]:

```
sns.relplot(x="Profit", y="Sales", hue="Category", data=df)
```

Out[48]:

```
<seaborn.axisgrid.FacetGrid at 0x17bbe62f040>
```



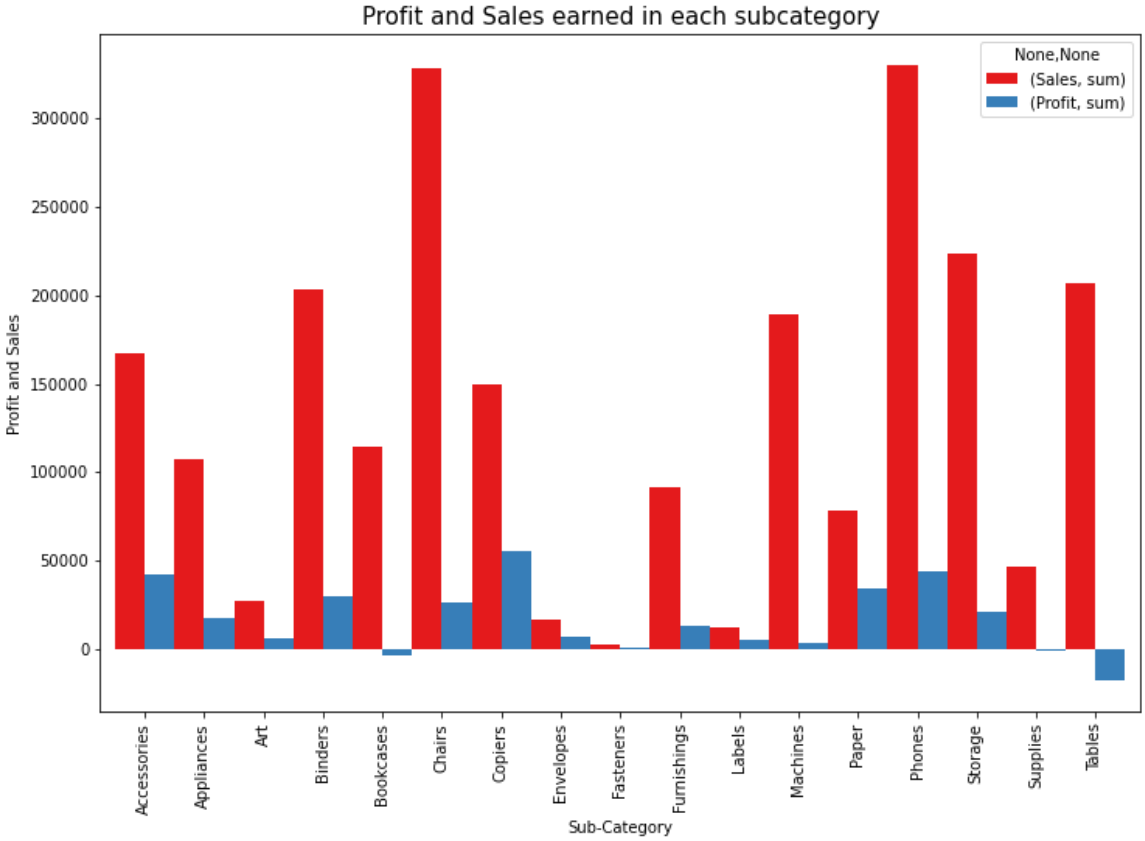
It shows how much profit is earned on sales in each category. As we can see high or less sales of a product has less impact on the profit.

In [49]:

```
s_p=df.groupby("Sub-Category")["Sales","Profit"].agg(["sum"])
s_p.plot.bar(width=1, figsize=(12,8))
plt.title("Profit and Sales earned in each subcategory", fontsize=15)
plt.xlabel("Sub-Category")
plt.ylabel("Profit and Sales")
```

Out[49]:

```
Text(0, 0.5, 'Profit and Sales')
```



The final representation of Profit and Sales of Products. In this we have a very clear picture of profit and sales. There is no specific margin of profit in respect to sales. If sales is very high of a product, it doesn't show extreme rise in profit and vice-versa. Profit increasing or decreasing with respect to the category of products and not much by sales of product.