**Aim:** Analysis of Superstore Dataset

**IDE:** Microsoft Excel, Tableau , Spyder

**Pre-Requisites:-**  Insert the Customer’s Full Name, Gender, City, and Country From the Customer Sheet Using VLOOKUP Function In Order Dataset In Excel Itself on the Basis of Customer ID.

**Now Import Necessary Libraries for Analysis:-**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

Dataset = pd.read\_excel("./Superstore.xlsx",'Orders')

Dataset['Order\_Date'] = pd.to\_datetime(Dataset['Order\_Date'])

Dataset["Year"] = Dataset["Order\_Date"].dt.year

Dataset["Quater"] = Dataset["Order\_Date"].dt.quarter

Dataset["Month"] = Dataset["Order\_Date"].dt.month

**Questions:**

1. **What are the product categories available for customers to shop?**

**Code:-**

Unique\_Categories = list(set(Dataset["Category"]))

plt.figure(figsize=(16, 8))

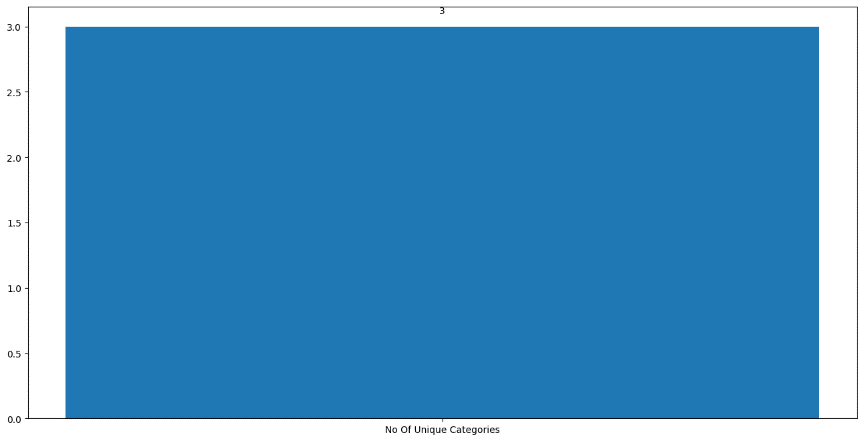
plt.bar(x=["No Of Unique Categories"], height=[len(Unique\_Categories)], width=0.5)

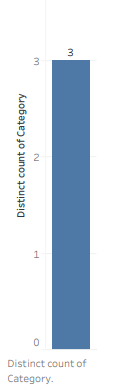
for i, v in enumerate([len(Unique\_Categories)]):

plt.text(i, v + 0.1, str(v), ha="center")

plt.show()

**Output :-**



**Tableau Workbook :-**

1. **How many products are there in each category?**

**Code:-**

plt.figure(figsize=(20, 10))

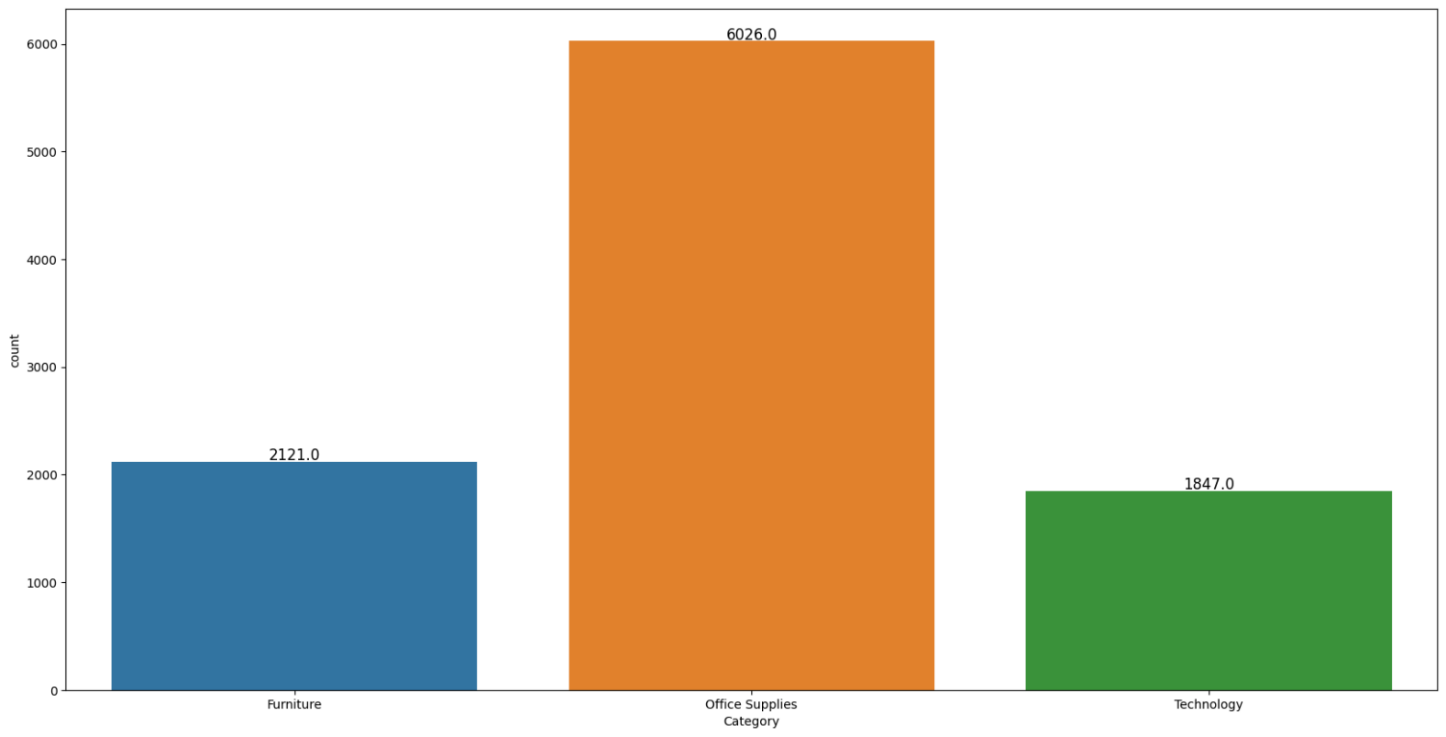
ax = sns.countplot(x=Dataset["Category"], hue=Dataset["Category"])

for p in ax.patches:

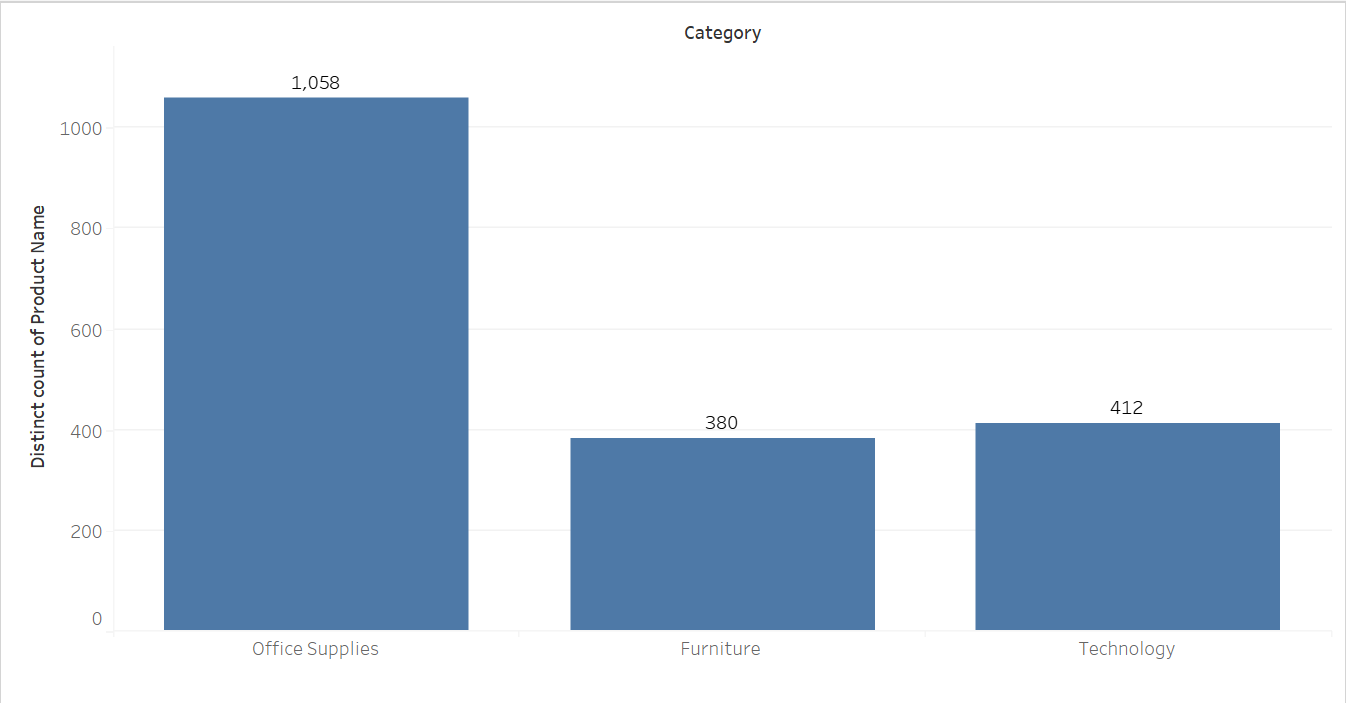
ax.annotate(f"{p.get\_height()}",(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),ha="center",va="center",fontsize=12,color="black",xytext=(0, 5),textcoords="offset points",)

plt.show()

**Output :-**

****

**Tableau Workbook :-**



1. **What is the count of products in each sub-category?**

**Code:-**

Sub\_categgory\_wise\_product\_distribuition = Dataset["Sub-Category"].value\_counts()

print(Sub\_categgory\_wise\_product\_distribuition)

plt.figure(figsize=(20, 10))

ax = sns.countplot(x=Dataset["Sub-Category"], hue=Dataset["Sub-Category"])

for p in ax.patches:

ax.annotate(

f"{p.get\_height()}",

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

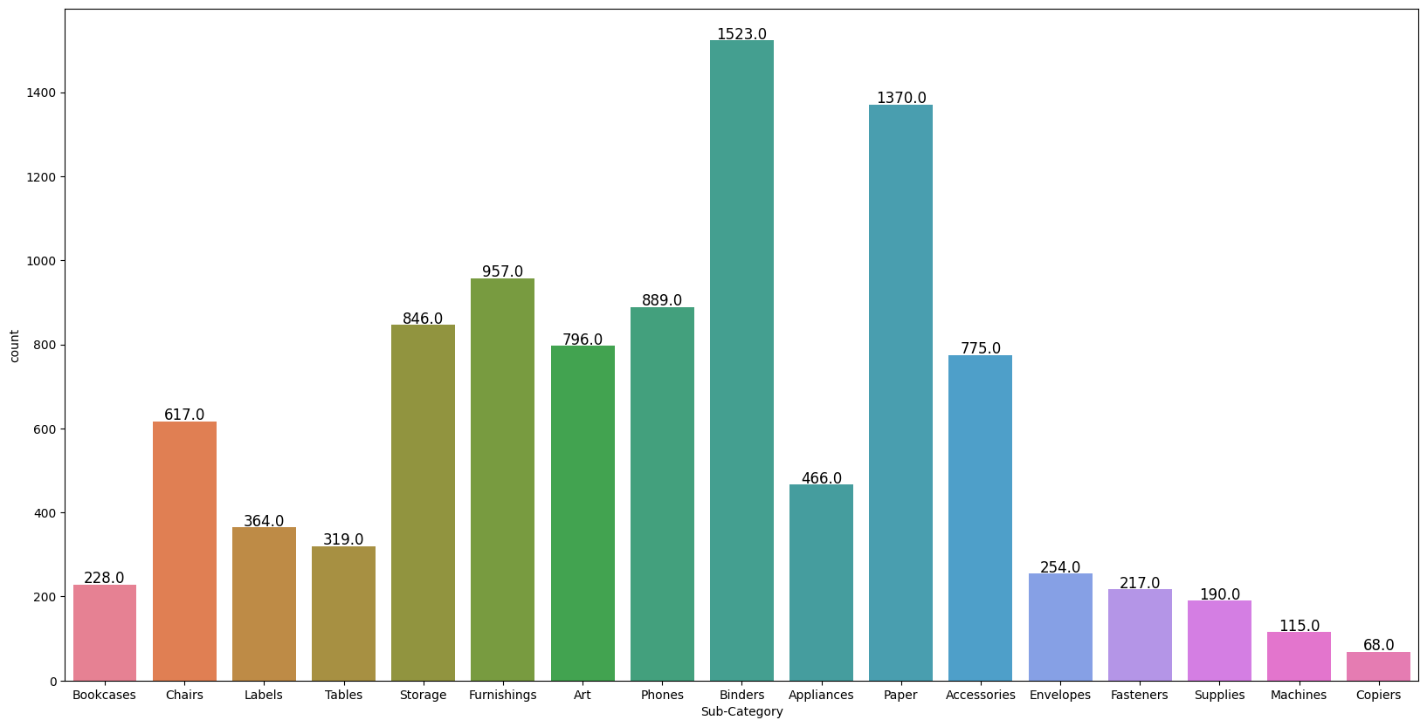
fontsize=12,

color="black",

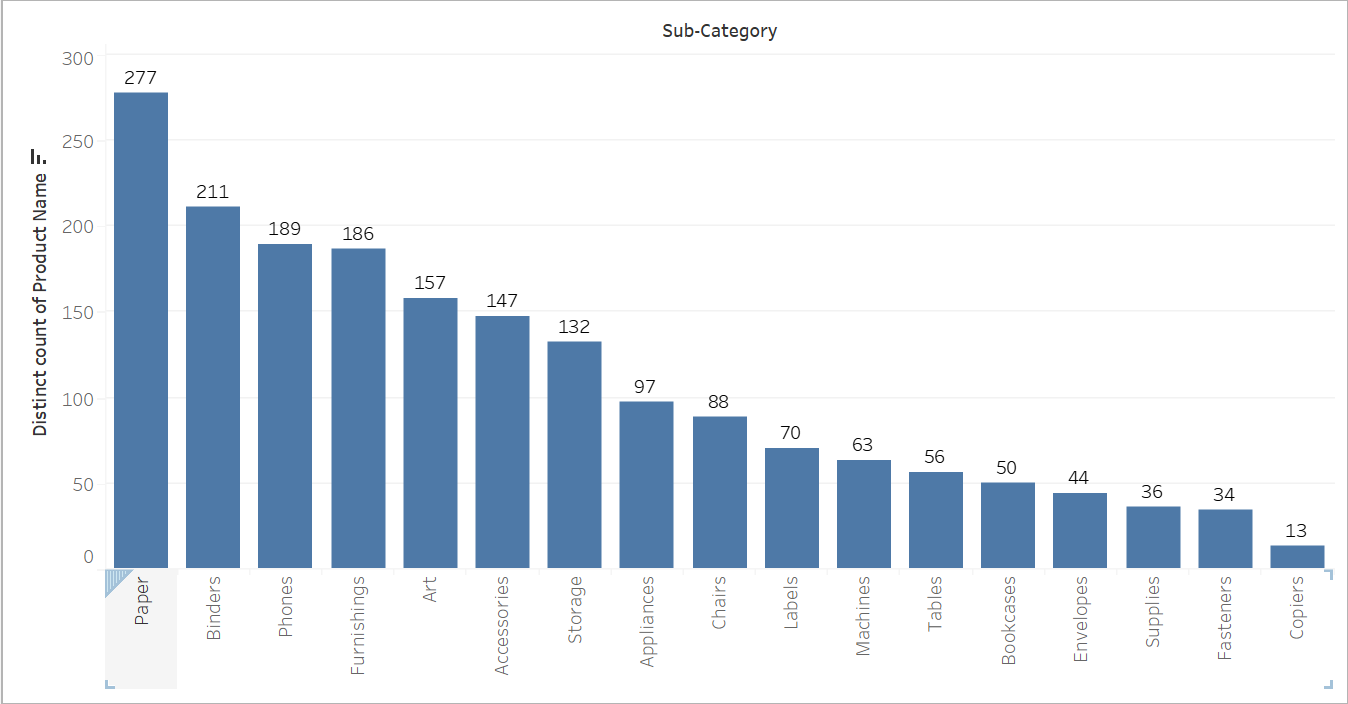
xytext=(0, 5),

textcoords="offset points",)

plt.show()

**Output :-**

**Tableau Workbook :-**

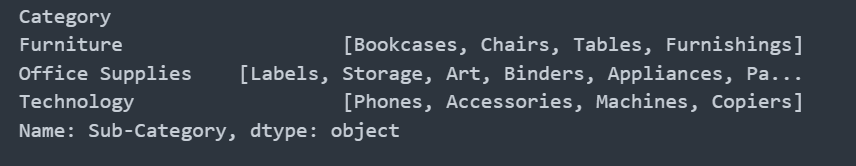


1. **How are sub-categories distributed in relation to their respective categories?**

**Code:-**

Distribution\_of\_sub\_categgory\_wrt\_Category = Dataset.groupby('Category')['Sub-Category'].unique()

print(Distribution\_of\_sub\_categgory\_wrt\_Category)

**Output :-**

**Tableau Workbook :-**

1. **What is the percentage distribution of varieties of Office Supplies?**

**Code:-**

Office\_Supplies\_Distribution = (

Dataset[Dataset["Category"] == "Office Supplies"]["Sub-Category"]

.value\_counts()

.rename("Counts")

)

total\_office\_supplies = Office\_Supplies\_Distribution.sum()

Office\_Supplies\_Distribution\_with\_percentage = (

Office\_Supplies\_Distribution / total\_office\_supplies

) \* 100

Office\_Supplies\_Distribution\_with\_percentage = (

Office\_Supplies\_Distribution\_with\_percentage.rename("Percentage")

)

Distribution\_of\_office\_Supplies = pd.concat(

[Office\_Supplies\_Distribution, Office\_Supplies\_Distribution\_with\_percentage], axis=1

)

plt.figure(figsize=(10, 6))

plt.pie(

Distribution\_of\_office\_Supplies["Counts"],

labels=Distribution\_of\_office\_Supplies.index,

autopct="%1.1f%%",

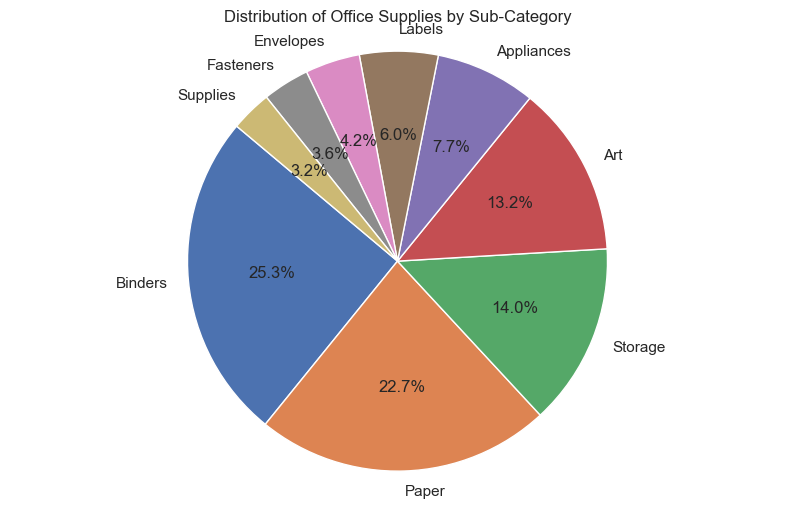
startangle=140,

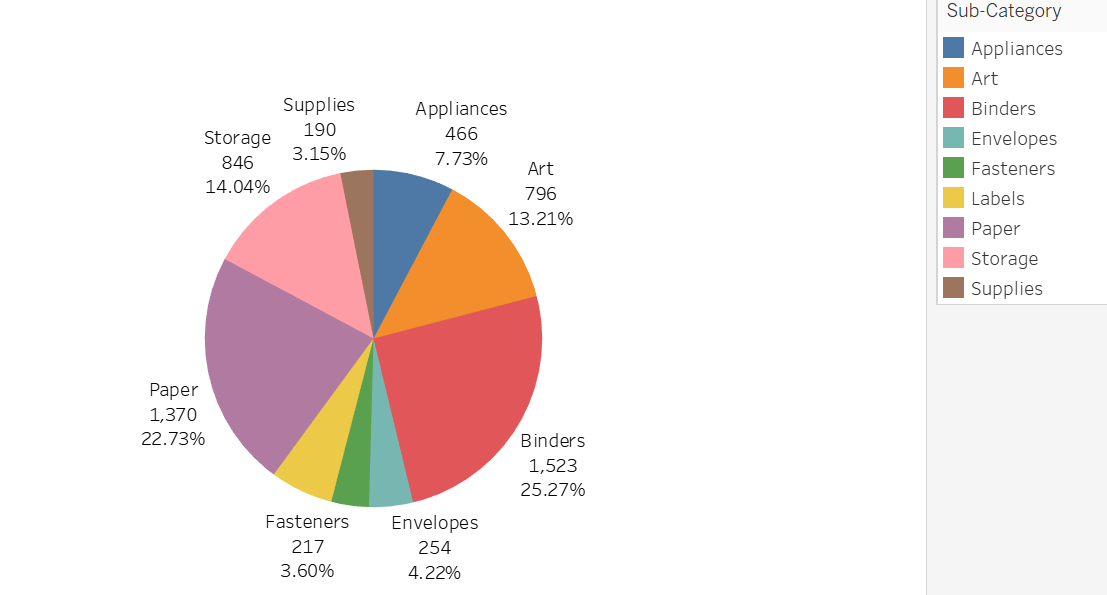
)

plt.title("Distribution of Office Supplies by Sub-Category")

plt.axis("equal")

plt.show()

**Output :-**

**Tableau Workbook:-**

1. **What is the percentage distribution of varieties of Technology?**

**Code:-**

Technology\_Distribution = (

Dataset[Dataset["Category"] == "Technology"]["Sub-Category"]

.value\_counts()

.rename("Counts")

)

total\_technology = Technology\_Distribution.sum()

Technology\_Distribution\_with\_percentage = (

Technology\_Distribution / total\_technology

) \* 100

Technology\_Distribution\_with\_percentage = (

Technology\_Distribution\_with\_percentage.rename("Percentage")

)

Distribution\_of\_Technology = pd.concat(

[Technology\_Distribution, Technology\_Distribution\_with\_percentage], axis=1

)

print(Distribution\_of\_Technology)

plt.figure(figsize=(10, 6))

plt.pie(

Distribution\_of\_Technology["Counts"],

labels=Distribution\_of\_Technology.index,

autopct="%1.1f%%",

startangle=140,

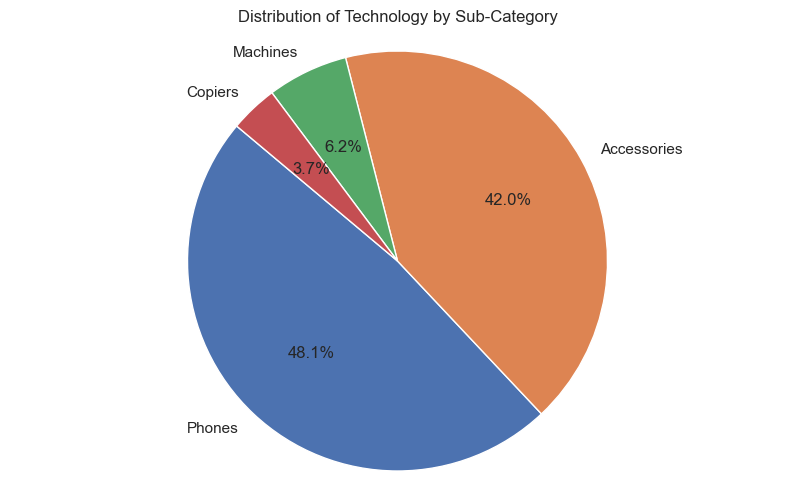
)

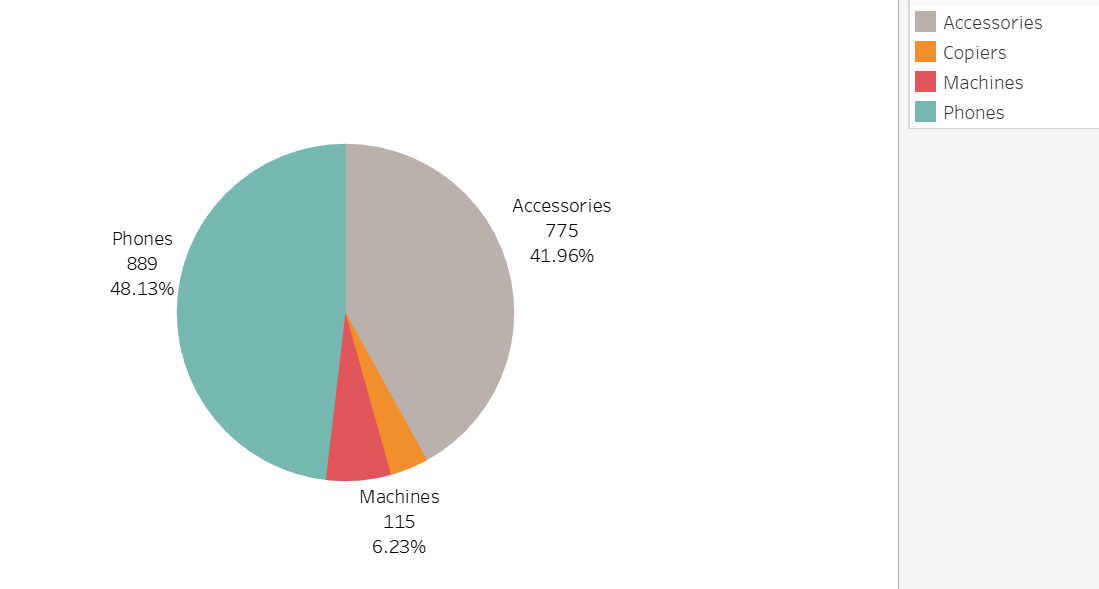
plt.title("Distribution of Technology by Sub-Category")

plt.axis("equal")

plt.show()

**Output:-**



**Tableau Workbook :-**

1. **What is the percentage distribution of varieties of Furniture?**

**Code:-**

Furniture\_Distribuition = (Dataset[Dataset["Category"] == "Furniture"]["Sub-Category"].value\_counts().rename("Counts"))

total\_furniture = Furniture\_Distribuition.sum()

Furniture\_Distribution\_with\_percentage = (Furniture\_Distribuition / total\_furniture) \* 100

Furniture\_Distribution\_with\_percentage = Furniture\_Distribution\_with\_percentage.rename("Percentage")

Distribution\_of\_Furniture = pd.concat([Furniture\_Distribuition, Furniture\_Distribution\_with\_percentage], axis=1)

print(Distribution\_of\_Furniture)

plt.figure(figsize=(10, 8))

plt.pie(

Distribution\_of\_Furniture["Counts"],

labels=Distribution\_of\_Furniture.index,

autopct="%1.1f%%",

startangle=140,

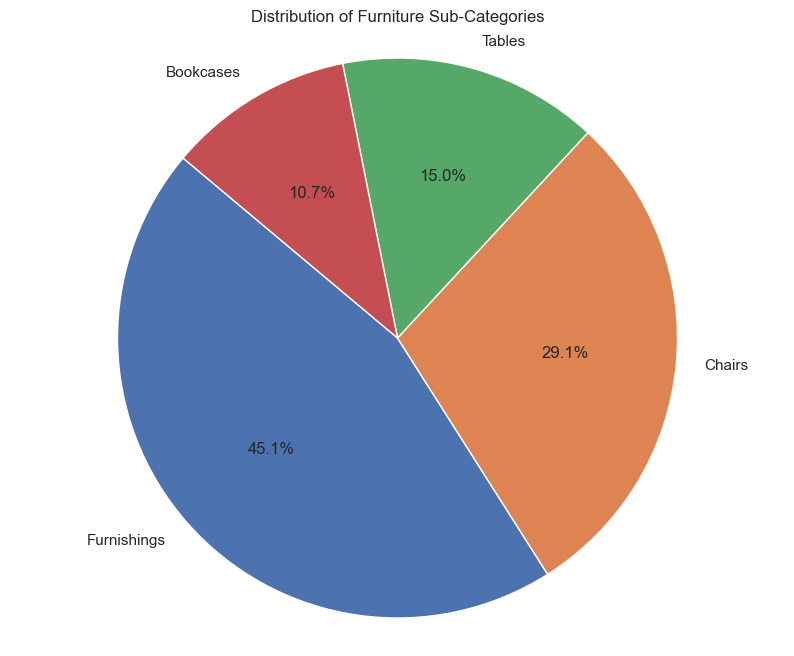
)

plt.title("Distribution of Furniture Sub-Categories")

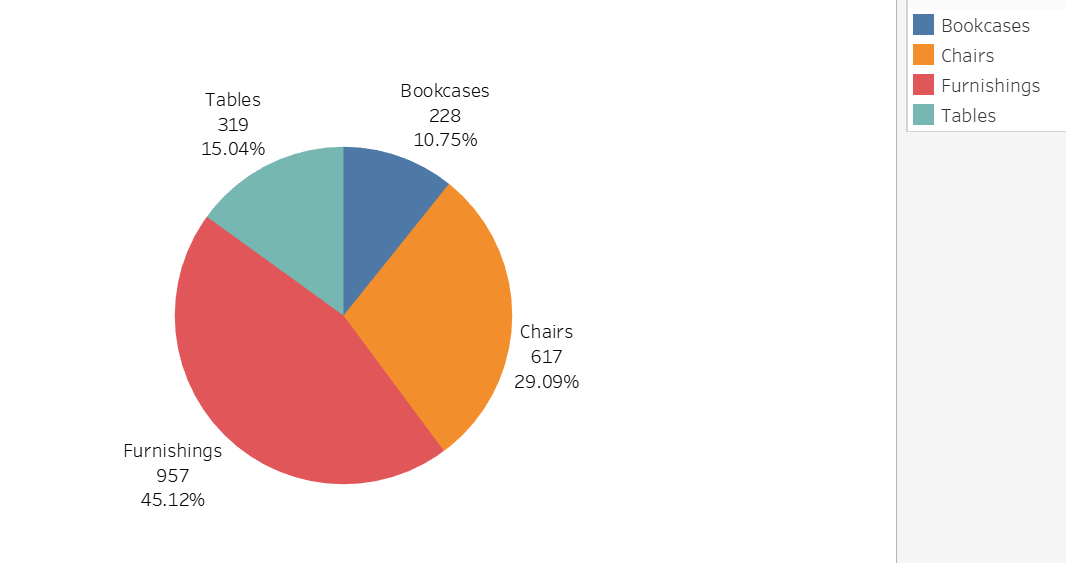
plt.axis("equal")

plt.show()

**Output :-**

****

**Tableau Workbook :-**



1. **What is the profit and sales data for each sub-category?**

**Code:-**

Category\_wise\_Sales\_and\_profit = Dataset.groupby("Sub-Category")[["Profit", "Sales"]].sum()

print(Category\_wise\_Sales\_and\_profit)

ax = Category\_wise\_Sales\_and\_profit.plot(kind="line", figsize=(12, 6))

plt.title("Category-wise Sales and Profit")

plt.xlabel("Sub-Category")

plt.ylabel("Amount")

plt.grid(True)

plt.legend(loc="upper right")

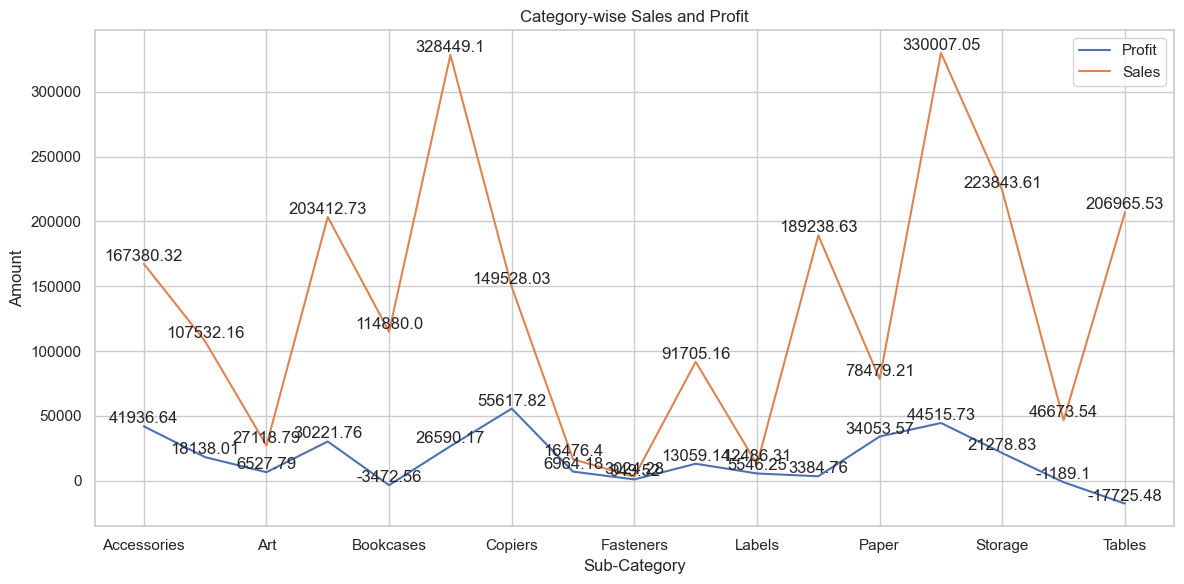
for column in Category\_wise\_Sales\_and\_profit.columns:

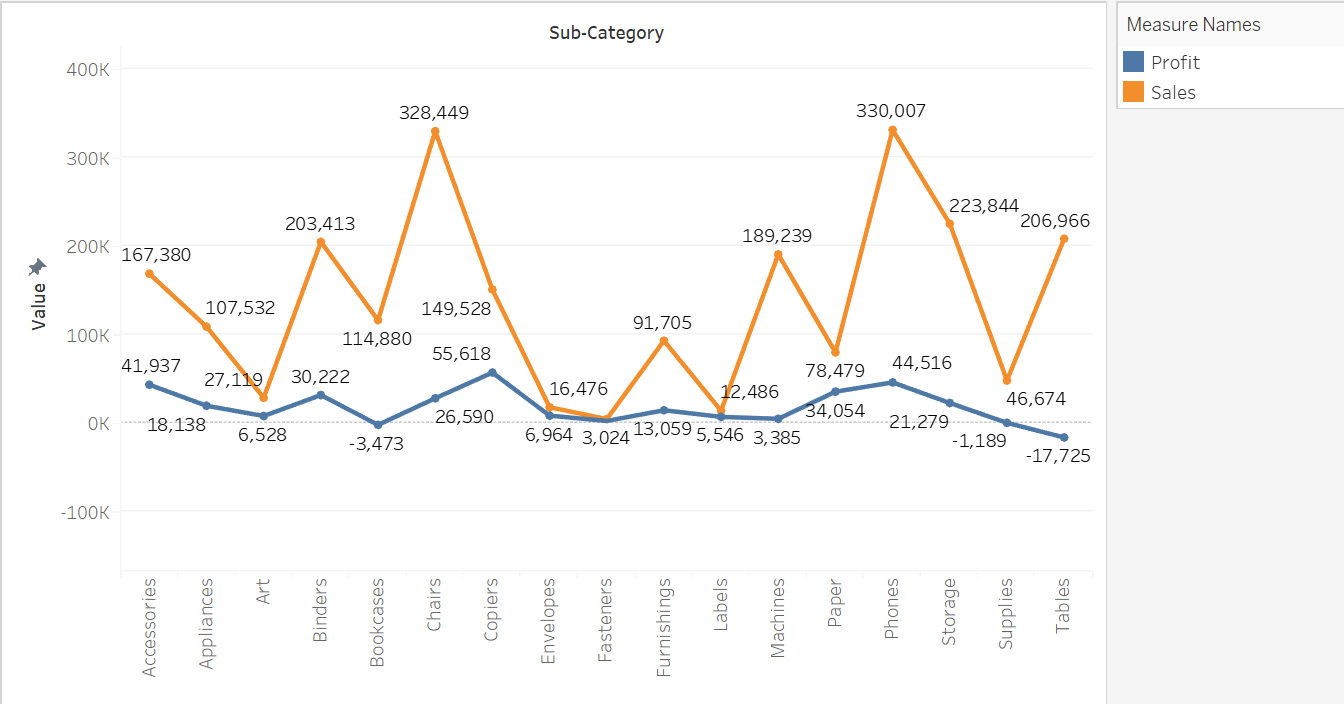
for index, value in enumerate(Category\_wise\_Sales\_and\_profit[column]):

ax.text(index, value, str(round(value, 2)), ha="center", va="bottom")

plt.tight\_layout()

plt.show()

**Output :-**

**Tableau Workbook :-**

1. **How many unique products have been ordered in the store?**

**Code:-**

Unique\_Products = Dataset['Product Name'].unique()

No\_of\_Unique\_Products = Dataset['Product Name'].nunique()

print(f"There are {No\_of\_Unique\_Products} Unique Products :-\n{Unique\_Products}")

plt.bar(

x=["No Of Unique Products"], height=[len(Unique\_Products)], width=0.5

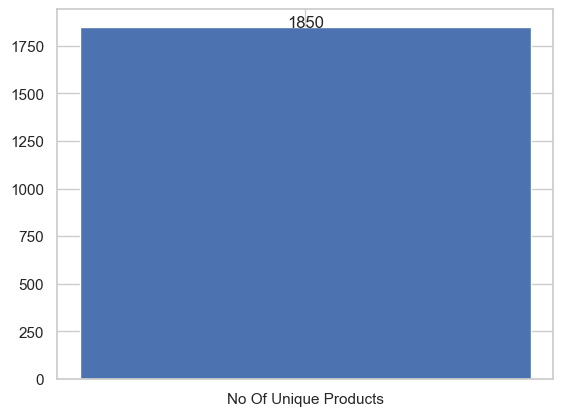
)

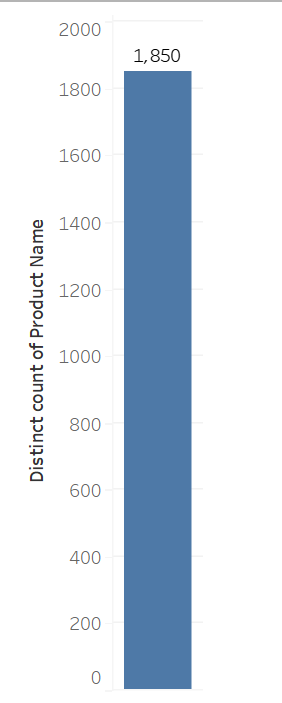
for i, v in enumerate([len(Unique\_Products)]):

plt.text(i, v + 0.5, str(v), ha="center")

plt.show()

**Output :-**



**Tableau Workbook :-**

1. **What is the total count of each product ordered in the store?**

**Code:-**

Count\_of\_Each\_Product = Dataset['Product Name'].value\_counts()

Data = Count\_of\_Each\_Product[:10]

Data\_df = pd.DataFrame({"Product Name": Data.index, "Count": Data.values})

plt.figure(figsize=(20, 10))

ax = sns.barplot(data=Data\_df, x="Product Name", y="Count", hue="Product Name")

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".0f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=12,

color="black",

)

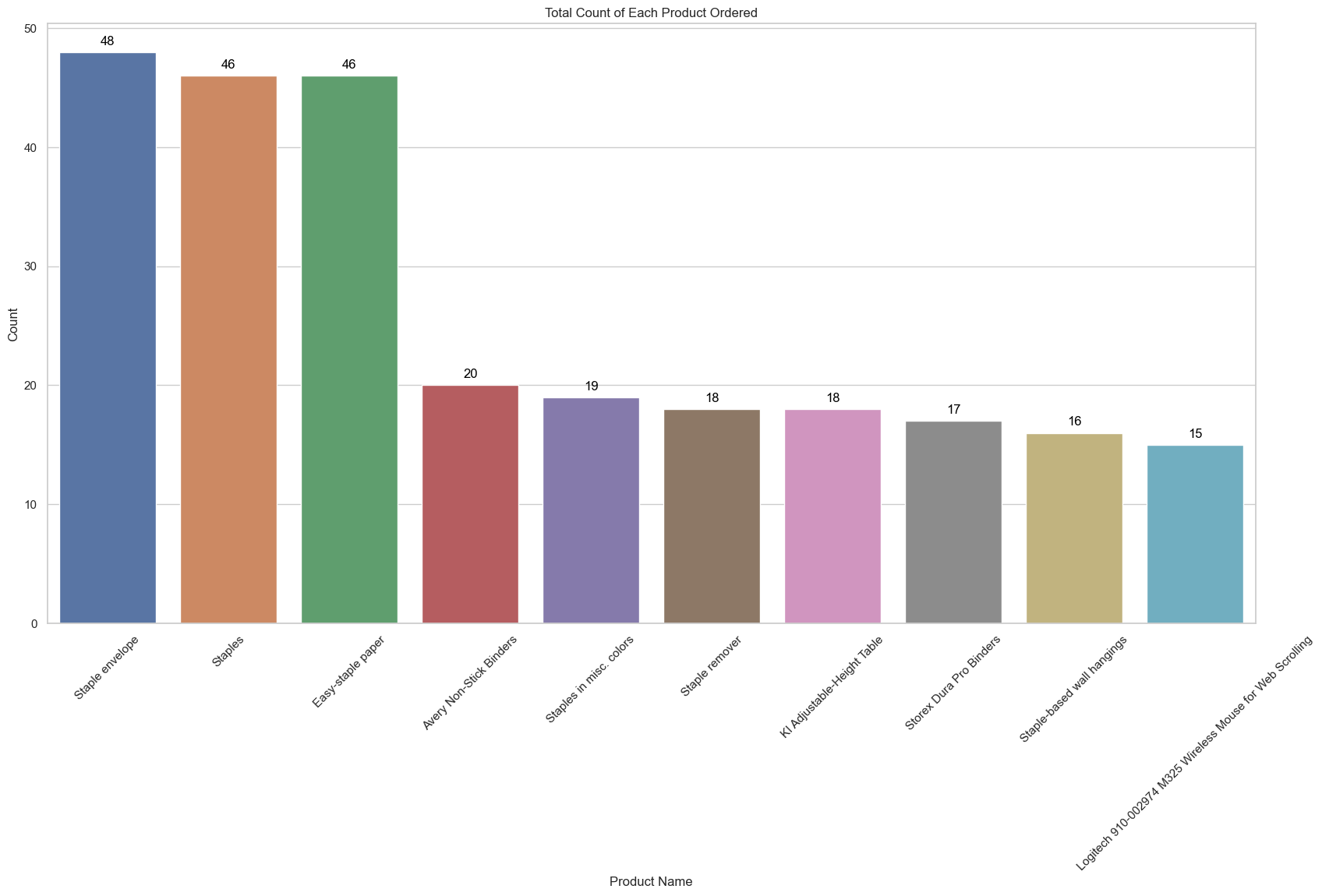
plt.xticks(rotation=45)

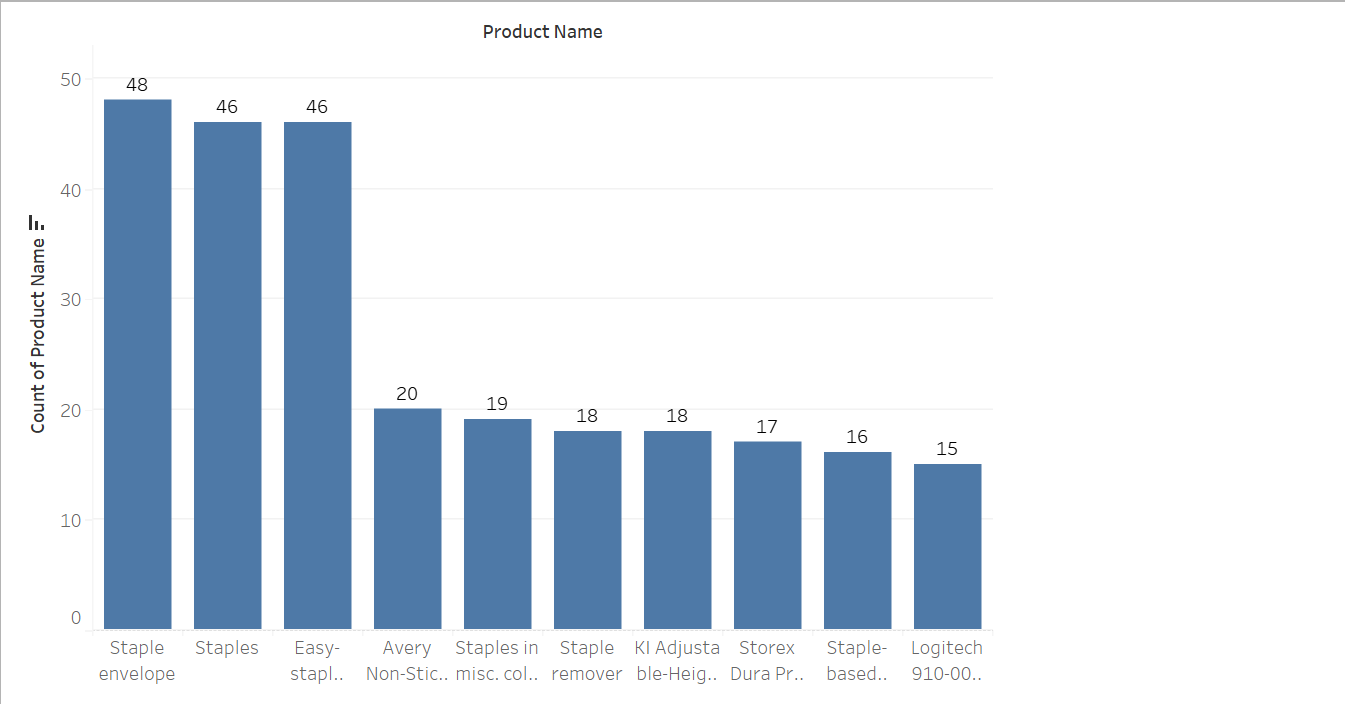
plt.xlabel("Product Name")

plt.ylabel("Count")

plt.title("Total Count of Each Product Ordered")

plt.show()

**Output :-**

**Tableau Workbook :-**

1. **What is the total count of each sub-category region-wise?**

**Code:-**

Region\_wise\_sub\_category\_count = Dataset.groupby(["Region", "Sub-Category"]).size().reset\_index(name="Count")

plt.figure(figsize=(20, 10))

ax = sns.barplot(

data=Region\_wise\_sub\_category\_count, x="Sub-Category", y="Count", hue="Region"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".0f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Sub-Category")

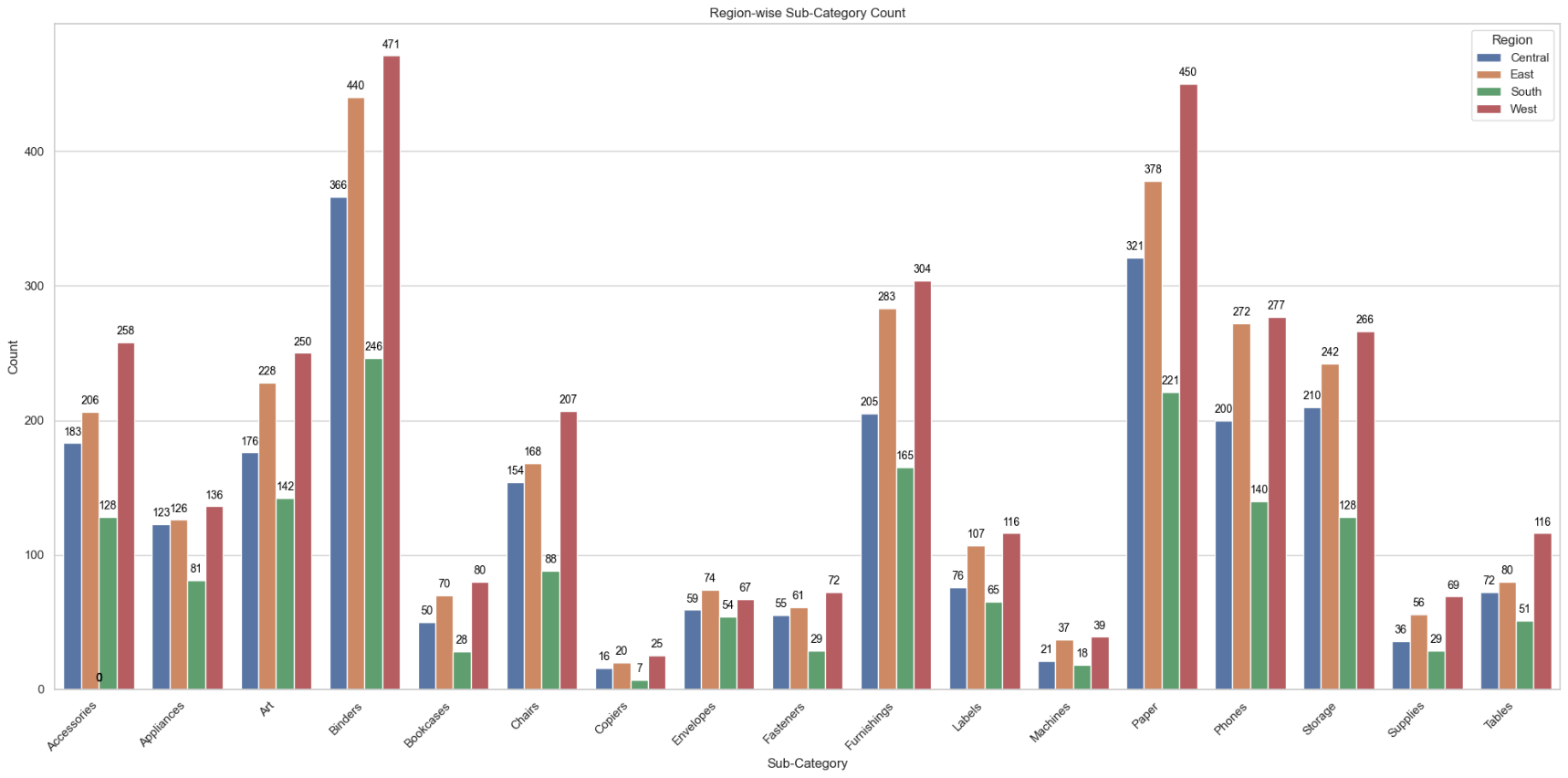
plt.ylabel("Count")

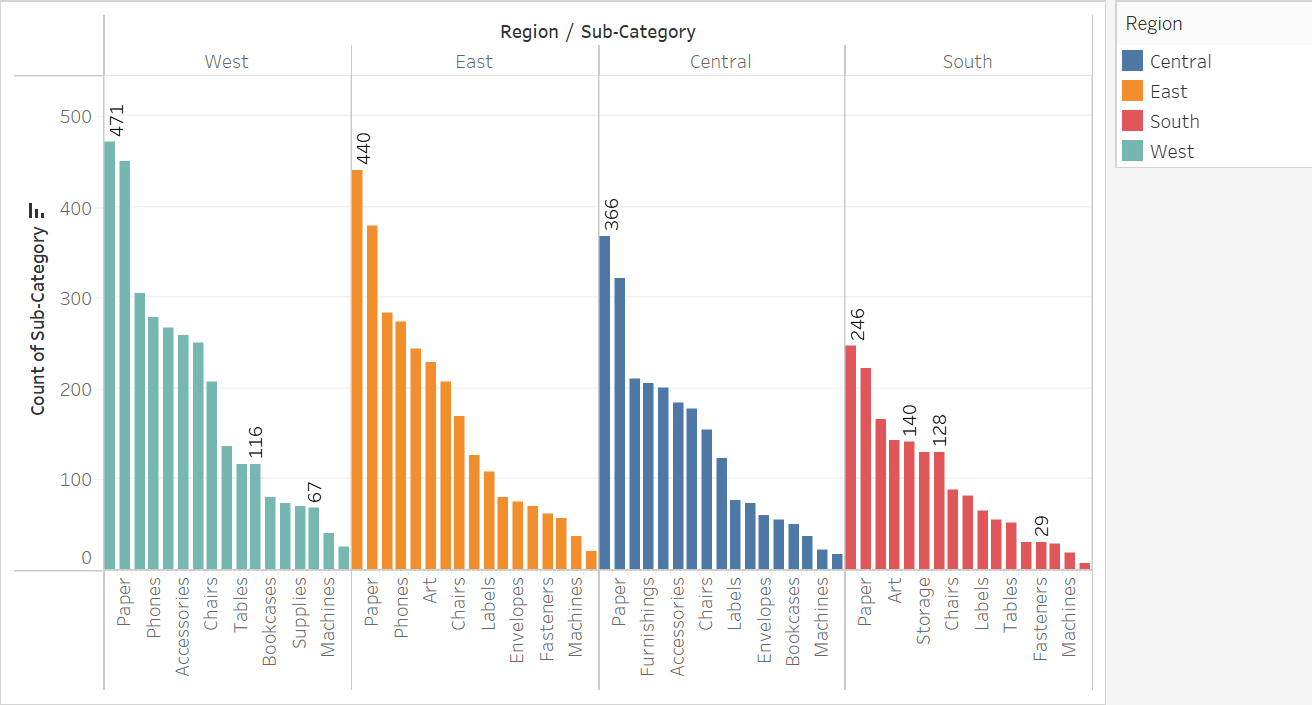
plt.title("Region-wise Sub-Category Count")

plt.legend(title="Region", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**

**Tableau Workbook :-**

1. **Which product has the highest profit?**

**Code:-**

Max\_Profitable\_Product = (

Dataset.groupby("Product Name")["Profit"].sum().sort\_values(ascending=False)

)[:10]

Max\_Profitable\_Product\_df = pd.DataFrame(

{"Product Name": Max\_Profitable\_Product.index, "Profit": Max\_Profitable\_Product.values}

)

plt.figure(figsize=(15, 8))

ax = sns.barplot(

data=Max\_Profitable\_Product\_df, x="Product Name", y="Profit", hue="Product Name"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".2f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Product Name")

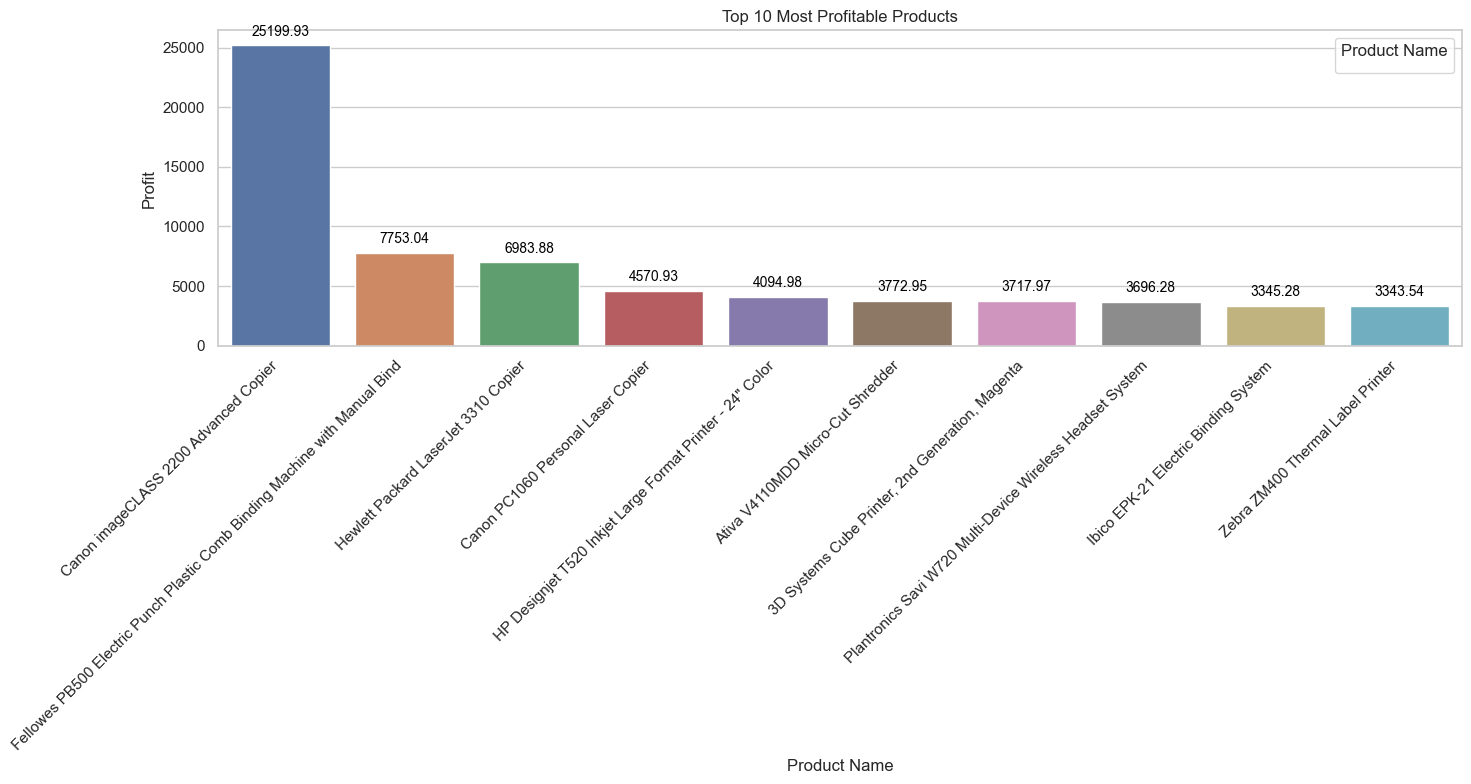
plt.ylabel("Profit")

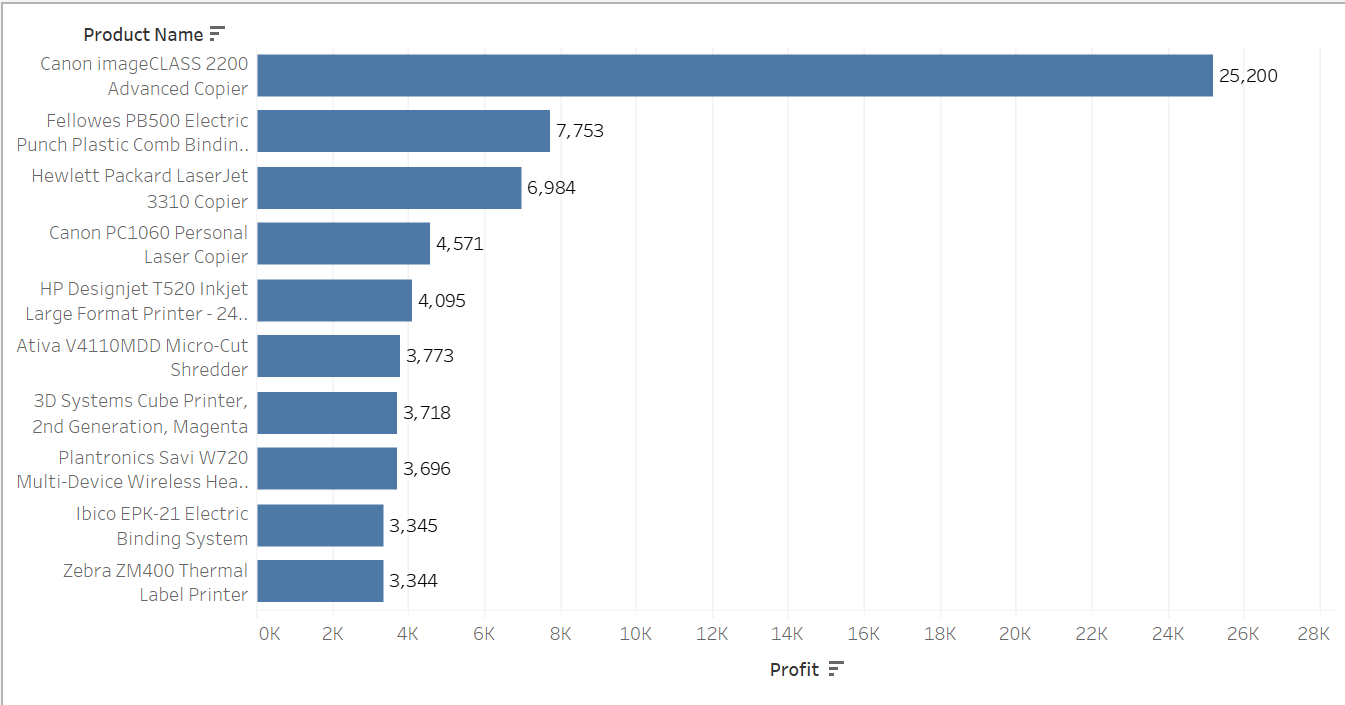
plt.title("Top 10 Most Profitable Products")

plt.legend(title="Product Name", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**

**Tableau Workbook :-**

1. **Which sub-category has the highest profit?**

**Code:-**

Max\_Profitable\_Sub\_Category = Dataset.groupby('Sub-Category')['Profit'].sum().sort\_values(ascending=False)

Max\_Profitable\_Sub\_Category\_df = pd.DataFrame(

{

"Sub-Category": Max\_Profitable\_Sub\_Category.index,

"Profit": Max\_Profitable\_Sub\_Category.values,

}

)

plt.figure(figsize=(15, 8))

ax = sns.barplot(

data=Max\_Profitable\_Sub\_Category\_df,

x="Sub-Category",

y="Profit",

hue="Sub-Category",

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".2f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Sub-Category")

plt.ylabel("Profit")

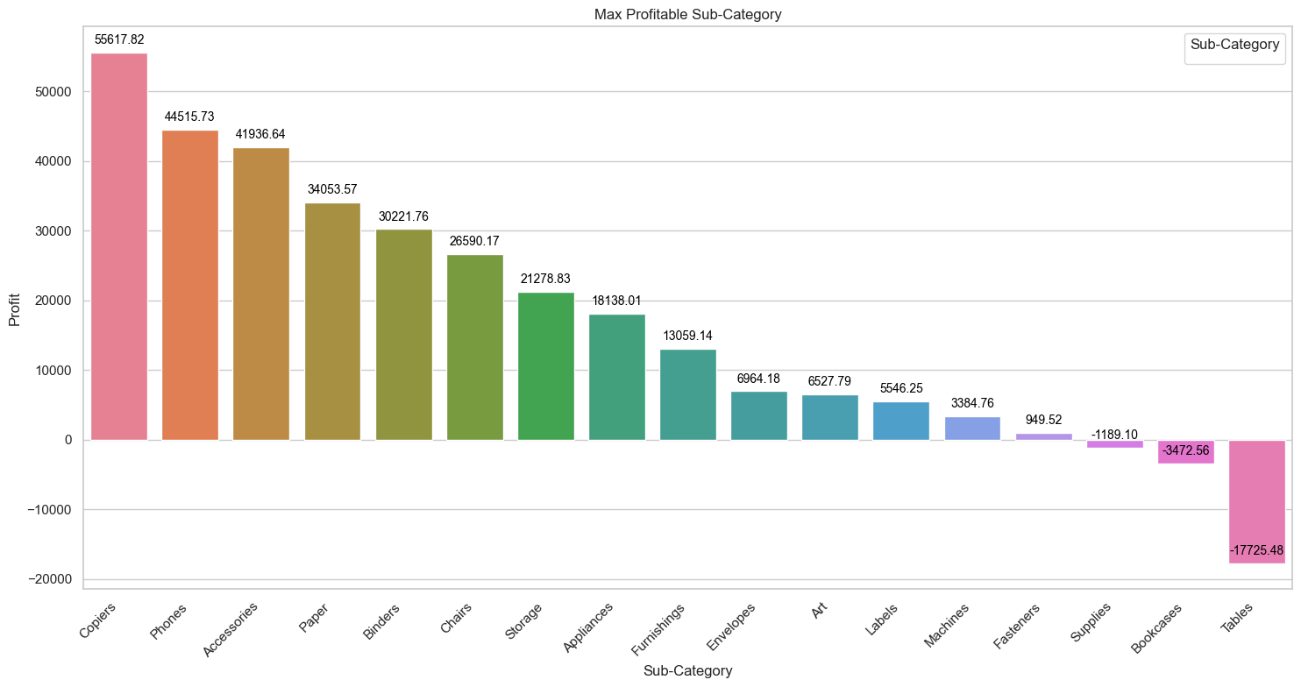
plt.title("Max Profitable Sub-Category")

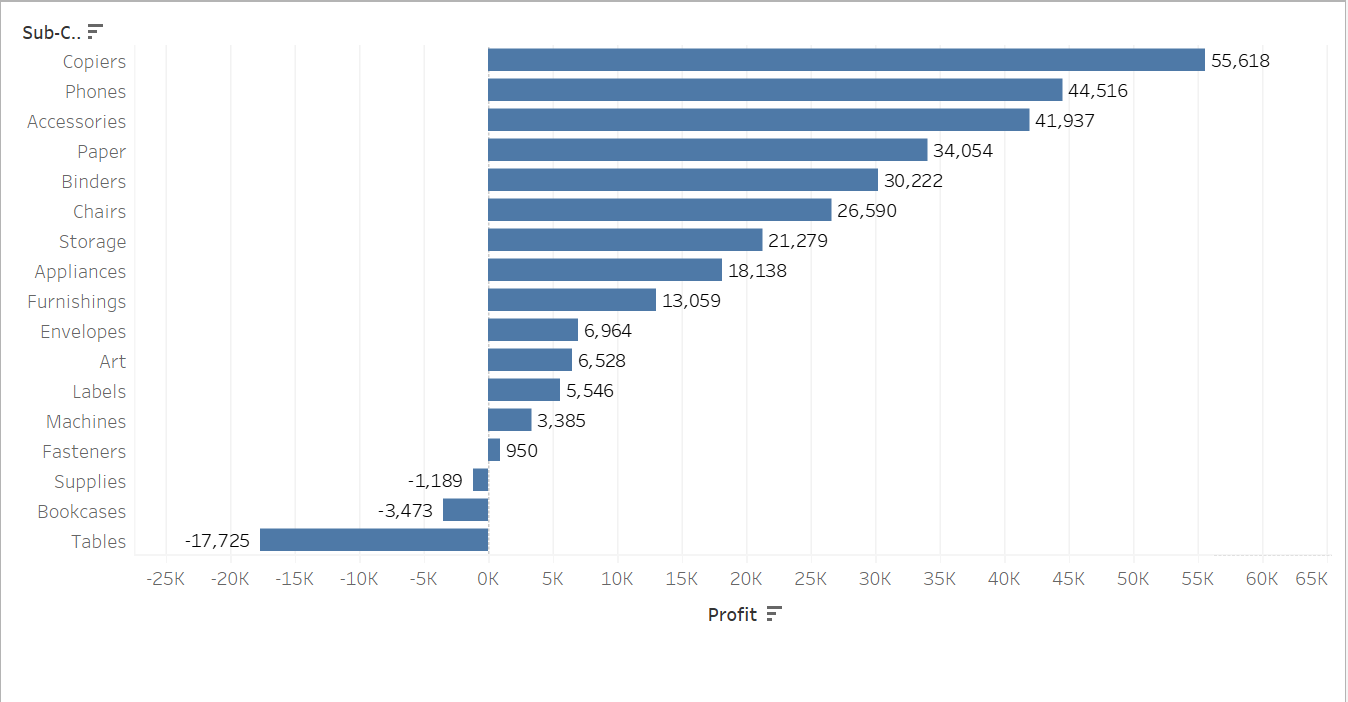
plt.legend(title="Sub-Category", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**



**Tableau Workbook :-**

1. **Which product has the highest loss?**

**Code:-**

Max\_Loss\_Product = Dataset.groupby("Product Name")["Profit"].sum().sort\_values(ascending=True)[:10]

Max\_Loss\_Product\_df = pd.DataFrame(

{

"Product Name": Max\_Loss\_Product.index,

"Profit": Max\_Loss\_Product.values,

}

)

plt.figure(figsize=(15, 8))

ax = sns.barplot(

data=Max\_Loss\_Product\_df, x="Product Name", y="Profit", hue="Product Name"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".2f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Product Name")

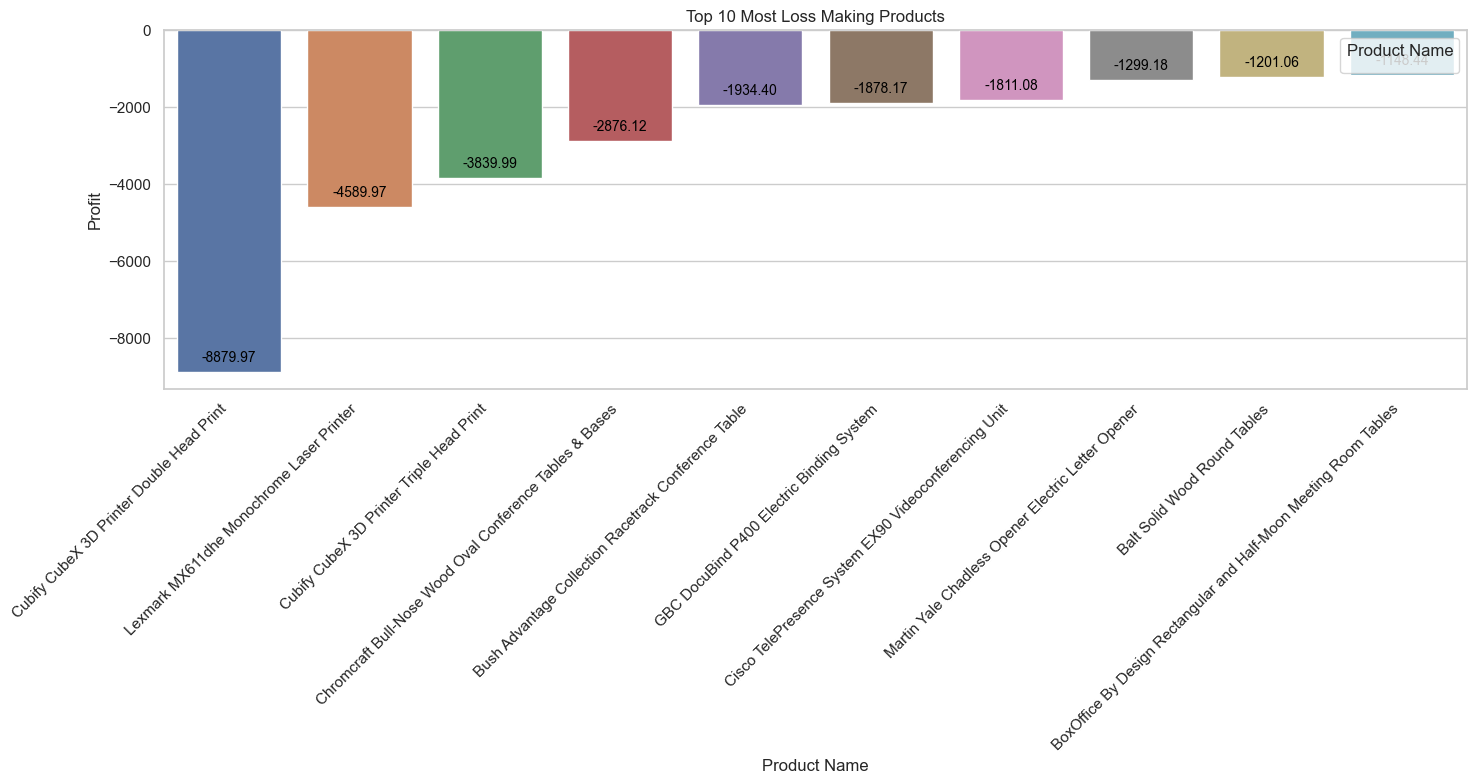
plt.ylabel("Profit")

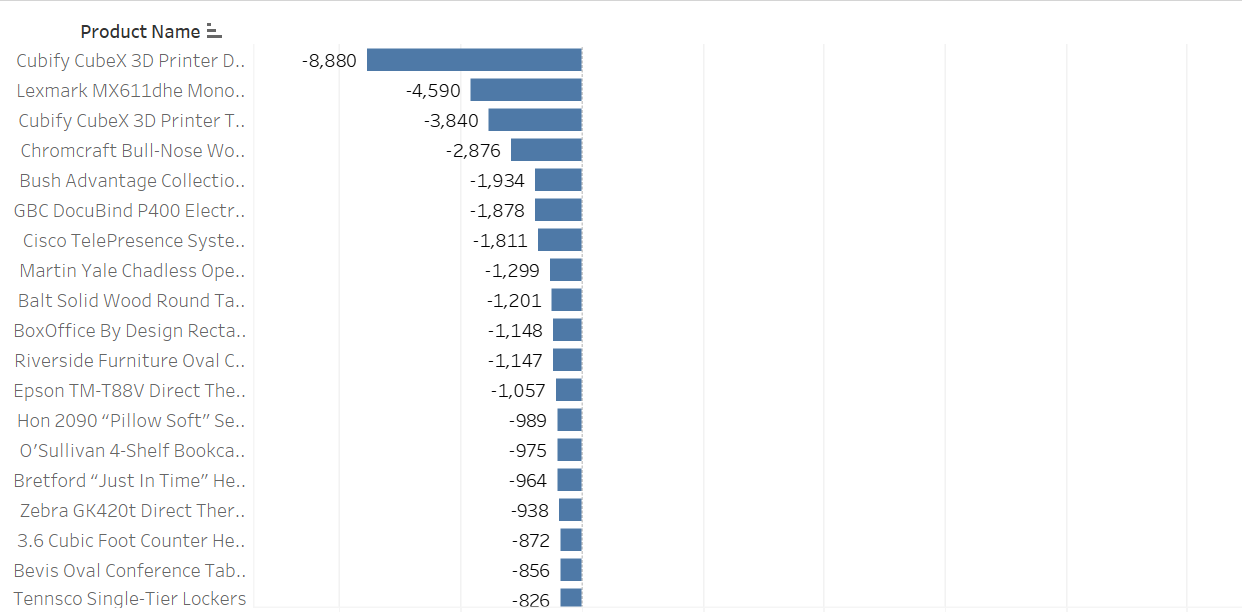
plt.title("Top 10 Most Loss Making Products")

plt.legend(title="Product Name", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**

**Tableau Workbook :-**

1. **Which sub-category has the highest loss?**

**Code:-**

Max\_Loss\_Sub\_Category = Dataset.groupby("Sub-Category")["Profit"].sum().sort\_values(ascending=True)

Max\_Loss\_Sub\_Category\_df = pd.DataFrame(

{

"Sub-Category": Max\_Loss\_Sub\_Category.index,

"Profit": Max\_Loss\_Sub\_Category.values,

}

)

plt.figure(figsize=(15, 8))

ax = sns.barplot(

data=Max\_Loss\_Sub\_Category\_df,

x="Sub-Category",

y="Profit",

hue="Sub-Category",

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".2f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Sub-Category")

plt.ylabel("Profit")

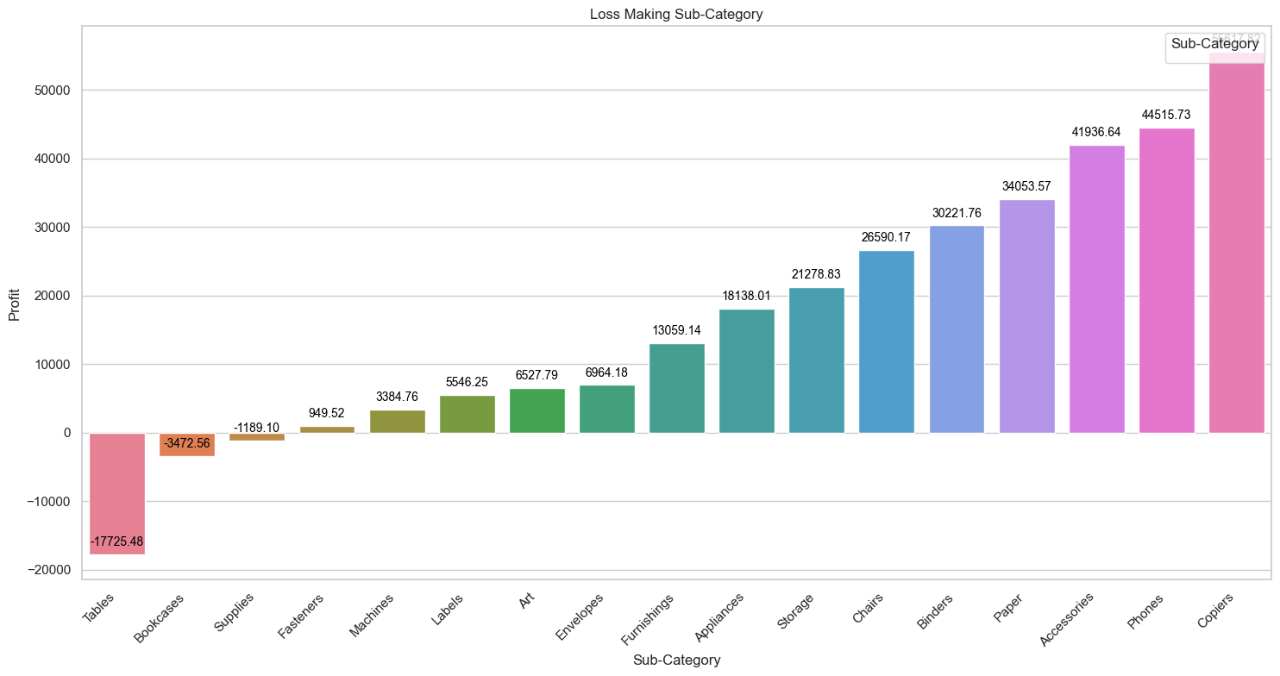
plt.title("Loss Making Sub-Category")

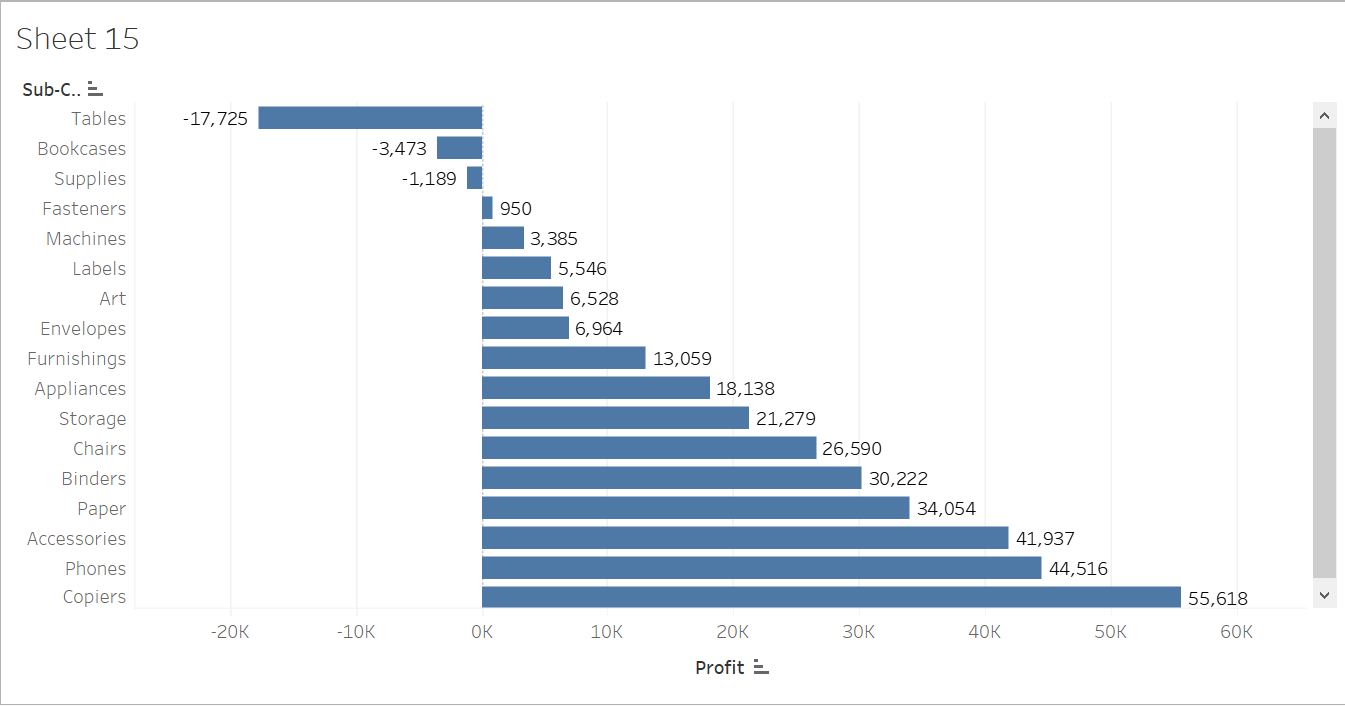
plt.legend(title="Sub-Category", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**



**Tableau Workbook :-**

1. **Who are the top 10 customers who order frequently?**

**Code:-**

Top\_10\_Customers = Dataset['Customer Name'].value\_counts()

Top\_10\_Customers\_df = pd.DataFrame({

"Customer Name": Top\_10\_Customers.index,

"Count": Top\_10\_Customers.values

})[:10]

plt.figure(figsize=(12, 8))

ax = sns.barplot(

data=Top\_10\_Customers\_df, x="Customer Name", y="Count", hue="Customer Name"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".0f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Customer Name")

plt.ylabel("Count")

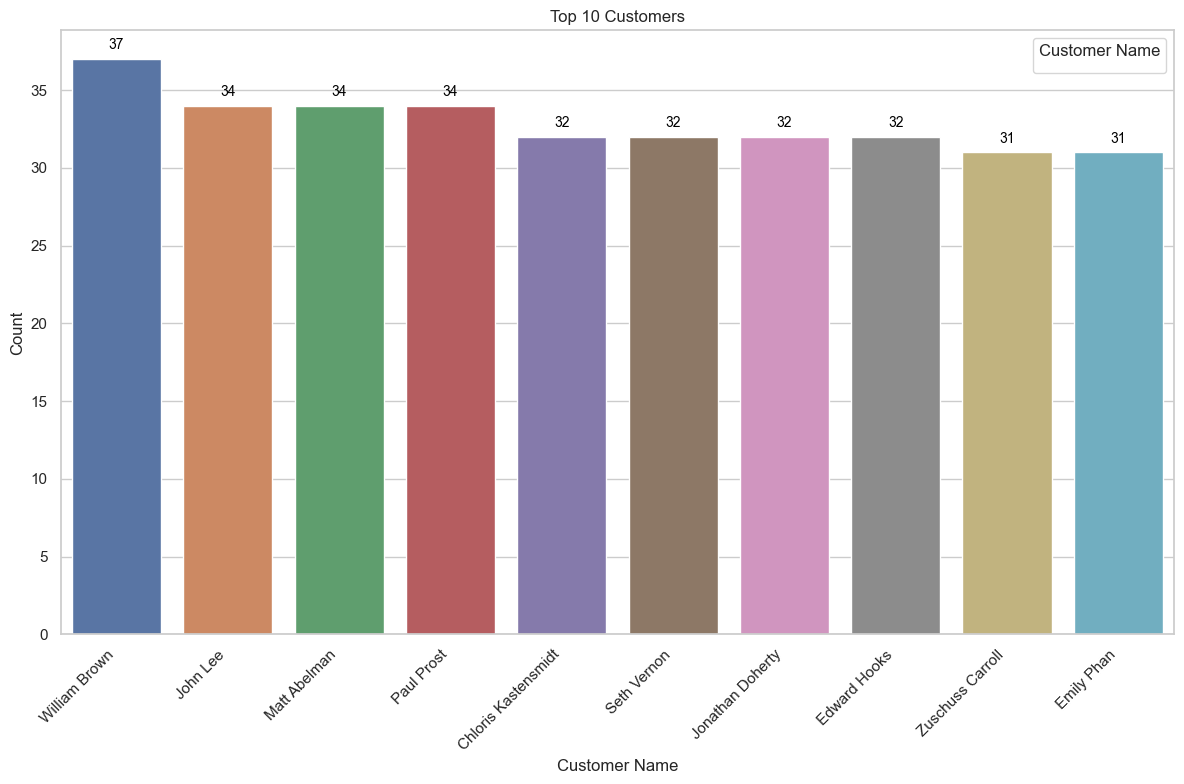
plt.title("Top 10 Customers")

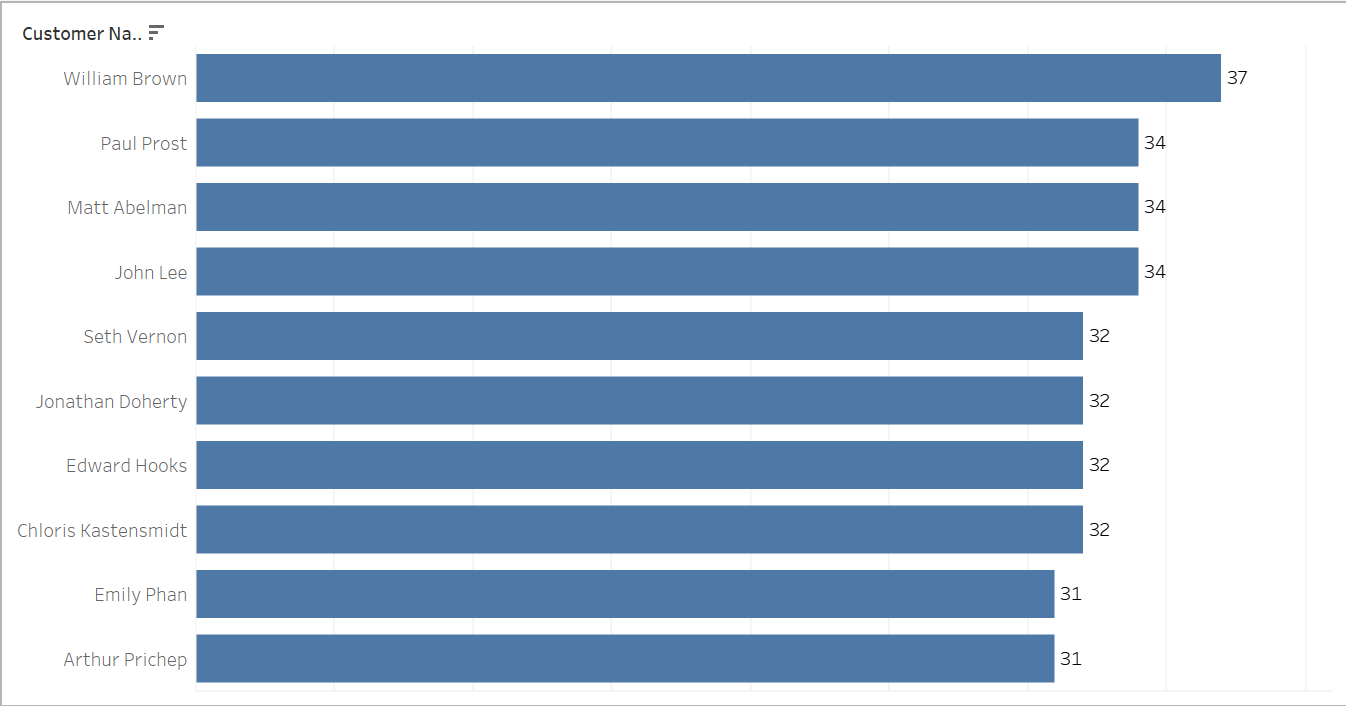
plt.legend(title="Customer Name", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**



**Tableau Workbook :-**

1. **How many unique customers are there in total?**

**Code:-**

unique\_customers\_count = Dataset["Customer Name"].nunique()

plt.figure(figsize=(8, 6))

bars = plt.bar(["Unique Customers"], [unique\_customers\_count], color="skyblue")

plt.xlabel("Category")

plt.ylabel("Count")

plt.title("Total Number of Unique Customers")

for bar in bars:

yval = bar.get\_height()

plt.text(

bar.get\_x() + bar.get\_width() / 2,

yval,

round(yval, 2),

va="bottom",

ha="center",

)

plt.xticks(rotation=45, ha="right")

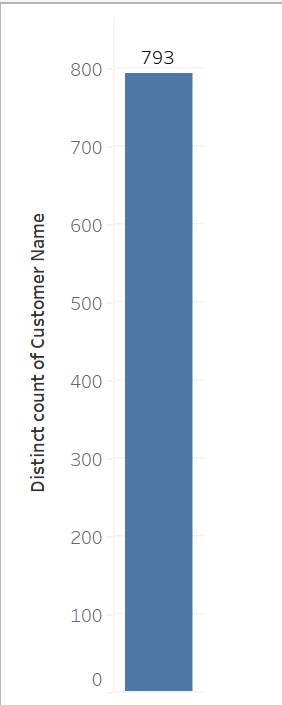
plt.tight\_layout()

plt.show()

**Output :-**

****

**Tableau Workbook :-**



1. **Who are the top 10 profitable customers in New York?**

**Code:-**

New\_York\_Customers = Dataset[Dataset["State"] == "New York"]

Profitable\_Customers = New\_York\_Customers.groupby("Customer Name")["Profit"].sum()

Top\_10\_Customers = pd.DataFrame(

{"Customer Name": Profitable\_Customers.index, "Profit": Profitable\_Customers.values}

).sort\_values(by="Profit", ascending=False)[:10]

plt.figure(figsize=(12, 8))

ax = sns.barplot(

data=Top\_10\_Customers, x="Customer Name", y="Profit", hue="Customer Name"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".0f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Customer Name")

plt.ylabel("Count")

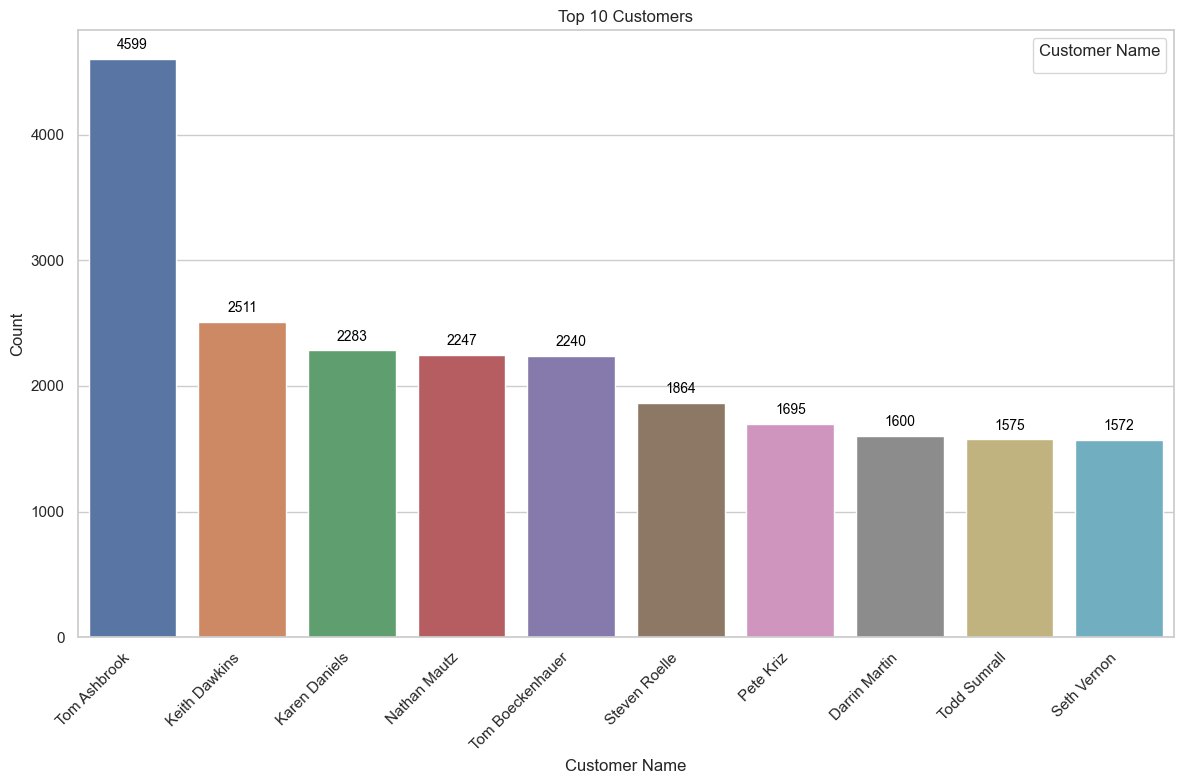
plt.title("Top 10 Customers")

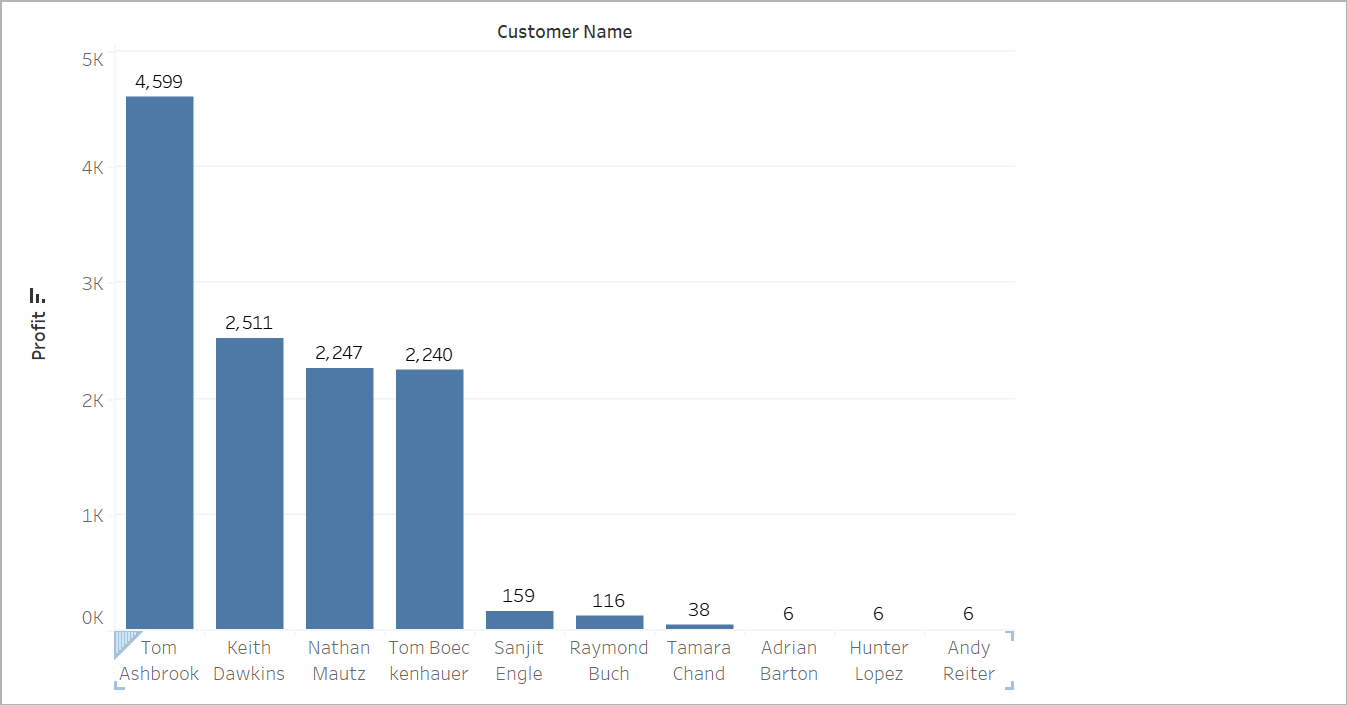
plt.legend(title="Customer Name", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**

****

**Tableau Workbook :-**

1. **Which product has the maximum time taken to ship?**

**Code:-**

Dataset['Time\_to\_Ship'] = (Dataset['Ship Date'] - Dataset['Order Date']).dt.days

Product\_wies\_Ship\_Day = Dataset.groupby("Product Name")["Time\_to\_Ship"].max()

Product\_wies\_Ship\_Day = pd.DataFrame(

{

"Product Name": Product\_wies\_Ship\_Day.index,

"Time to Ship": Product\_wies\_Ship\_Day.values,

}

).sort\_values(by="Time to Ship", ascending=False)[:10]

plt.figure(figsize=(12, 8))

ax = sns.barplot(

data=Product\_wies\_Ship\_Day, x="Product Name", y="Time to Ship", hue="Product Name"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".0f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Product Name")

plt.ylabel("Time to Ship")

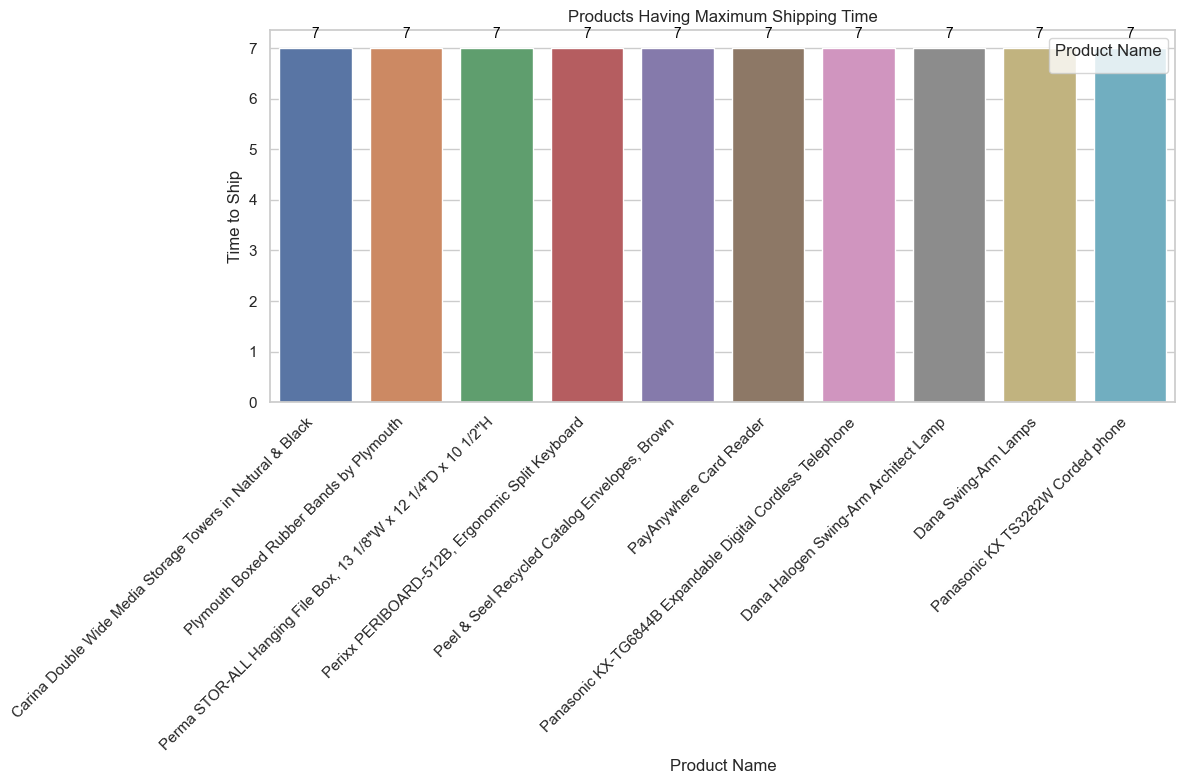
plt.title("Products Having Maximum Shipping Time")

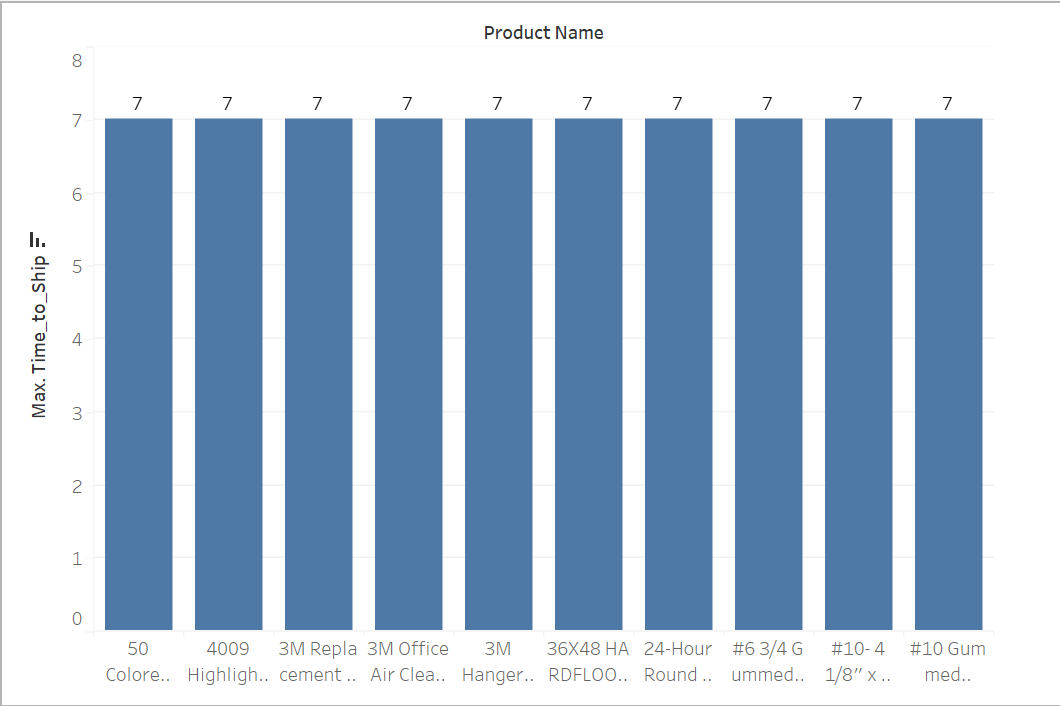
plt.legend(title="Product Name", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**

****

**Tableau Workbook :-**

1. **Which category has the maximum time taken to ship?**

**Code:-**

Dataset['Time\_to\_Ship'] = (Dataset['Ship Date'] - Dataset['Order Date']).dt.days

Category\_Ship\_Wise\_Groupping = Dataset.groupby('Category')['Time\_to\_Ship'].max()

Category\_Ship\_Wise\_Groupping\_df = pd.DataFrame(

{

"Category": Category\_Ship\_Wise\_Groupping.index,

"Time to Ship": Category\_Ship\_Wise\_Groupping.values,

}

).sort\_values(by="Time to Ship", ascending=False)[:10]

plt.figure(figsize=(12, 8))

ax = sns.barplot(

data=Category\_Ship\_Wise\_Groupping\_df, x="Category", y="Time to Ship", hue="Category"

)

for p in ax.patches:

ax.annotate(

format(p.get\_height(), ".0f"),

(p.get\_x() + p.get\_width() / 2.0, p.get\_height()),

ha="center",

va="center",

xytext=(0, 10),

textcoords="offset points",

fontsize=10,

color="black",

)

plt.xticks(rotation=45, ha="right")

plt.xlabel("Category")

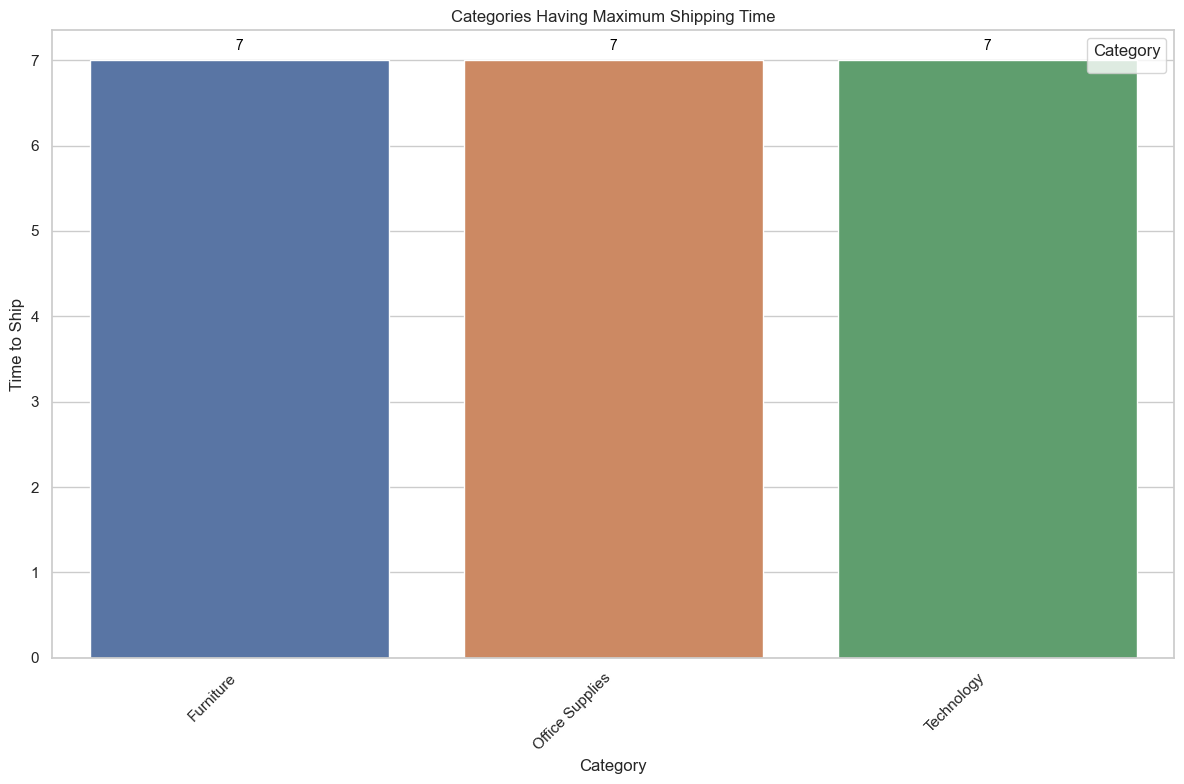
plt.ylabel("Time to Ship")

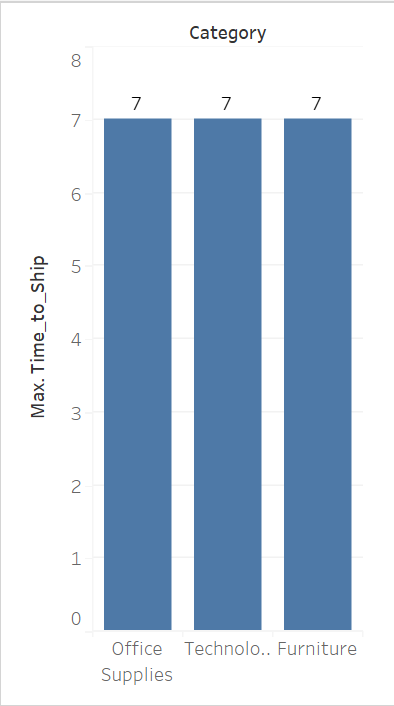
plt.title("Categories Having Maximum Shipping Time")

plt.legend(title="Category", loc="upper right")

plt.tight\_layout()

plt.show()

**Output :-**

**Tableau Workbook:-**

1. **What are the sales and profit figures for each year across all categories?**

**Code:-**

Dataset['Year'] = Dataset['Order Date'].dt.year

Category\_Ship\_Wise\_Groupping = Dataset.groupby('Category')['Time\_to\_Ship'].max()

Year\_wise\_profit\_and\_sales = Dataset.groupby(["Year", 'Category']).agg(

{"Profit": "sum", "Sales": "sum"})

grouped\_data = Dataset.groupby(['Year', 'Category'])[['Sales', 'Profit']].sum().reset\_index()

pivot\_data = grouped\_data.pivot\_table(index='Year', columns='Category', values=['Sales', 'Profit'])

pivot\_data.plot(kind='bar', figsize=(12, 6))

plt.title('Sales and Profit for each year (for all categories)')

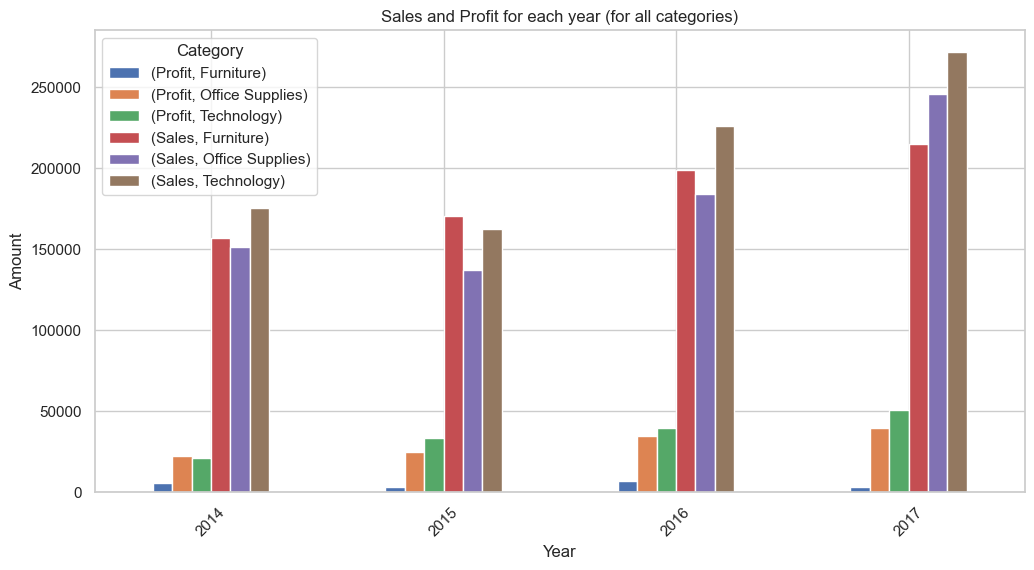
plt.xlabel('Year')

plt.ylabel('Amount')

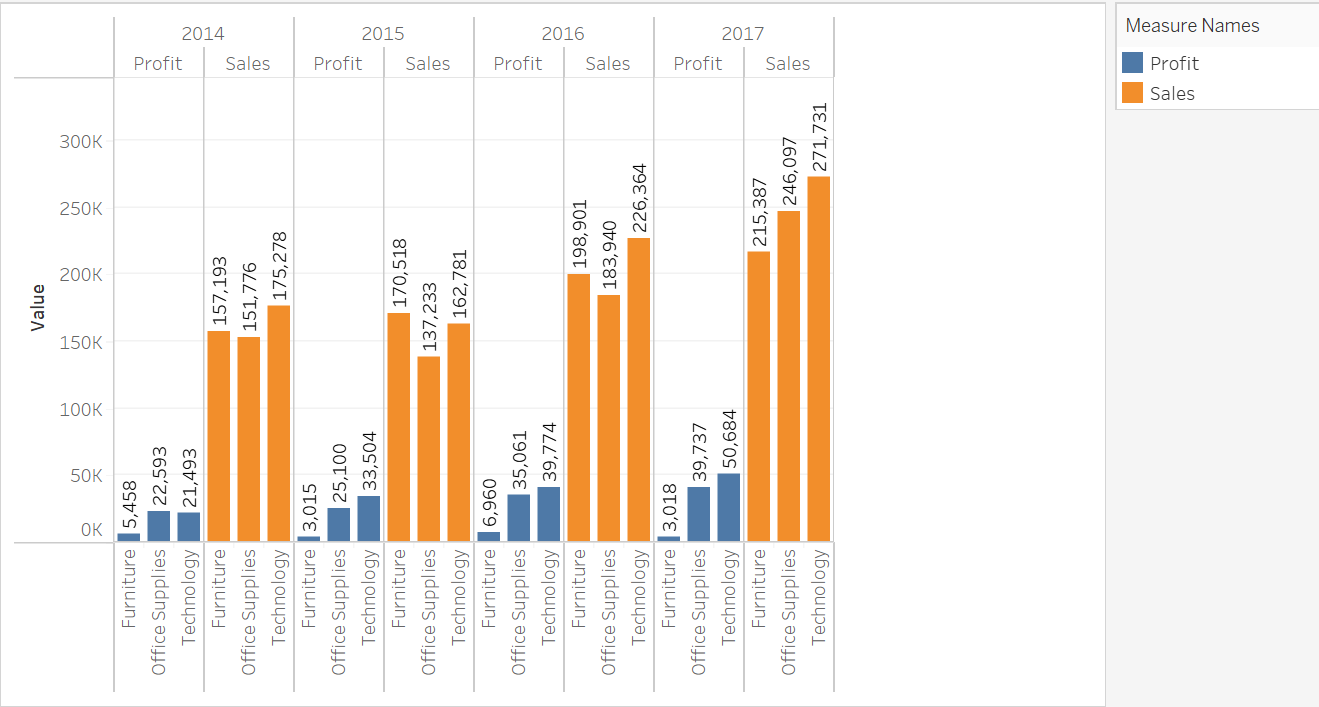
plt.legend(title='Category')

plt.xticks(rotation=45)

plt.show()

**Output :-**

**Tableau Workbook :-**

****

1. **Design the Dashboard of Questions 1, 16, 8, 9, and 21.**

**Code:-**

fig, axs = plt.subplots(3, 2, figsize=(30, 15))

Unique\_Categories = list(set(Dataset["Category"]))

axs[0, 0].bar(x=["No Of Unique Categories"], height=[len(Unique\_Categories)], width=0.5)

axs[0, 0].set\_title("Unique Categories")

Category\_wise\_Sales\_and\_profit = Dataset.groupby("Sub-Category")[

["Profit", "Sales"]

].sum()

Category\_wise\_Sales\_and\_profit.plot(kind="line", ax=axs[0, 1])

axs[0, 1].set\_title("Category-wise Sales and Profit")

axs[0, 1].set\_xlabel("Sub-Category")

axs[0, 1].set\_ylabel("Amount")

axs[0, 1].grid(True)

axs[0, 1].legend(loc="upper right")

Unique\_Products = Dataset["Product Name"].unique()

No\_of\_Unique\_Products = Dataset["Product Name"].nunique()

axs[1, 0].bar(x=["No Of Unique Products"], height=[len(Unique\_Products)], width=0.5)

axs[1, 0].set\_title("Unique Products")

Top\_10\_Customers = Dataset["Customer Name"].value\_counts()

Top\_10\_Customers\_df = pd.DataFrame(

{"Customer Name": Top\_10\_Customers.index, "Count": Top\_10\_Customers.values}

)[:10]

sns.barplot(

data=Top\_10\_Customers\_df,

x="Customer Name",

y="Count",

hue="Customer Name",

ax=axs[1, 1],

)

axs[1, 1].set\_title("Top 10 Customers")

axs[1, 1].set\_xlabel("Customer Name")

axs[1, 1].set\_ylabel("Count")

axs[1, 1].legend(title="Customer Name", loc="upper right")

axs[1, 1].tick\_params(axis="x", rotation=45)

pivot\_data.plot(kind="bar", ax=axs[2, 0], figsize=(16, 6), legend=True)

axs[2, 0].set\_title("Sales and Profit for each year (for all categories)")

axs[2, 0].set\_xlabel("Year")

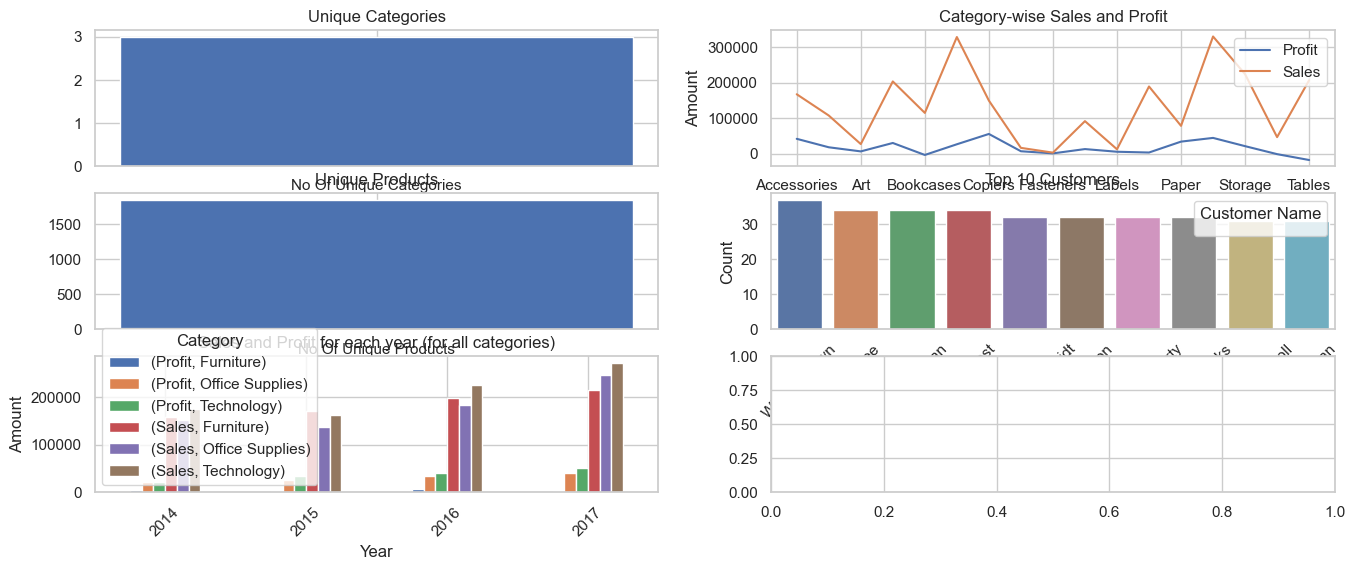
axs[2, 0].set\_ylabel("Amount")

axs[2, 0].legend(title="Category")

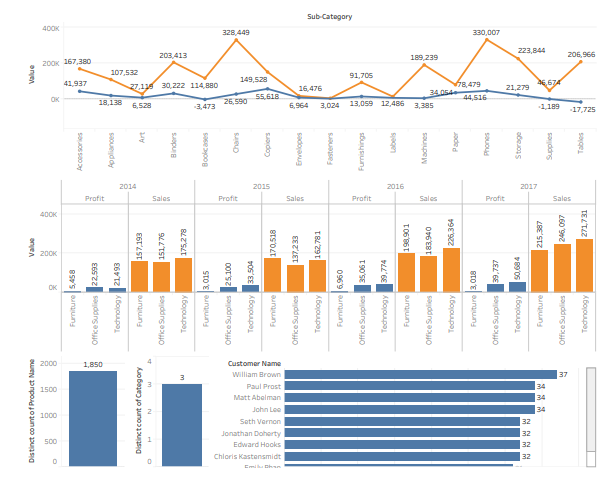
axs[2, 0].tick\_params(axis="x", rotation=45)

plt.show()

**Output :-**



**Tableau Workbook :-**

****