EXPLORATORY DATA ANALYSIS ON E-COMMERCE SALES

Masters of Computer Application

SUBMITTED BY:
Nirmal Kuttan
Reg No:- 221347040

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SGT UNIVERSITY, GURUGRAM (14pt.)

(Batch: 2022-2024)

1. INTRODUCTION

1.1 CONCEPTUAL STUDY OF THE PROJECT:

Exploratory data analysis is an approach to analysing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis(IDA. Which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA. The primary goal of EDA is to maximize the analyst's insight into a data set and into the underlying structure of a data set, while providing all of the specific items that an analyst would want to extract from a data set, such as: a good-fitting, parsimonious model. a list of outliers.

1.2 OBJECTIVES OF THE PROJECT:-

- To understand which order has the highest and lowest priority
- To identify which Ship mode was used more and used less.
- To identify which ship mode was used for as specific product category.
- To identify which product category was sold more and less.
- To identify which customer segment has ordered the products more and less.
- To identify which year has the most sales done.

1.3 SCOPE OF THE PROJECT:-

- Identifying which product has highest sales and lowest sales.
- Identifying profit and loss.
- To identify which customer segment has ordered the products more and less.
- To identify which year has the most sales done.

2. ABOUT DATASET:

2.1 DATA IDENTIFIED FROM:-

This dataset consists of E-Commerce Sales dataset. The dataset is collected from kaggle.
Superstore_USA | Kaggle
https://www.kaggle.com/datasets/anuragupadhyay6212/superstore-usadataset

.

2.2 DETAILS ABOUT THE ATTRIBUTES IN DATASET:-

No of Rows & Columns:- (9426, 24)

Columns Names:-

- Row ID
- Order Priority
- Discount
- Unit Price
- Shipping Cost
- Customer ID
- Customer Name
- Ship Mode
- Customer Segment
- Product Category
- Product Sub-Category
- Product Container
- Product Name
- Product Base Margin
- Region
- State or Province
- City
- Postal Code
- Order Date
- Ship Date
- Profit
- Quantity ordered new
- Sales
- Order ID

3. BASIC DATA EXPLORATION:-

- df.head()
- df.info()
- df.describe()

4. VARIOUS ANALYSIS PERFORMED:-

- Checking for null values.
- Handling missing values.
- Checking for outliers.
- Handlin Outliers that are present.

5. VISULAIZATIONS:-

- Order Priority.
- Ship Mode.
- Product Category.
- Customer Segment.
- Order Date.
- Profit
- State, Region, City
- Profit Base Margin

6. PROJECT SCREENSHOTS:-

jupyter Ecommerce Last Checkpoint: Last Monday at 7:38 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [4]:
```
df = pd.read_excel("Superstore_USA.xlsx")
```

In [5]:
```
df.head(3)
```

Out[5]:

	Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Category	...	Region	State or Province	City	Postal Code	Order Date	Ship Date	Pr
0	18606	Not Specified	0.01	2.88	0.50	2	Janice Fletcher	Regular Air	Corporate	Office Supplies	...	Central	Illinois	Addison	60101	2012-05-28	2012-05-30	1
1	20847	High	0.01	2.84	0.93	3	Bonnie Potter	Express Air	Corporate	Office Supplies	...	West	Washington	Anacortes	98221	2010-07-07	2010-07-08	4
2	23086	Not Specified	0.03	6.68	6.15	3	Bonnie Potter	Express Air	Corporate	Office Supplies	...	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	-47

3 rows × 24 columns

In [6]:
```
df.shape
```

Out[6]: (9426, 24)

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Row ID                  0
Order Priority          0
Discount                0
Unit Price              0
Shipping Cost           0
Customer ID             0
Customer Name           0
Ship Mode               0
Customer Segment        0
Product Category        0
Product Sub-Category    0
Product Container       0
Product Name            0
Product Base Margin     72
Region                  0
State or Province       0
City                    0
Postal Code             0
Order Date              0
Ship Date               0
Profit                  0
Quantity ordered new    0
Sales                   0
Order ID                0
dtype: int64
```

Order Priority

In [8]:
```
df['Order Priority'].value_counts()
```

Out[8]:
```
High             1970
Low              1926
Not Specified    1881
Medium           1844
Critical         1804
Critical            1
Name: Order Priority, dtype: int64
```

In [9]:
```
#two critical values are there in order priority
df['Order Priority'].unique()
```

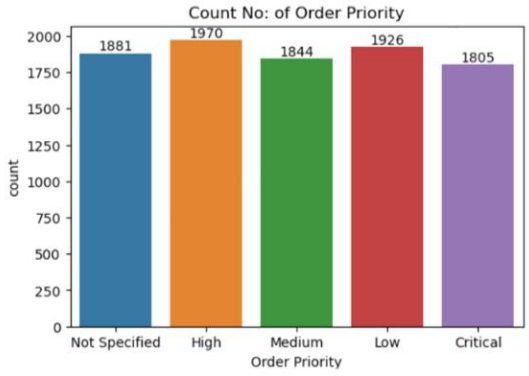
Out[9]:
```
array(['Not Specified', 'High', 'Medium', 'Low', 'Critical', 'Critical '],
      dtype=object)
```

In [10]:
```
df['Order Priority']=df['Order Priority'].replace("Critical ","Critical")
```

In [11]:
```
df['Order Priority'].value_counts()
```

Out[11]:
```
High             1970
Low              1926
Not Specified    1881
Medium           1844
Critical         1805
Name: Order Priority, dtype: int64
```

In [49]:
```
plt.figure(figsize=(6,4))
ax=sns.countplot(x="Order Priority",data=df)
for bars in ax.containers:
    ax.bar_label(bars)
plt.title("Count No: of Order Priority")
plt.savefig("Count Of Order Priority.jpg")
plt.show()
```



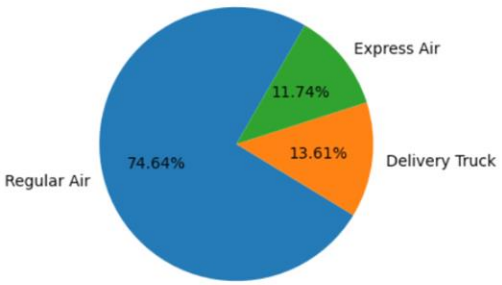
Ship Mode

In [13]:
```
df['Ship Mode'].value_counts()
```

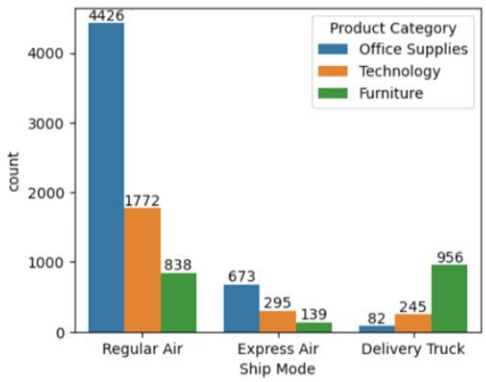
Out[13]:
```
Regular Air       7036
Delivery Truck    1283
Express Air       1107
Name: Ship Mode, dtype: int64
```

In [14]:
```
x=df['Ship Mode'].value_counts().index
y=df['Ship Mode'].value_counts().values
```

In [15]:
```
plt.figure(figsize=(5,4))
plt.pie(y,labels=x,startangle=60,autopct="%0.2f%%")
plt.show()
```

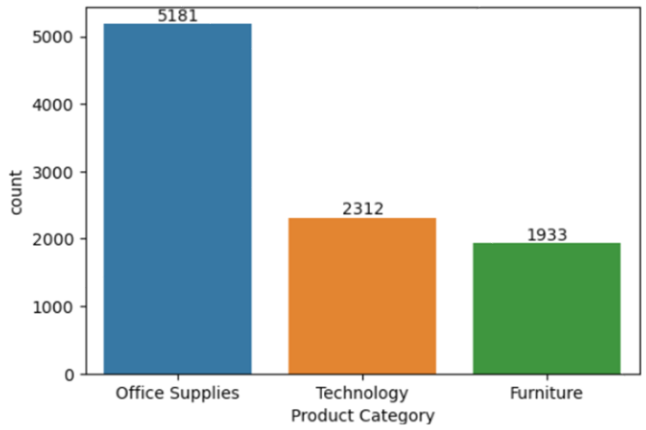


In [46]:
```
plt.figure(figsize=(5,4))
ax=sns.countplot(x="Ship Mode",data=df,hue="Product Category")
for bars in ax.containers:
    ax.bar_label(bars)
plt.savefig("ShipModes_ProdCat.jpg")
plt.show()
```



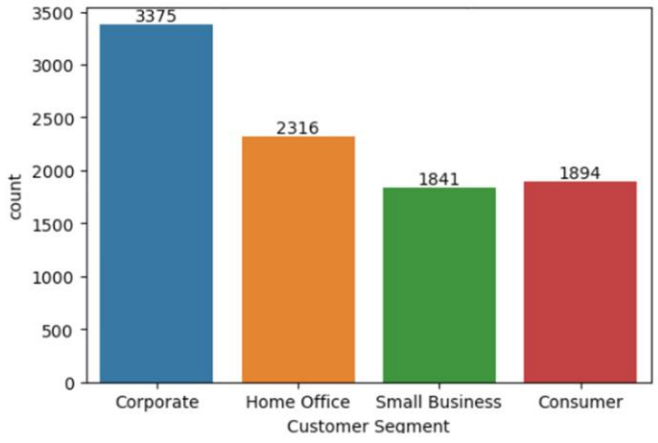
Product Category

In [40]:
```
plt.figure(figsize=(6,4))
ax=sns.countplot(x="Product Category",data=df)
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



Customer Segment

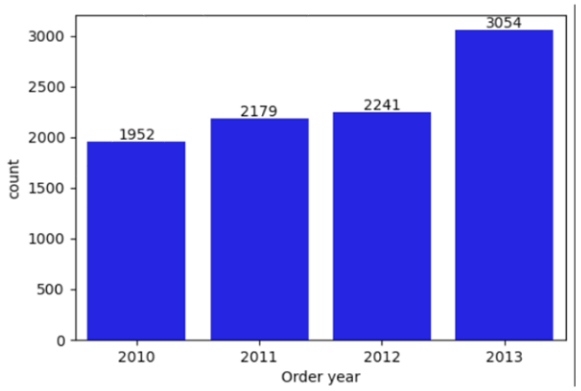
In [50]:
```
plt.figure(figsize=(6,4))
ax=sns.countplot(x="Customer Segment",data=df)
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



In [65]:
```
#Sales in a year
df['Order year'].value_counts()
```

Out[65]:
```
2013    3054
2012    2241
2011    2179
2010    1952
Name: Order year, dtype: int64
```

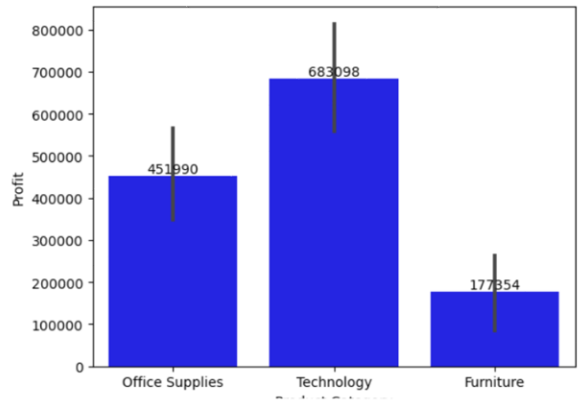
In [67]:
```
plt.figure(figsize=(6,4))
ax=sns.countplot(x="Order year",data=df,color='Blue')
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



Profit

Product Category wise Profut

In [73]:
```
ax=sns.barplot(x="Product Category",y="Profit",data=df,estimator='sum',color='Blue')
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



Profir Base Margin

In [81]:
```
ax=sns.barplot(x="Product Category",y="Product Base Margin",data=df,estimator='sum',color='Blue'
plt.show()
```