

1. INTRODUCTION

1.1 Background:

Exploratory Data Analysis (EDA) is a critical step in the data analysis process. It is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. The goal of EDA is to gain a deeper understanding of the data, identify patterns and relationships, and uncover any anomalies or unexpected results.

EDA is typically performed using a combination of statistical methods and visualizations. This includes summarizing the distributions of variables, computing summary statistics (e.g mean, median, and standard deviation), and creating visualizations such as histograms, scatter plots, and box plots.

One of the key benefits of the EDA is that it allows you to quickly get a feel for your data and identify any potential issues or problems. For Example, if you find that one variable is highly skewed or has extreme outliers, you can adjust your analysis accordingly.

Overall, EDA is an important step in the data analysis process that helps you gain a deeper understanding of your data and identify areas for further investigation.

The following are the important reasons to perform EDA.

1. Understanding the data: EDA provides a deeper understanding of the data, including the distribution of variables, relationships between variables, and any anomalies between them.
2. Communication: EDA is an effective way to communicate the results of your analysis and share insights with others. By using visualization and statistical methods, you can effectively communicate the results of your analysis to stakeholders.
3. Saving time: By Performing EDA, you can identify issues with the data and adjust your analysis accordingly, which can save time and resources.

EDA is an iterative process that involves multiple rounds of visualizing and summarizing and transforming data.

The major steps involved in EDA are

1. Data Cleaning: This is also a part of data pre-processing. This includes removing missing values, removing outliers, and transforming variables.
2. Univariate Analysis: This is the study of distributions of individual variables to understand the central tendency and spread across the data. They can be visualized with Histogram plots and density plots.
3. Bivariate Analysis: This is the study of the relation between two variables. Scatter plots, Box plots are used in Bivariate Analysis. This helps to identify patterns in the data.
4. Multivariate Analysis: This is the study of relations between multiple variables. This is more achieved by my cluster analysis, Principal component analysis and is, Dimensionality Reduction.
5. Hypothesis Testing: Here, We make certain Hypotheses after visualizing the data and we try to test them.
6. Model Building: After the Assumptions are validated, now it's time to train the data and test the data. After many attempts, we may arrive at the required result. Now, the model is built.

Throughout the entire EDA, the most important step is Data visualization and to gain insights from the data. As this would help to understand the data better and helps to train the model accurately. For any Data Science Project / Application, EDA helps to gain a better view of the given data.

Sometimes, We may also have to pre-process the data, before we perform EDA. As data from the real world may not always be perfect and clean.

1.2 Problem Statement:

As a Business Manager, try to find out the weak areas where you can work to make a profit.

Perform ‘Exploratory Data Analysis on SampleSuperStore’.

Derive insights from the data and help the business to find its flaws and create dashboards to visualize the data more clearly.

You are free to choose (Python/R/Power BI/Tabulea/Excel/SAP/SAS) to Visualize the given data and create dashboards to understand and analyze the data.

1.3 Significance of the work

EDA helps to better understand the data. It helps to analyze the data more rapidly and accurately as we would be doing data pre-processing and cleaning the data.

We would also be creating Univariate, Bivariate and multivariate Analysis to better understand the relations between variables. This helps us to identify the variables that effect the model hugely and the variables which may impact less.

After creating dashboards, we would be able to clearly understand the data within no time and we would be able to discuss more on what to do next.

EDA mainly uses Statistical methods. So, we would be able to get more information on large datasets which in general clumsy to understand.

We will also use Principal Component Analysis, and Dimensionality Reduction which would help to remove the variables that are less significant.

2. System Analysis

2.1 Requirement Specification:

The following are the specifications required to complete the given task.

2.2.1 Functional Requirements:

Table: 2.2.1 Functional Requirements.

S.No	Name	Functional/Non-Functional	Usage
1	Access to Data	Functional	To Perform EDA, we need to have the data.
2	Data cleaning	Functional	Should have the ability to clean the data.
3	Data Visualization	Functional	Should have the ability to create dashboards.
4	Statistical Analysis	Functional	Should have the ability to perform statistical analysis on the data
5	Data Exploration	Functional	Should have the ability to identify relations.

2.2.2 System Requirements:

Software Requirements:

Operating System: Windows 11

IDE: Jupyter Notebook / Anaconda

Language: Python

Hardware Requirements:

- . Personal laptop
- . Installed RAM: 8 GB
- . Storage: 250 GB

2.2.3 Non-Functional Requirements:

Table 2.2.3 Non-Functional Requirements.

S.no	Name	Functional/ Non-Functional	Usage
1	Performance	Non-Functional	System should be efficient and handle large datasets.
2	User Experience	Non - Functional	Application had to be user-friendly for better usage.
3	Availability	Non- Functional	Applications had to be available before performing EDA.

3. System Design

3.1 Architecture of Proposed System:

1. Data Cleaning and Preparation:

This performs data cleaning and preparation. This remove missing values, converting categorical variables, handling outliers.

2. Data Storage:

This should contain the data either in local system/ remote repository/should get the data using APIs.

3. Data Visualizations:

This will visualize the data and create dashboards.

4. Statistical Analysis:

This will perform the statistical calculation on the given data set.

5. Communication:

This researcher should be able to communicate effectively the outcome of the EDA and help the stake holder to gain better understanding.

6. Access to Technical Support:

This should allow the researcher to look into documentation when stuck or when needed help.

3.2 Flow of Work.

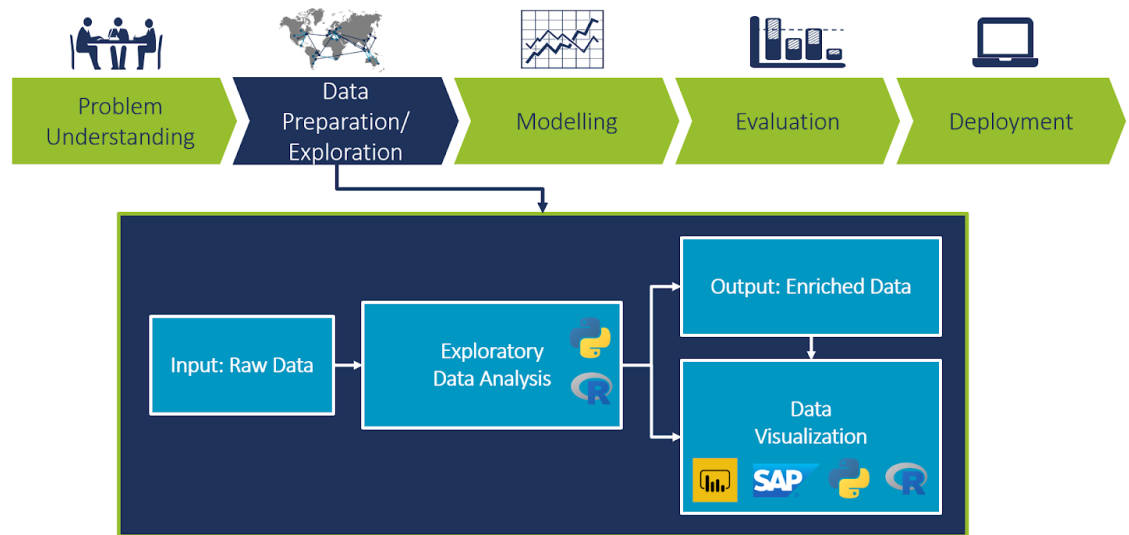


Fig: 3.2.1 Flow of work

EDA is used in the Data Preparation Stage. It helps to understand the data better. It is before the modeling phase. After Analyzing the data, we could accurately model the data.

4. Implementation And Testing

Step 1: Import all the Required Libraries.

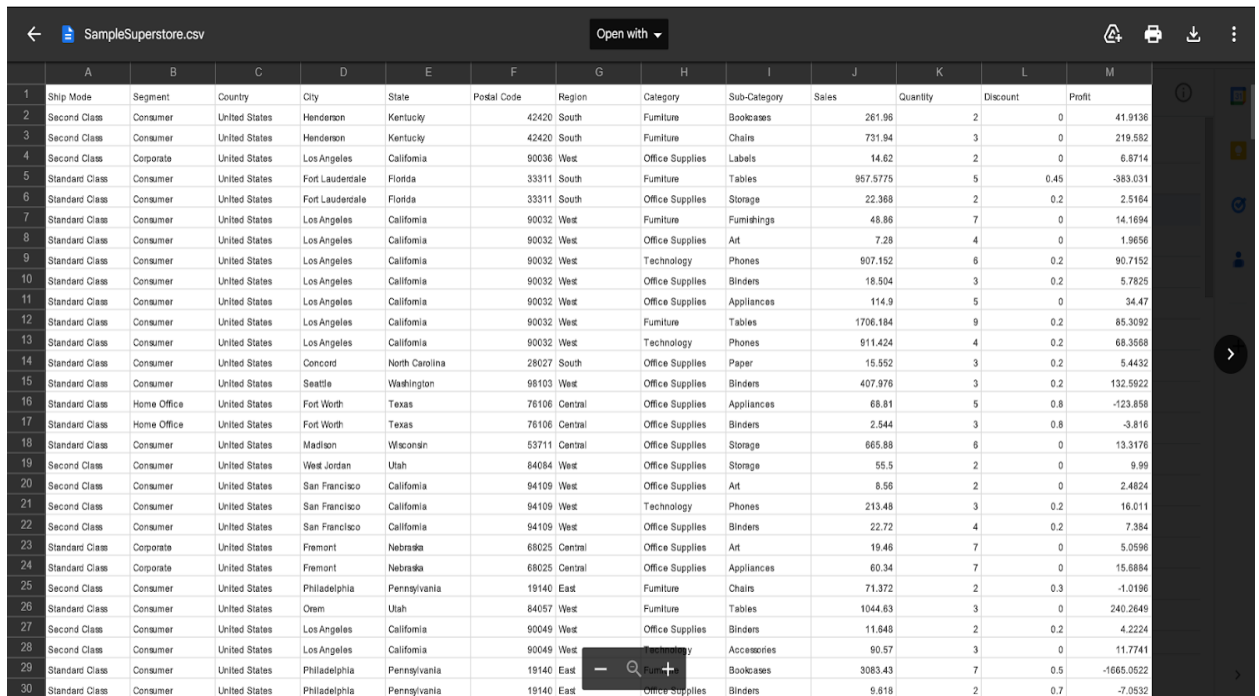
```
#filter out warning
import warnings
warnings.filterwarnings('ignore')

#importing the required libraries
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Fig: 4.1 Importing Libraries.

Step 2: Import the Data set.

The given dataset is:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
2	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.96	2	0	41.9136
3	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.94	3	0	219.582
4	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.62	2	0	6.8714
5	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.031
6	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.368	2	0.2	2.5164
7	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Furniture	Furnishings	48.86	7	0	14.1694
8	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Office Supplies	Art	7.28	4	0	1.9656
9	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Technology	Phones	907.152	6	0.2	90.7152
10	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Office Supplies	Binders	18.504	3	0.2	5.7826
11	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Office Supplies	Appliances	114.9	5	0	34.47
12	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Furniture	Tables	1706.184	9	0.2	85.3092
13	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Technology	Phones	911.424	4	0.2	68.3568
14	Standard Class	Consumer	United States	Concord	North Carolina	28027	South	Office Supplies	Paper	15.552	3	0.2	5.4432
15	Standard Class	Consumer	United States	Seattle	Washington	98103	West	Office Supplies	Binders	407.976	3	0.2	132.5922
16	Standard Class	Home Office	United States	Fort Worth	Texas	76106	Central	Office Supplies	Appliances	68.81	5	0.8	-123.858
17	Standard Class	Home Office	United States	Fort Worth	Texas	76106	Central	Office Supplies	Binders	2.544	3	0.8	-3.816
18	Standard Class	Consumer	United States	Madison	Wisconsin	53711	Central	Office Supplies	Storage	665.88	6	0	13.3176
19	Second Class	Consumer	United States	West Jordan	Utah	84084	West	Office Supplies	Storage	55.5	2	0	9.99
20	Second Class	Consumer	United States	San Francisco	California	94109	West	Office Supplies	Art	8.56	2	0	2.4824
21	Second Class	Consumer	United States	San Francisco	California	94109	West	Technology	Phones	213.48	3	0.2	16.011
22	Second Class	Consumer	United States	San Francisco	California	94109	West	Office Supplies	Binders	22.72	4	0.2	7.384
23	Standard Class	Corporate	United States	Fremont	Nebraska	68025	Central	Office Supplies	Art	19.46	7	0	5.0596
24	Standard Class	Corporate	United States	Fremont	Nebraska	68025	Central	Office Supplies	Appliances	60.34	7	0	15.8884
25	Second Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Furniture	Chairs	71.372	2	0.3	-1.0186
26	Standard Class	Consumer	United States	Orem	Utah	84057	West	Furniture	Tables	1044.63	3	0	240.2649
27	Second Class	Consumer	United States	Los Angeles	California	90049	West	Office Supplies	Binders	11.648	2	0.2	4.2224
28	Second Class	Consumer	United States	Los Angeles	California	90049	West	Office Supplies	Accessories	90.57	3	0	11.7741
29	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Bookcases	3063.43	7	0.5	-1665.0522
30	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Binders	9.618	2	0.7	-7.0532

Fig: 4.2 Dataset

View the imported data

```
# To read Data where we will get top 5 rows.  
retail.head()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

Fig: 4.3 Data Importing

The different columns in the data are:

```
# To check the columns of the data  
retail.columns
```

```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',  
      'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',  
      'Profit'],  
      dtype='object')
```

Fig: 4.4 Different Columns in Data

The data types of those columns are:

```
# To check the data type  
retail.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 9994 entries, 0 to 9993  
Data columns (total 13 columns):  
#   Column                Non-Null Count  Dtype    
---  ---                  
0   Ship Mode             9994 non-null   object   
1   Segment               9994 non-null   object   
2   Country               9994 non-null   object   
3   City                  9994 non-null   object   
4   State                 9994 non-null   object   
5   Postal Code           9994 non-null   int64    
6   Region                9994 non-null   object   
7   Category              9994 non-null   object   
8   Sub-Category          9994 non-null   object   
9   Sales                 9994 non-null   float64  
10  Quantity              9994 non-null   int64    
11  Discount              9994 non-null   float64  
12  Profit                9994 non-null   float64  
dtypes: float64(3), int64(2), object(8)  
memory usage: 1015.1+ KB
```

Fig: 4.5 Datatypes of columns

To check the size of a given dataset we can use:

```
# To check the structure of Dataset i.e the number of columns and rows  
retail.shape
```

```
(9994, 13)
```

Fig: 4.6 Size of data given

Data Preprocessing:

The first step in data pre-processing is to check for null values.

```
#To check for missing values in the data  
retail.isnull().sum()
```

```
Ship Mode      0  
Segment        0  
Country        0  
City           0  
State          0  
Postal Code    0  
Region         0  
Category       0  
Sub-Category   0  
Sales          0  
Quantity       0  
Discount       0  
Profit         0  
dtype: int64
```

Fig: 4.7 Checking for null values

From this we can derive that, There aren't any Null values in the dataset.

The next step is to check for duplicate values.

```
# To check whether we have any duplicacy in Dataset or not  
retail.duplicated().sum()
```

17

Fig: 4.8 checking for duplicate values.

Here, we find that we had 17 duplicate values. So, we had to remove those to analyze them better.

The duplicates can be dropped as below.

```
# Since we have 17 duplicate rows we need to drop them for further Analysis  
retail=retail.drop_duplicates()
```

Fig: 4.9 Removing Duplicates.

Now, let's use check the shape of the data again.

```
#Chceking the shape of the data after dropping the duplicate records  
retail.shape
```

(9977, 13)

Fig: 4.10 Checking shape again

From above we can observe that 17 rows have been deleted from the original one.

Now, let's use check the aggregates of the given data.

```
# To check the aggregates of the Data Set  
retail.describe()
```

	Postal Code	Sales	Quantity	Discount	Profit
count	9977.000000	9977.000000	9977.000000	9977.000000	9977.000000
mean	55154.964117	230.148902	3.790719	0.156278	28.69013
std	32058.266816	623.721409	2.226657	0.206455	234.45784
min	1040.000000	0.444000	1.000000	0.000000	-6599.97800
25%	23223.000000	17.300000	2.000000	0.000000	1.72620
50%	55901.000000	54.816000	3.000000	0.200000	8.67100
75%	90008.000000	209.970000	5.000000	0.200000	29.37200
max	99301.000000	22638.480000	14.000000	0.800000	8399.97600

Fig: 4.11 Data Describe

Let's find the maximum sales and profit:

```
: # To find out Sales and Profit generated by the Superstore  
print("TOTAL SALES:",retail['Sales'].sum())  
print("TOTAL PROFIT:",retail['Profit'].sum())
```

TOTAL SALES: 2296195.5903

TOTAL PROFIT: 286241.4226

Fig: 4.12 sales and profit

Regional Analysis

With the given data, let's check the regions which regions had the most transactions were made.

```
#To check in which Region maximum transactions were made  
retail['Region'].value_counts().plot.pie(colors=['r','g','b','y'],autopct="%.1f%%")
```

<AxesSubplot:ylabel='Region'>

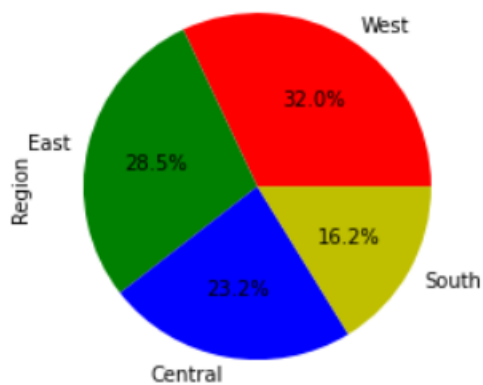


Fig: 4.13 Regional Analysis

From the pie chart, we could understand that most transactions were made in the west and then followed by the east.

Now, Let's find sales vs profit from each region

```
#Next we will check what amount of sales and profits were being made by each Region
plt.style.use('dark_background')
retail.groupby(['Region'])['Sales','Profit'].sum().plot.bar(color=['#00FFFF','#DC143C'],figsize=(12,6))
plt.xlabel("REGION",fontdict={'color':'white','fontsize':15})
plt.title("REGION WISE SALES AND PROFIT",fontdict={'color':'white','fontsize':20})
```

```
Text(0.5, 1.0, 'REGION WISE SALES AND PROFIT')
```

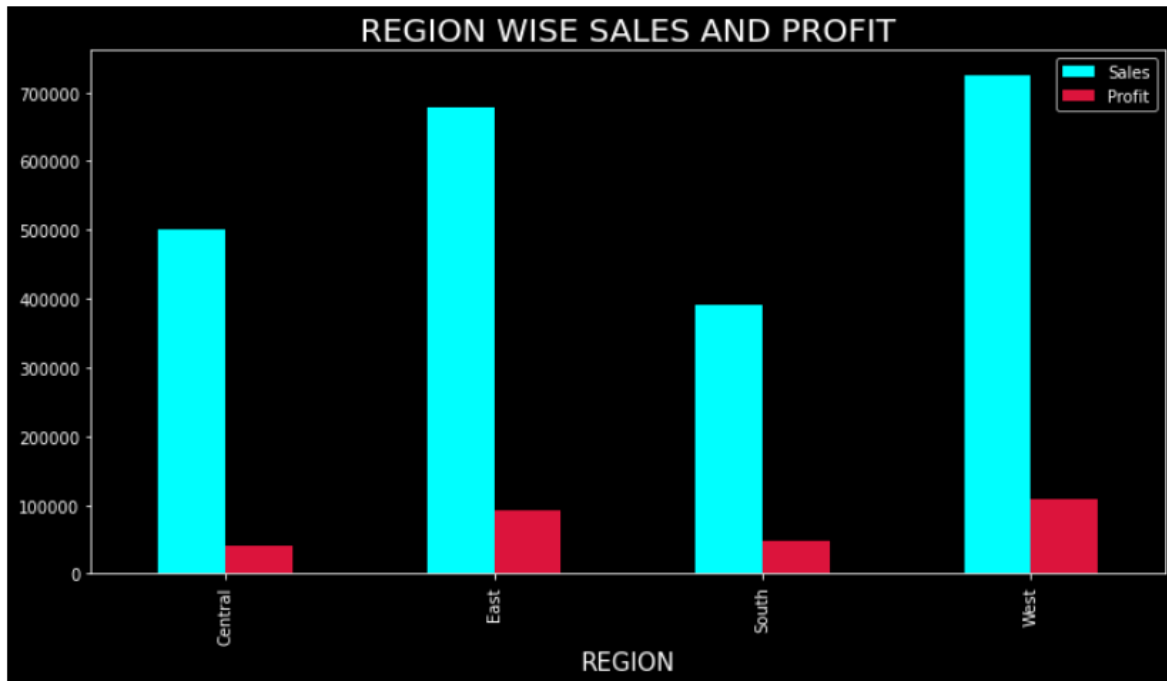


Fig: 4.14 Regions wise sales and profit

From the Bar plot, we can analyze that sales and profit were maximum in the western region.

Now Let's Analyze the type of customer who made maximum sales and profit.

```
# To check maximum Sales and Profit in each segment
plt.style.use('dark_background')
retail.groupby(['Segment'])['Sales', 'Profit'].sum().plot.bar(color=['lightgreen', 'yellow'], figsize=(12,6))
plt.xlabel("SEGMENT", fontdict={'color': 'white', 'fontsize': 15})
plt.title("SEGMENT WISE SALES AND PROFIT", fontdict={'color': 'white', 'fontsize': 20})
```

Text(0.5, 1.0, 'SEGMENT WISE SALES AND PROFIT')

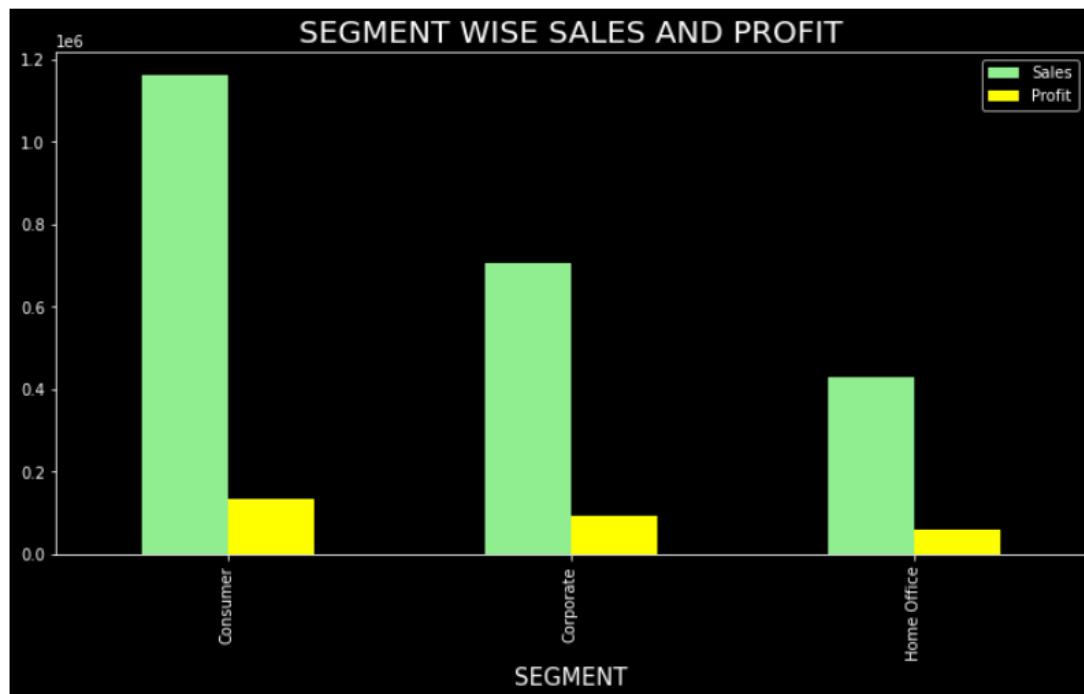


Fig: 4.15 segment wise sales and profit

From the Plotting, we could conclude that “Consumers” had maximum buying capacity and also gave maximum profit to the stores. Also “Home Offices” purchases are less and they also gave fewer profits.

Let's check the segment wise ship mode. This would help us understand under which class had the maximum shippings.

```
retail=retail.rename(columns={'Ship Mode':'Ship_mode'})
```

```
#Checking Ship Mode Segment wise
plt.style.use('dark_background')
plt.subplots(figsize=(16,8))
sns.countplot(x='Segment',hue='Ship_mode',data=retail)
plt.xlabel('SEGMENT',fontdict={'color':'white','fontsize':15})
plt.title("SEGMENT WISE SHIP MODE",fontdict={'color':'white','fontsize':15})
```

Text(0.5, 1.0, 'SEGMENT WISE SHIP MODE')

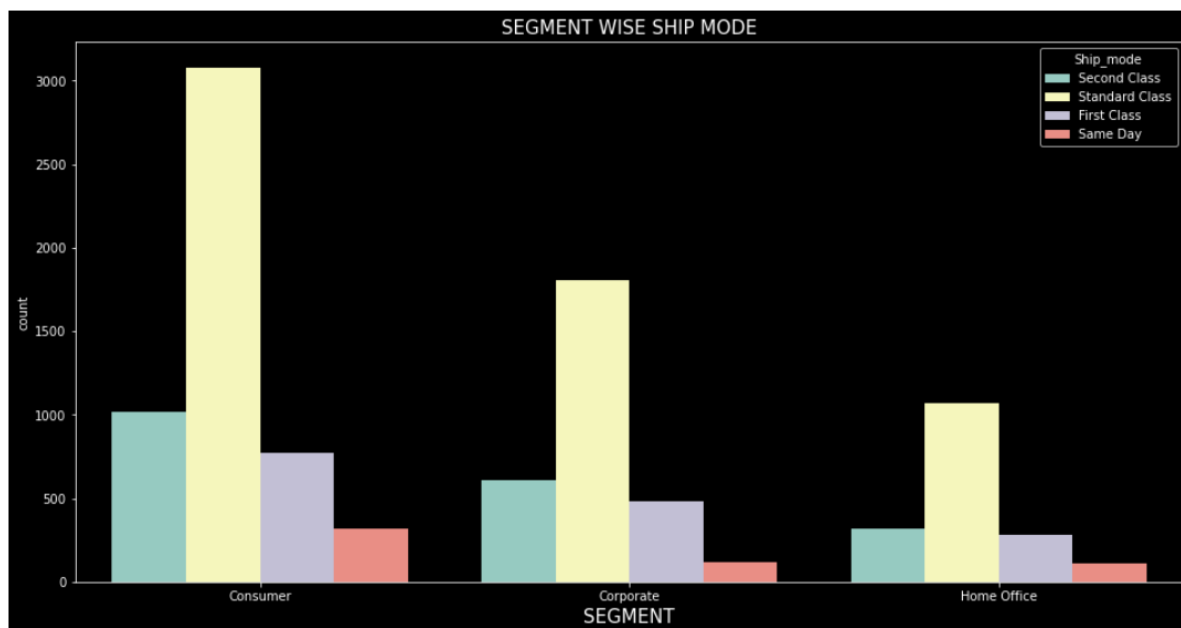


Fig: 4.16 segment wise ship more

From the plot, we find that most shipping are from standard class.

Let's now analyze the product. Let's try to understand which products had the maximum profits and sales.

```
# To check profit and sales Product wise
plt.style.use('dark_background')
retail.groupby(['Category'])['Sales', 'Profit'].sum().plot.bar(color=['#FF8C00', '#3CB371'], figsize=(12,6))
plt.xlabel("CATEGORY", fontdict={'color': 'white', 'fontsize': 15})
plt.title("PRODUCT CATEGORY WISE SALES AND PROFIT", fontdict={'color': 'white', 'fontsize': 20})
```

Text(0.5, 1.0, 'PRODUCT CATEGORY WISE SALES AND PROFIT')

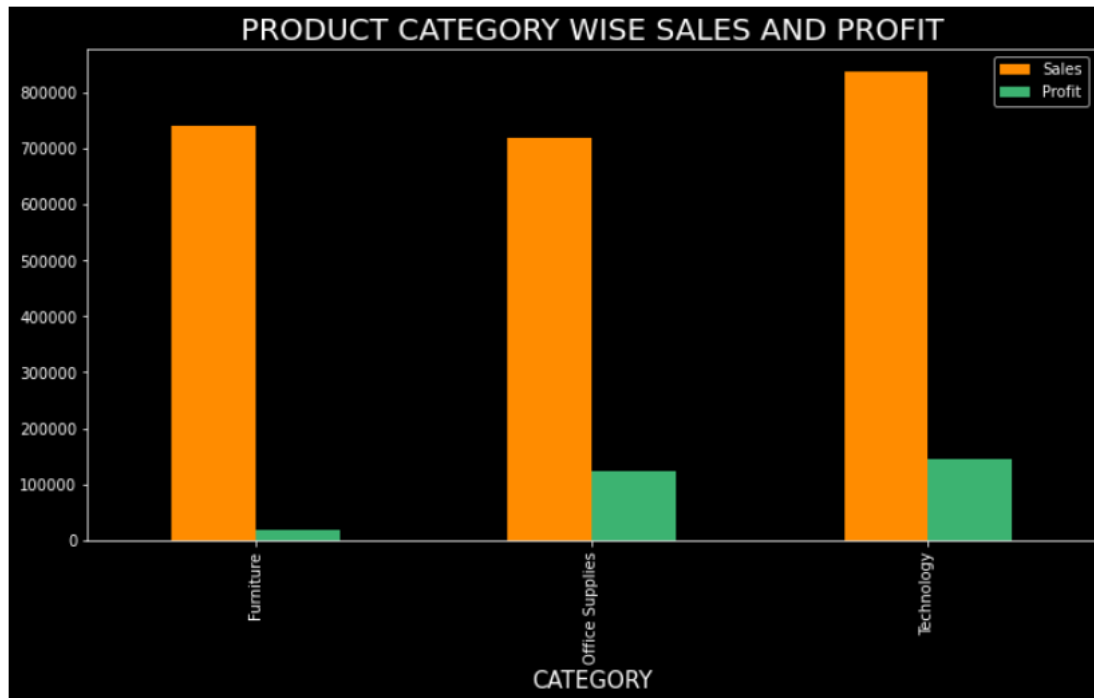


Fig: 4.17 Product Category wise sales and profit

From Plotting, We derive that Technology had the highest sales and profits but coming to the case of Furniture, It had good sales but the profits we low.

This section may have a problem in it as we were not able to get profits in spite of sales recorded.

So, Let's focus on "Furniture" Products and analyze them to gain profits.

Since, "Furniture" is unusual, Let's analyze sub-category wise in the Furniture.

```
#Since Furniture showed an unusual trend, now looking at sales and profit for subcategories of Furniture
plt.style.use('dark_background')
retail[retail['Category']=='Furniture'].groupby(['Sub-Category'])['Sales','Profit'].sum().plot.bar(color=['#FF1493','#E0FFFF'],figsize=(12,6))
plt.xlabel("SUB CATEGORY",fontdict={'color':'white','fontSize':15})
plt.title("SUB-CATEGORY WISE SALES AND PROFIT FOR FURNITURE",fontdict={'color':'white','fontSize':20})
```

Text(0.5, 1.0, 'SUB-CATEGORY WISE SALES AND PROFIT FOR FURNITURE')

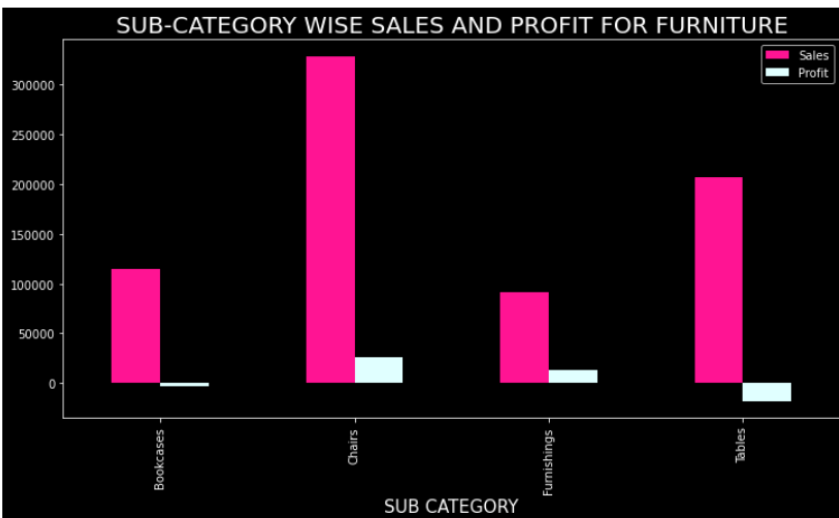


Fig: 4.18 Sub-Category wise sales and profit for furniture

There are 4 sub-categories in Furniture.

- BookCases
- Chairs
- Tables
- Furnishings.

From the Plotting, we could notice that though there were sales in Tables and Bookcases, still losses were incurring.

This is the reason why the entire "Furniture" is showing unusual trends.

The question we had to answer, is why are losses still incurring even though there are good sales.

The probable answer to this would be due to the discounts given to them.

So, Let's Visualize the discounts given under each sub-category of "Furniture"

```
#Now we need to check irrespective of high sales why are we incurring loss?
#To check the probable reason of Loss Lets look at the discount given to each Furniture sub-category
plt.style.use('dark_background')
retail[retail['Category']=='Furniture'].groupby(['Sub-Category'])['Discount'].mean().plot.bar(color='#0000FF',figsize=(12,6))
plt.xlabel("SUB CATEGORY",fontdict={'color':'white','fontsize':15})
plt.title("DISCOUNT GIVEN IN FURNITURE CATEGORY",fontdict={'color':'white','fontsize':20})

: Text(0.5, 1.0, 'DISCOUNT GIVEN IN FURNITURE CATEGORY')
```



Fig: 4.19 Discount in Furniture.

From the Plotting, we can conclude that high discounts were given for Tables, and book cases.

We may assume that due to high discounts, we may get losses. But this is just an assumption.

To Test our assumption, let's calculate the correlation. Correlation tells how the variables are related to each other

```
corr_data=retail.corr()
display(corr_data)
```

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023476	0.013110	0.059225	-0.029892
Sales	-0.023476	1.000000	0.200722	-0.028311	0.479067
Quantity	0.013110	0.200722	1.000000	0.008678	0.066211
Discount	0.059225	-0.028311	0.008678	1.000000	-0.219662
Profit	-0.029892	0.479067	0.066211	-0.219662	1.000000

Fig: 4.20 Correlation

Let's us use heat map to understand correlation better.

```
plt.subplots(figsize=(14,7))
sns.heatmap(corr_data,annot=True)
```

<AxesSubplot:>

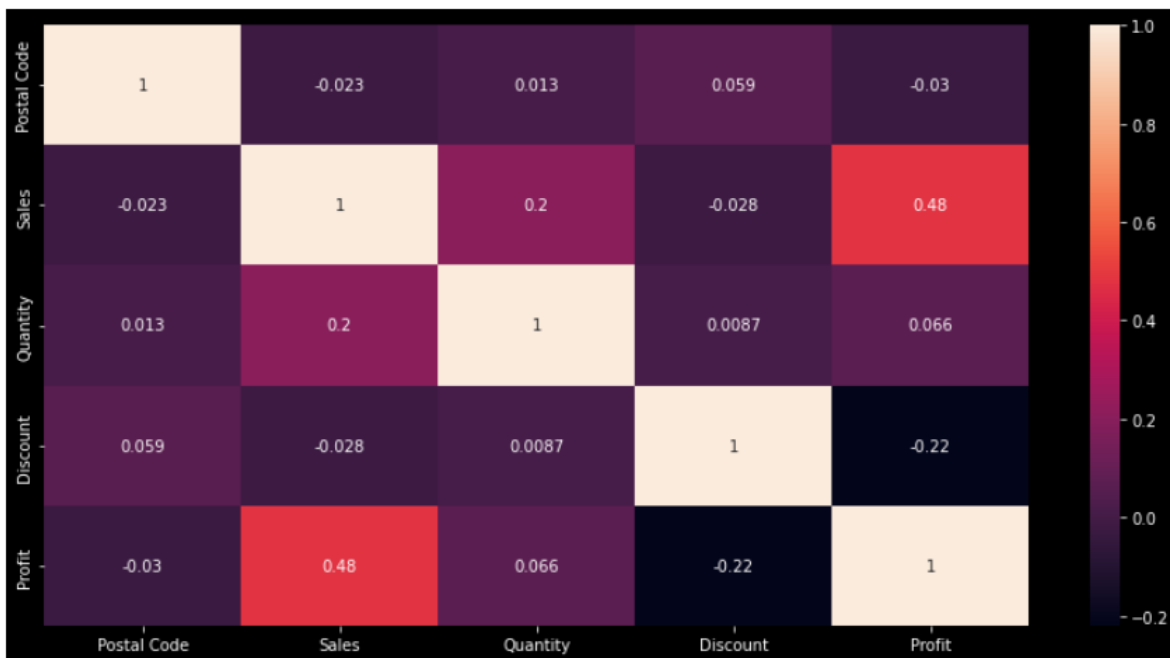


Fig: 4.21 Heat map

From the heat map, we can derive a negative correlation between profit and discount,
And a positive correlation between profit and sales.

Relation between sales and profit.

```
plt.style.use('dark_background')
plt.subplots(figsize=(12,6))
sns.regplot(x=retail['Sales'],y=retail['Profit'],color='orange',line_kws={'color':'red'})
plt.xlabel("SALES",fontdict={'color':'white','fontSize':15})
plt.ylabel("PROFIT",fontdict={'color':'white','fontSize':15})
plt.title("RELATIONSHIP BETWEEN SALES AND PROFIT",fontdict={'color':'white','fontSize':20})
```

Text(0.5, 1.0, 'RELATIONSHIP BETWEEN SALES AND PROFIT')

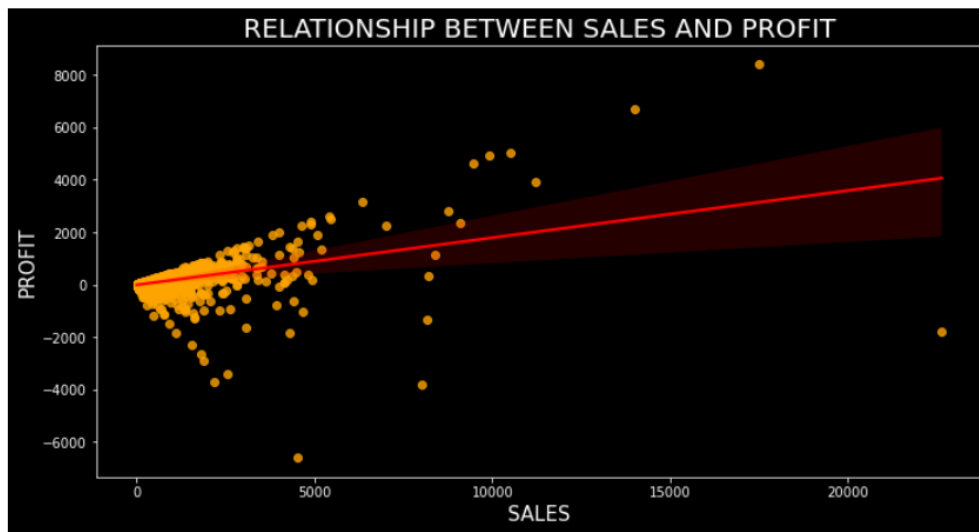


Fig: 4.22 Relation between sales and profit

From the Plotting, we could conclude that there is a strong positive relation between sales and profit.

Sub Category Wise Sales

```
# Now we will check the Top Products Sold
plt.style.use('dark_background')
retail.groupby(['Sub-Category'])['Sales'].sum().sort_values(ascending=False).plot.bar(color=['#00FF00'],figsize=(12,6))
plt.xlabel("SUB CATEGORY",fontdict={'color':'white','fontsize':15})
plt.ylabel("SALES",fontdict={'color':'white','fontsize':15})
plt.title("SUB-CATEGORY WISE SALES",fontdict={'color':'white','fontsize':20})
```

Text(0.5, 1.0, 'SUB-CATEGORY WISE SALES')

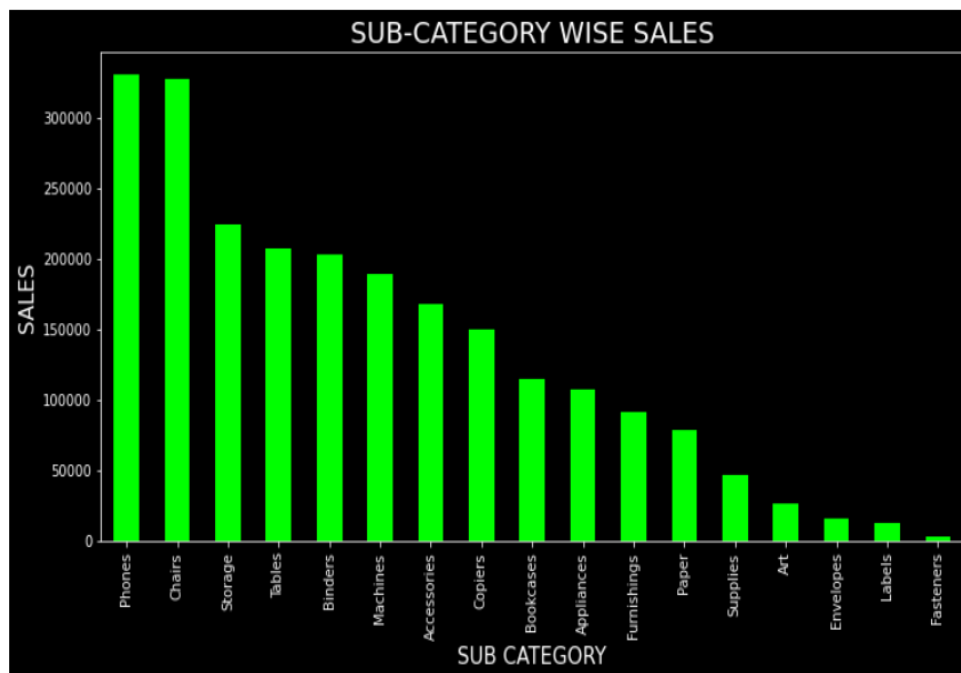


Fig: 4.23 Sub-category-wise sales

From this we conclude that Phones, Chairs, Storage, tablets, and binders are sold the most. Whereas Fasteners, Labels, and Envelopes were sold the least

Sub Category Wise Profit

```
#To check the profit earned from all the Sub-Categories
plt.style.use('dark_background')
retail.groupby(['Sub-Category'])['Profit'].sum().sort_values(ascending=False).plot.bar(color='#F4A460',figsize=(12,6))
plt.xlabel("SUB CATEGORY",fontdict={'color':'white','fontsize':15})
plt.ylabel("PROFIT",fontdict={'color':'white','fontsize':15})
plt.title("SUB-CATEGORY WISE PROFIT",fontdict={'color':'white','fontsize':20})
```

Text(0.5, 1.0, 'SUB-CATEGORY WISE PROFIT')

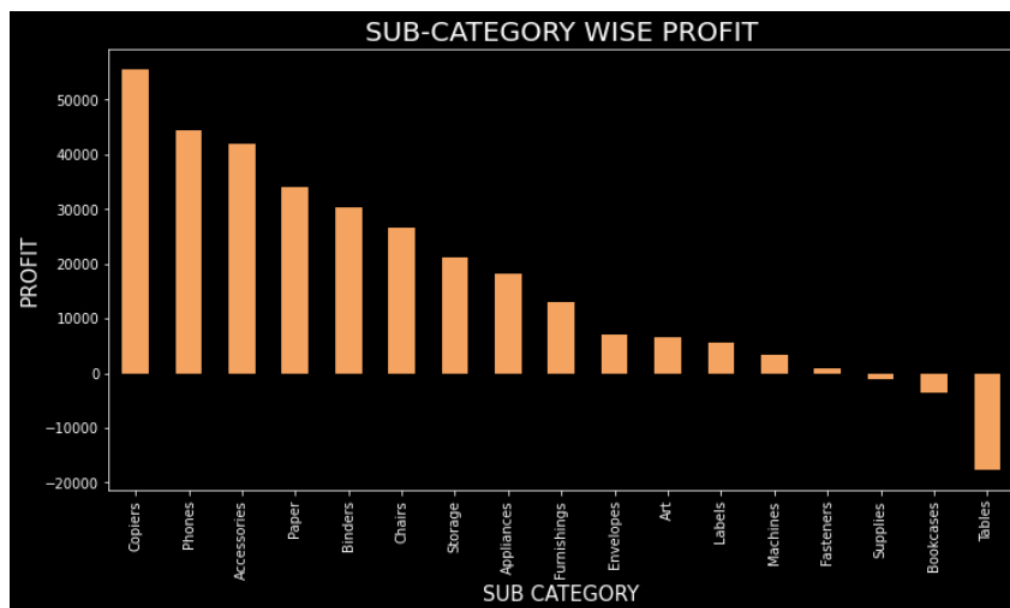


Fig: 4.24 sub-category-wise profit

From the plotting, We could understand that Copiers, Phones, and Accessories are the top profit-giving products to the store. Whereas Store is incurring losses due to Tables, Book Cases and Suppliers.

City Wise Sales.

```
#To check top 5 cities based on Sales
plt.style.use('dark_background')
retail.groupby(['City'])['Sales'].sum().sort_values(ascending=False).head().plot.bar(color=['#FF00FF'],figsize=(12,6))
plt.xlabel("CITY",fontdict={'color':'white','fontsize':15})
plt.ylabel("SALES",fontdict={'color':'white','fontsize':15})
plt.title("CITY WISE SALES",fontdict={'color':'white','fontsize':20})
```

Text(0.5, 1.0, 'CITY WISE SALES')



Fig: 4.25 city wise sales

From the plotting, New York City is giving the store the highest sales followed by Los Angeles and Seattle.

City Wise Profit.

```
#To check the profit earned from the top 5 cities
plt.style.use('dark_background')
retail.groupby(['City'])['Profit'].sum().sort_values(ascending=False).head().plot.bar(color=['#00FA9A'],figsize=(12,6))
plt.xlabel("CITY",fontdict={'color':'white','fontsize':15})
plt.ylabel("PROFIT",fontdict={'color':'white','fontsize':15})
plt.title("CITY WISE PROFIT",fontdict={'color':'white','fontsize':20})
```

Text(0.5, 1.0, 'CITY WISE PROFIT')



Fig: 4.26 City Wise Profit

We Could conclude that New York City is having highest profit followed by Los Angeles and Seattle.

City Wise Discount

```
#To check top 5 cities based on total discount given
plt.style.use('dark_background')
retail.groupby(['City'])['Discount'].sum().sort_values(ascending=False).head().plot.bar(color='#1E90FF',figsize=(12,6))
plt.xlabel("CITY",fontdict={'color':'white','fontsize':15})
plt.ylabel("DISCOUNT",fontdict={'color':'white','fontsize':15})
plt.title("CITY WISE DISCOUNT",fontdict={'color':'white','fontsize':20})
```

```
Text(0.5, 1.0, 'CITY WISE DISCOUNT')
```



Fig: 4.27 City-wise Discount

Highest discounts were in Philadelphia followed by Houston and Chicago. But from the above observation, we could conclude that these cities don't lead to sales and profit table.

5. Results

The below insights were drawn after performing Exploratory Data Analysis on the Given Retail Dataset.

1. Maximum Transactions were made in the western region.
2. Maximum Sales in the West region.
3. Maximum Profits in the West Region.
4. Maximum Sales and Profits in the Consumer segment.
5. Maximum Transactions were shipped in the Standard class irrespective of the segment.
6. Least Profit is incurred in Furniture Category irrespective of good amount of Sales.
7. Under Furniture, Tables and Book Cases are incurring losses and effecting the total Profit of Furniture Category.
8. High Discount is being offered in Tables and Book Cases which is somewhere the probable reason of losses.
9. City Contributing to the maximum profit and sales is New York City.
10. Maximum discount is given in the city of Philadelphia.
11. Philadelphia contributes the least towards profit
12. Positive Correlation between Sales and Profits.
13. Negative Correlation between Profit and Discounts.

6. Conclusion and Future Work

From the observations we conclude that Furniture Category is the weak area where we need to work upon. As for Furniture, we have Tables and Book Cases where we losses incurred even with huge discounts. So, We need to reduce the discount in order to increase the profit.

Maximum discounts were given to Philadelphia followed by Houston and Chicago. Despite this fact, Philadelphia is contributing the least towards profit followed by Houston. So, we can reduce the discounts in these cities.

Now, we could make assumptions and use different modeling techniques to train the model and test it. Upon repeating the training phase, we could arrive at a model which would increase profit and help the business.

7. References

- [1] Callus O.Wilke (2019). Fundamentals of Data Visualization,[O'Reilly Media, Inc.](#)
- [2] Suresh Kumar Mukhiya, Usman Ahmed (2020). Hands-On Exploratory Data Analysis with Python. Packt Publishing
- [3] Boris Paskhaver (2021) Pandas in Action. Manning Publications.
- [4] Jack Dougherty, Ilya Ilyankou. Hands-on Data Visualization. [O'Reilly Media, Inc.](#)