# PROJECT TITLE

# Exploratory Data Analysis on Superstore Sales Dataset

In [26]:
```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [27]:
```
#load excel file
df = pd.read_excel("C:\\Users\\HEENA\\OneDrive\\Desktop\\power bi lecture\\1.sup
```

In [25]:
```
##Data overview
df.columns
```

Out[25]:
```
Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
       'segment', 'state', 'country', 'market', 'region', 'product_id',
       'category', 'sub_category', 'product_name', 'sales', 'quantity',
       'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
      dtype='object')
```

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   order_id        51290 non-null  object
 1   order_date      51290 non-null  datetime64[ns]
 2   ship_date       51290 non-null  datetime64[ns]
 3   ship_mode       51290 non-null  object
 4   customer_name   51290 non-null  object
 5   segment         51290 non-null  object
 6   state           51290 non-null  object
 7   country         51290 non-null  object
 8   market          51290 non-null  object
 9   region          51290 non-null  object
 10  product_id      51290 non-null  object
 11  category        51290 non-null  object
 12  sub_category    51290 non-null  object
 13  product_name    51290 non-null  object
 14  sales           51290 non-null  float64
 15  quantity        51290 non-null  int64
 16  discount        51290 non-null  float64
 17  profit          51290 non-null  float64
 18  shipping_cost   51290 non-null  float64
 19  order_priority  51290 non-null  object
 20  year            51290 non-null  int64
dtypes: datetime64[ns](2), float64(4), int64(2), object(13)
memory usage: 8.2+ MB
```

In [16]:
```
df.head()
```

Out[16]:

| | order_id | order_date | ship_date | ship_mode | customer_name | segment | state |
|---|---|---|---|---|---|---|---|
| **0** | AG-2011-2040 | 2011-01-01 | 2011-01-06 | Standard Class | Toby Braunhardt | Consumer | Constantine |
| **1** | IN-2011-47883 | 2011-01-01 | 2011-01-08 | Standard Class | Joseph Holt | Consumer | New South Wales |
| **2** | HU-2011-1220 | 2011-01-01 | 2011-01-05 | Second Class | Annie Thurman | Consumer | Budapest |
| **3** | IT-2011-3647632 | 2011-01-01 | 2011-01-05 | Second Class | Eugene Moren | Home Office | Stockholm |
| **4** | IN-2011-47883 | 2011-01-01 | 2011-01-08 | Standard Class | Joseph Holt | Consumer | New South Wales |

5 rows × 21 columns

In [17]:
```
df.describe()
```

Out[17]:

| | order_date | ship_date | sales | quantity | discount |
|---|---|---|---|---|---|
| **count** | 51290 | 51290 | 51290.000000 | 51290.000000 | 51290.000000 |
| **mean** | 2013-05-11 21:26:49.155780864 | 2013-05-15 20:42:42.745174528 | 246.490581 | 3.476545 | 0.142908 |
| **min** | 2011-01-01 00:00:00 | 2011-01-03 00:00:00 | 0.444000 | 1.000000 | 0.000000 |
| **25%** | 2012-06-19 00:00:00 | 2012-06-23 00:00:00 | 30.758625 | 2.000000 | 0.000000 |
| **50%** | 2013-07-08 00:00:00 | 2013-07-12 00:00:00 | 85.053000 | 3.000000 | 0.000000 |
| **75%** | 2014-05-22 00:00:00 | 2014-05-26 00:00:00 | 251.053200 | 5.000000 | 0.200000 |
| **max** | 2014-12-31 00:00:00 | 2015-01-07 00:00:00 | 22638.480000 | 14.000000 | 0.850000 |
| **std** | NaN | NaN | 487.565361 | 2.278766 | 0.212280 |

In [18]:
```
df.shape
```

Out[18]:
```
(51290, 21)
```

In [19]:
```
#check missing value
df.isnull().sum()
```

Out[19]:
```
order_id          0
order_date        0
ship_date         0
ship_mode         0
customer_name     0
segment           0
state             0
country           0
market            0
region            0
product_id        0
category          0
sub_category      0
product_name      0
sales             0
quantity          0
discount          0
profit            0
shipping_cost     0
order_priority    0
year              0
dtype: int64
```

In [20]:
```
df.duplicated().sum()
```

Out[20]:
```
np.int64(0)
```

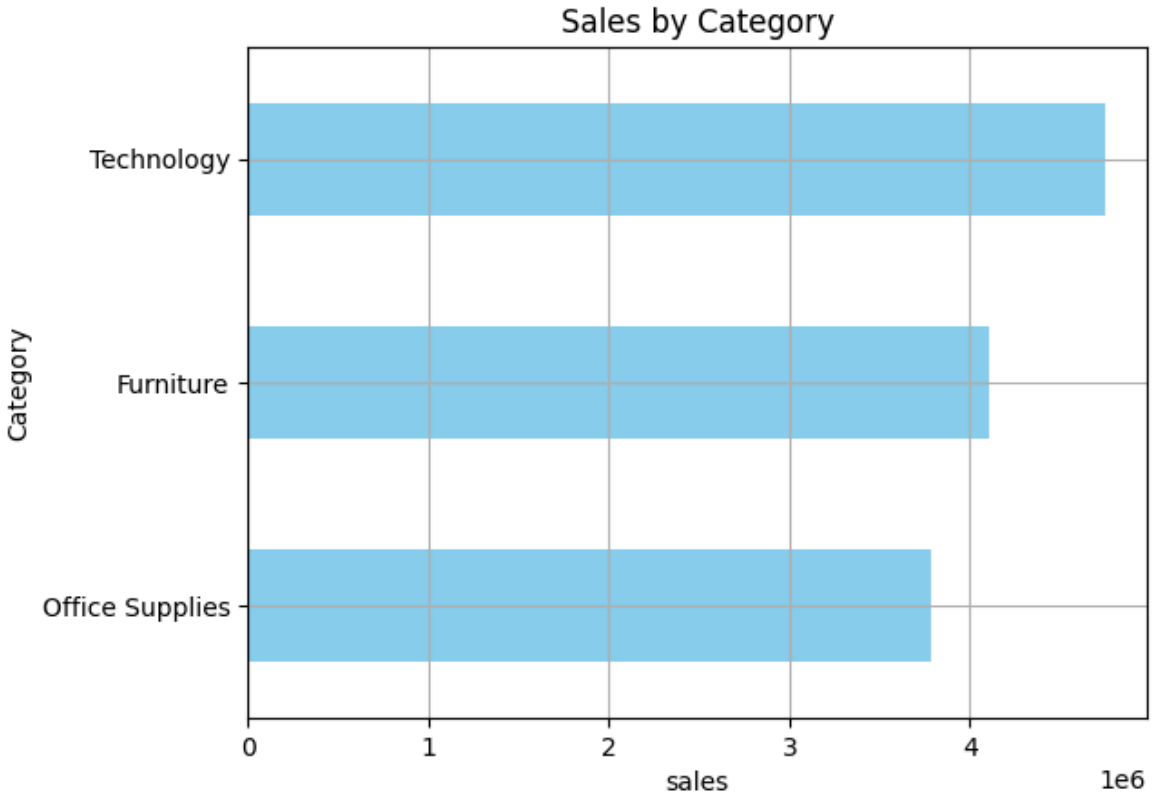
In [21]:
```
df.dtypes
```

Out[21]:
```
order_id                object
order_date      datetime64[ns]
ship_date       datetime64[ns]
ship_mode               object
customer_name           object
segment                 object
state                   object
country                 object
market                  object
region                  object
product_id              object
category                object
sub_category            object
product_name            object
sales                  float64
quantity                 int64
discount               float64
profit                 float64
shipping_cost          float64
order_priority          object
year                     int64
dtype: object
```

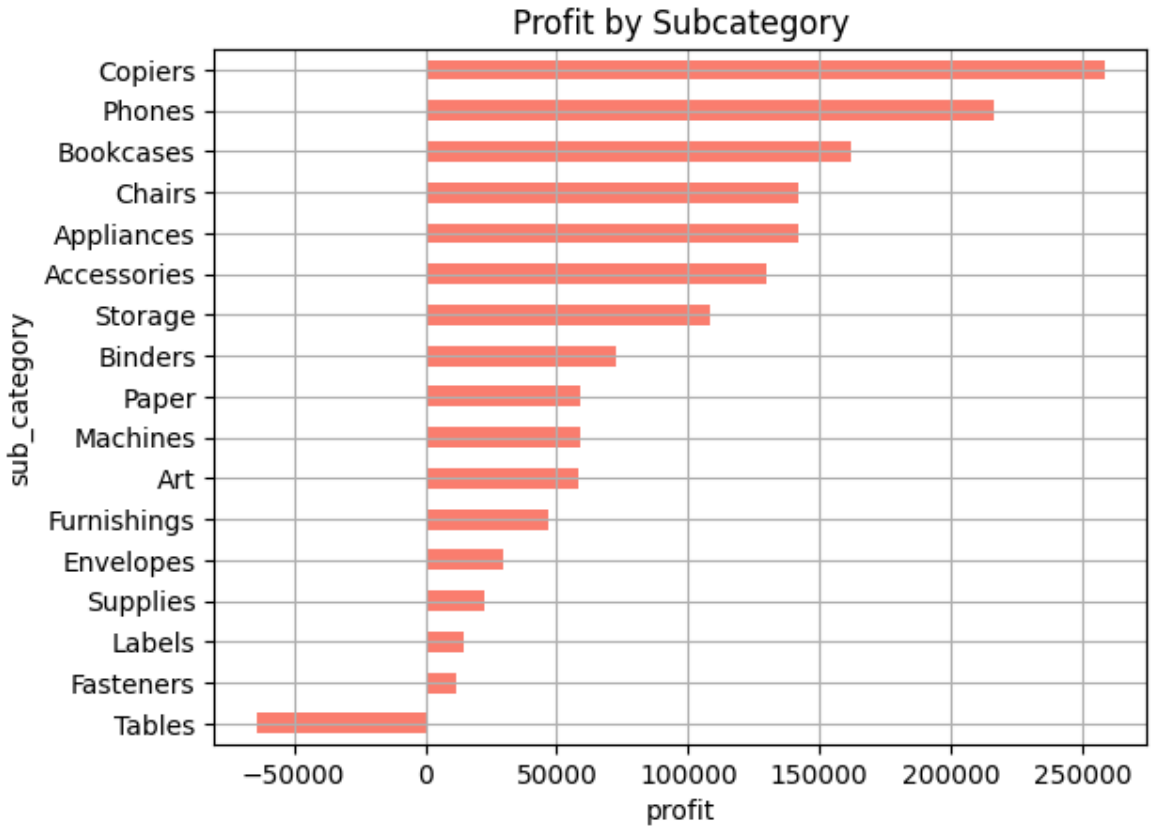
## Visualization Section

In [37]:
```
#sales by category
category_sales=df.groupby("category")["sales"].sum().sort_values()
category_sales.plot(kind="barh",color="skyblue")
plt.title("Sales by Category")
```

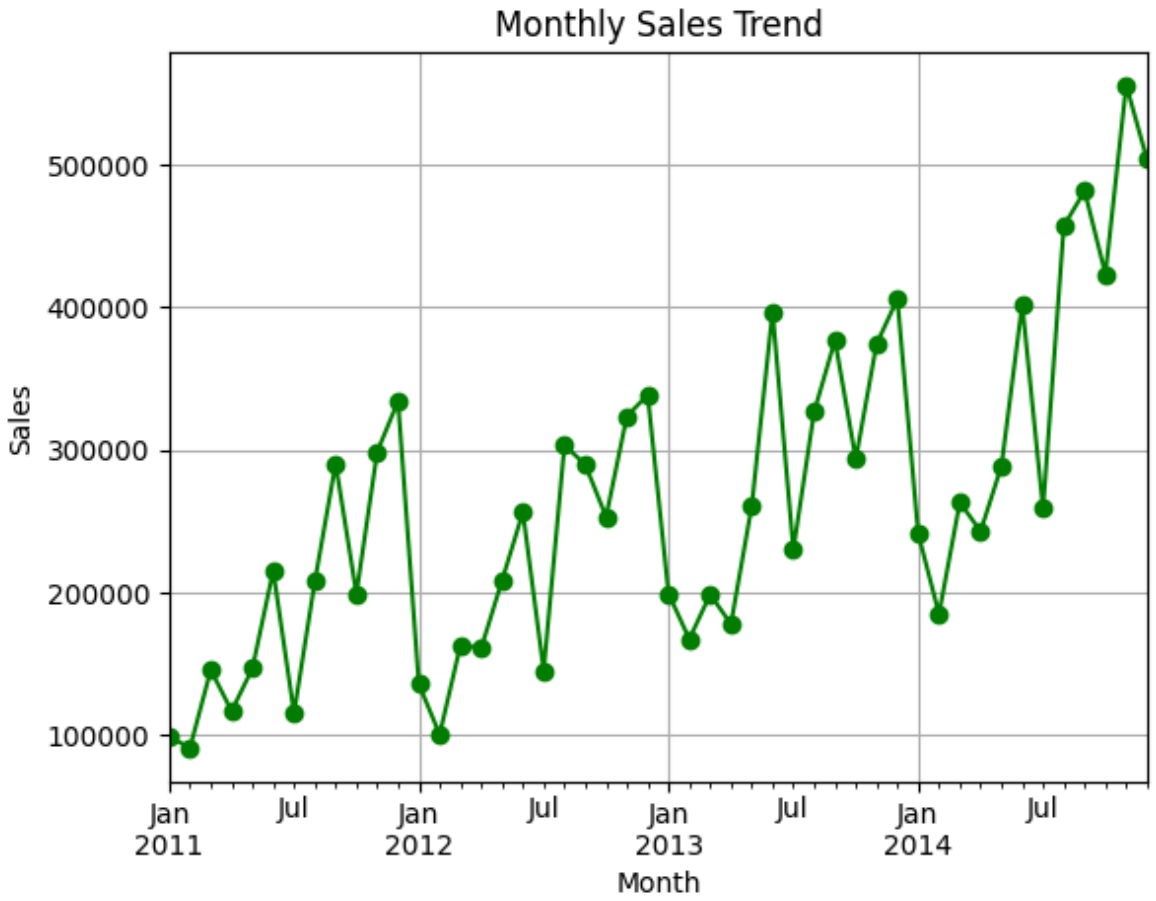
```
plt.xlabel("sales")
plt.ylabel("Category")
plt.grid(True)
plt.show()
```




In [46]:
```
#**Profit by subcategory**
subc_profit = df.groupby("sub_category")["profit"].sum().sort_values()
subc_profit.plot(kind="barh",color="salmon")
plt.title("Profit by Subcategory")
plt.xlabel("profit")
plt.ylabel("sub_category")
plt.grid(True)
plt.show()
```



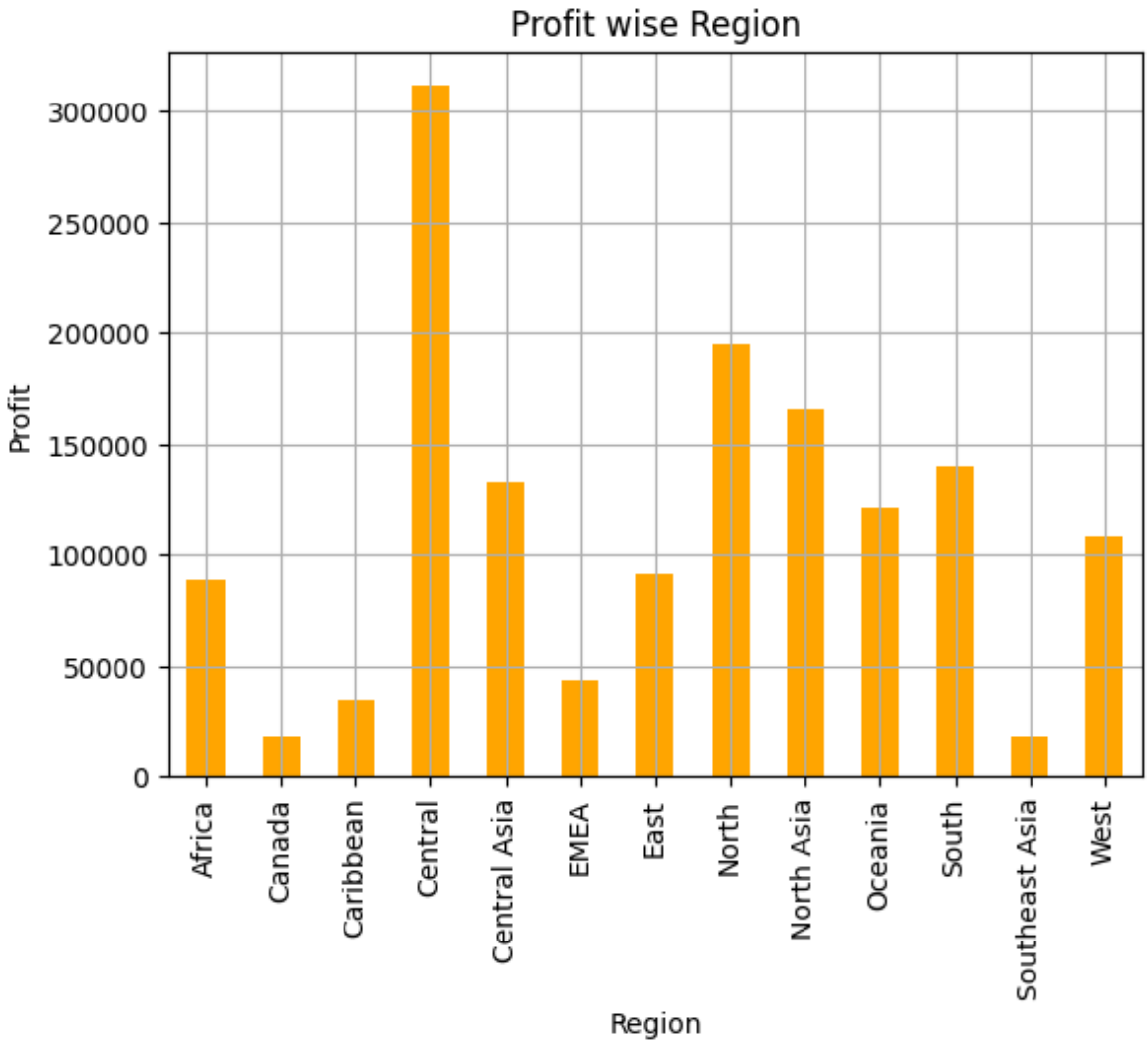

In [62]:
```
#Monthly Sales Trend
df['Month'] =  df['order_date'].dt.to_period("M")
Monthly_sales= df.groupby("Month")['sales'].sum()
Monthly_sales.plot(kind='line', marker='o',color='green')
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel('Sales')
plt.grid()
plt.show()
```



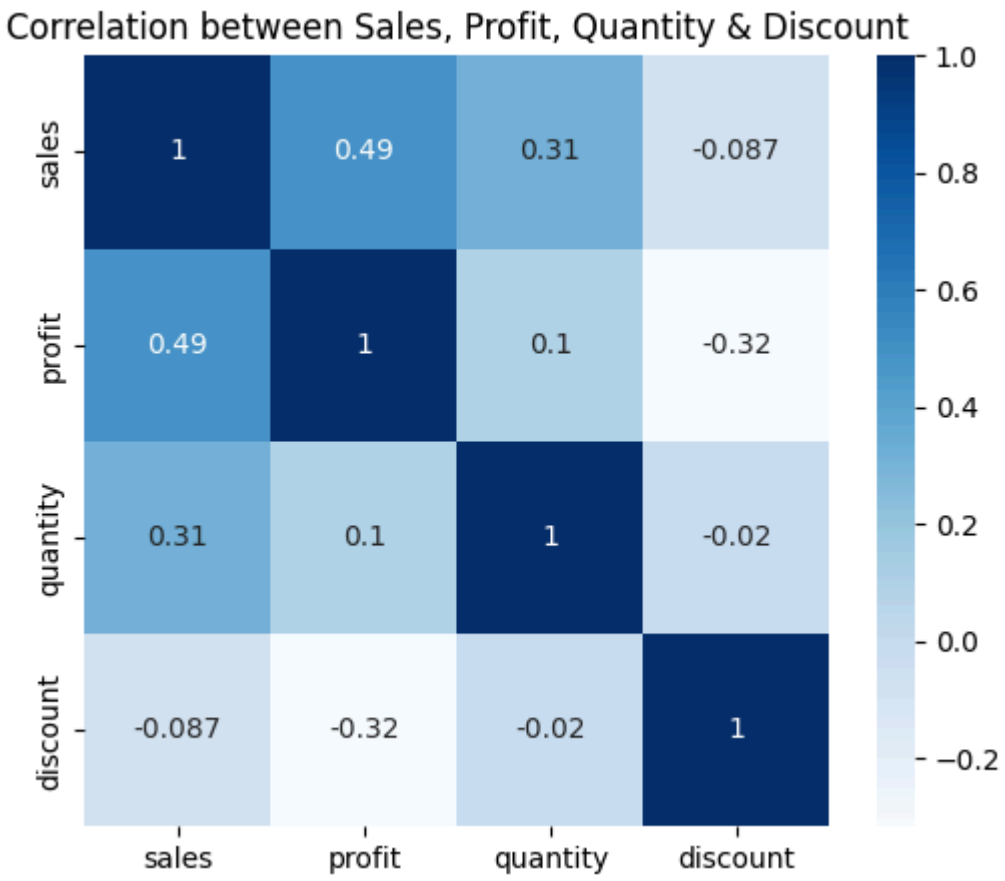

In [55]:
```
#to show column names
df.columns
```

Out[55]:
```
Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
       'segment', 'state', 'country', 'market', 'region', 'product_id',
       'category', 'sub_category', 'product_name', 'sales', 'quantity',
       'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
      dtype='object')
```

In [68]:
```
#REGION WISE PROFIT
region_profit=df.groupby("region")['profit'].sum()
region_profit.plot(kind="bar",color="orange")
plt.title("Profit wise Region")
plt.ylabel("Profit")
plt.xlabel("Region")
plt.grid()
plt.show()
```

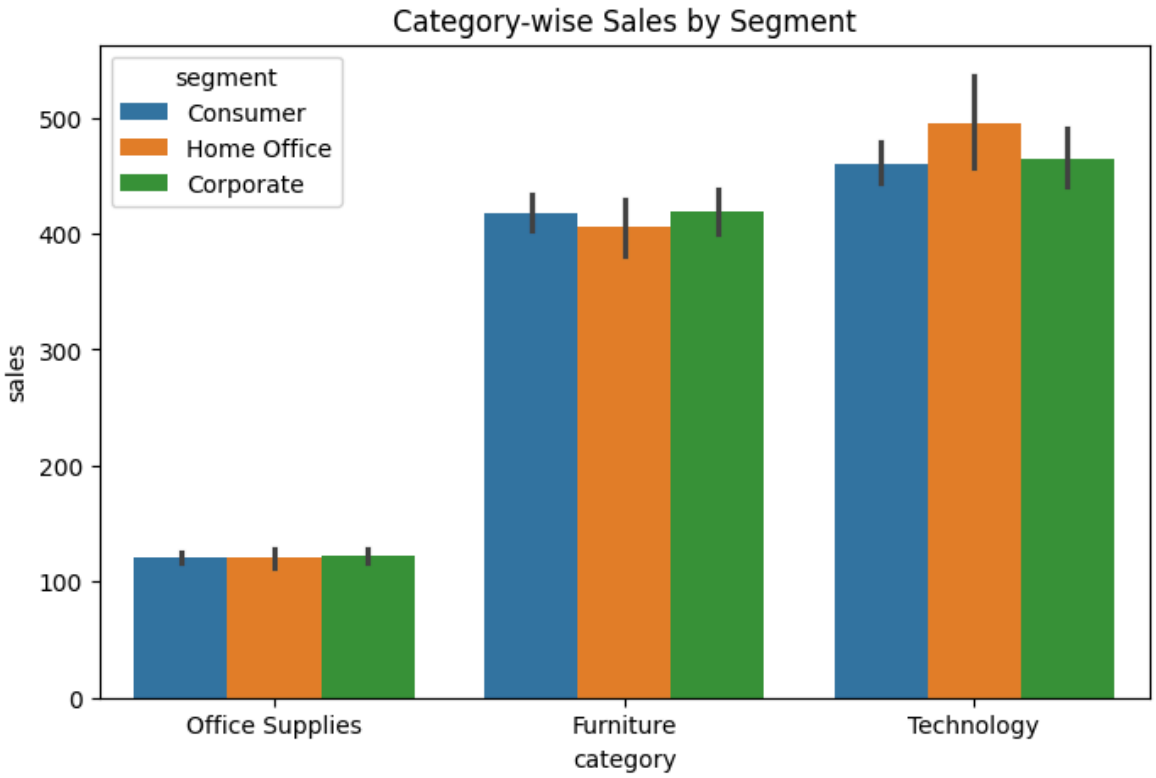



In [73]:
```
# Correlation Heatmap
plt.figure(figsize=(6,5))
sns.heatmap(df[['sales', 'profit', 'quantity', 'discount']].corr(), annot=True,
plt.title("Correlation between Sales, Profit, Quantity & Discount")
plt.show()
```




In [74]:
```
#Category vs Segment Analysis

plt.figure(figsize=(8,5))
sns.barplot(data=df, x='category', y='sales', hue='segment')
plt.title("Category-wise Sales by Segment")
plt.show()
```



## Conclusion

- Technology is the most profitable category
- West region gives highest profit
- December has peak sales
- Discount and Profit are negatively correlated
- Tables and Bookcases are least profitable

In [ ]: