Problem Statement

With growing demands and cut-throat competitions in the market, a Superstore Giant is seeking your knowledge in understanding what works best for them. They would like to understand which products, regions, categories and customer segments they should target or avoid

Questions related to data

- What are the top selling products in the superstore?
- What are the top profitable products in the superstore?
- Which category of products generates the highest revenue and profit?
- Which category of products has a most sales?
- Which region generates the most sales?
- Which region generates the most profits?
- What is the impact of discounts and promotions on sales?
- What is the impact on profit after giveing the discounts ?
- What is the average profit margin for each product category?
- Which sub-category of products has the highest demand?
- Which sub-category of products has the highest profit?
- Which customer segments should target?
- Which customer segment gives most profit?

Column Description

- Row ID => Unique ID for each row.
- Order ID => Unique Order ID for each Customer.
- Order Date => Order Date of the product.
- Ship Date => Shipping Date of the Product.
- Ship Mode=> Shipping Mode specified by the Customer.
- Customer ID => Unique ID to identify each Customer.
- Customer Name => Name of the Customer.
- Segment => The segment where the Customer belongs.
- Country => Country of residence of the Customer.
- City => City of residence of of the Customer.
- State => State of residence of the Customer.
- Postal Code => Postal Code of every Customer.
- Region => Region where the Customer belong.
- Product ID => Unique ID of the Product.
- Category => Category of the product ordered.
- Sub-Category => Sub-Category of the product ordered.
- Product Name => Name of the Product
- Sales => Sales of the Product.
- Quantity => Quantity of the Product.
- Discount => Discount provided. *Profit => Profit/Loss incurred.

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import datetime as dt
import plotly.express as px
```

In [2]:
```
df2 = pd.read_csv("Sample - Superstore.csv", encoding = 'unicode_escape')
```

In [3]:
```
df2
```

Out[3]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Ca
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Fu
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Fu
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	S
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Fu
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	S
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	33180	South	FUR-FU-10001889	Fu
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	FUR-FU-10000747	Fu
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	TEC-PH-10003645	Tech
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	OFF-PA-10004041	S
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	92683	West	OFF-AP-10002684	S

9994 rows × 21 columns

In [4]:
```
df=df2.drop(['Row ID'], axis=1)
df
```

Out[4]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Ca
0	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-BO-10001798	F
1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-CH-10000454	F
2	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West	OFF-LA-10000240	S
3	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	F
4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	S
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9989	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	Florida	33180	South	FUR-FU-10001889	F
9990	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California	92627	West	FUR-FU-10000747	F
9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California	92627	West	TEC-PH-10003645	Tecl
9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California	92627	West	OFF-PA-10004041	S
9993	CA-2017-119914	5/4/2017	5/9/2017	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	California	92683	West	OFF-AP-10002684	S

9994 rows × 20 columns

In [5]:
```
df.shape
```

Out[5]:
```
(9994, 20)
```

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

Out[6]:
```
Index(['Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer ID',
       'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name',
       'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 20 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Order ID       9994 non-null   object 
 1   Order Date     9994 non-null   object 
 2   Ship Date      9994 non-null   object 
 3   Ship Mode      9994 non-null   object 
 4   Customer ID    9994 non-null   object 
 5   Customer Name  9994 non-null   object 
 6   Segment        9994 non-null   object 
 7   Country        9994 non-null   object 
 8   City           9994 non-null   object 
 9   State          9994 non-null   object 
 10  Postal Code    9994 non-null   int64  
 11  Region         9994 non-null   object 
 12  Product ID     9994 non-null   object 
 13  Category       9994 non-null   object 
 14  Sub-Category   9994 non-null   object 
 15  Product Name   9994 non-null   object 
 16  Sales          9994 non-null   float64
 17  Quantity       9994 non-null   int64  
 18  Discount       9994 non-null   float64
 19  Profit         9994 non-null   float64
dtypes: float64(3), int64(2), object(15)
memory usage: 1.5+ MB
```

In [8]:
```
df.isna().sum()
```

Out[8]:
```
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

In [9]:
```
df.duplicated().sum()
```

Out[9]:
```
1
```

In [10]:
```
df.drop_duplicates(keep=False, inplace=True)
```

In [11]:
```
df.describe()
```

Out[11]:

	Postal Code	Sales	Quantity	Discount	Profit
count	9992.000000	9992.000000	9992.000000	9992.000000	9992.000000
mean	55192.773619	229.847690	3.789932	0.156174	28.665046
std	32066.455778	623.307052	2.225188	0.206463	234.282845
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.272000	2.000000	0.000000	1.732125
50%	56560.000000	54.432000	3.000000	0.200000	8.671250
75%	90008.000000	209.932500	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [12]:
```
# Changeing data type of date column

df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])
```

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9992 entries, 0 to 9993
Data columns (total 20 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order ID       9992 non-null   object
 1   Order Date     9992 non-null   datetime64[ns]
 2   Ship Date      9992 non-null   datetime64[ns]
 3   Ship Mode      9992 non-null   object
 4   Customer ID    9992 non-null   object
 5   Customer Name  9992 non-null   object
 6   Segment        9992 non-null   object
 7   Country        9992 non-null   object
 8   City           9992 non-null   object
 9   State          9992 non-null   object
 10  Postal Code    9992 non-null   int64
 11  Region         9992 non-null   object
 12  Product ID     9992 non-null   object
 13  Category       9992 non-null   object
 14  Sub-Category   9992 non-null   object
 15  Product Name   9992 non-null   object
 16  Sales          9992 non-null   float64
 17  Quantity       9992 non-null   int64
 18  Discount       9992 non-null   float64
 19  Profit         9992 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(2), object(13)
memory usage: 1.6+ MB
```

In [14]:
```
df.head()
```

Out[14]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category
0	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-BO-10001798	Furniture	Bookcases
1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-CH-10000454	Furniture	Chairs
2	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West	OFF-LA-10000240	Office Supplies	Labels
3	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	Furniture	Tables
4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	Office Supplies	Storage

In [15]:
```
# Separate Date column into Year ,  month and day

df['Year'] = df['Order Date'].dt.year
df['Month'] = df['Order Date'].dt.month_name()
df['Day'] = df['Order Date'].dt.day_name()
```

In [16]:
```
# Drop date column because we separate Date colum into year,month,day
df.drop(['Order Date'], axis=1,inplace=True)
```

PERFORMING EXPLORATOTY DATA ANALYSIS (EDA)

Q1 = What are the top selling products in the superstore?

In [17]:
```
df['Product Name'].value_counts()
```

Out[17]:
```
Product Name
Staple envelope                                                                   48
Staples                                                                           46
Easy-staple paper                                                                 46
Avery Non-Stick Binders                                                           20
Staples in misc. colors                                                           19
                                                                                  ..
Boston 1900 Electric Pencil Sharpener                                              1
RCA ViSYS 25423RE1 Corded phone                                                    1
Canon Color ImageCLASS MF8580Cdw Wireless Laser All-In-One Printer, Copier, Scanner    1
Newell 342                                                                         1
Eldon Jumbo ProFile Portable File Boxes Graphite/Black                             1
Name: count, Length: 1850, dtype: int64
```

In [18]:
```
df['Sales'].value_counts()
```

Out[18]:
```
Sales
12.960     56
19.440     39
15.552     39
25.920     36
10.368     36
           ..
4.240       1
319.960     1
646.740     1
81.940      1
243.160     1
Name: count, Length: 5824, dtype: int64
```

In [19]:
```
Top_selling_product = df.groupby(['Product Name'])['Sales'].sum().sort_values(ascending=False).reset_index()
Top_selling_product
```

Out[19]:

	Product Name	Sales
0	Canon imageCLASS 2200 Advanced Copier	61599.824
1	Fellowes PB500 Electric Punch Plastic Comb Bin...	27453.384
2	Cisco TelePresence System EX90 Videoconferenci...	22638.480
3	HON 5400 Series Task Chairs for Big and Tall	21870.576
4	GBC DocuBind TL300 Electric Binding System	19823.479
...	...	...
1845	Avery Hi-Liter Pen Style Six-Color Fluorescent...	7.700
1846	Grip Seal Envelopes	7.072
1847	Xerox 20	6.480
1848	Avery 5	5.760
1849	Eureka Disposable Bags for Sanitaire Vibra Gro...	1.624

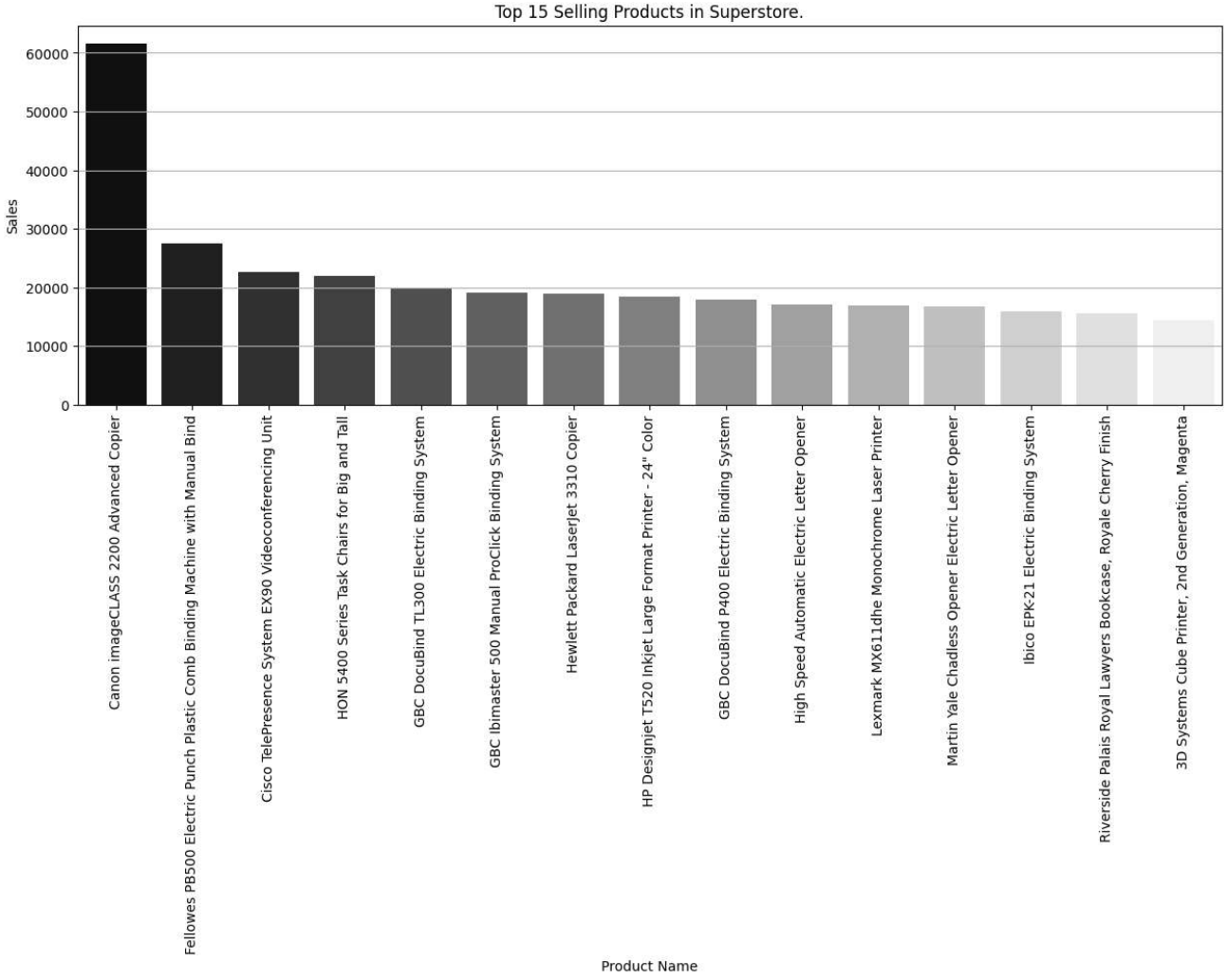
1850 rows × 2 columns

In [20]:
```
top_15_selling_products = pd.DataFrame(Top_selling_product.head(15))
top_15_selling_products
```

Out[20]:

	Product Name	Sales
0	Canon imageCLASS 2200 Advanced Copier	61599.8240
1	Fellowes PB500 Electric Punch Plastic Comb Bin...	27453.3840
2	Cisco TelePresence System EX90 Videoconferenci...	22638.4800
3	HON 5400 Series Task Chairs for Big and Tall	21870.5760
4	GBC DocuBind TL300 Electric Binding System	19823.4790
5	GBC Ibimaster 500 Manual ProClick Binding System	19024.5000
6	Hewlett Packard LaserJet 3310 Copier	18839.6860
7	HP Designjet T520 Inkjet Large Format Printer ...	18374.8950
8	GBC DocuBind P400 Electric Binding System	17965.0680
9	High Speed Automatic Electric Letter Opener	17030.3120
10	Lexmark MX611dhe Monochrome Laser Printer	16829.9010
11	Martin Yale Chadless Opener Electric Letter Op...	16656.2000
12	Ibico EPK-21 Electric Binding System	15875.9160
13	Riverside Palais Royal Lawyers Bookcase, Royal...	15610.9656
14	3D Systems Cube Printer, 2nd Generation, Magenta	14299.8900

In [21]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Product Name',y='Sales',data=top_15_selling_products,palette='gist_yarg_r')
plt.title('Top 15 Selling Products in Superstore.')
plt.xlabel('Product Name',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 90)
plt.show()
```



Q2 = What are the top profitable products in the superstore?

In [22]:
```
Top_Profitable_product = df.groupby(['Product Name'])['Profit'].sum().sort_values(ascending=False).reset_index()
Top_Profitable_product
```

Out[22]:

	Product Name	Profit
0	Canon imageCLASS 2200 Advanced Copier	25199.9280
1	Fellowes PB500 Electric Punch Plastic Comb Bin...	7753.0390
2	Hewlett Packard LaserJet 3310 Copier	6983.8836
3	Canon PC1060 Personal Laser Copier	4570.9347
4	HP Designjet T520 Inkjet Large Format Printer ...	4094.9766
...	...	...
1845	Bush Advantage Collection Racetrack Conference...	-1934.3976
1846	Chromcraft Bull-Nose Wood Oval Conference Tabl...	-2876.1156
1847	Cubify CubeX 3D Printer Triple Head Print	-3839.9904
1848	Lexmark MX611dhe Monochrome Laser Printer	-4589.9730
1849	Cubify CubeX 3D Printer Double Head Print	-8879.9704

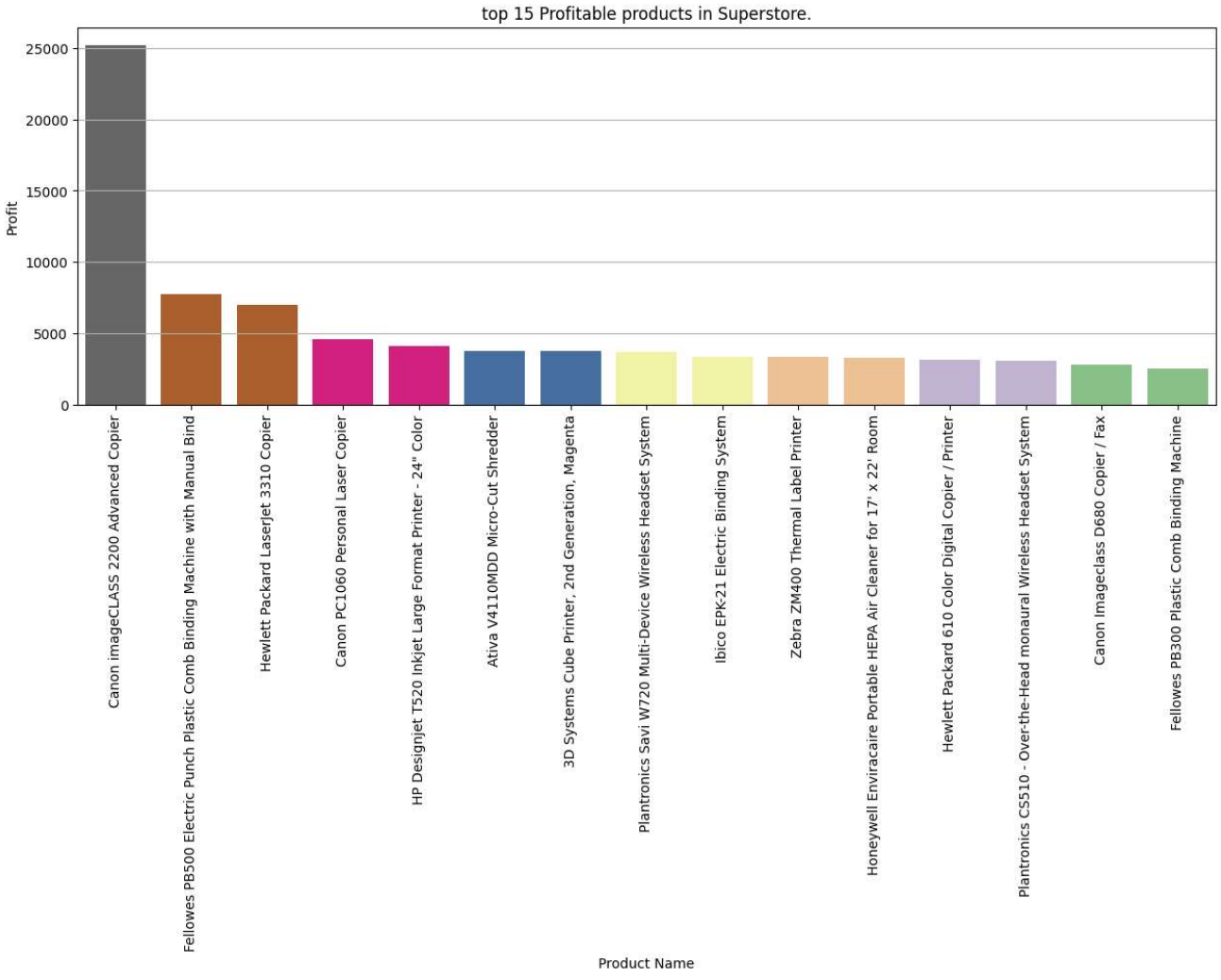
1850 rows × 2 columns

In [23]:
```
top_15_Profitable_products = pd.DataFrame(Top_Profitable_product.head(15))
top_15_Profitable_products
```

Out[23]:

	Product Name	Profit
0	Canon imageCLASS 2200 Advanced Copier	25199.9280
1	Fellowes PB500 Electric Punch Plastic Comb Bin...	7753.0390
2	Hewlett Packard LaserJet 3310 Copier	6983.8836
3	Canon PC1060 Personal Laser Copier	4570.9347
4	HP Designjet T520 Inkjet Large Format Printer ...	4094.9766
5	Ativa V4110MDD Micro-Cut Shredder	3772.9461
6	3D Systems Cube Printer, 2nd Generation, Magenta	3717.9714
7	Plantronics Savi W720 Multi-Device Wireless He...	3696.2820
8	Ibico EPK-21 Electric Binding System	3345.2823
9	Zebra ZM400 Thermal Label Printer	3343.5360
10	Honeywell Enviracaire Portable HEPA Air Cleane...	3247.0200
11	Hewlett Packard 610 Color Digital Copier / Pri...	3124.9375
12	Plantronics CS510 - Over-the-Head monaural Wir...	3085.0325
13	Canon Imageclass D680 Copier / Fax	2799.9600
14	Fellowes PB300 Plastic Comb Binding Machine	2518.0551

In [24]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Product Name',y='Profit',data=top_15_Profitable_products,palette='Accent_r')
plt.title('top 15 Profitable products in Superstore.')
plt.xlabel('Product Name',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 90)
plt.show()
```



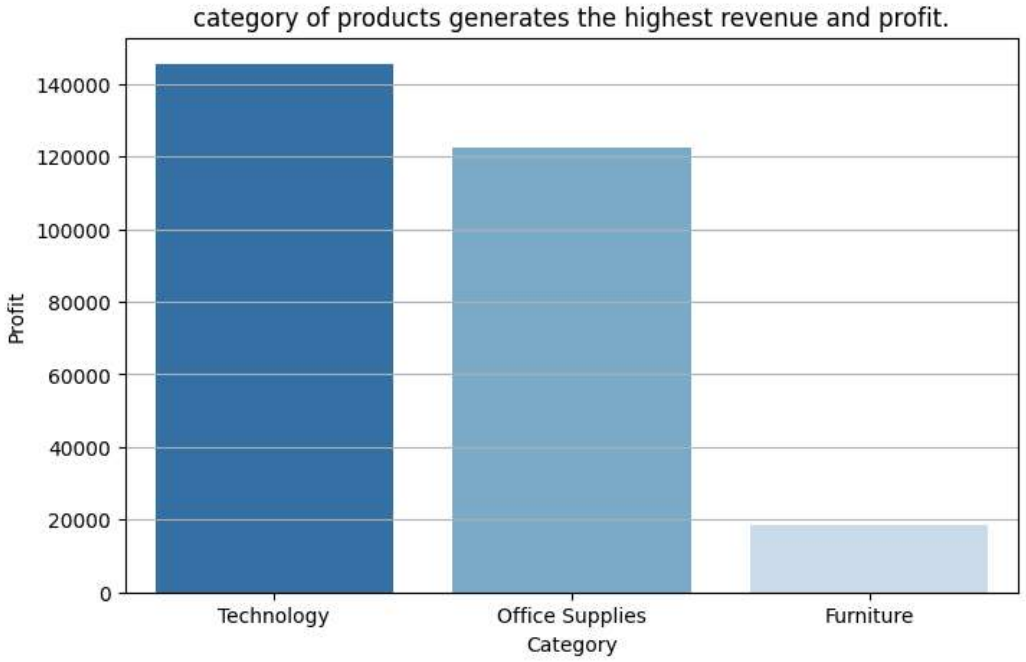
Q3 = Which category of products generates the highest revenue and profit?

In [25]:
```
Top_Profitable_Category = df.groupby(['Category'])['Profit'].sum().sort_values(ascending=False).reset_index()
Top_Profitable_Category
```

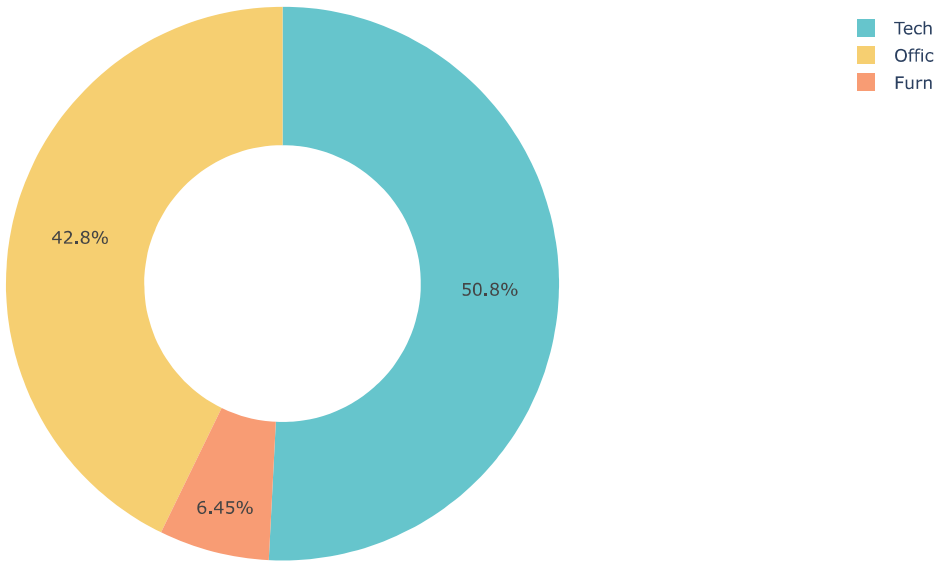
Out[25]:

	Category	Profit
0	Technology	145454.9481
1	Office Supplies	122490.8008
2	Furniture	18475.3904

In [26]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Category',y='Profit',data=Top_Profitable_Category,palette='Blues_r')
plt.title('category of products generates the highest revenue and profit.')
plt.xlabel('Category',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.show()
```



In [27]:
```
px.pie(Top_Profitable_Category,
             values='Profit',
             names='Category',
             hole=0.5,
             color_discrete_sequence=px.colors.qualitative.Pastel)
```



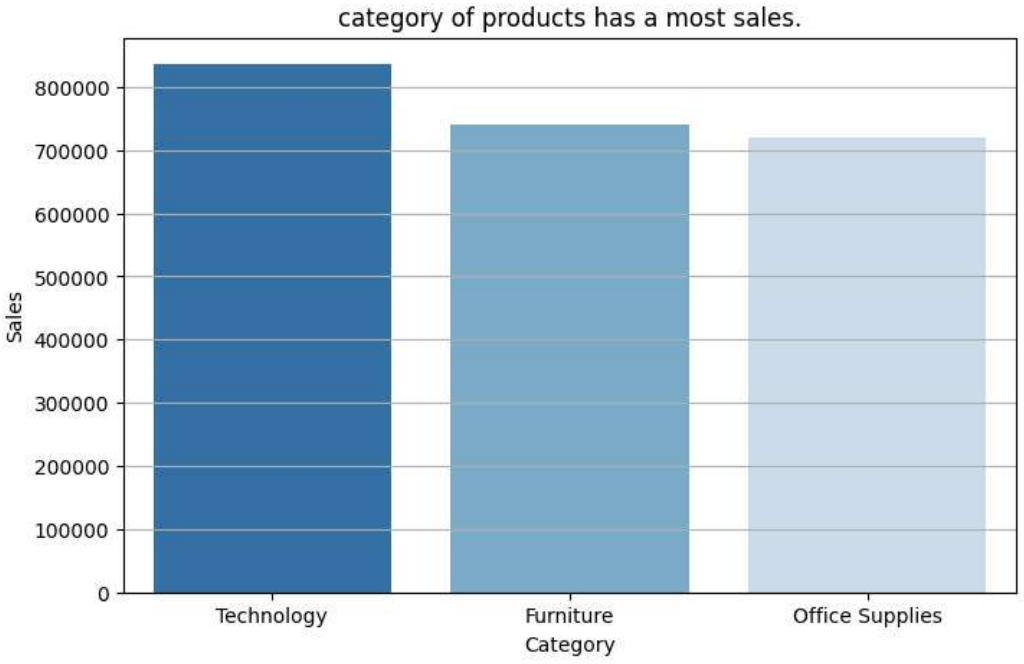
Q4 = Which category of products has a most sales?

In [28]:
```
Top_selling_Category = df.groupby(['Category'])['Sales'].sum().sort_values(ascending=False).reset_index()
Top_selling_Category
```

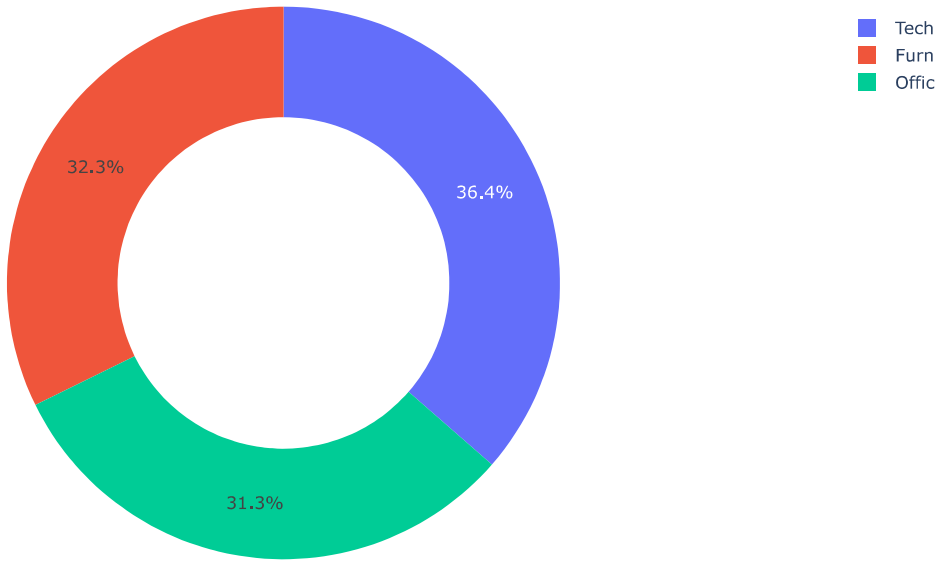
Out[28]:

	Category	Sales
0	Technology	836154.0330
1	Furniture	741437.0513
2	Office Supplies	719047.0320

In [29]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Category',y='Sales',data=Top_selling_Category,palette='Blues_r')
plt.title('category of products has a most sales.')
plt.xlabel('Category',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.show()
```



In [30]:
```
px.pie(Top_selling_Category,values='Sales',names='Category',hole=0.6)
```



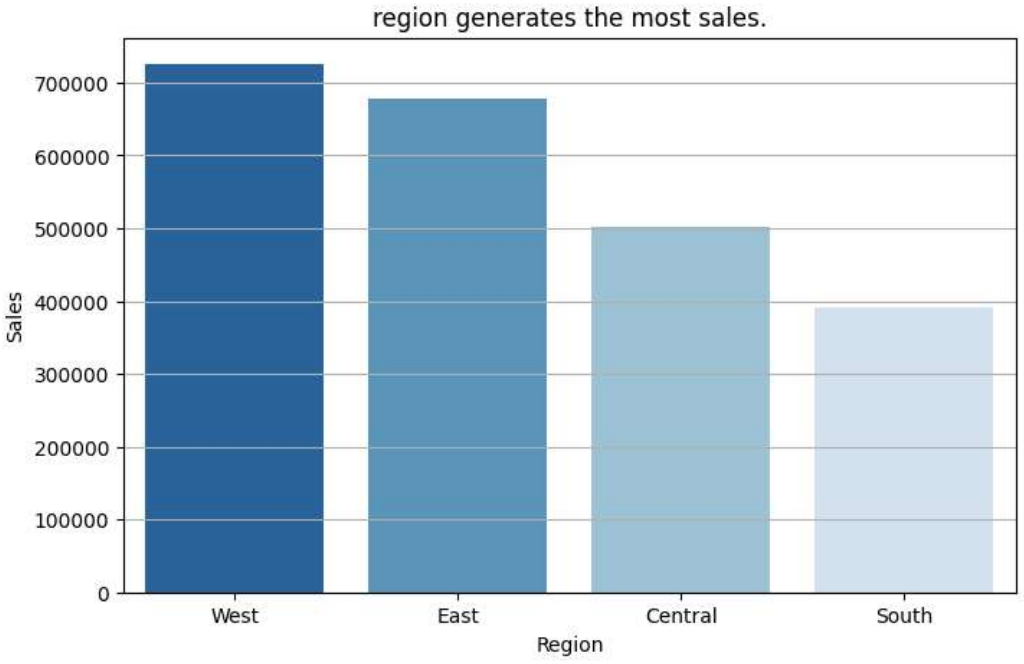
Q5 = Which region generates the most sales?

In [31]:
```
Top_region_sale = df.groupby(['Region'])['Sales'].sum().sort_values(ascending=False).reset_index()
Top_region_sale
```

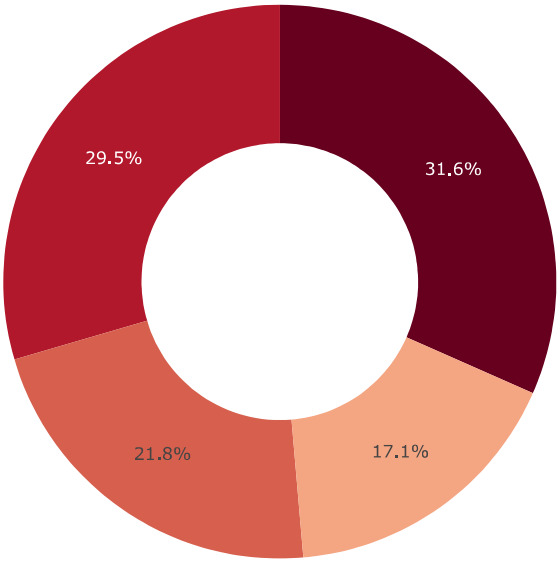
Out[31]:

	Region	Sales
0	West	725457.8245
1	East	678218.4960
2	Central	501239.8908
3	South	391721.9050

In [32]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Region',y='Sales',data=Top_region_sale,palette='Blues_r')
plt.title('region generates the most sales.')
plt.xlabel('Region',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.show()
```



In [33]:
```
px.pie(Top_region_sale,values='Sales',names='Region',hole=0.5,color_discrete_sequence=px.colors.sequential.RdBu)
```



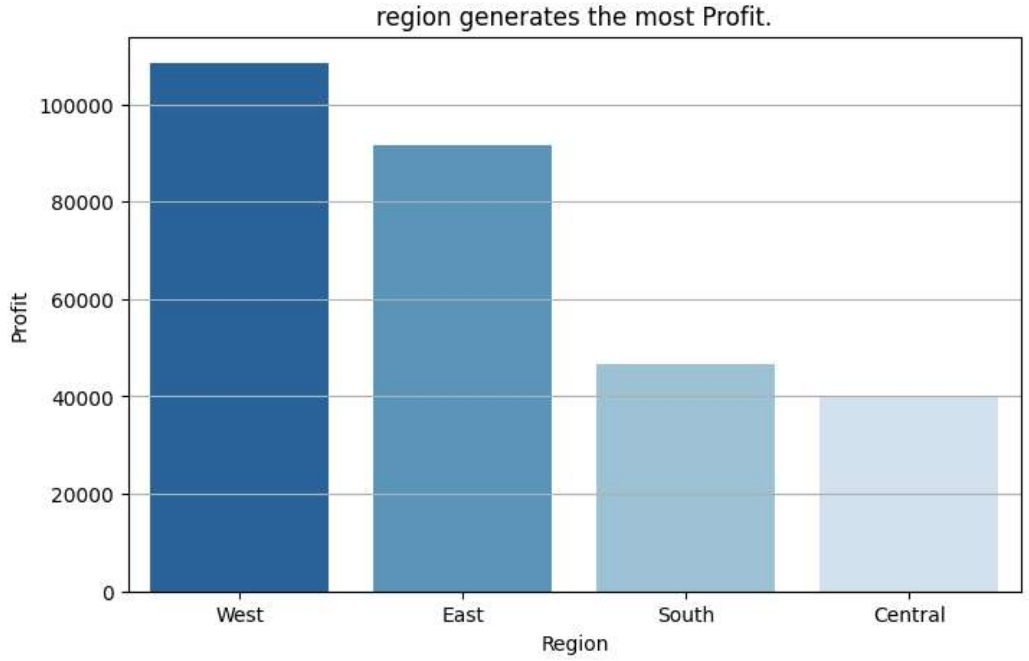
Q6 = Which region generates the most profits?

In [34]:
```
Top_region_Profit = df.groupby(['Region'])['Profit'].sum().sort_values(ascending=False).reset_index()
Top_region_Profit
```

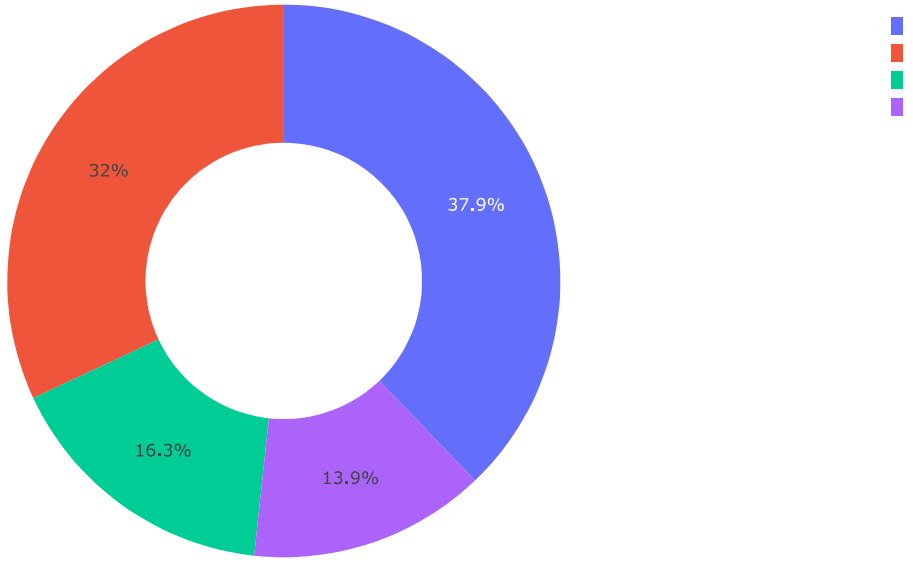
Out[34]:

	Region	Profit
0	West	108418.4489
1	East	91546.8976
2	South	46749.4303
3	Central	39706.3625

In [35]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Region',y='Profit',data=Top_region_Profit,palette='Blues_r')
plt.title('region generates the most Profit.')
plt.xlabel('Region',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.show()
```



In [36]:
```
px.pie(Top_region_Profit,values='Profit',names='Region',hole=0.5)
```



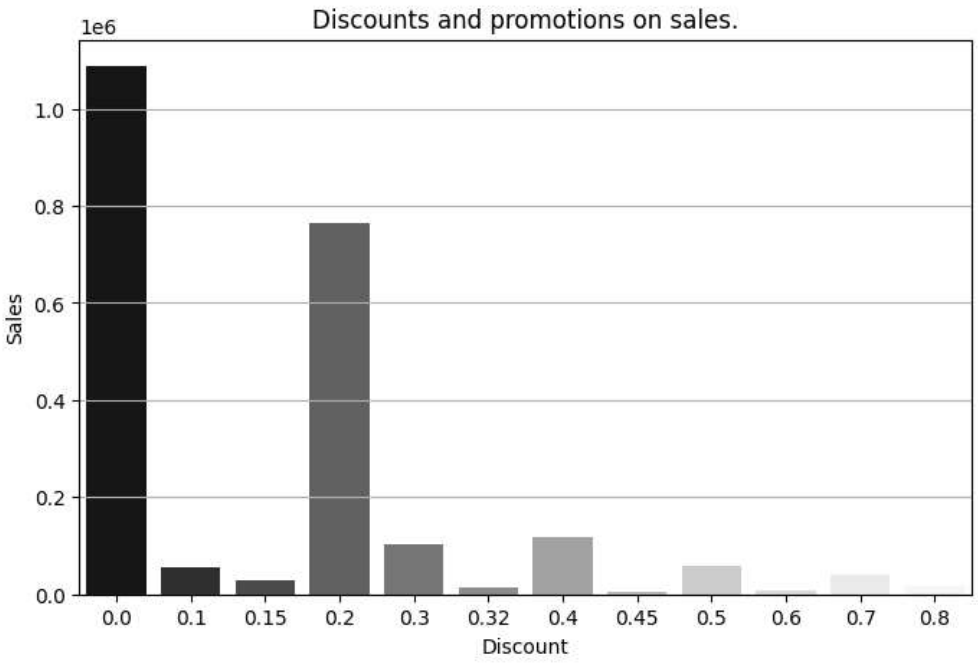
Q7 = What is the impact of discounts and promotions on sales?

In [37]:
```
Discounts_On_Sales = df.groupby(['Discount'])['Sales'].sum().sort_values(ascending=False).reset_index()
Discounts_On_Sales
```

Out[37]:

	Discount	Sales
0	0.00	1.087908e+06
1	0.20	7.645944e+05
2	0.40	1.164178e+05
3	0.30	1.026639e+05
4	0.50	5.891854e+04
5	0.10	5.436935e+04
6	0.70	4.062028e+04
7	0.15	2.755852e+04
8	0.80	1.696376e+04
9	0.32	1.449346e+04
10	0.60	6.644700e+03
11	0.45	5.484974e+03

In [38]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Discount',y='Sales',data=Discounts_On_Sales,palette='Greys_r')
plt.title('Discounts and promotions on sales.')
plt.xlabel('Discount',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.show()
```



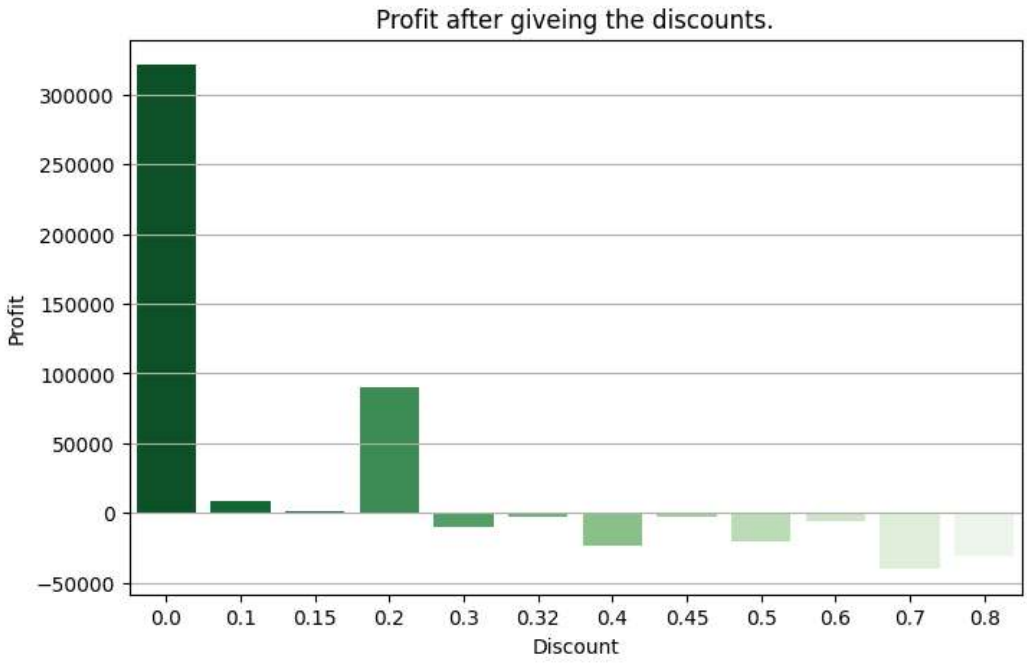
Q8 = What is the impact on profit after giveing the discounts ?

In [39]:
```
Discounts_On_Sales = df.groupby(['Discount'])['Profit'].sum().sort_values(ascending=False).reset_index()
Discounts_On_Sales
```

Out[39]:

	Discount	Profit
0	0.00	320987.6032
1	0.20	90337.3060
2	0.10	9029.1770
3	0.15	1418.9915
4	0.32	-2391.1377
5	0.45	-2493.1111
6	0.60	-5944.6552
7	0.30	-10345.1598
8	0.50	-20506.4281
9	0.40	-23057.0504
10	0.80	-30539.0392
11	0.70	-40075.3569

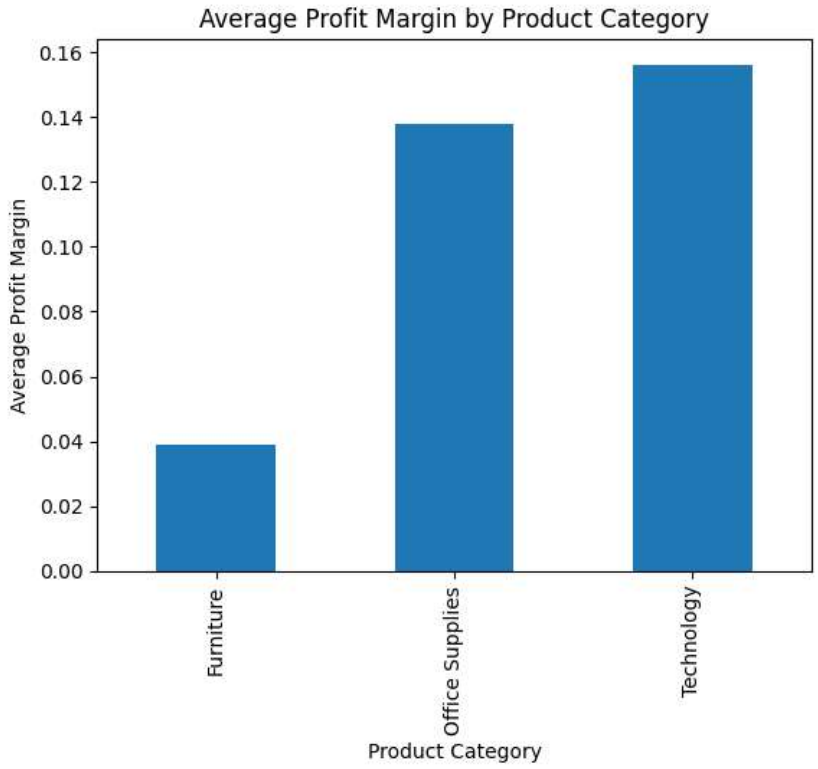
In [40]:
```
plt.figure(figsize=(8,5))
sns.barplot(x='Discount',y='Profit',data=Discounts_On_Sales,palette='Greens_r')
plt.title('Profit after giveing the discounts.')
plt.xlabel('Discount',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.show()
```



Q9 = What is the average profit margin for each product category?

In [41]:
```
df['Profit Margin'] = df['Profit'] / df['Sales']
avg_profit_margin_by_category = df.groupby('Category')['Profit Margin'].mean()
avg_profit_margin_by_category.plot(kind='bar')
plt.title("Average Profit Margin by Product Category")
plt.xlabel("Product Category")
plt.ylabel("Average Profit Margin")

plt.show()
```



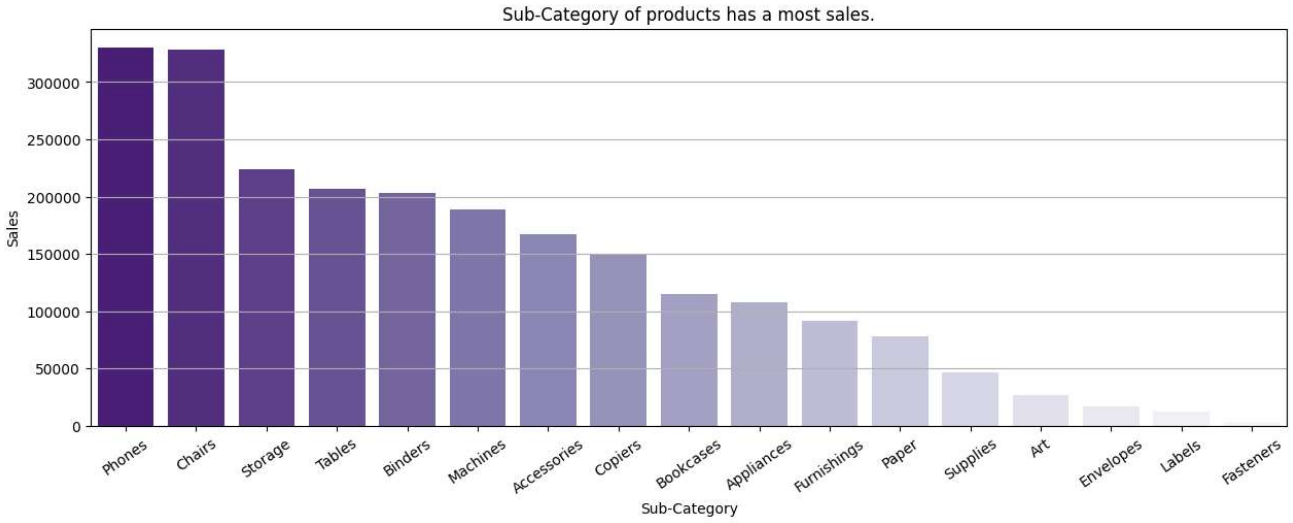
Q10 = Which sub-category of products has the highest demand?

In [42]:
```
sub_category_in_demand_sales = df.groupby(['Sub-Category'])['Sales'].sum().sort_values(ascending=False).reset_index()
sub_category_in_demand_sales
```

Out[42]:

	Sub-Category	Sales
0	Phones	330007.0540
1	Chairs	327886.3590
2	Storage	223843.6080
3	Tables	206965.5320
4	Binders	203412.7330
5	Machines	189238.6310
6	Accessories	167380.3180
7	Copiers	149528.0300
8	Bookcases	114879.9963
9	Appliances	107532.1610
10	Furnishings	91705.1640
11	Paper	78479.2060
12	Supplies	46673.5380
13	Art	27118.7920
14	Envelopes	16476.4020
15	Labels	12486.3120
16	Fasteners	3024.2800

In [43]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Sub-Category',y='Sales',data=sub_category_in_demand_sales,palette='Purples_r')
plt.title('Sub-Category of products has a most sales.')
plt.xlabel('Sub-Category',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 35)
plt.show()
```



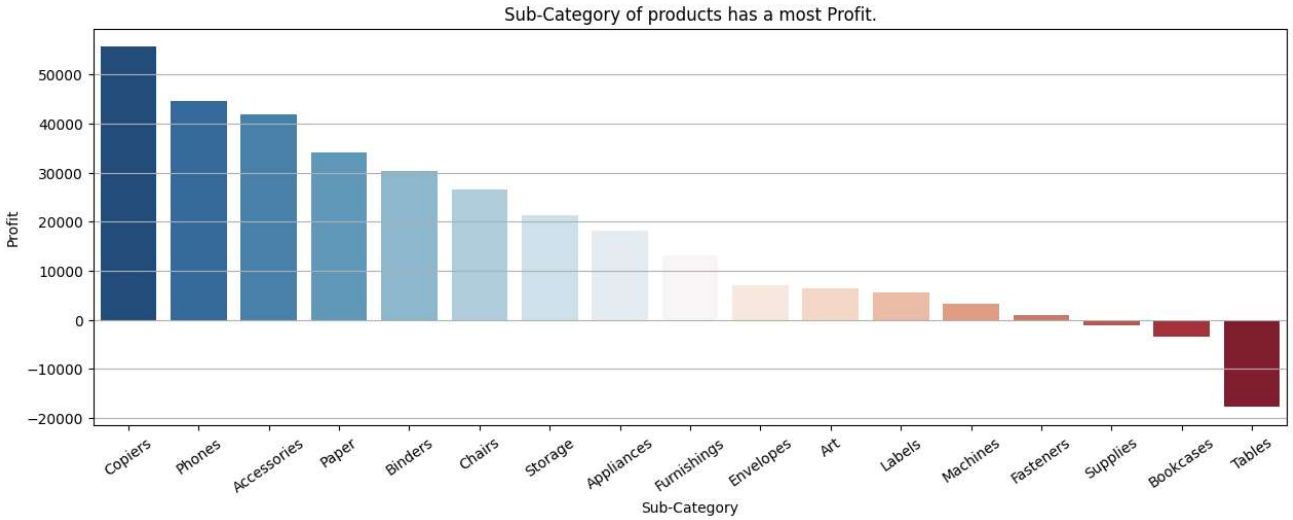
Q11 = Which sub-category of products has the highest profit?

In [44]:
```
sub_category_in_demand_profit = df.groupby(['Sub-Category'])['Profit'].sum().sort_values(ascending=False).reset_index(
sub_category_in_demand_profit
```

Out[44]:

	Sub-Category	Profit
0	Copiers	55617.8249
1	Phones	44515.7306
2	Accessories	41936.6357
3	Paper	34053.5693
4	Binders	30221.7633
5	Chairs	26614.2839
6	Storage	21278.8264
7	Appliances	18138.0054
8	Furnishings	13059.1436
9	Envelopes	6964.1767
10	Art	6527.7870
11	Labels	5546.2540
12	Machines	3384.7569
13	Fasteners	949.5182
14	Supplies	-1189.0995
15	Bookcases	-3472.5560
16	Tables	-17725.4811

In [45]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Sub-Category',y='Profit',data=sub_category_in_demand_profit,palette='RdBu_r')
plt.title('Sub-Category of products has a most Profit.')
plt.xlabel('Sub-Category',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 35)
plt.show()
```



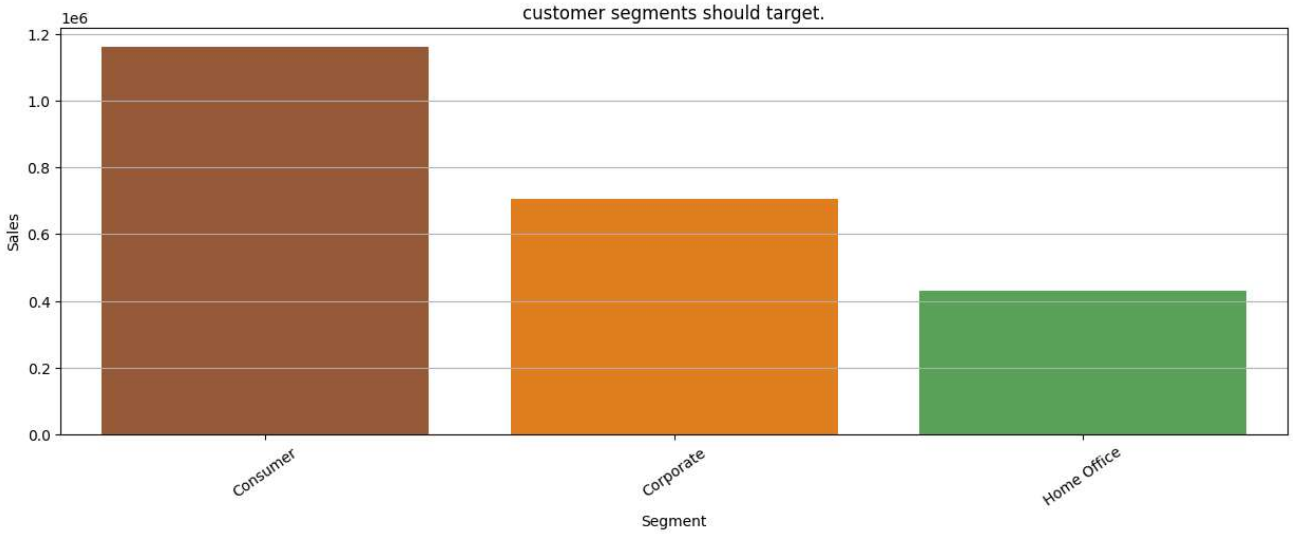
Q12 = Which customer segments should target?

In [46]:
```
customer_segments_target = df.groupby(['Segment'])['Sales'].sum().sort_values(ascending=False).reset_index()
customer_segments_target
```

Out[46]:

	Segment	Sales
0	Consumer	1.161401e+06
1	Corporate	7.061464e+05
2	Home Office	4.290904e+05

In [47]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Segment',y='Sales',data=customer_segments_target,palette='Set1_r')
plt.title('customer segments should target.')
plt.xlabel('Segment',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 35)
plt.show()
```



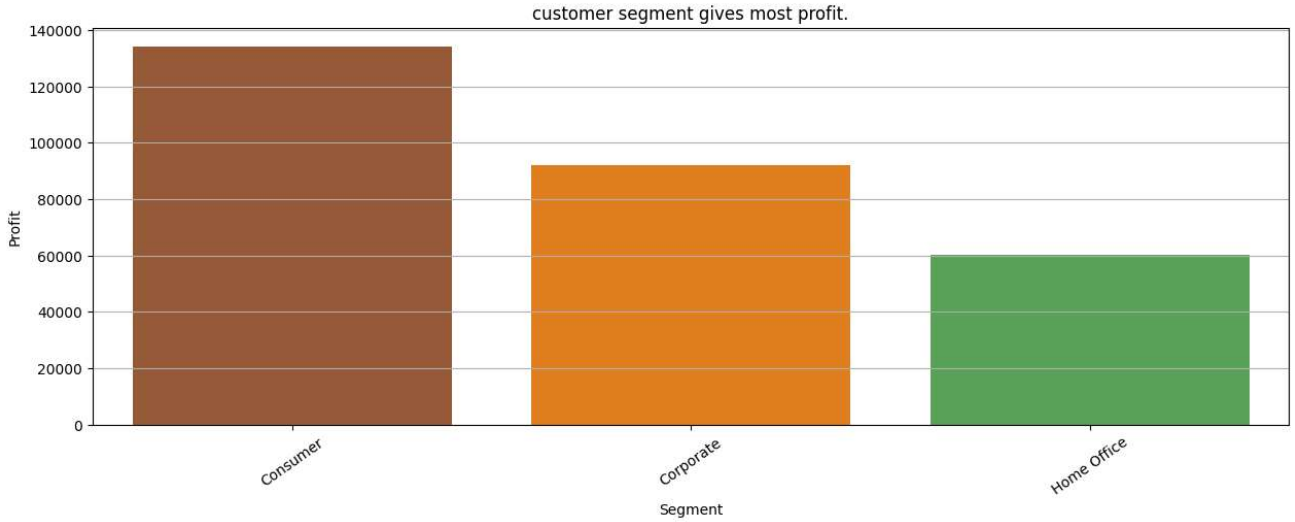
Q13 = Which customer segment gives most profit?

In [48]:
```
customer_segments_profit = df.groupby(['Segment'])['Profit'].sum().sort_values(ascending=False).reset_index()
customer_segments_profit
```

Out[48]:

	Segment	Profit
0	Consumer	134119.2092
1	Corporate	91979.1340
2	Home Office	60322.7961

In [49]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Segment',y='Profit',data=customer_segments_profit,palette='Set1_r')
plt.title('customer segment gives most profit.')
plt.xlabel('Segment',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 35)
plt.show()
```

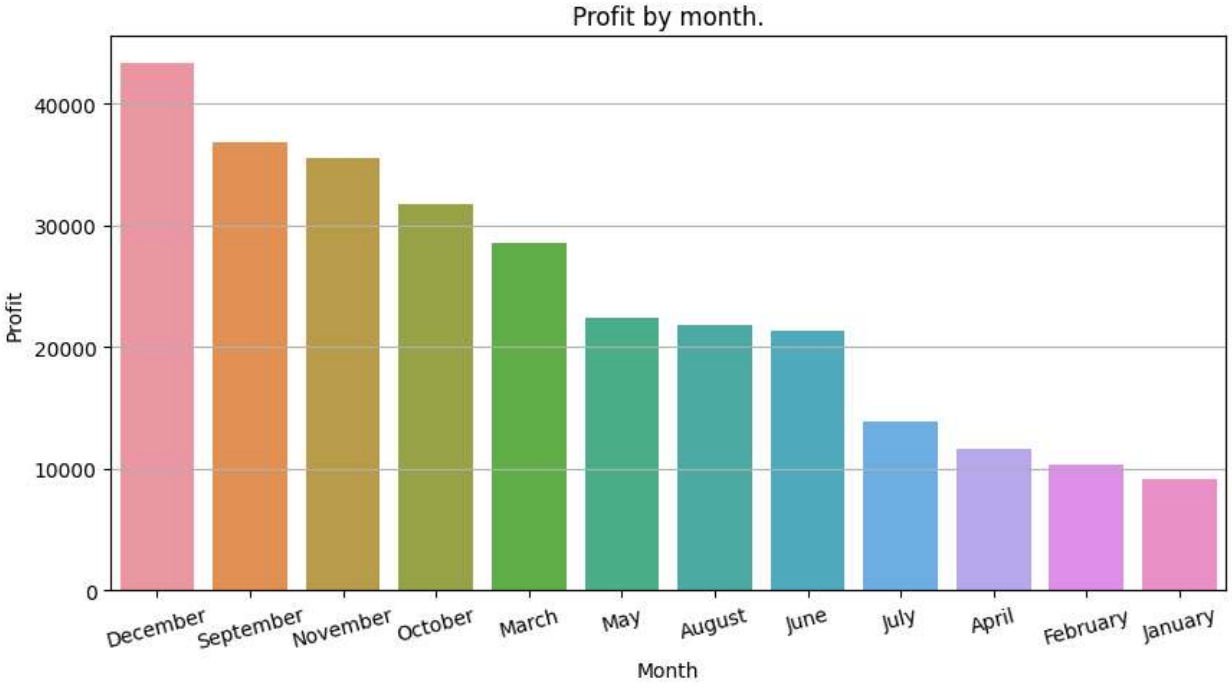


In [50]:
```
Profit_by_month = df.groupby(['Month'])['Profit'].sum().sort_values(ascending=False).reset_index()
Profit_by_month
```

Out[50]:

	Month	Profit
0	December	43369.1919
1	September	36857.4753
2	November	35468.4265
3	October	31784.0413
4	March	28594.6872
5	May	22411.3078
6	August	21776.9384
7	June	21285.7954
8	July	13832.6648
9	April	11611.5539
10	February	10294.6107
11	January	9134.4461

In [51]:
```
plt.figure(figsize=(10,5))
sns.barplot(x='Month',y='Profit',data=Profit_by_month)
plt.title('Profit by month.')
plt.xlabel('Month',fontsize=10)
plt.ylabel('Profit',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 15)
plt.show()
```

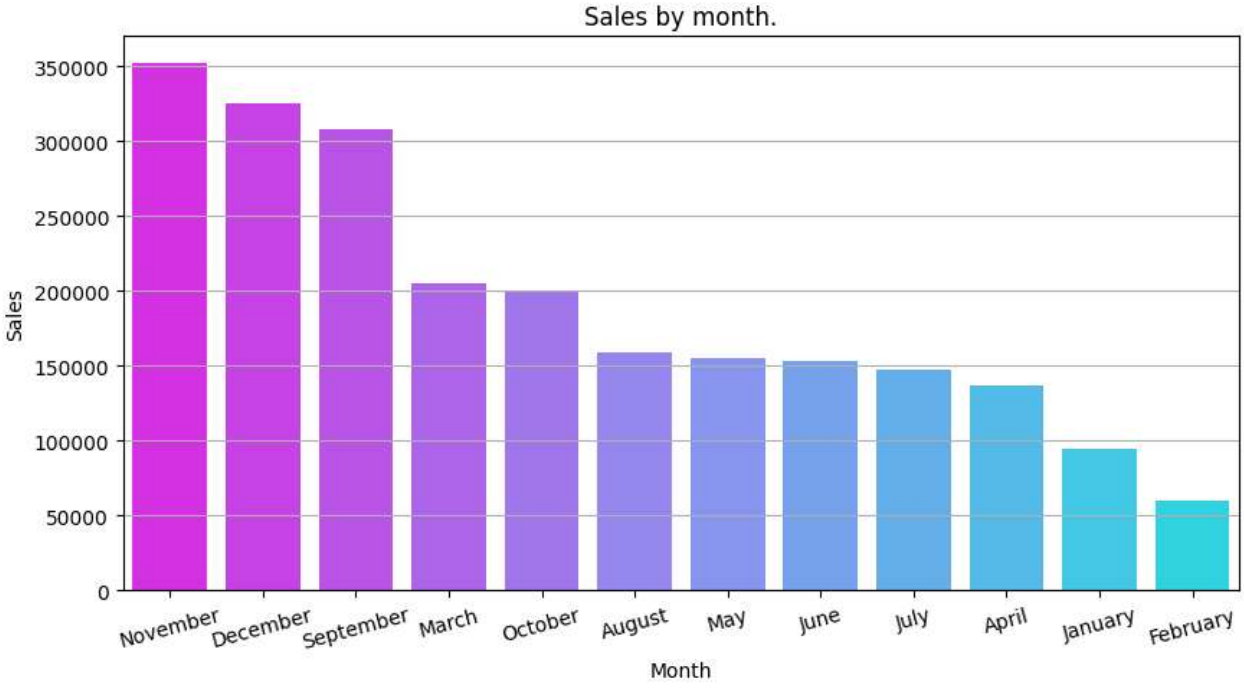


In [52]:
```
Sale_by_month = df.groupby(['Month'])['Sales'].sum().sort_values(ascending=False).reset_index()
Sale_by_month
```

Out[52]:

	Month	Sales
0	November	352461.0710
1	December	325293.5035
2	September	307649.9457
3	March	205005.4888
4	October	200322.9847
5	August	159044.0630
6	May	155028.8117
7	June	152718.6793
8	July	147238.0970
9	April	137199.3846
10	January	94924.8356
11	February	59751.2514

In [53]:
```
plt.figure(figsize=(10,5))
sns.barplot(x='Month',y='Sales',data=Sale_by_month,palette='cool_r')
plt.title('Sales by month.')
plt.xlabel('Month',fontsize=10)
plt.ylabel('Sales',fontsize=10)
plt.grid(axis='y')
plt.xticks(rotation = 15)
plt.show()
```



Observation for Problem Statement

Most selling products in SuperStore is

- Canon imageCLASS 2200 Advanced Copier-
- Fellowes P8500 Electric Punch Plastic Comb Binding Machine with Manual Bind
- Cisco TelePresence System EX90 Videoconferencing Unit
- HON 5400 Series Task Chairs for Big and Tall
- GBC DocuBind TL300 Electric Binding System
- GBC Ibimaster 500 Manual ProClick Binding System
- Hewlett Packard Laserjet 3310 Copier
- HP Designjet T520 Inkjet Large Format Printer - 24" Color
- GBC DocuBind P400 Electric Binding System
- High Speed Automatic Electric Letter Opener
- Lexmark MX611dhe Monochrome Laser Printer
- Martin Yale Chadless Opener Electric Letter Opener
- ibico EPK-21 Electric Binding System
- Riverside Palais Royal Lawyers Bookcase, Royale Cherry Finish
- 3D Systems Cube Printer, 2nd Generation Magenta

Most Profitable products in SuperStore is

- Canon imageCLASS 2200 Advanced Copier
- Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind
- Hewlett Packard Laserjet 3310 Copier
- Canon PC1060 Personal Laser Copier
- HP Designjet T520 Inkjet Large Format Printer - 24" Color
- Ativa V4110MDD Micro-Cut Shredder
- 3D Systems Cube Printer, 2nd Generation, Magenta
- Plantronics Savi W720 Multi-Device Wireless Headset System
- Ibico EPK-21 Electric Binding System
- Zebra ZM400 Thermal Label Printer
- Honeywell Enviracaire Portable HEPA Air Cleaner for 17' x 22' Room

- Hewlett Packard 610 Color Digital Copier/Printer
- Plantronics CS510-Over-the-Head monaural Wireless Headset System
- Canon Imageclass D680 Copier/ Fax
- Fellowes PB300 Plastic Comb Binding Machine

SUPERSTORE SHOULD TARGET THIS 15 PRODUCT FOR GROWTH

- West and East are the two regions where super stores have maximum sales.
- West and East are the two regions where super stores make the most profit.

SUPERSTORE SHOULD TARGET THIS TWO REGIONS FOR GROWTH

- Technologies and office supplies are the most profitable category as compared to furniture.
- Technology and furniture have the highest sales as compared to office supplies.

/ SUPERSTORE SHOULD TARGET Technologies and office supplies FOR GENERATE PROFIT..... / SUPERSTORE SHOULD TARGET Technology and furniture FOR INCREASE SALES.

- Consumer and Corporate this to customer segment superstore need to target

SUPERSTORE SHOULD TARGET CONSUMER AND CORPORATE CUSTOMER SEGMENT FOR GROWTH

Solution of Problem Statement

- SUPERSTORE SHOULD TARGET THIS 15 PRODUCT FOR GROWTH.
- SUPERSTORE SHOULD TARGET West and East THIS TWO REGIONS FOR GROWTH.
- SUPERSTORE SHOULD TARGET Technologies and office supplies FOR GENERATE PROFIT.
- SUPERSTORE SHOULD TARGET Technology and furniture FOR INCREASE SALES.
- SUPERSTORE SHOULD TARGET CONSUMER AND CORPORATE CUSTOMER SEGMENT FOR GROWTH

In [ ]: