In [3]:
```
import pandas as pd
import numpy as np
```

In [5]:
```
file_s = 'C:\\Users\\Administrator\\Documents\\Superstore.xls'

all_sheets = pd.read_excel(file_s, sheet_name=None)

orders_df = all_sheets['Orders']
returns_df = all_sheets['Returns']
people_df = all_sheets['People']

print("Orders Sheet:")
print(orders_df.head())
print("\nReturns Sheet:")
print(returns_df.head())
print("\nPeople Sheet:")
print(people_df.head())
```

```
Orders Sheet:
   Row ID        Order ID Order Date  Ship Date       Ship Mode Customer ID  \
0       1  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
1       2  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
2       3  CA-2016-138688 2016-06-12 2016-06-16    Second Class    DV-13045
3       4  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335
4       5  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335

     Customer Name    Segment        Country             City  ...  \
0      Claire Gute   Consumer  United States        Henderson  ...
1      Claire Gute   Consumer  United States        Henderson  ...
2  Darrin Van Huff  Corporate  United States      Los Angeles  ...
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...

  Postal Code  Region       Product ID         Category Sub-Category  \
0       42420   South  FUR-BO-10001798        Furniture    Bookcases
1       42420   South  FUR-CH-10000454        Furniture       Chairs
2       90036    West  OFF-LA-10000240  Office Supplies       Labels
3       33311   South  FUR-TA-10000577        Furniture       Tables
4       33311   South  OFF-ST-10000760  Office Supplies      Storage

                                        Product Name Sales ($)  Quantity  \
0                  Bush Somerset Collection Bookcase  261.9600         2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400         3
2  Self-Adhesive Address Labels for Typewriters b...   14.6200         2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                     Eldon Fold 'N Roll Cart System   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 rows x 21 columns]

Returns Sheet:
  Returned        Order ID
0      Yes  CA-2017-153822
1      Yes  CA-2017-129707
2      Yes  CA-2014-152345
3      Yes  CA-2015-156440
4      Yes  US-2017-155999

People Sheet:
              Person   Region
0      Anna Andreadi     West
1        Chuck Magee     East
2     Kelly Williams  Central
3  Cassandra Brandow    South
```

In [29]:
```
orders_df.head()
```

Out[29]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales ($)	Quantity	Discount ($)	Profit ($)
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.9600	2	0.00	41.9136
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	0.00	219.5820
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	0.00	6.8714
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	0.45	-383.0310
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	22.3680	2	0.20	2.5164

5 rows × 21 columns

In [31]:
```
orders_df.isnull().sum()
```

Out[31]:
```
Row ID           0
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 [73]:
```
orders_df.duplicated().sum()
```

Out[73]: 0

In [33]:
```
orders_df.dtypes
```

Out[33]:
```
Row ID                    int64
Order ID                 object
Order Date       datetime64[ns]
Ship Date        datetime64[ns]
Ship Mode                object
Customer ID              object
Customer Name            object
Segment                  object
Country                  object
City                     object
State                    object
Postal Code               int64
Region                   object
Product ID               object
Category                 object
Sub-Category             object
Product Name             object
Sales ($)               float64
Quantity                  int64
Discount ($)            float64
Profit ($)              float64
dtype: object
```

In [35]:
```
orders_df.info()
```

```
&lt;class 'pandas.core.frame.DataFrame'&gt;
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales ($)      9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount ($)   9994 non-null   float64
 20  Profit ($)     9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

In [37]:
```
orders_df.describe()
```

Out[37]:

	Row ID	Order Date	Ship Date	Postal Code	Sales ($)	Quantity	Discount ($)	Profit ($)
count	9994.000000	9994	9994	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	2016-04-30 00:07:12.259355648	2016-05-03 23:06:58.571142912	55190.379428	229.858001	3.789574	0.156203	28.656896
min	1.000000	2014-01-03 00:00:00	2014-01-07 00:00:00	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	2015-05-23 00:00:00	2015-05-27 00:00:00	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	2016-06-26 00:00:00	2016-06-29 00:00:00	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	2017-05-14 00:00:00	2017-05-18 00:00:00	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	2017-12-30 00:00:00	2018-01-05 00:00:00	99301.000000	22638.480000	14.000000	0.800000	8399.976000
std	2885.163629	NaN	NaN	32063.693350	623.245101	2.225110	0.206452	234.260108

In [40]:
```
returns_df.head()
```

Out[40]:

	Returned	Order ID
0	Yes	CA-2017-153822
1	Yes	CA-2017-129707
2	Yes	CA-2014-152345
3	Yes	CA-2015-156440
4	Yes	US-2017-155999

In [42]:
```
returns_df.isnull().sum()
```

Out[42]:
```
Returned    0
Order ID    0
dtype: int64
```

In [44]:
```
people_df.head()
```

Out[44]:

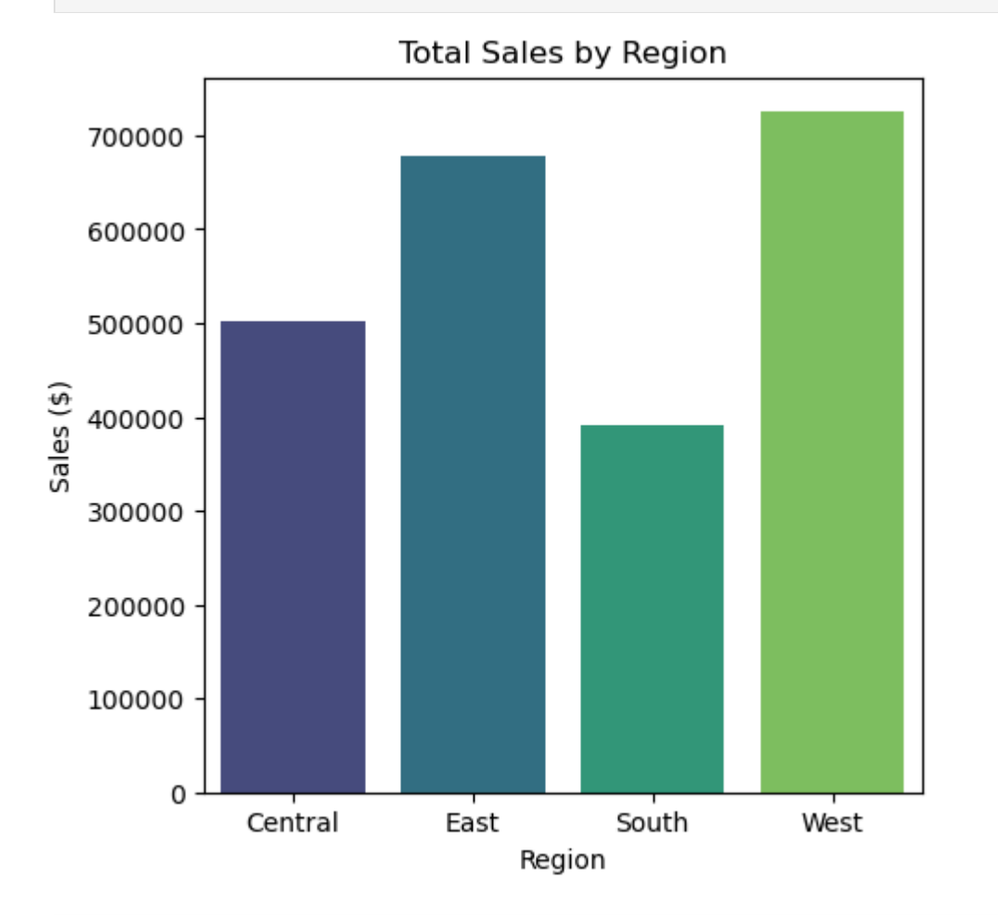
	Person	Region
0	Anna Andreadi	West
1	Chuck Magee	East
2	Kelly Williams	Central
3	Cassandra Brandow	South

In [23]:
```
#TOTAL SALES AND PROFIT BY REGION
region_sales_profit = orders_df.groupby('Region').agg({
    'Sales ($)': 'sum',
    'Profit ($)': 'sum'
}).reset_index()

region_sales_profit
```

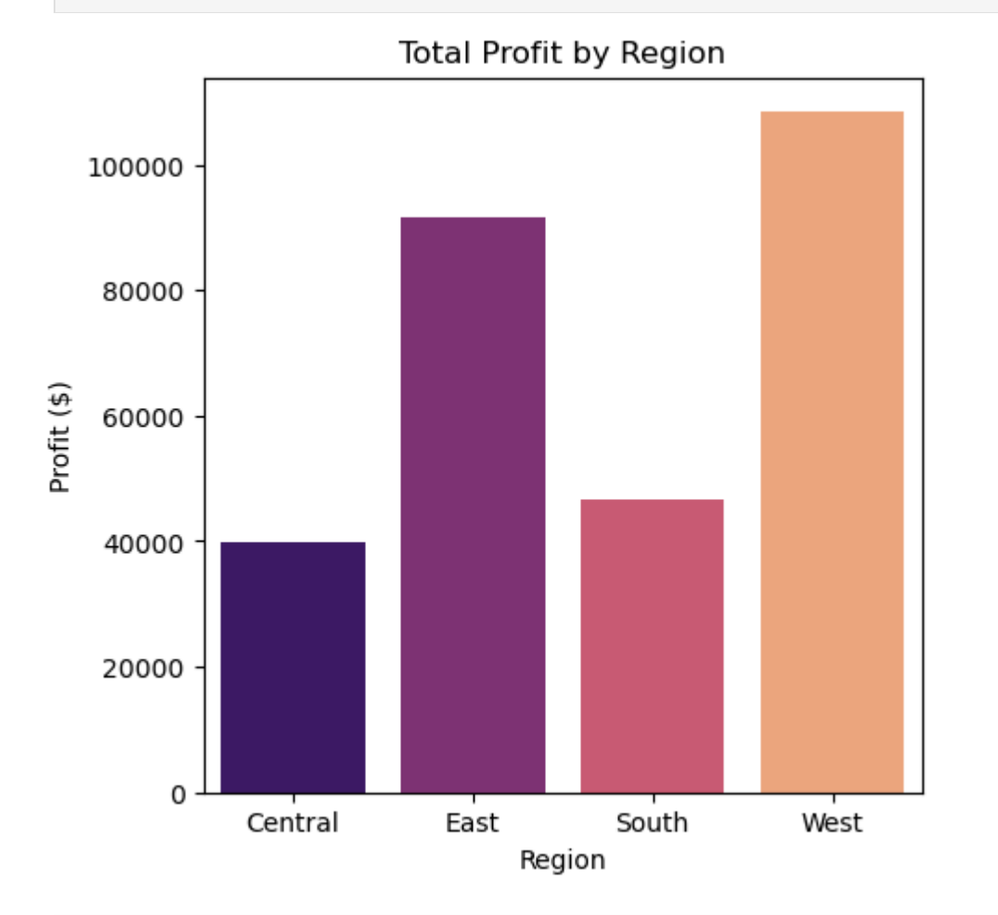
Out[23]:

	Region	Sales ($)	Profit ($)
0	Central	501239.8908	39706.3625
1	East	678781.2400	91522.7800
2	South	391721.9050	46749.4303
3	West	725457.8245	108418.4489

In [33]:
```
#Total SALES BY REGION
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(5, 5))
sns.barplot(x='Region', y='Sales ($)', data=region_sales_profit, palette='viridis')
plt.title('Total Sales by Region')
plt.show()
```



In [39]:
```
#TOTAL PROFIT BY REGION
plt.figure(figsize=(5, 5))
sns.barplot(x='Region', y='Profit ($)', data=region_sales_profit, palette='magma')
plt.title('Total Profit by Region')
plt.show()
```

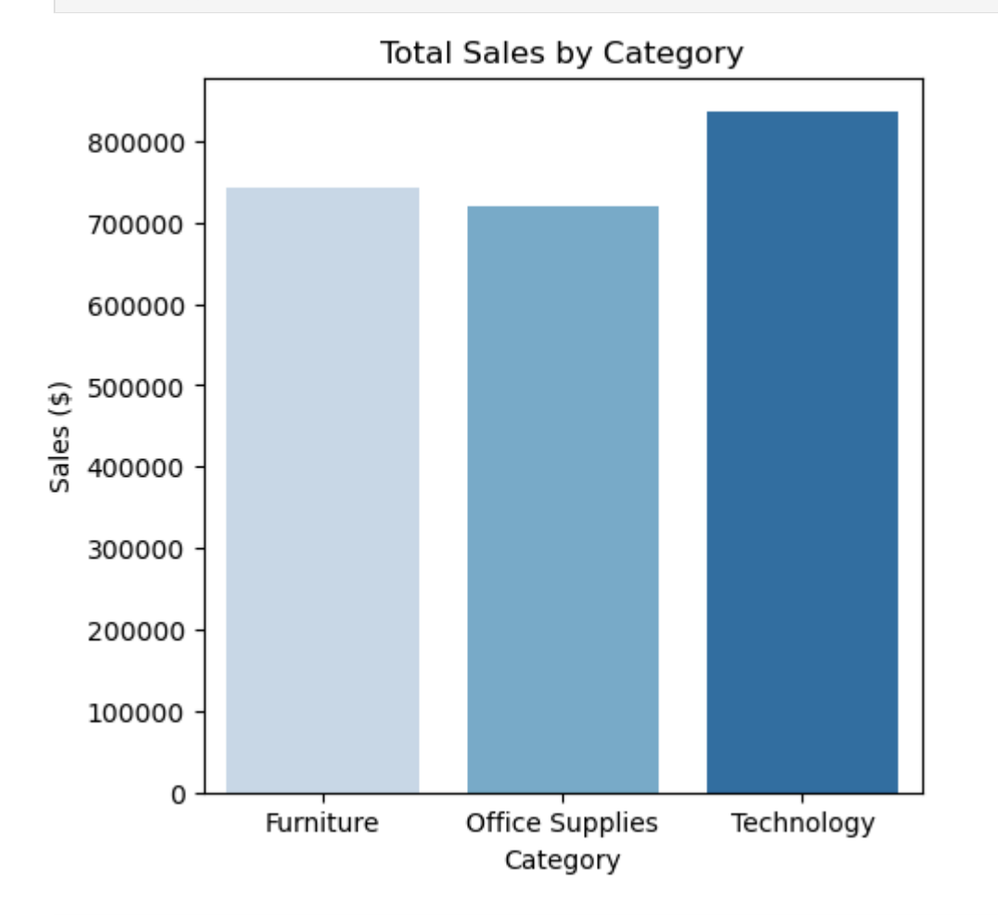


In [37]:
```
#TOTAL SALES AND PROFIT BY CATEGORY
category_sales_profit = orders_df.groupby('Category').agg({
    'Sales ($)': 'sum',
    'Profit ($)': 'sum'
}).reset_index()
category_sales_profit
```

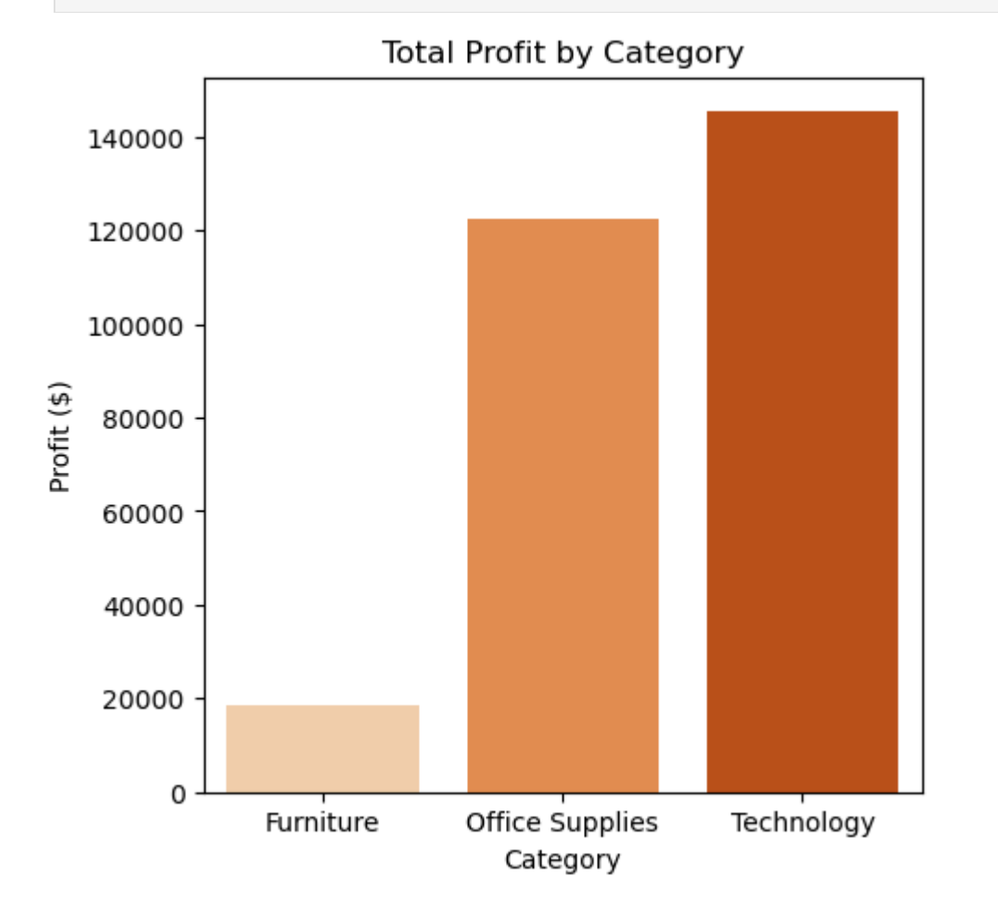
Out[37]:

	Category	Sales ($)	Profit ($)
0	Furniture	741999.7953	18451.2728
1	Office Supplies	719047.0320	122490.8008
2	Technology	836154.0330	145454.9481

In [43]:
```
#SALES BY CATEGORY

plt.figure(figsize=(5, 5))
sns.barplot(x='Category', y='Sales ($)', data=category_sales_profit, palette='Blues')
plt.title('Total Sales by Category')
plt.show()
```



In [45]:
```
#PROFIT BY CATEGORY

plt.figure(figsize=(5, 5))
sns.barplot(x='Category', y='Profit ($)', data=category_sales_profit, palette='Oranges')
plt.title('Total Profit by Category')
plt.show()
```

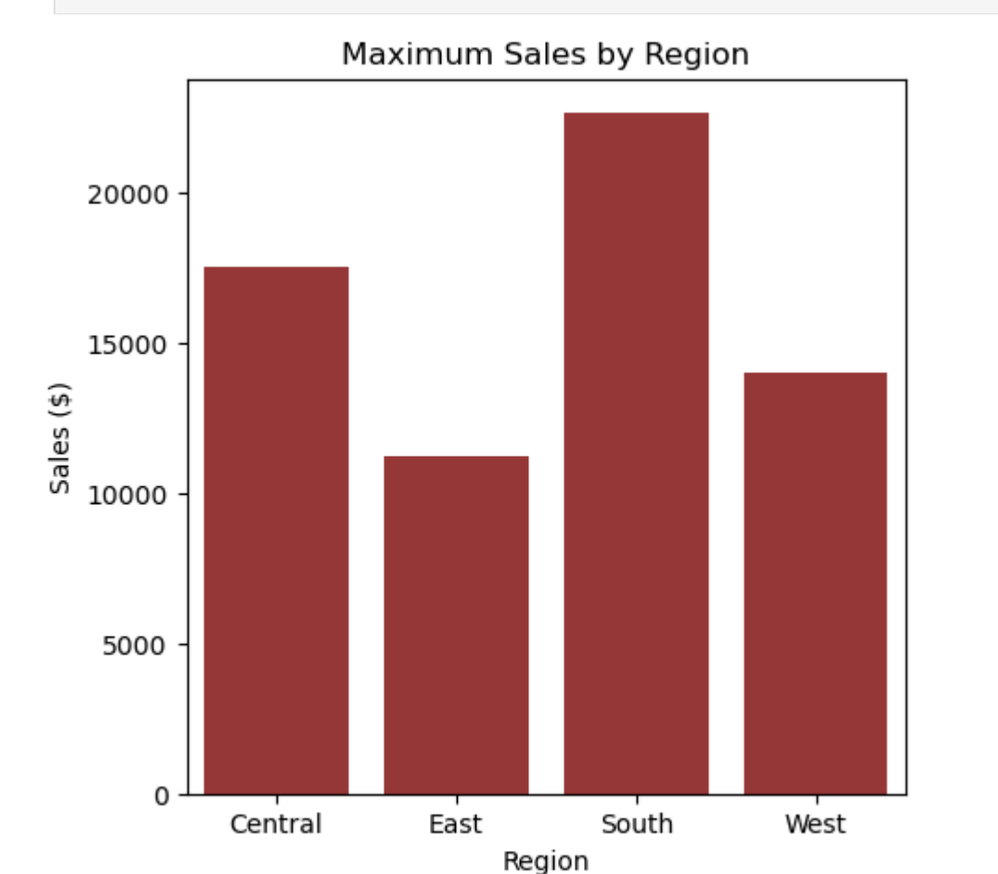


In [49]:
```
#MAXIMUM SALES AND PROFIT BY REGION
Maximum_sp_region= orders_df.groupby('Region').max({
    'Sales ($)',
    'Profit ($)'
}).reset_index()
Maximum_sp_region
```

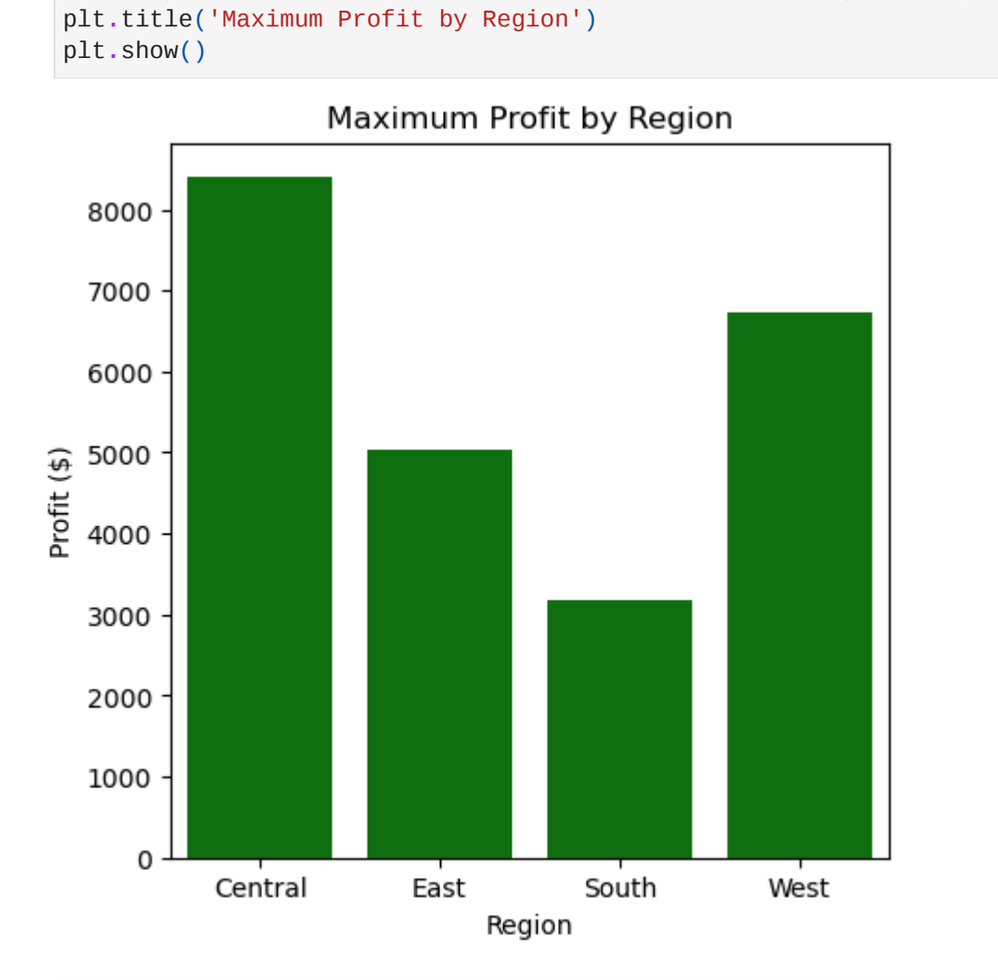
Out[49]:

	Region	Row ID	Postal Code	Sales ($)	Quantity	Discount ($)	Profit ($)
0	Central	9984	79907	17499.950	14	0.8	8399.9760
1	East	9986	45503	11199.968	14	0.7	5039.9856
2	South	9990	72762	22638.480	14	0.7	3177.4750
3	West	9994	99301	13999.960	14	0.7	6719.9808

In [55]:
```
#MAXIMUM SALES BY REGION
plt.figure(figsize=(5, 5))
sns.barplot(x='Region', y='Sales ($)', data=Maximum_sp_region, color='Brown')
plt.title('Maximum Sales by Region')
plt.show()
```



In [59]:
```
#MAXIMUM PROFIT BY REGION
plt.figure(figsize=(5, 5))
sns.barplot(x='Region', y='Profit ($)', data=Maximum_sp_region, color='Green')
plt.title('Maximum Profit by Region')
plt.show()
```



In [15]:
```
returns_df.duplicated().sum()
```

Out[15]: 0

In [91]:
```
#Merge Orders and Returns data on 'Order ID' to identify returned orders
merge_or = pd.merge(orders_df, returns_df, on='Order ID', how='left')
merge_or.head(2)
```

Out[91]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Product ID	Category	Sub-Category	Product Name	Sales ($)	Quantity	Discount ($)	Profit ($)	Returned
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2	0.0	41.9136	NaN
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.94	3	0.0	219.5820	NaN

2 rows × 22 columns

In [93]:
```
merge_or['Returned'].fillna('No', inplace =True )
merge_or.head(2)
```

Out[93]:

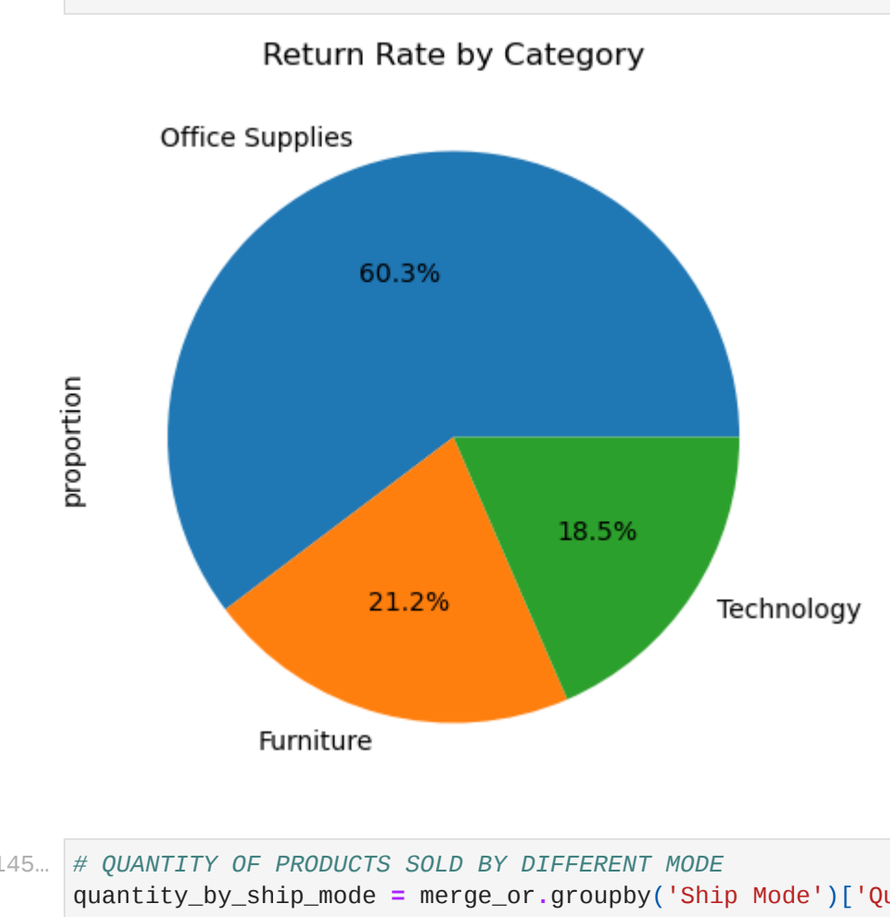
	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Product ID	Category	Sub-Category	Product Name	Sales ($)	Quantity	Discount ($)	Profit ($)	Returned
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2	0.0	41.9136	No
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.94	3	0.0	219.5820	No

2 rows × 22 columns

In [95]:
```
#CALCULATING RETURN RATE BY CATEGORY
return_rate_by_category = merge_or['Category'].value_counts(normalize=True) * 100
return_rate_by_category
```

Out[95]:
```
Category
Office Supplies    60.296178
Furniture          21.222734
Technology         18.481089
Name: proportion, dtype: float64
```

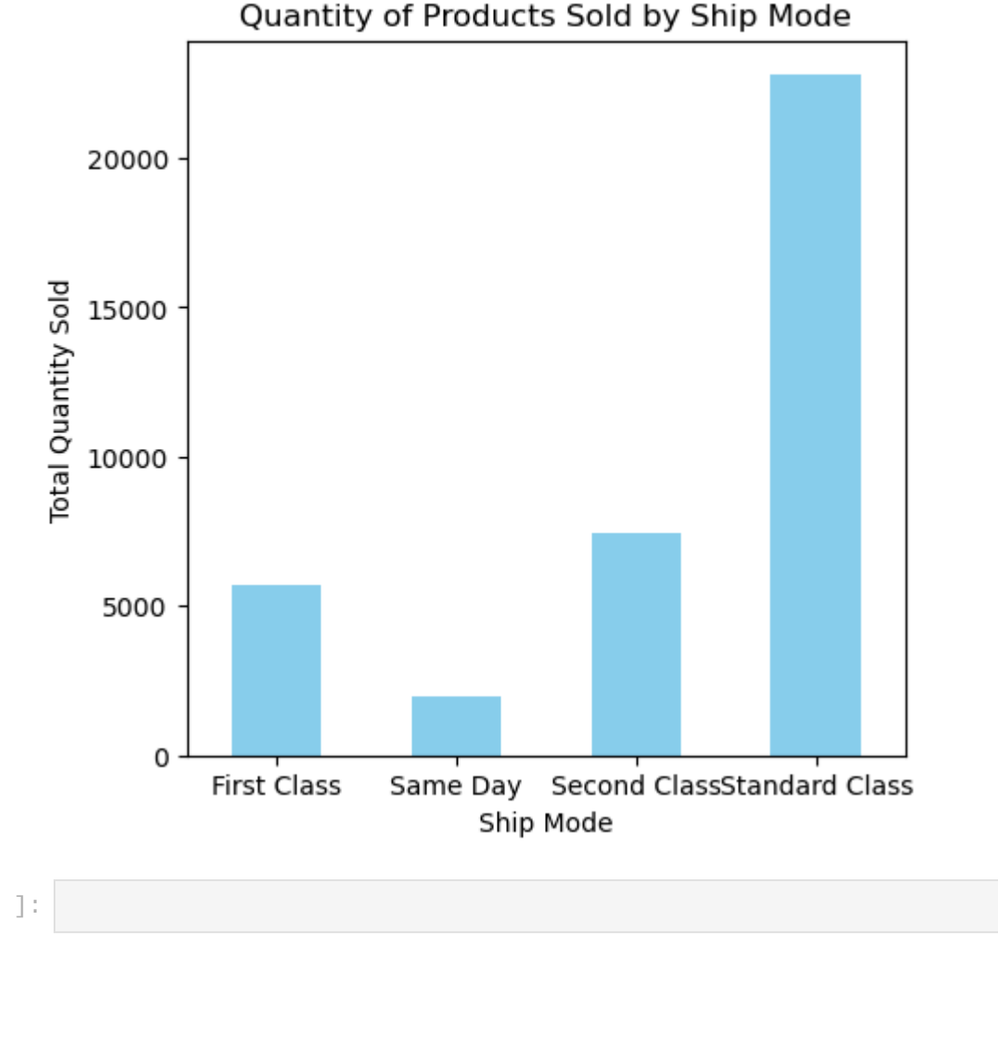
In [99]:
```
#PIE CHART SHOWING RETURN RATE PERCENT BY CATEGORY
return_rate_by_category.plot(kind='pie', autopct='%1.1f%%', figsize=(5, 5), title='Return Rate by Category')
plt.show()
```



In [145…
```
# QUANTITY OF PRODUCTS SOLD BY DIFFERENT MODE
quantity_by_ship_mode = merge_or.groupby('Ship Mode')['Quantity'].sum()
quantity_by_ship_mode
```

Out[145…
```
Ship Mode
First Class        5693
Same Day           1960
Second Class       7423
Standard Class    22797
Name: Quantity, dtype: int64
```

In [147…
```
plt.figure(figsize=(5, 5))
quantity_by_ship_mode.plot(kind='bar', color='skyblue')
plt.title('Quantity of Products Sold by Ship Mode')
plt.xlabel('Ship Mode')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation=0)
plt.show()
```



In [ ]: