HNG

March 3, 2025

[140]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

[3]:

```
data_df = pd.read_csv(r"C:\Users\essie\Desktop\HNG\SampleSuperstore.csv")
```

[4]:

```
data_df.head()
```

[4]:

```
        Ship Mode    Segment        Country             City       State  \
0    Second Class   Consumer  United States        Henderson    Kentucky
1    Second Class   Consumer  United States        Henderson    Kentucky
2    Second Class  Corporate  United States      Los Angeles  California
3  Standard Class   Consumer  United States  Fort Lauderdale     Florida
4  Standard Class   Consumer  United States  Fort Lauderdale     Florida

   Postal Code Region         Category Sub-Category     Sales  Quantity  \
0        42420  South        Furniture    Bookcases  261.9600         2
1        42420  South        Furniture       Chairs  731.9400         3
2        90036   West  Office Supplies       Labels   14.6200         2
3        33311  South        Furniture       Tables  957.5775         5
4        33311  South  Office Supplies      Storage   22.3680         2

   Discount    Profit
0      0.00   41.9136
1      0.00  219.5820
2      0.00    6.8714
3      0.45 -383.0310
4      0.20    2.5164
```

[5]:

```
data_df.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
```

[7]:

```
data_df['Postal Code'] = data_df['Postal Code'].astype(object)
```

[8]:

```
data_df.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   object
 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(1), object(9)
memory usage: 1015.1+ KB
```

[9]:

```
data_df.describe()
```

[9]:

```
              Sales     Quantity     Discount       Profit
count   9994.000000  9994.000000  9994.000000  9994.000000
mean     229.858001     3.789574     0.156203    28.656896
std      623.245101     2.225110     0.206452   234.260108
min        0.444000     1.000000     0.000000 -6599.978000
25%       17.280000     2.000000     0.000000     1.728750
```

```
50%         54.490000     3.000000     0.200000     8.666500
75%        209.940000     5.000000     0.200000    29.364000
max      22638.480000    14.000000     0.800000  8399.976000
```

[11]:

```
if data_df.duplicated().sum() > 0:
    print('duplicates found')
else:
    print('no duplicates')
```

```
duplicates found
```

[126]:

```
# 1. Check for Duplicates
duplicates = data_df.duplicated()
print(f"Number of duplicate rows: {duplicates.sum()}")

# Optional: Display duplicate rows
if duplicates.sum() > 0:
    print("Duplicate rows:")
    print(data_df[duplicates])
```

```
Number of duplicate rows: 17
Duplicate rows:
           Ship Mode      Segment        Country           City         State  \
950   Standard Class  Home Office  United States   Philadelphia  Pennsylvania
3406  Standard Class  Home Office  United States       Columbus          Ohio
3670  Standard Class     Consumer  United States          Salem        Oregon
4117  Standard Class     Consumer  United States    Los Angeles    California
4553  Standard Class     Consumer  United States  San Francisco    California
5905        Same Day  Home Office  United States  San Francisco    California
6146  Standard Class    Corporate  United States  San Francisco    California
6334  Standard Class     Consumer  United States  New York City      New York
6357  Standard Class    Corporate  United States        Seattle    Washington
7608  Standard Class     Consumer  United States  San Francisco    California
7735  Standard Class    Corporate  United States        Seattle    Washington
7759  Standard Class    Corporate  United States        Houston         Texas
8032     First Class     Consumer  United States        Houston         Texas
8095    Second Class     Consumer  United States        Seattle    Washington
9262  Standard Class     Consumer  United States        Detroit      Michigan
9363  Standard Class  Home Office  United States        Seattle    Washington
9477    Second Class    Corporate  United States        Chicago      Illinois

     Postal Code   Region         Category Sub-Category    Sales  Quantity  \
950        19120     East  Office Supplies        Paper   15.552         3
3406       43229     East        Furniture       Chairs  281.372         2
3670       97301     West  Office Supplies        Paper   10.368         2
4117       90036     West  Office Supplies        Paper   19.440         3
4553       94122     West  Office Supplies        Paper   12.840         3
5905       94122     West  Office Supplies       Labels   41.400         4
```

```
6146       94122     West  Office Supplies          Art   11.760         4
6334       10011     East  Office Supplies        Paper   49.120         4
6357       98103     West  Office Supplies        Paper   25.920         4
7608       94122     West  Office Supplies        Paper   25.920         4
7735       98105     West  Office Supplies        Paper   19.440         3
7759       77041  Central  Office Supplies        Paper   15.552         3
8032       77041  Central  Office Supplies        Paper   47.952         3
8095       98115     West  Office Supplies        Paper   12.960         2
9262       48227  Central        Furniture       Chairs  389.970         3
9363       98105     West        Furniture  Furnishings   22.140         3
9477       60653  Central  Office Supplies      Binders    3.564         3

      Discount   Profit
950        0.2   5.4432
3406       0.3 -12.0588
3670       0.2   3.6288
4117       0.0   9.3312
4553       0.0   5.7780
5905       0.0  19.8720
6146       0.0   3.1752
6334       0.0  23.0864
6357       0.0  12.4416
7608       0.0  12.4416
7735       0.0   9.3312
7759       0.2   5.4432
8032       0.2  16.1838
8095       0.0   6.2208
9262       0.0  35.0973
9363       0.0   6.4206
9477       0.8  -6.2370
```

[127]:
```
cleaned_data = data_df.drop_duplicates()
```

[128]:
```
print(f"Number of rows after removing duplicates: {len(cleaned_data)}")
```

```
Number of rows after removing duplicates: 9977
```

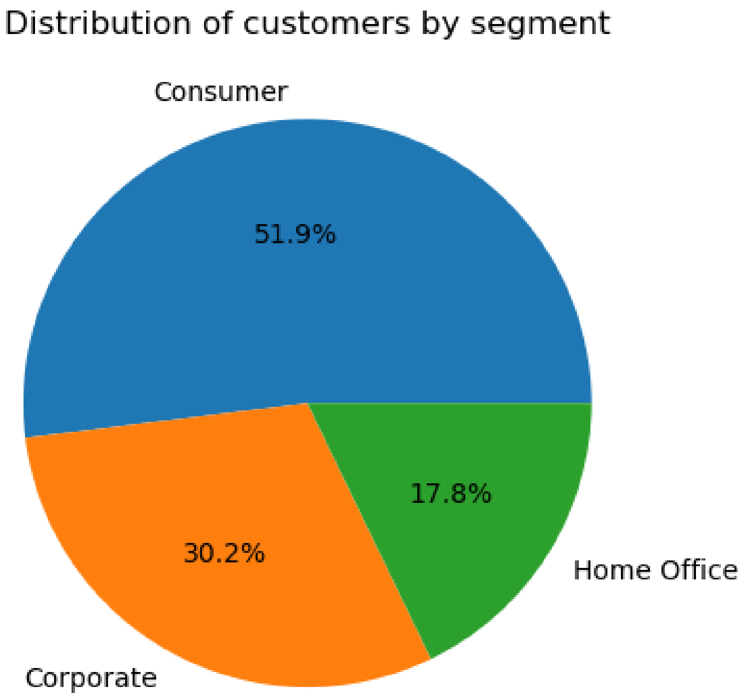
Exploratory Analysis

[129]:
```
#number of customers in each segment
no_customers = cleaned_data['Segment'].value_counts().reset_index()
no_customers = no_customers.rename(columns={'Segment':'customer_type','count':
  ↪'Total_customers'})
print(no_customers)
```

```
  customer_type  Total_customers
0      Consumer             5183
1     Corporate             3015
2   Home Office             1779
```

[130]:

```
plt.
 ↪pie(no_customers['Total_customers'],labels=no_customers['customer_type'],autopct='%1.
 ↪1f%%')
plt.title('Distribution of customers by segment')
plt.show()
```



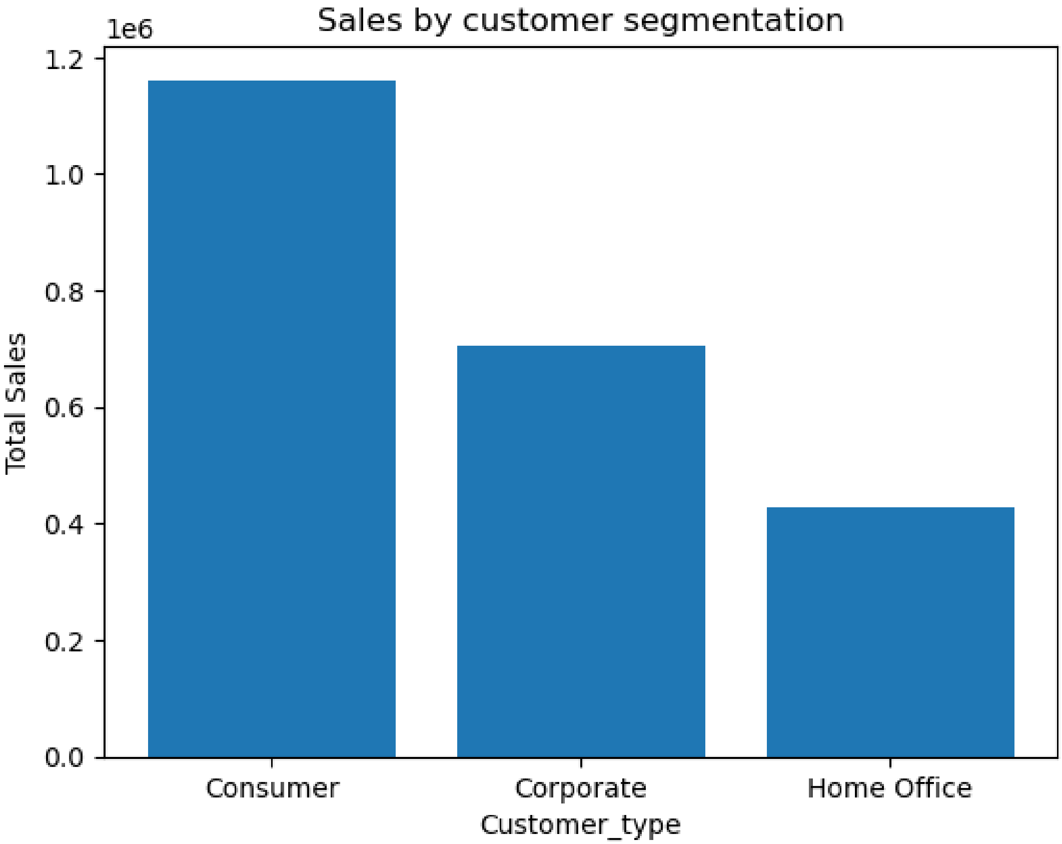
[131]:

```
#sales by categopry
sales_by_category = cleaned_data.groupby('Segment')['Sales'].sum().reset_index()
sales_by_category = sales_by_category.rename(columns={'Segment':
 ↪'Customer_type','Sales':'Total_sales'})
print(sales_by_category)
```

```
  Customer_type   Total_sales
0      Consumer  1.160833e+06
1     Corporate  7.060701e+05
2   Home Office  4.292927e+05
```

[52]:

```
plt.bar(sales_by_category['Customer_type'], sales_by_category['Total_sales']) ␣
 ↪# x, height

plt.title('Sales by customer segmentation')
plt.xlabel('Customer_type')
plt.ylabel('Total Sales')
plt.show()
```



shipping Mode

[54]:

```
type_of_shipping = cleaned_data['Ship Mode'].unique()
print(type_of_shipping)
```

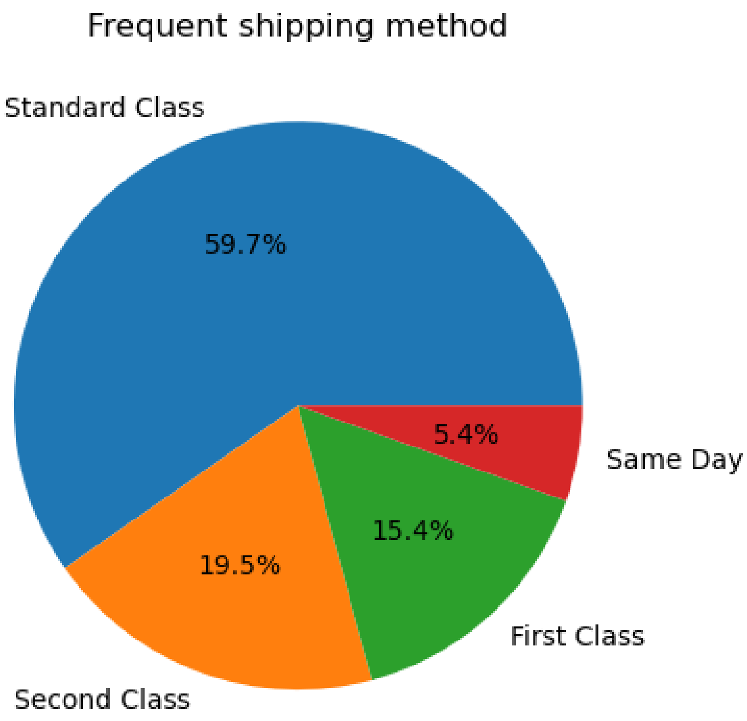
```
['Second Class' 'Standard Class' 'First Class' 'Same Day']
```

[58]:

```
#frquency of use
shipping_mode =cleaned_data['Ship Mode'].value_counts().reset_index()
shipping_mode = shipping_mode.rename(columns={'Ship Mode':'Mode of shipment',␣
  ↪'count':'use frequency'})
print(shipping_mode)
```

```
  Mode of shipment  use frequency
0   Standard Class           5955
1     Second Class           1943
2      First Class           1537
3         Same Day            542
```

```
[62]: plt.pie(shipping_mode['use frequency'],labels=shipping_mode['Mode of␣
        ↪shipment'],autopct='%1.1f%%')
      plt.title('Frequent shipping method')
      plt.show()
```



Geogrphical Analysis

```
[132]: States =cleaned_data['State'].value_counts().reset_index()
       States = States.rename(columns={'count':'Number of customers'})
       print(States.head())
```

```
         State  Number of customers
0   California                 1996
1     New York                 1127
2        Texas                  983
3 Pennsylvania                  586
4   Washington                  502
```

```
[71]: City =cleaned_data['City'].value_counts().reset_index()
      City = City.rename(columns={'count':'Number of customers'})
      print(City.head())
```

```
           City  Number of customers
0 New York City                  914
```

```
1     Los Angeles                  746
2    Philadelphia                  536
3   San Francisco                  506
4         Seattle                  424
```

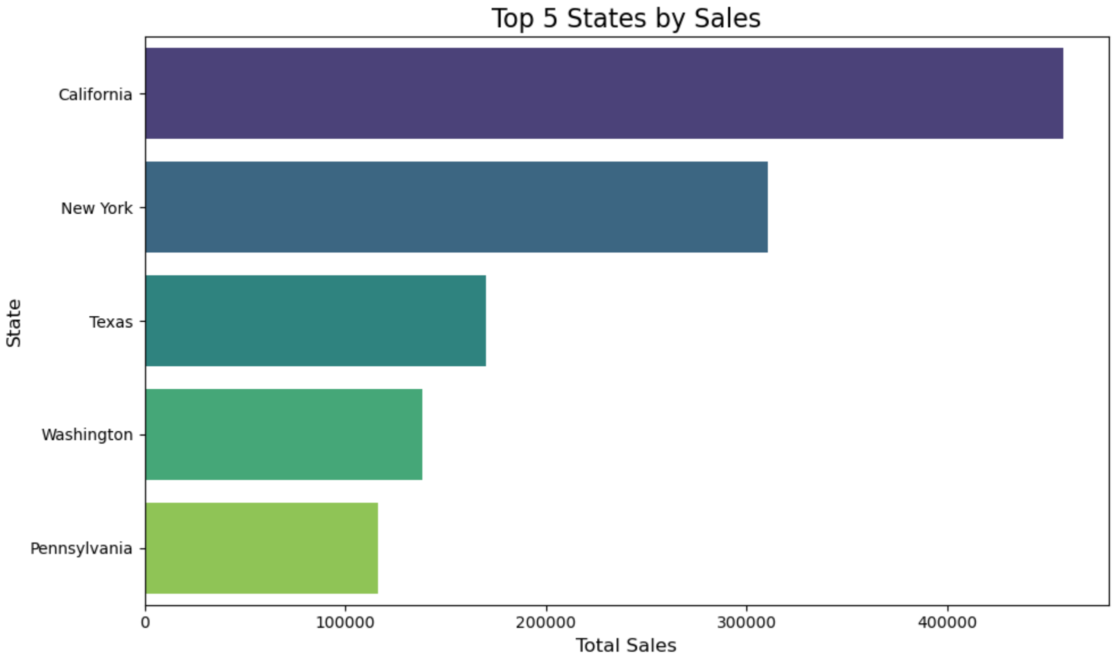
[136]:

```
#Sales by States
State_sales = cleaned_data.groupby(['State'])['Sales'].sum().reset_index()
top_states_sales=State_sales.sort_values(by='Sales',ascending =False)
print(top_states_sales.head(5).reset_index(drop=True))
```

```
          State        Sales
0    California  457576.2715
1      New York  310827.1510
2         Texas  170124.5418
3    Washington  138560.8100
4  Pennsylvania  116496.3620
```

[141]:

```
import seaborn as sns

top_5_states_sales = top_states_sales.head(5)

plt.figure(figsize=(10, 6))
sns.barplot(
    x='Sales',
    y='State',
    data=top_5_states_sales,
    palette='viridis'
)
plt.title('Top 5 States by Sales', fontsize=16)
plt.xlabel('Total Sales', fontsize=12)
plt.ylabel('State', fontsize=12)

plt.tight_layout()
plt.show()
```



Product analysis

[79]:
```
product_category = cleaned_data['Category'].unique()
print(product_category)
```

```
['Furniture' 'Office Supplies' 'Technology']
```

[86]:
```
subcategory_count = cleaned_data.groupby('Category')['Sub-Category'].nunique().
  ↪reset_index()
subcategory_count = subcategory_count.sort_values(by ='Sub-Category', ascending␣
  ↪= False)
print(subcategory_count.reset_index(drop=True))
```

```
          Category  Sub-Category
0  Office Supplies             9
1        Furniture             4
2       Technology             4
```

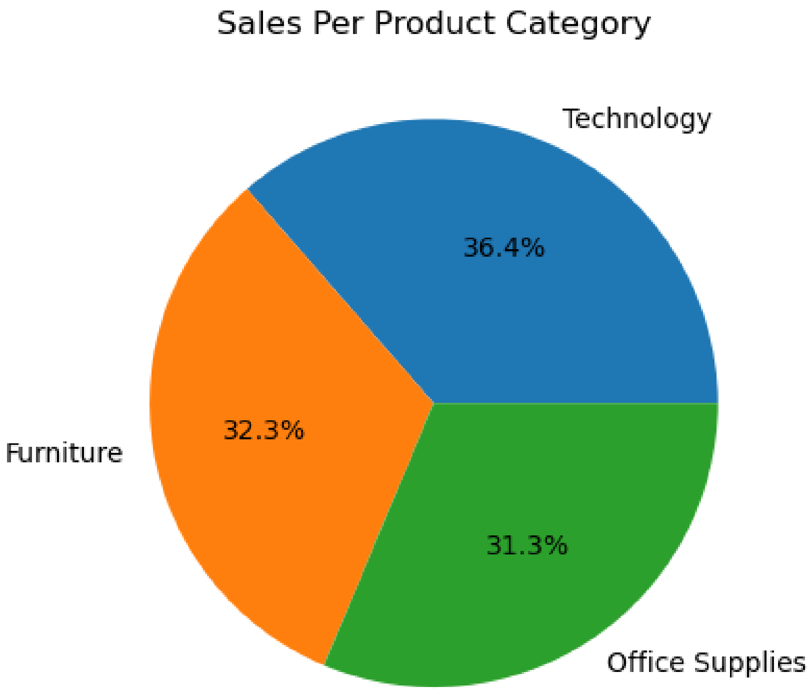
[92]:
```
#Sales per category

Sales_per_category = cleaned_data.groupby(['Category'])['Sales'].sum().
  ↪reset_index()
Sales_per_category = Sales_per_category.sort_values(by = 'Sales', ascending =␣
  ↪False)
print(Sales_per_category.reset_index(drop=True))
```

```
          Category        Sales
```

```
0        Technology  836154.0330
1         Furniture  741306.3133
2   Office Supplies  718735.2440
```

[167]:

```
plt.
 ↪pie(Sales_per_category['Sales'],labels=Sales_per_category['Category'],autopct='%1.
 ↪1f%%')
plt.title('Sales Per Product Category')
plt.show()
```



[95]:

```
#Total Sales
Total_sales = cleaned_data['Sales'].sum()
print(Total_sales)
```

```
2296195.5903
```

[96]:

```
# Total quantity
Total_quantity =cleaned_data['Quantity'].sum()
print(Total_quantity)
```

```
37820
```

[98]:

```
#Total profit
Total_profit = cleaned_data['Profit'].sum()
print(Total_profit)
```

```
286241.4226
```

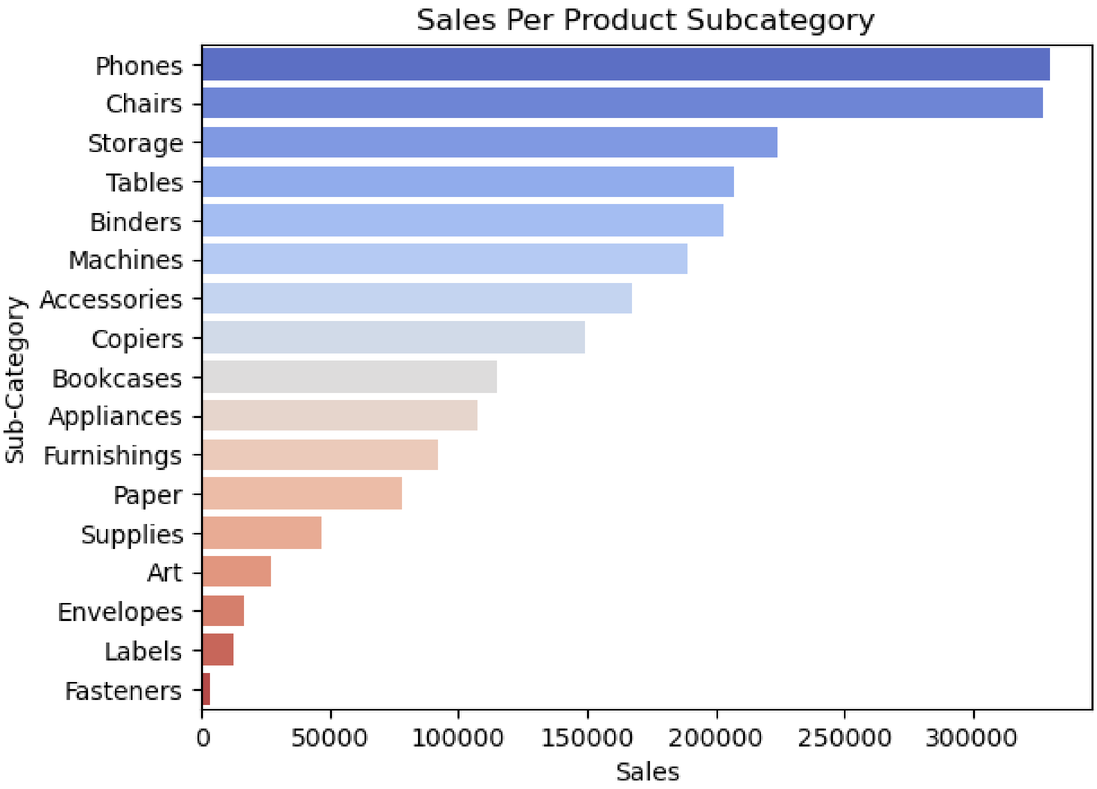
[146]:

```
product_subcategory = cleaned_data.groupby(['Sub-Category'])['Sales'].sum().
  ↪reset_index()
Top_product_subcategory = product_subcategory.sort_values(by='Sales', ascending␣
  ↪=False)
print(Top_product_subcategory.reset_index(drop=True))
```

```
   Sub-Category        Sales
0        Phones  330007.0540
1        Chairs  327777.7610
2       Storage  223843.6080
3        Tables  206965.5320
4       Binders  203409.1690
5      Machines  189238.6310
6   Accessories  167380.3180
7       Copiers  149528.0300
8     Bookcases  114879.9963
9    Appliances  107532.1610
10  Furnishings   91683.0240
11        Paper   78224.1420
12     Supplies   46673.5380
13          Art   27107.0320
14    Envelopes   16476.4020
15       Labels   12444.9120
16    Fasteners    3024.2800
```

[161]:

```
Top_product_subcategory = product_subcategory.sort_values(by='Sales', ascending␣
  ↪=False)
plt.
  ↪barh(Top_product_subcategory['Sub-Category'],Top_product_subcategory['Sales'])
plt.xlabel('Sales')
plt.ylabel('Product Subcategory')
plt.title('Sales Per Product Subcategory')
sns.barplot(
    x='Sales',
    y='Sub-Category',
    data=Top_product_subcategory,
    palette='coolwarm'
)
plt.show()
```



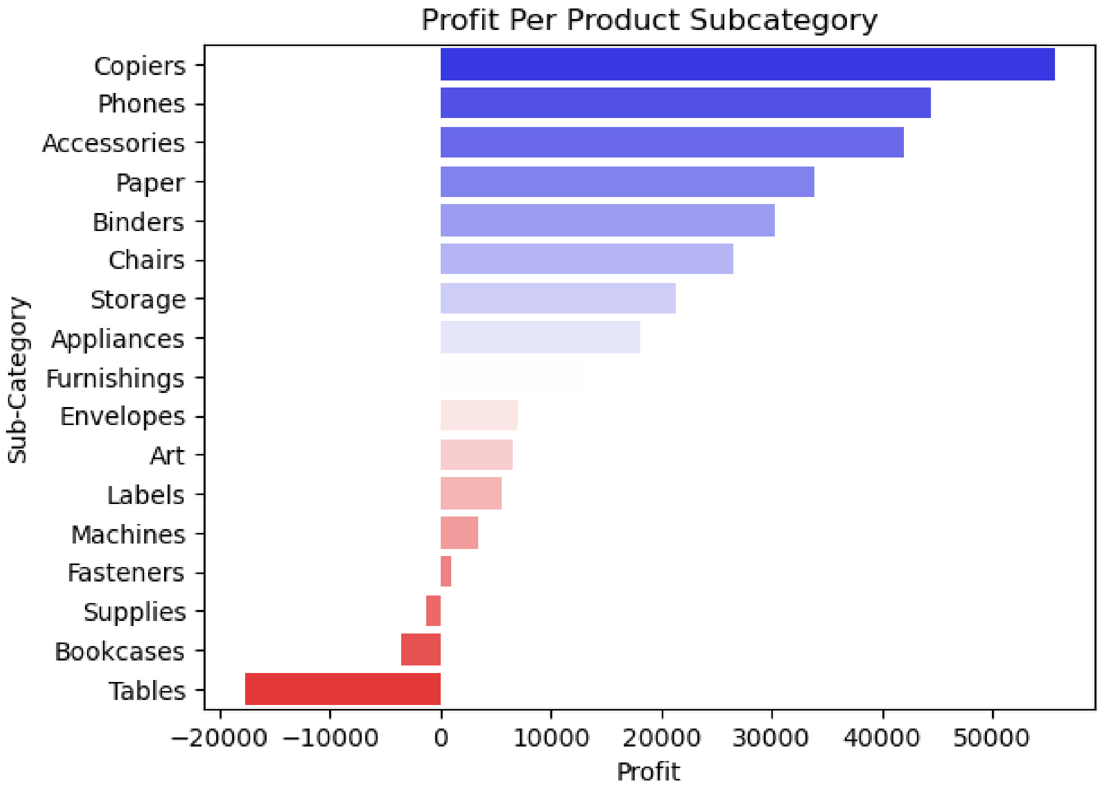
```
[162]: product_subcategory = cleaned_data.groupby(['Sub-Category'])['Profit'].sum().
         ↪reset_index()
       Top_product_subcategory = product_subcategory.sort_values(by='Profit',␣
         ↪ascending =False)
       print(Top_product_subcategory.reset_index(drop=True))
```

```
   Sub-Category       Profit
0       Copiers   55617.8249
1        Phones   44515.7306
2   Accessories   41936.6357
3         Paper   33944.2395
4       Binders   30228.0003
5        Chairs   26567.1278
6       Storage   21278.8264
7    Appliances   18138.0054
8   Furnishings   13052.7230
9     Envelopes    6964.1767
10          Art    6524.6118
11       Labels    5526.3820
12     Machines    3384.7569
13    Fasteners     949.5182
14     Supplies   -1189.0995
```

```
15     Bookcases  -3472.5560
16        Tables -17725.4811
```

[164]:

```
Top_product_subcategory = product_subcategory.sort_values(by='Profit',␣
 ↪ascending =False)
plt.
 ↪barh(Top_product_subcategory['Sub-Category'],Top_product_subcategory['Profit'])
plt.xlabel('Profit')
plt.ylabel('Product Subcategory')
plt.title('Profit Per Product Subcategory')
sns.barplot(
    x='Profit',
    y='Sub-Category',
    data=Top_product_subcategory,
    palette='bwr'
)
plt.show()
```



[165]:

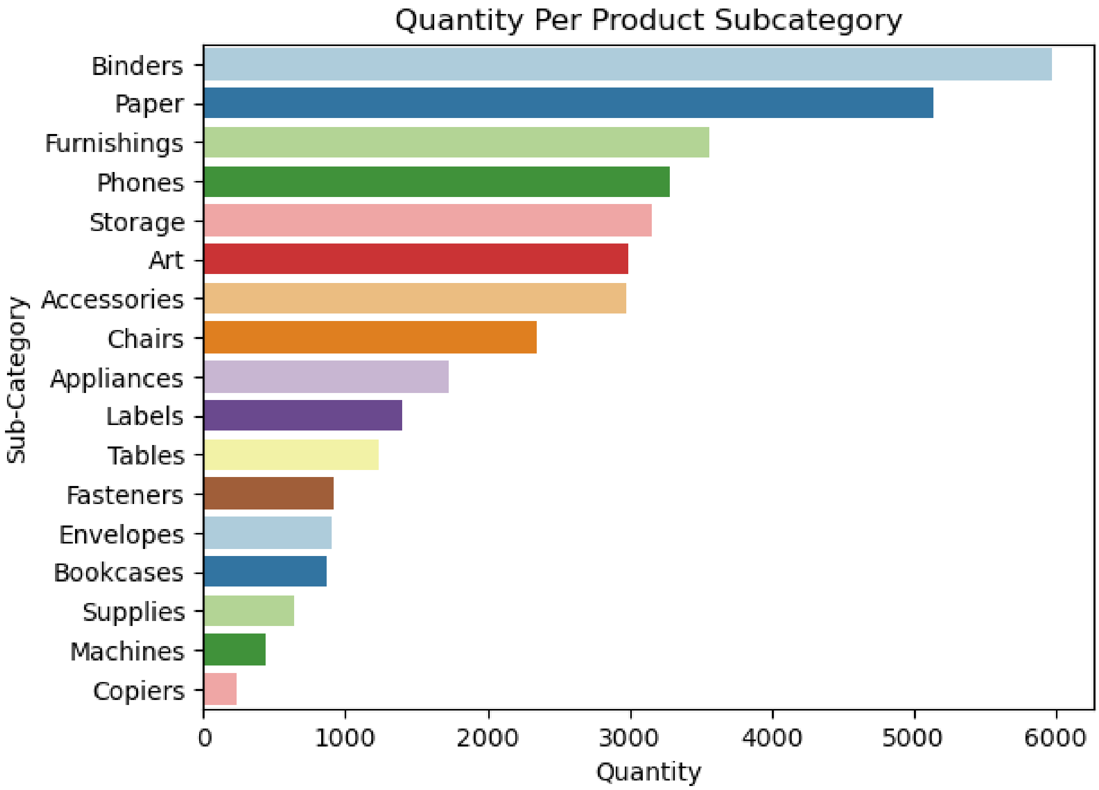
```
product_subcategory = cleaned_data.groupby(['Sub-Category'])['Quantity'].sum().
 ↪reset_index()
```

```
Top_product_subcategory = product_subcategory.sort_values(by='Quantity',␣
  ↪ascending =False)
print(Top_product_subcategory.reset_index(drop=True))
```

```
   Sub-Category  Quantity
0       Binders      5971
1         Paper      5144
2   Furnishings      3560
3        Phones      3289
4       Storage      3158
5           Art      2996
6   Accessories      2976
7        Chairs      2351
8    Appliances      1729
9        Labels      1396
10       Tables      1241
11    Fasteners       914
12    Envelopes       906
13    Bookcases       868
14     Supplies       647
15     Machines       440
16      Copiers       234
```

[166]:

```
Top_product_subcategory = product_subcategory.sort_values(by='Quantity',␣
  ↪ascending =False)
plt.
  ↪barh(Top_product_subcategory['Sub-Category'],Top_product_subcategory['Quantity'])
plt.xlabel('Quantity')
plt.ylabel('Product Subcategory')
plt.title('Quantity Per Product Subcategory')
sns.barplot(
    x='Quantity',
    y='Sub-Category',
    data=Top_product_subcategory,
    palette='Paired'
)
plt.show()
```


Quantity Per Product Subcategory
Binders
Paper
Furnishings
Phones
Storage
Art
Accessories
Chairs
Appliances
Labels
Tables
Fasteners
Envelopes
Bookcases
Supplies
Machines
Copiers
Sub-Category
0
1000
2000
3000
4000
5000
6000
Quantity

[ ]: