In [2]:

```python
import os
print(os.getcwd())
```

```
d:\Data Analyst\INTERNSHIP\project 2 retail
```

In [4]:

```python
import pandas as pd

file_path = r"Sample Superstore\Sample - Superstore.csv"
df = pd.read_csv(file_path, encoding='latin1')
df.head()
```

Out[4]:

| | Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | Country | City | ... | Postal Code | Region | Prod |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | CA-2016-152156 | 11/8/2016 | 11/11/2016 | Second Class | CG-12520 | Claire Gute | Consumer | United States | Henderson | ... | 42420 | South | FUR-I<br>10001 |
| **1** | 2 | CA-2016-152156 | 11/8/2016 | 11/11/2016 | Second Class | CG-12520 | Claire Gute | Consumer | United States | Henderson | ... | 42420 | South | FUR-(<br>10000 |
| **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-<br>10000 |
| **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-<br>10000 |
| **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-<br>10000 |

5 rows × 21 columns

In [5]:

```python
df.info()
df.describe()
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
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   object 
 3   Ship Date      9994 non-null   object 
 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: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

Out[5]:

```
Index(['Row ID', '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 [6]:

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

In [7]:

```
df.dtypes[['Order Date', 'Ship Date']]
```

Out[7]:

```
Order Date    datetime64[ns]
Ship Date     datetime64[ns]
dtype: object
```

In [8]:

```
df['Order Year'] = df['Order Date'].dt.year
df['Order Month'] = df['Order Date'].dt.month
```

In [9]:

```
df[['Order Date', 'Order Year', 'Order Month']].head()
```

Out[9]:

|  | Order Date | Order Year | Order Month |
|---|---|---|---|
| **0** | 2016-11-08 | 2016 | 11 |
| **1** | 2016-11-08 | 2016 | 11 |
| **2** | 2016-06-12 | 2016 | 6 |
| **3** | 2015-10-11 | 2015 | 10 |
| **4** | 2015-10-11 | 2015 | 10 |

In [10]:

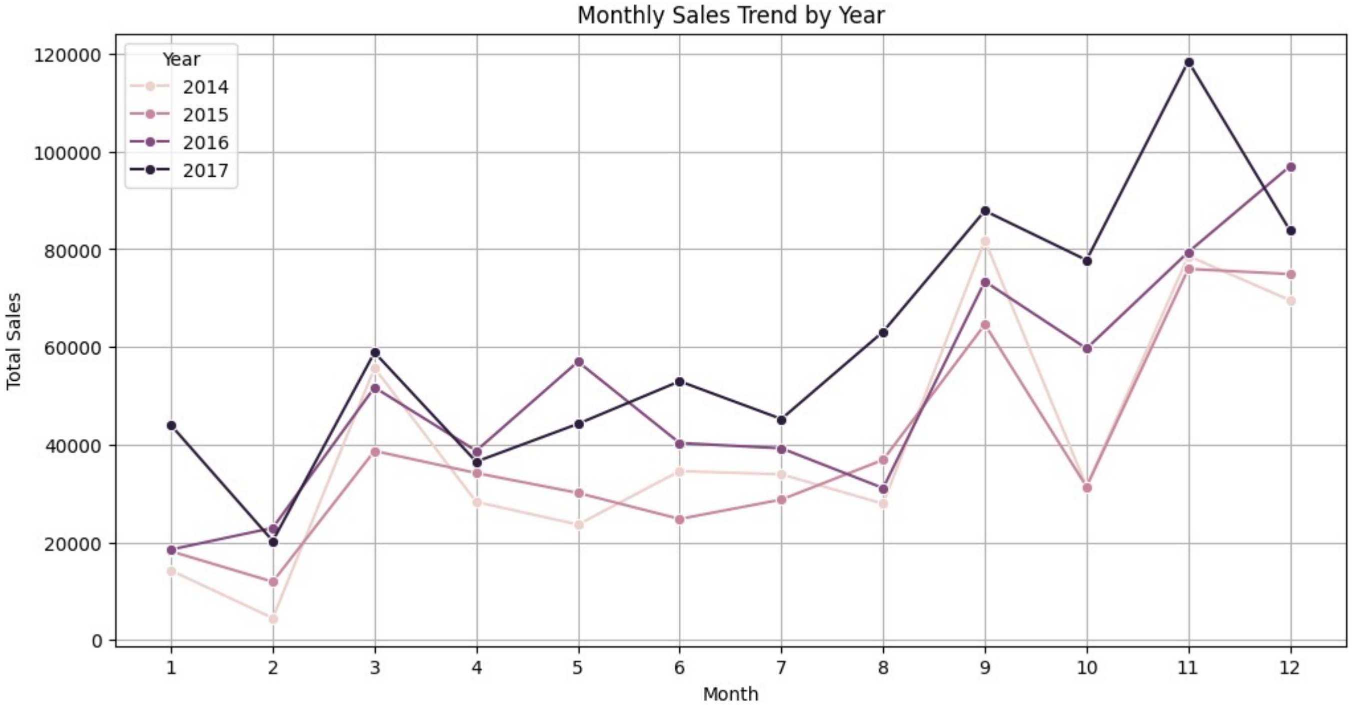
```
monthly_sales = df.groupby(['Order Year', 'Order Month'])['Sales'].sum().reset_index()
monthly_sales.columns = ['Year', 'Month', 'Total Sales']
monthly_sales.head()
```

Out[10]:

|  | Year | Month | Total Sales |
|---|---|---|---|
| **0** | 2014 | 1 | 14236.895 |
| **1** | 2014 | 2 | 4519.892 |
| **2** | 2014 | 3 | 55691.009 |
| **3** | 2014 | 4 | 28295.345 |
| **4** | 2014 | 5 | 23648.287 |

In [11]:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12,6))
sns.lineplot(data=monthly_sales, x='Month', y='Total Sales', hue='Year', marker='o')
plt.title('Monthly Sales Trend by Year')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.xticks(range(1, 13))
plt.grid(True)
plt.show()
```

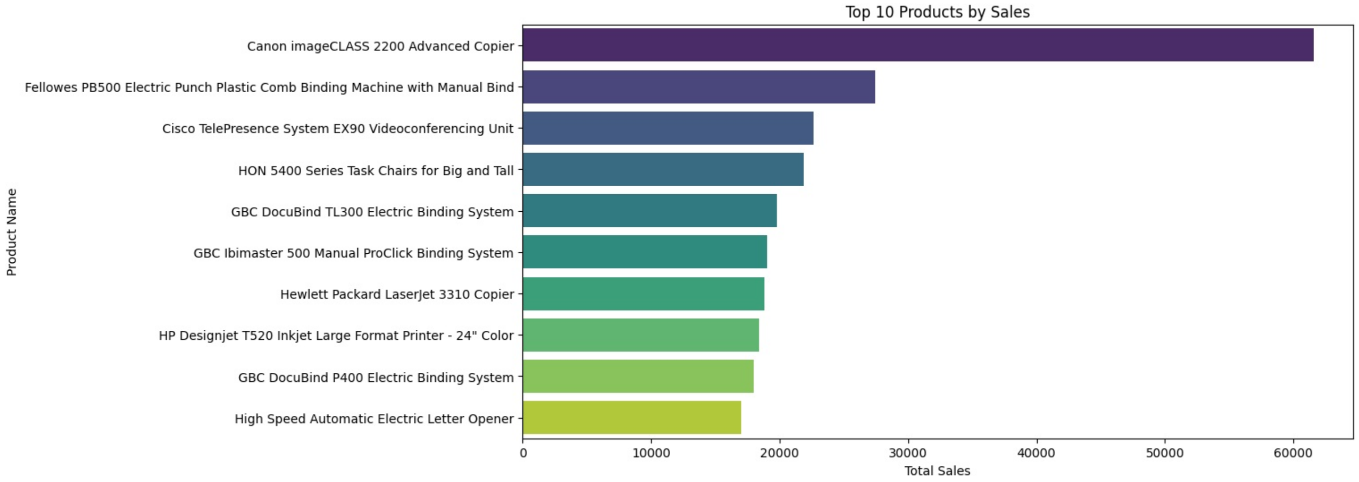


In [12]:

```
top_products = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_products.values, y=top_products.index, palette='viridis')
```

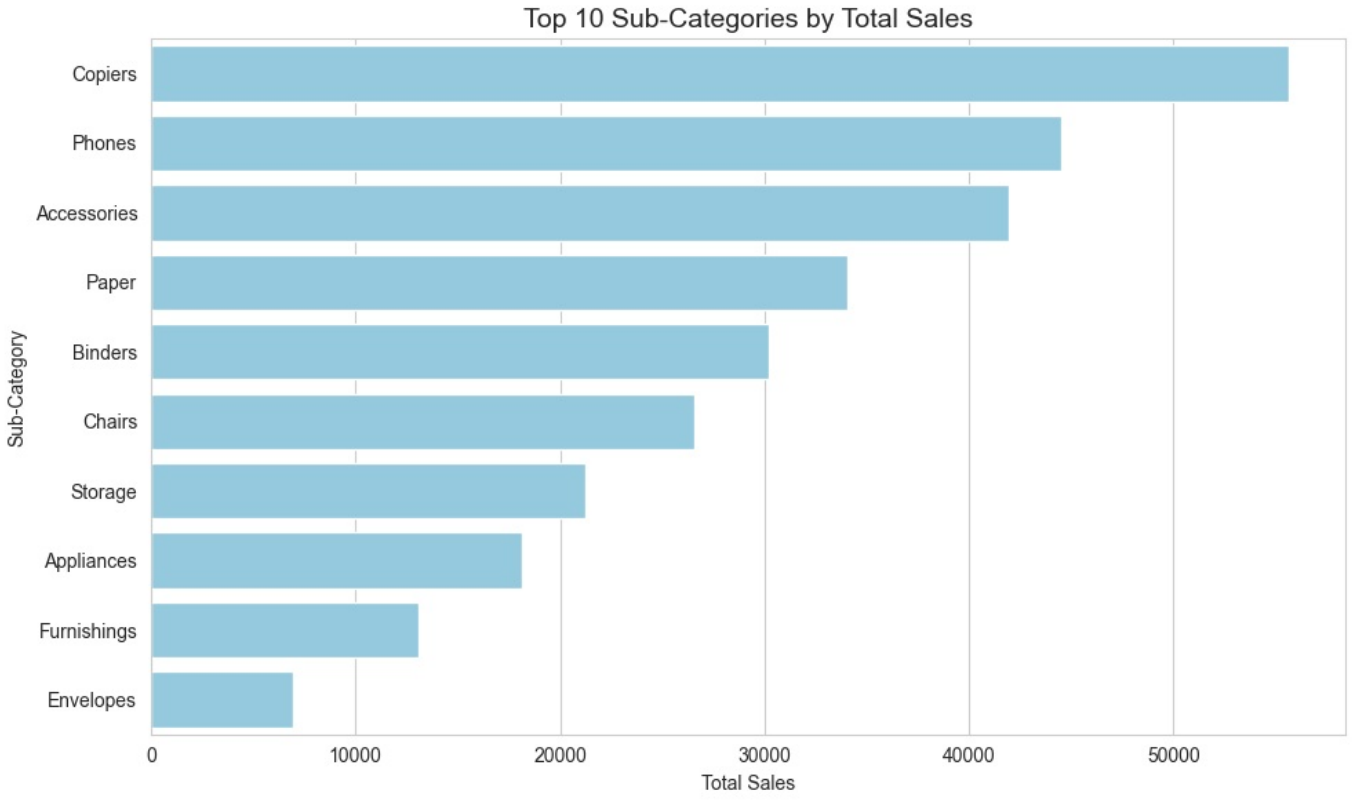
```
plt.title('Top 10 Products by Sales')
plt.xlabel('Total Sales')
plt.ylabel('Product Name')
plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_6796\484867223.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_products.values, y=top_products.index, palette='viridis')
```



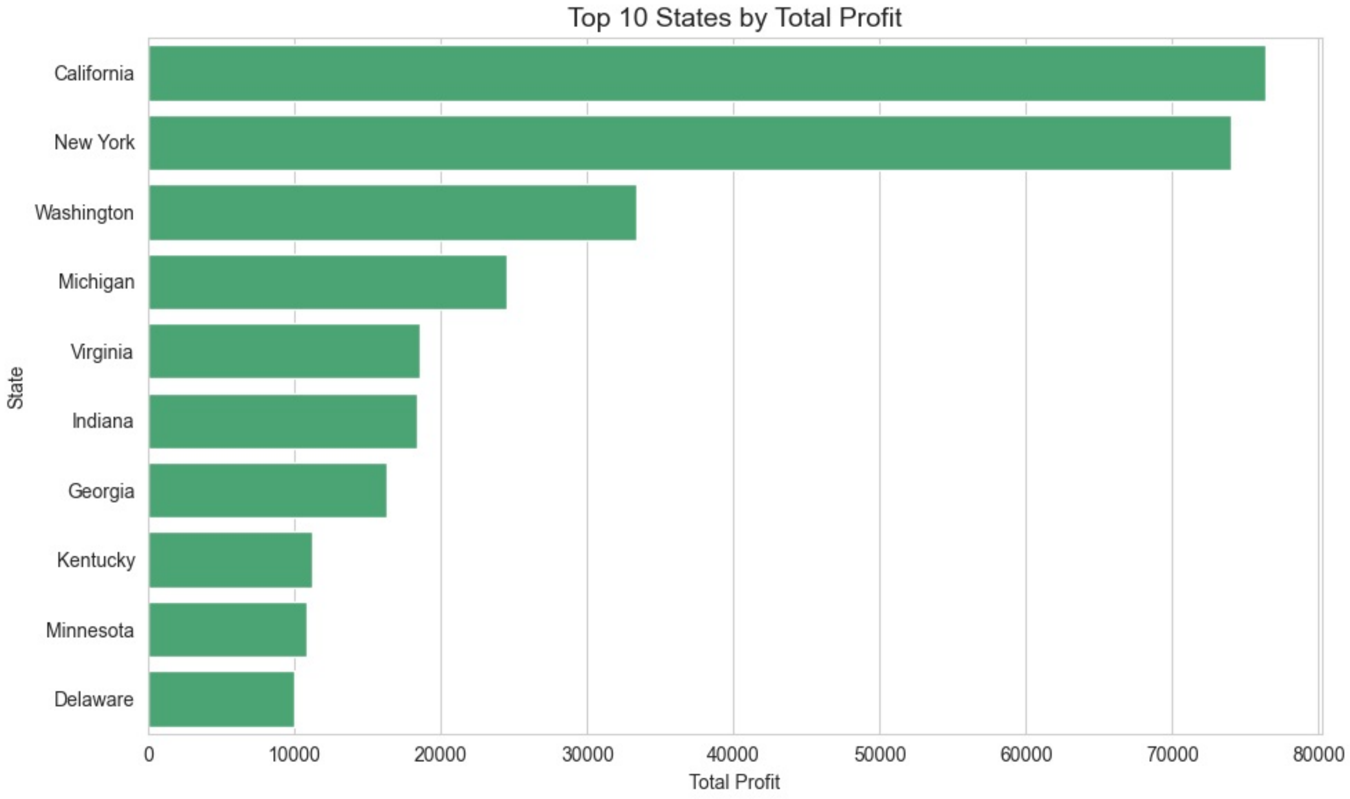
In [22]:

```
import matplotlib.pyplot as plt
import seaborn as sns

import matplotlib.pyplot as plt
import seaborn as sns

# Set figure size and style
plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")

# Barplot with yellow color
sns.barplot(x=top_subcategories.values, y=top_subcategories.index, color='skyblue')

# Titles and labels
plt.title('Top 10 Sub-Categories by Total Sales', fontsize=14)
plt.xlabel('Total Sales')
plt.ylabel('Sub-Category')

# Show plot
plt.tight_layout()
plt.show()
```



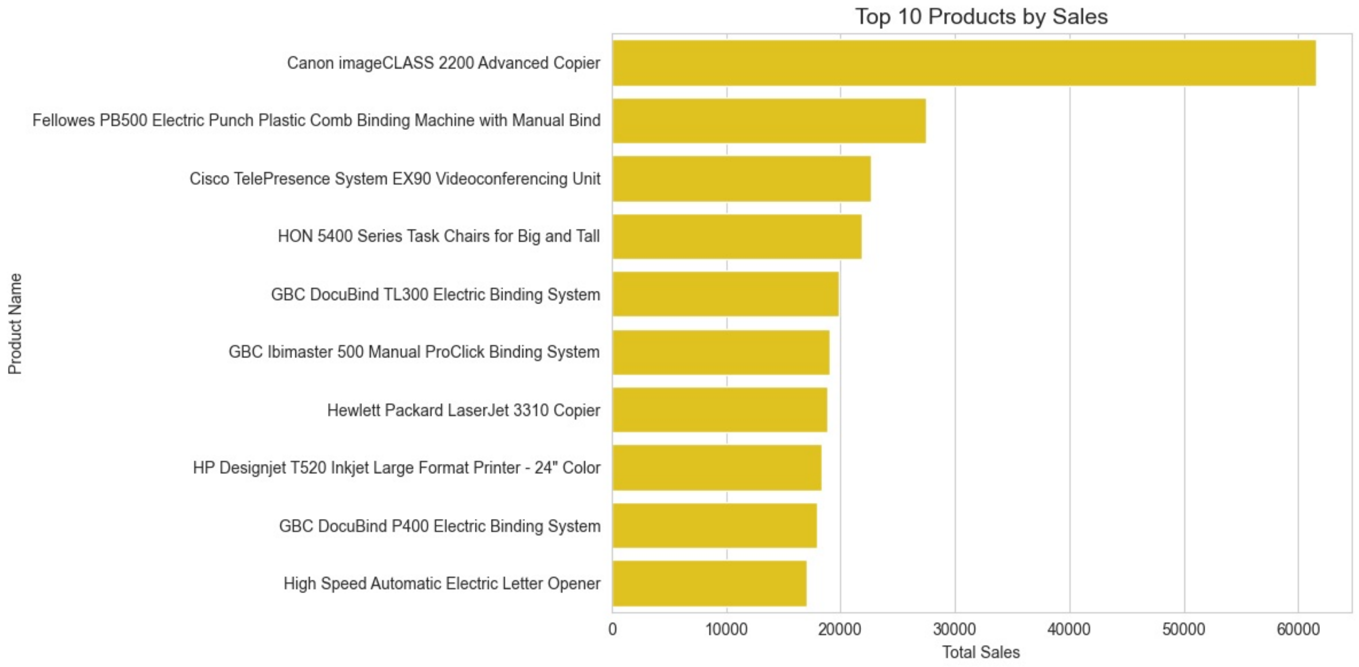
In [24]:

```
# Group data by State and calculate total profit
state_profit = df.groupby('State')['Profit'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=state_profit.values, y=state_profit.index, color='mediumseagreen')

# Add titles and labels
plt.title('Top 10 States by Total Profit', fontsize=14)
plt.xlabel('Total Profit')
plt.ylabel('State')

# Show plot
plt.tight_layout()
plt.show()
```



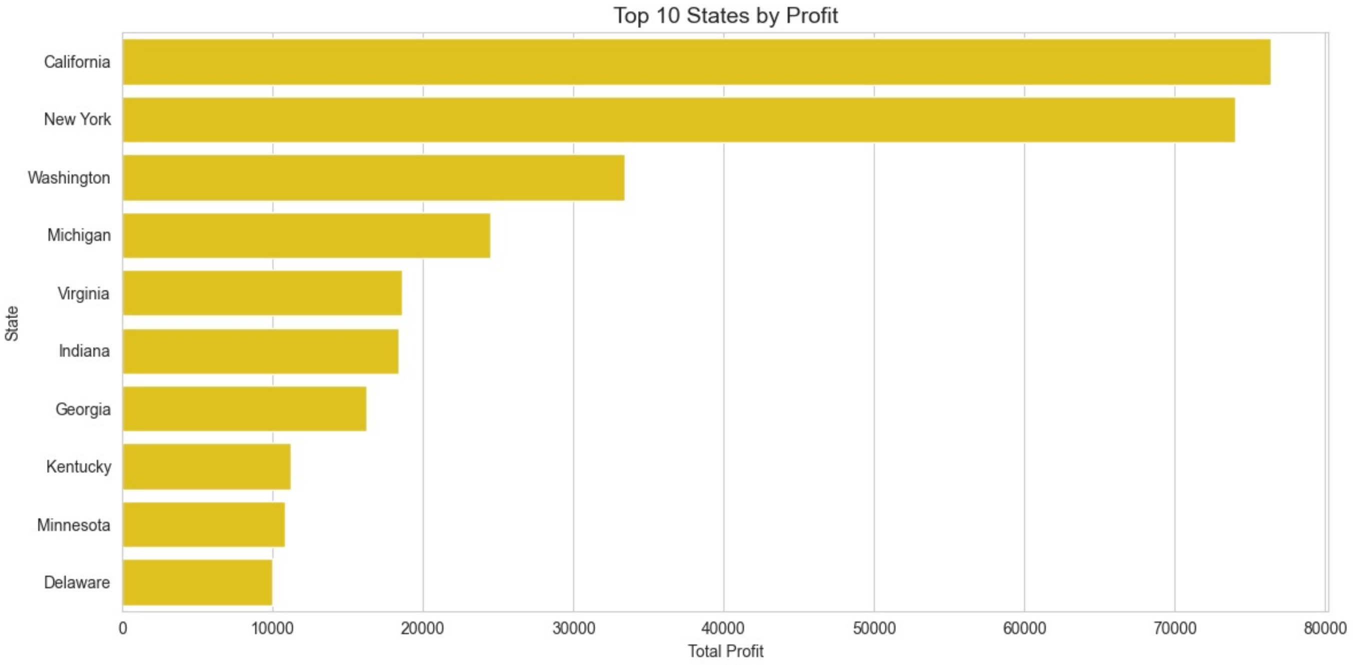
In [25]:

```
# Group data by Product Name and calculate total sales
product_sales = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=product_sales.values, y=product_sales.index, color='gold')

# Add titles and labels
plt.title('Top 10 Products by Sales', fontsize=14)
plt.xlabel('Total Sales')
plt.ylabel('Product Name')

# Show plot
plt.tight_layout()
plt.show()
```



In [26]:

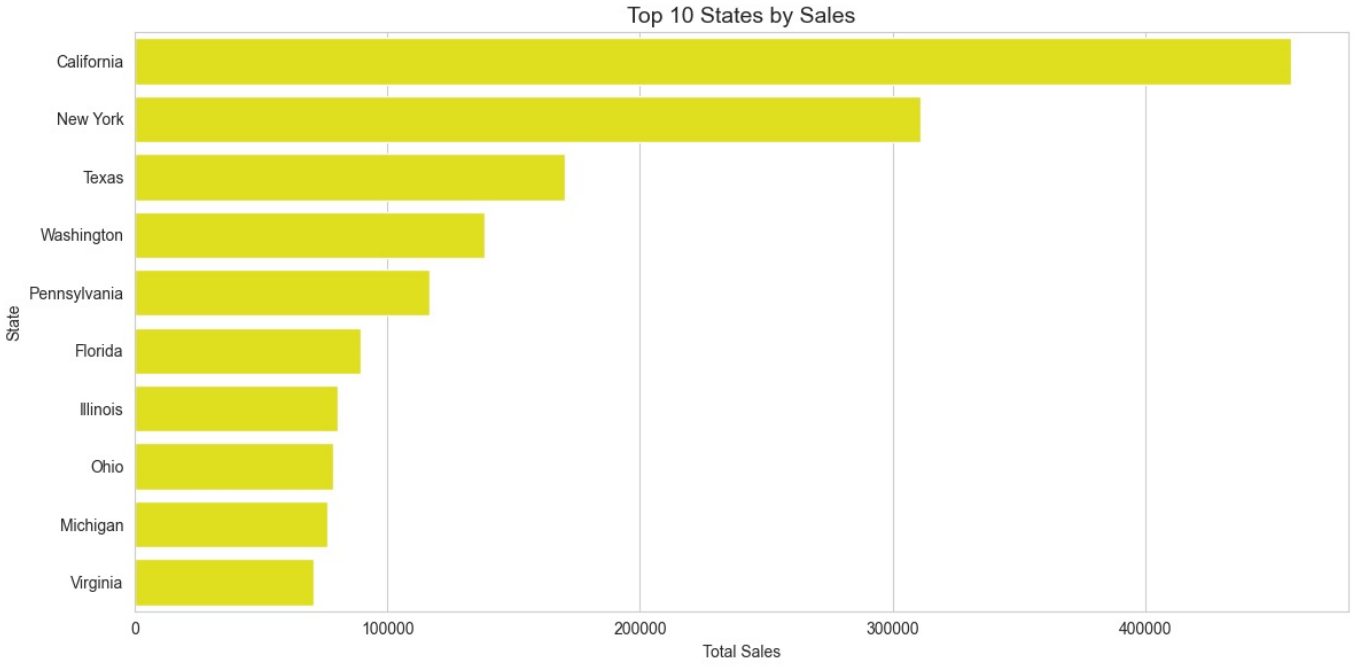
```
# Group data by State and calculate total profit
state_profit = df.groupby('State')['Profit'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=state_profit.values, y=state_profit.index, color='gold')

# Add titles and labels
plt.title('Top 10 States by Profit', fontsize=14)
plt.xlabel('Total Profit')
plt.ylabel('State')

# Show plot
plt.tight_layout()
plt.show()
```



In [27]:

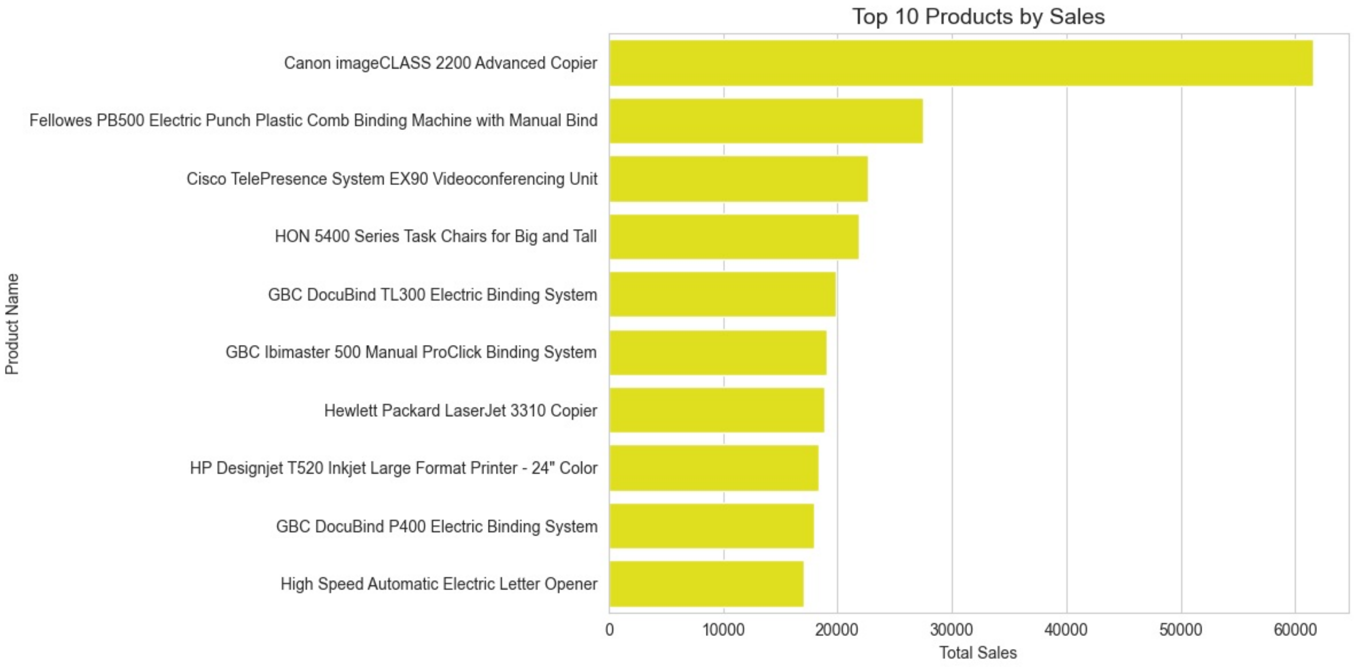
```
# Group data by State and calculate total sales
state_sales = df.groupby('State')['Sales'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=state_sales.values, y=state_sales.index, color='yellow')

# Add titles and labels
```

```
plt.title('Top 10 States by Sales', fontsize=14)
plt.xlabel('Total Sales')
plt.ylabel('State')

# Show plot
plt.tight_layout()
plt.show()
```



In [28]:

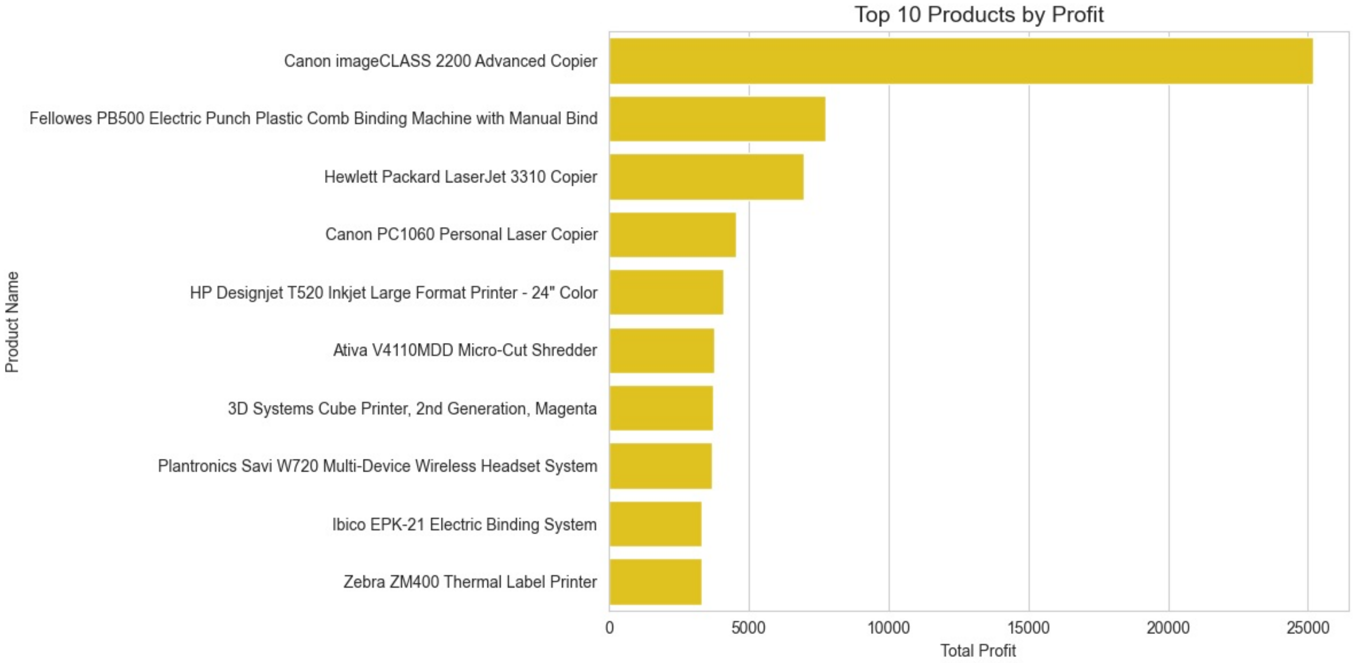
```
# Group data by Product Name and calculate total sales
product_sales = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=product_sales.values, y=product_sales.index, color='yellow')

# Add titles and labels
plt.title('Top 10 Products by Sales', fontsize=14)
plt.xlabel('Total Sales')
plt.ylabel('Product Name')

# Show plot
plt.tight_layout()
plt.show()
```



In [29]:

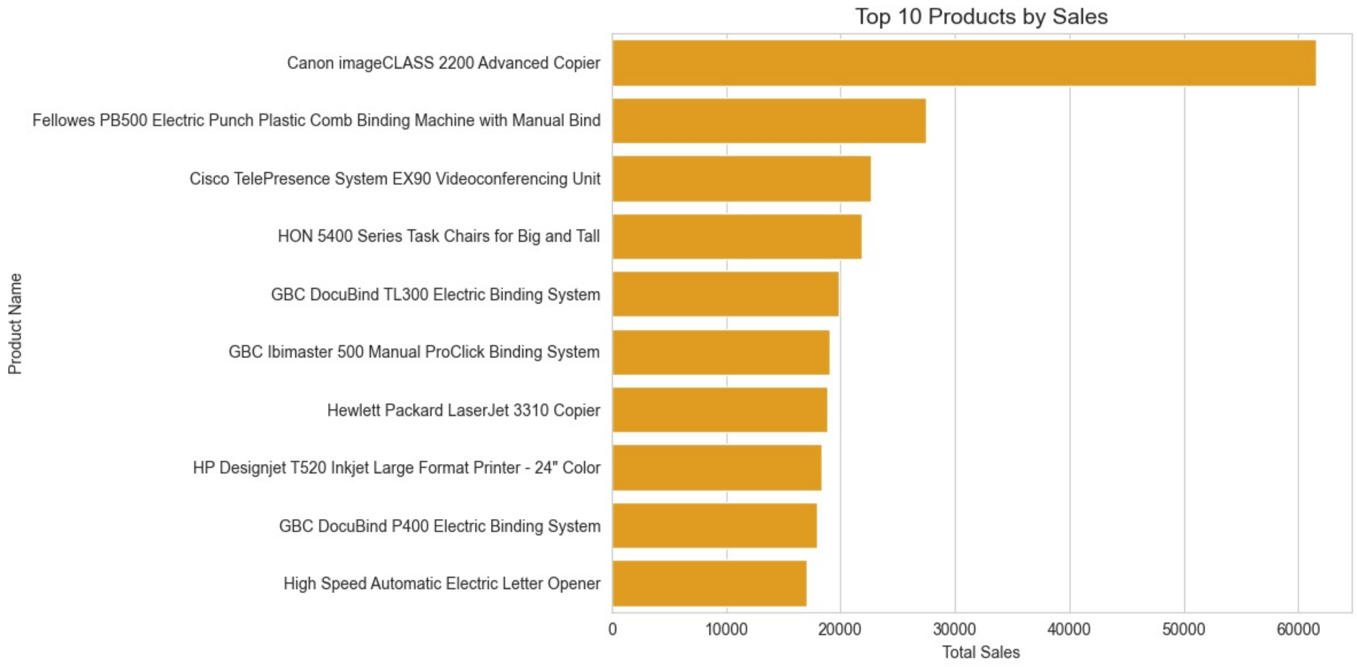
```
# Group data by Product Name and calculate total profit
product_profit = df.groupby('Product Name')['Profit'].sum().sort_values(ascending=False).head(10)
```

```python
# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=product_profit.values, y=product_profit.index, color='gold')

# Add titles and labels
plt.title('Top 10 Products by Profit', fontsize=14)
plt.xlabel('Total Profit')
plt.ylabel('Product Name')

# Show plot
plt.tight_layout()
plt.show()
```



In [30]:

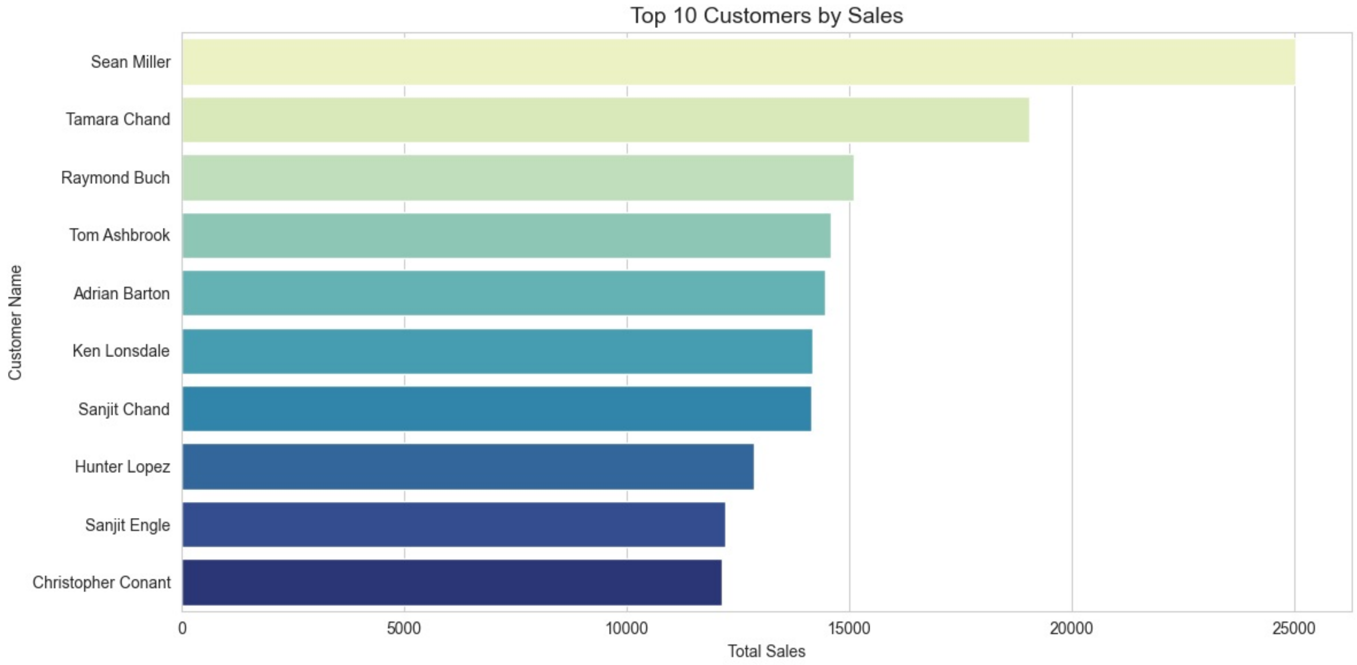
```python
# Group data by Product Name and calculate total sales
product_sales = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=product_sales.values, y=product_sales.index, color='orange')

# Add titles and labels
plt.title('Top 10 Products by Sales', fontsize=14)
plt.xlabel('Total Sales')
plt.ylabel('Product Name')

# Show plot
plt.tight_layout()
plt.show()
```



In [31]:

```
# Group data by Customer Name and calculate total sales
customer_sales = df.groupby('Customer Name')['Sales'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=customer_sales.values, y=customer_sales.index, palette="YlGnBu")

# Add titles and labels
plt.title('Top 10 Customers by Sales', fontsize=14)
plt.xlabel('Total Sales')
plt.ylabel('Customer Name')

# Show plot
plt.tight_layout()
plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_6796\191650058.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=customer_sales.values, y=customer_sales.index, palette="YlGnBu")
```

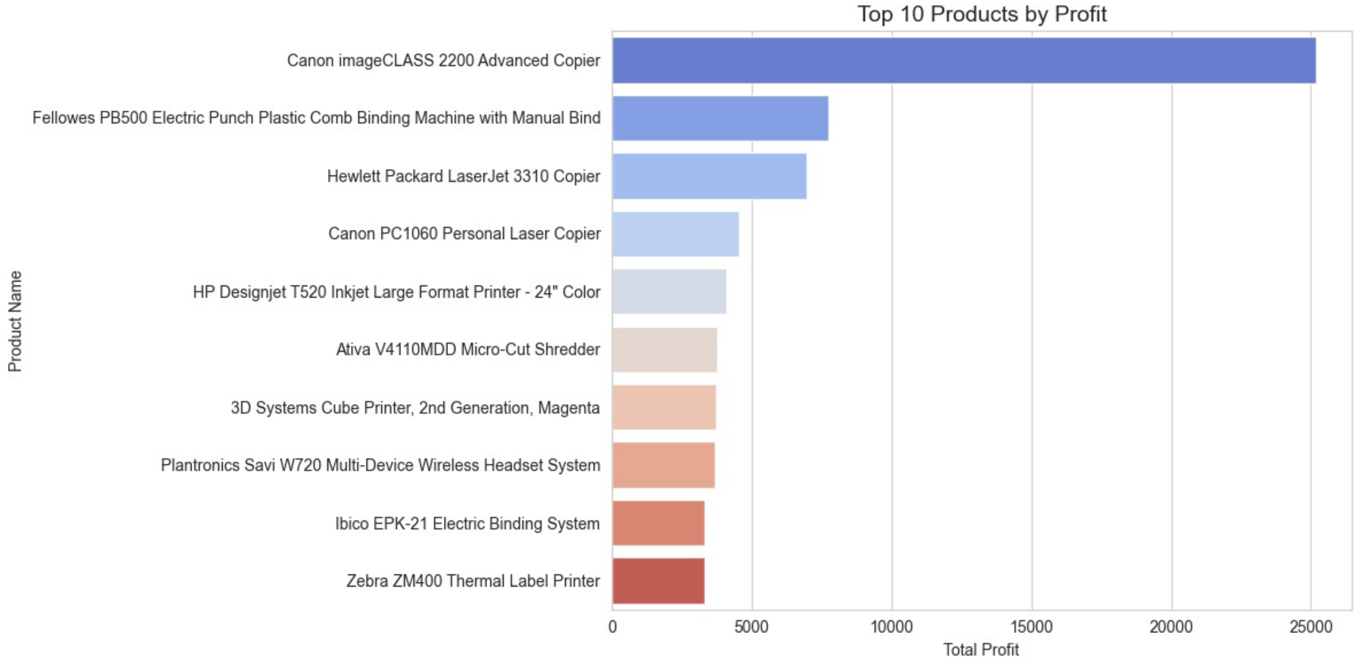


In [32]:

```
# Group data by Product Name and calculate total profit
product_profit = df.groupby('Product Name')['Profit'].sum().sort_values(ascending=False).head(10)

# Set figure size and style
```

```
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

# Create a barplot
sns.barplot(x=product_profit.values, y=product_profit.index, palette="coolwarm")

# Add titles and labels
plt.title('Top 10 Products by Profit', fontsize=14)
plt.xlabel('Total Profit')
plt.ylabel('Product Name')

# Show plot
plt.tight_layout()
plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_6796\1413953051.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=product_profit.values, y=product_profit.index, palette="coolwarm")
```



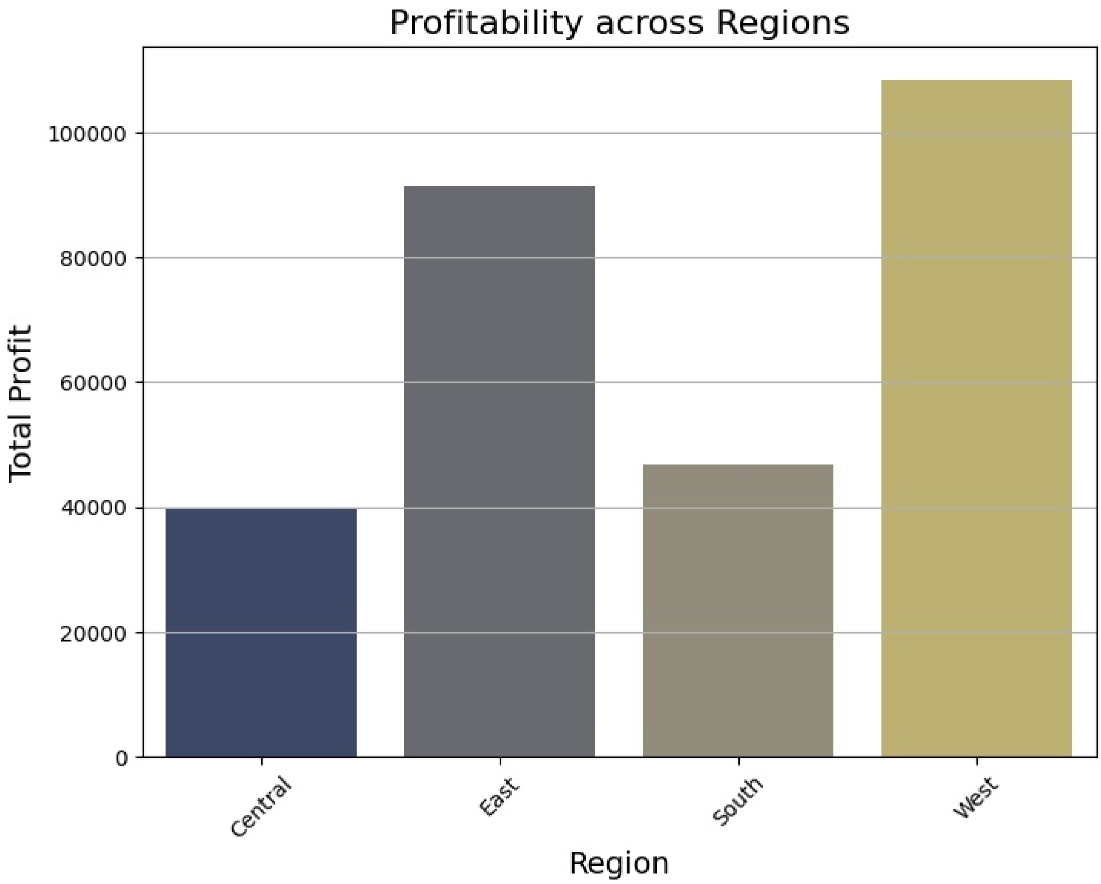
In [3]:

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

# Load your data
file_path = r"Sample Superstore\Sample - Superstore.csv"
df = pd.read_csv(file_path, encoding='latin1')  # Use the same encoding you used before

# Convert 'Order Date' and 'Ship Date' to datetime again
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])
```

In [4]:

```
# Profitability across Regions

region_profit = df.groupby('Region')['Profit'].sum().reset_index()

plt.figure(figsize=(8,6))
sns.barplot(x='Region', y='Profit', data=region_profit, palette='cividis')

plt.title('Profitability across Regions', fontsize=16)
plt.xlabel('Region', fontsize=14)
plt.ylabel('Total Profit', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

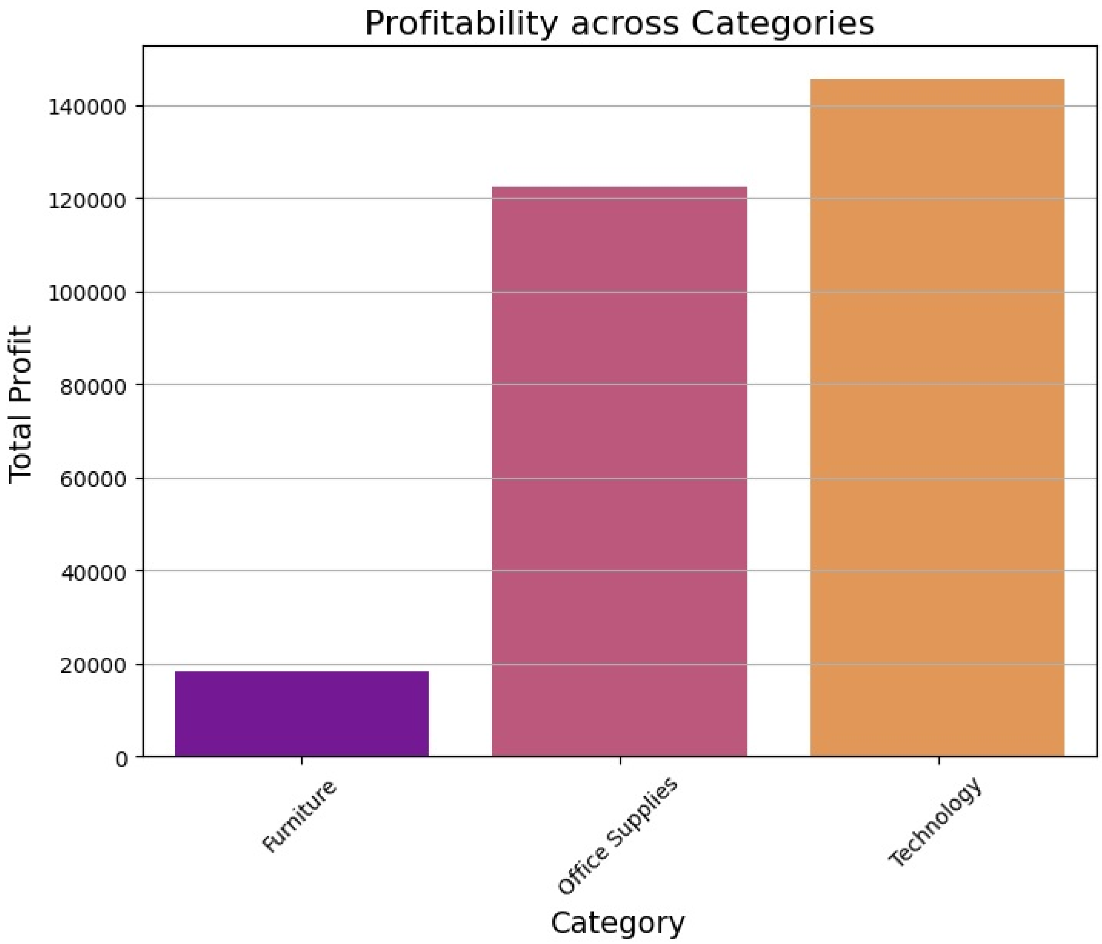
```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\1863692214.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Region', y='Profit', data=region_profit, palette='cividis')
```



In [5]:

```
# Profitability across Categories

category_profit = df.groupby('Category')['Profit'].sum().reset_index()

plt.figure(figsize=(8,6))
sns.barplot(x='Category', y='Profit', data=category_profit, palette='plasma')

plt.title('Profitability across Categories', fontsize=16)
plt.xlabel('Category', fontsize=14)
plt.ylabel('Total Profit', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

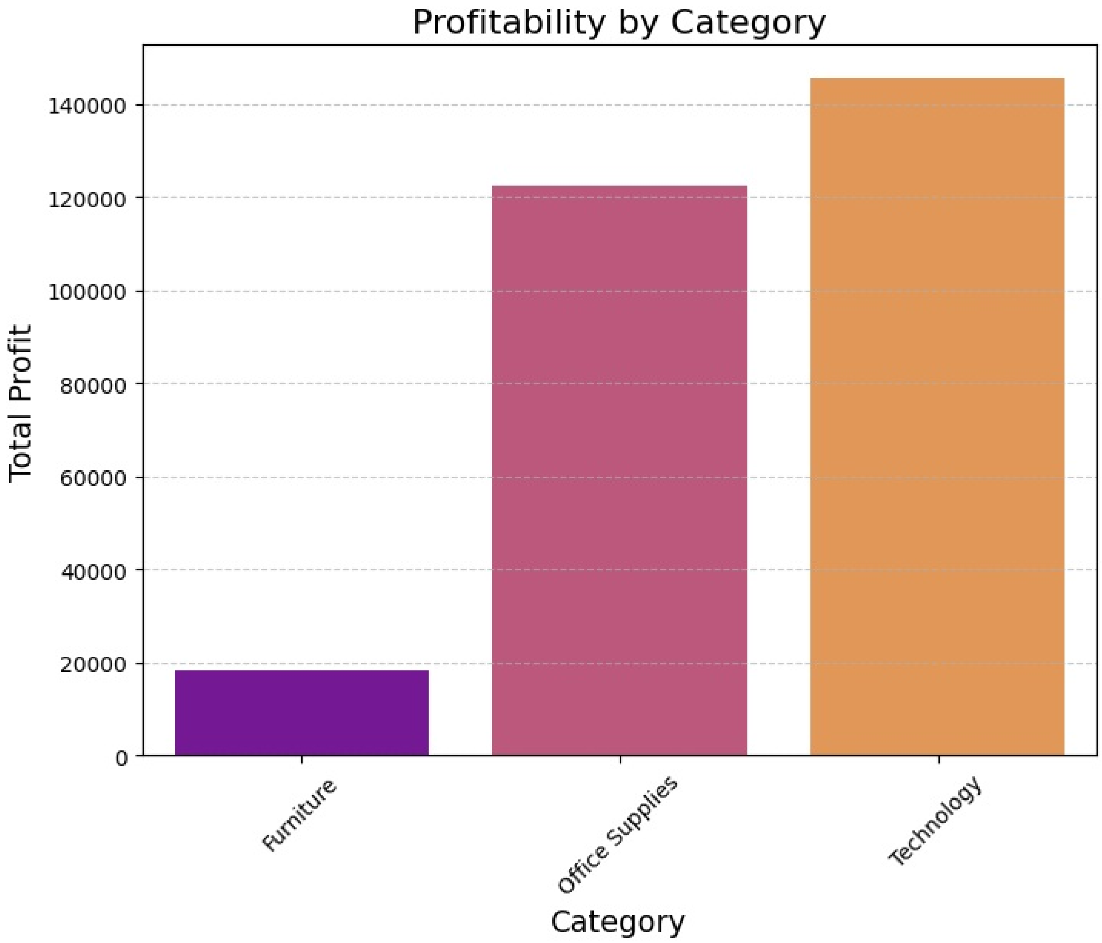
```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\486982649.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Category', y='Profit', data=category_profit, palette='plasma')
```



In [6]:

```
# Profitability across Categories
category_profit = df.groupby('Category')['Profit'].sum().reset_index()

plt.figure(figsize=(8,6))
sns.barplot(x='Category', y='Profit', data=category_profit, palette='plasma')

plt.title('Profitability by Category', fontsize=16)
plt.xlabel('Category', fontsize=14)
plt.ylabel('Total Profit', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```

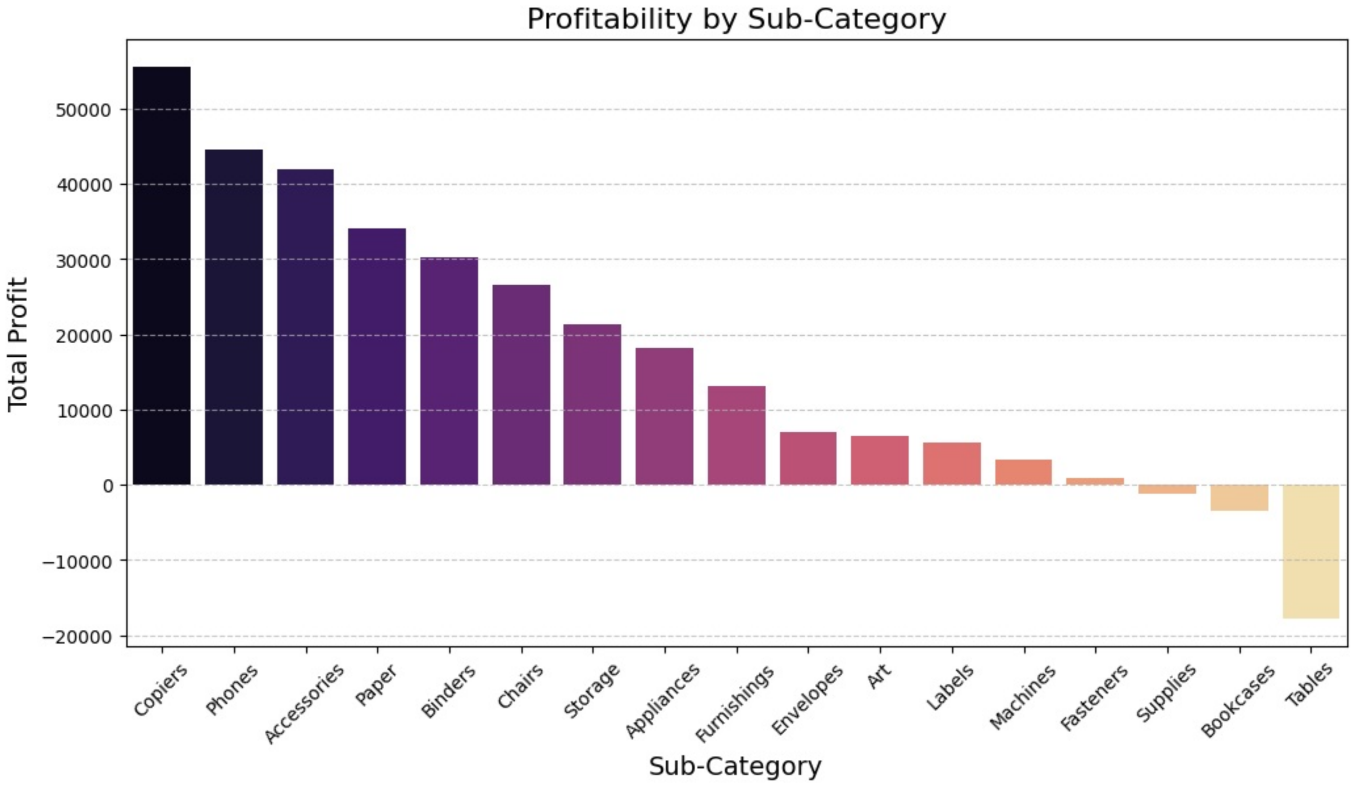
```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\3950630039.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Category', y='Profit', data=category_profit, palette='plasma')
```



In [7]:

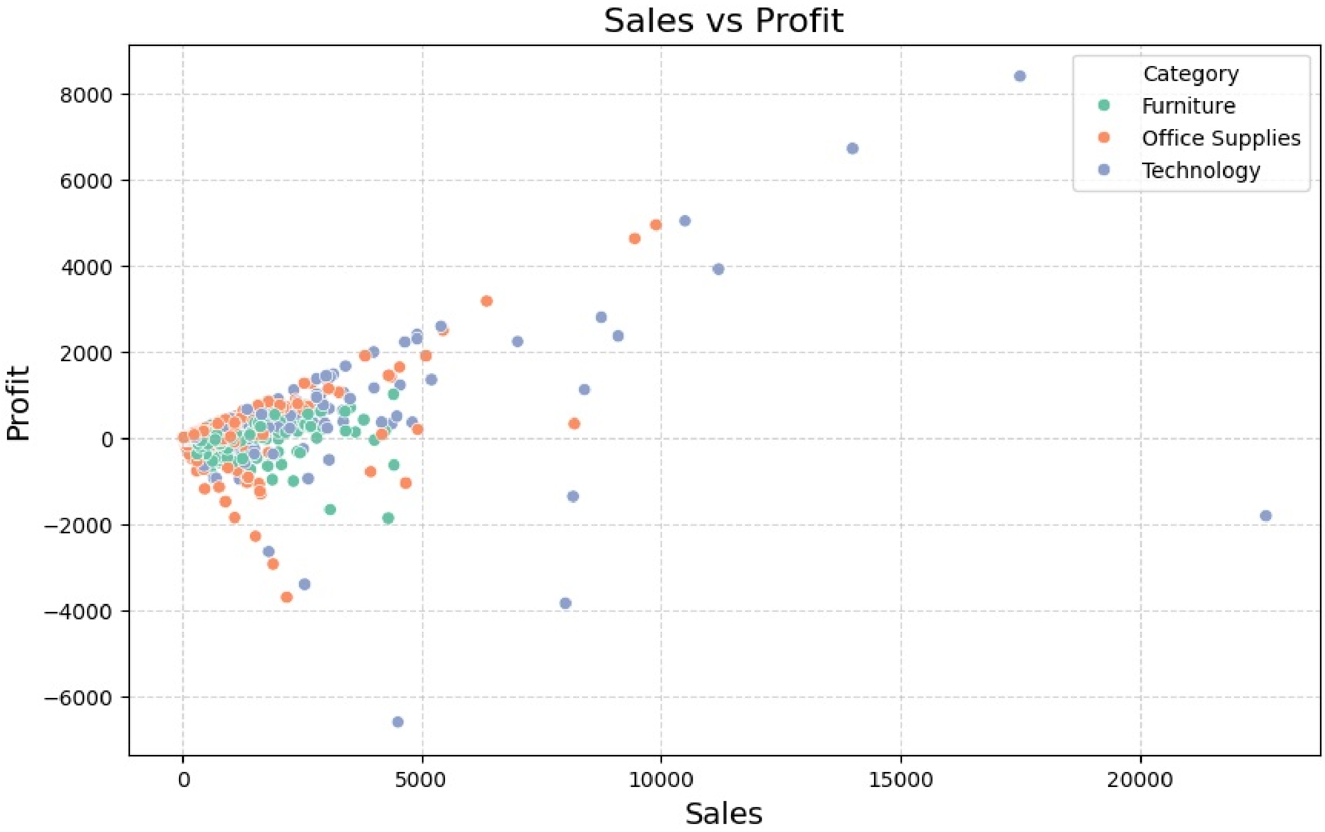
```
# Profitability across Sub-Categories
subcategory_profit = df.groupby('Sub-Category')['Profit'].sum().reset_index().sort_values(by='Profit', ascendin

plt.figure(figsize=(12,6))
sns.barplot(x='Sub-Category', y='Profit', data=subcategory_profit, palette='magma')

plt.title('Profitability by Sub-Category', fontsize=16)
plt.xlabel('Sub-Category', fontsize=14)
plt.ylabel('Total Profit', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\917434741.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Sub-Category', y='Profit', data=subcategory_profit, palette='magma')
```



In [9]:

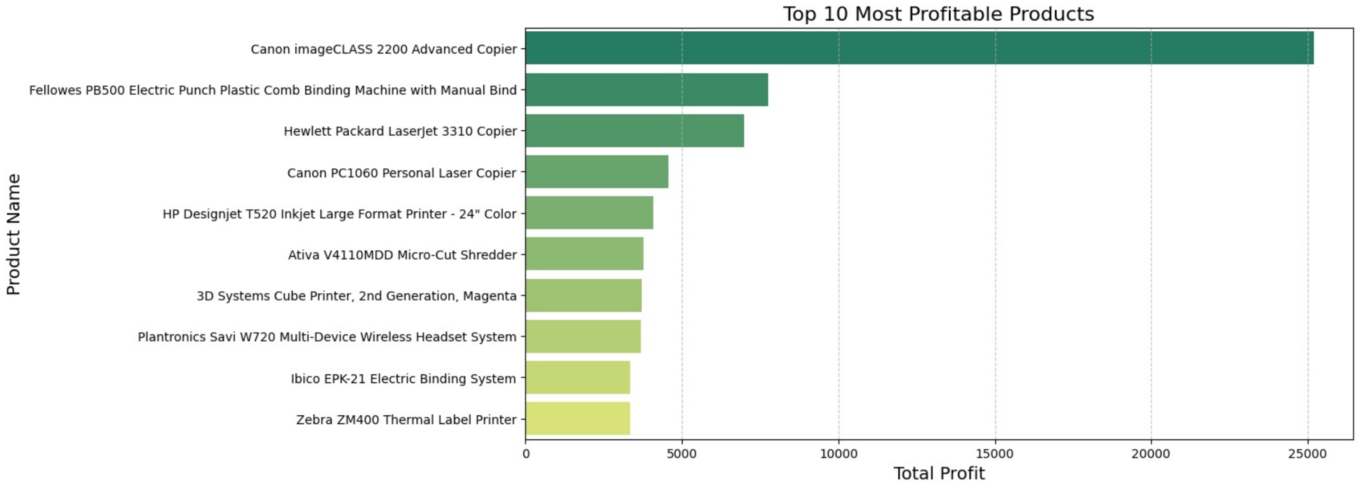
```python
# Sales vs Profit Scatter Plot
plt.figure(figsize=(10,6))
sns.scatterplot(x='Sales', y='Profit', data=df, hue='Category', palette='Set2')

plt.title('Sales vs Profit', fontsize=16)
plt.xlabel('Sales', fontsize=14)
plt.ylabel('Profit', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title='Category')

plt.show()
```



In [10]:

```
# Top 10 Most Profitable Products
top_products = df.groupby('Product Name')['Profit'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_products.values, y=top_products.index, palette='summer')

plt.title('Top 10 Most Profitable Products', fontsize=16)
plt.xlabel('Total Profit', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)

plt.show()
```

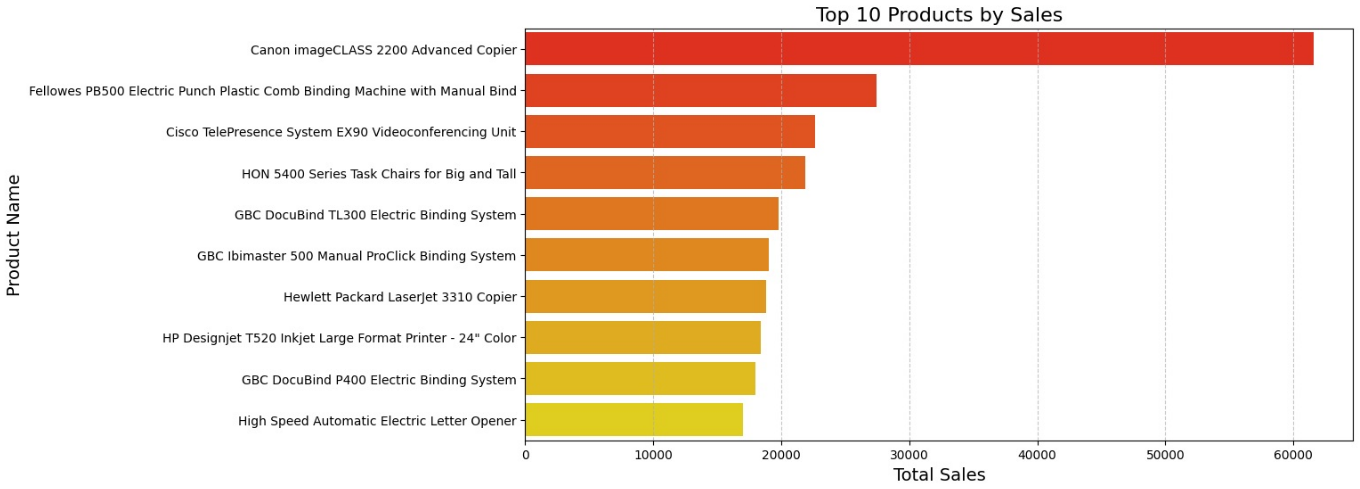
```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\2936808375.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_products.values, y=top_products.index, palette='summer')
```



In [11]:

```
# Top 10 Products by Sales
top_sales_products = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_sales_products.values, y=top_sales_products.index, palette='autumn')

plt.title('Top 10 Products by Sales', fontsize=16)
plt.xlabel('Total Sales', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.show()
```

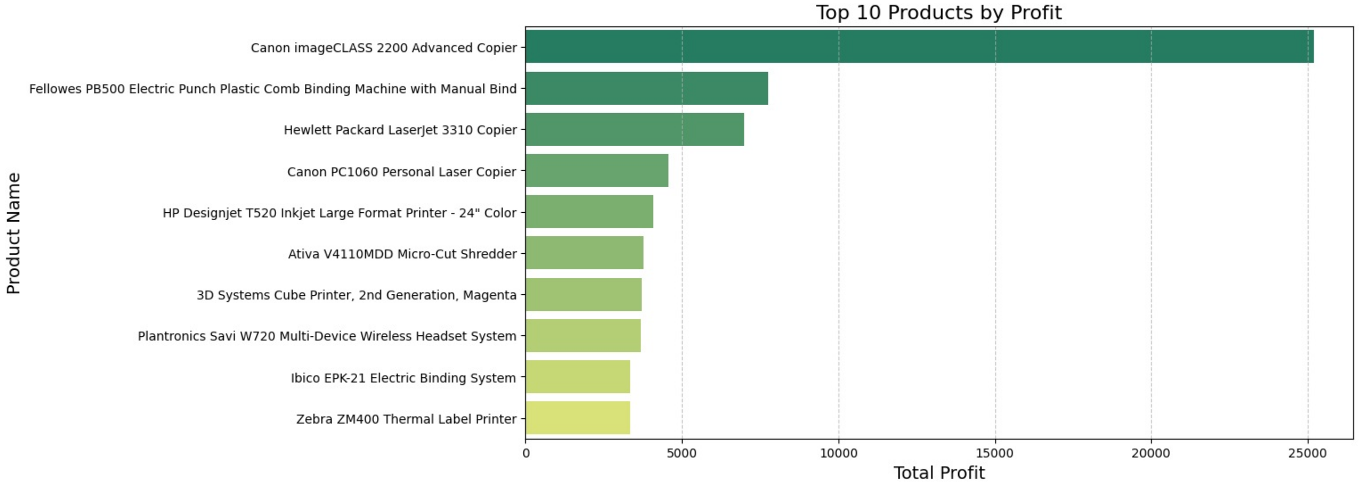
```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\3949513252.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_sales_products.values, y=top_sales_products.index, palette='autumn')
```



In [12]:

```
# Top 10 Products by Profit
top_profit_products = df.groupby('Product Name')['Profit'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_profit_products.values, y=top_profit_products.index, palette='summer')

plt.title('Top 10 Products by Profit', fontsize=16)
plt.xlabel('Total Profit', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)

plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_11960\1562844007.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_profit_products.values, y=top_profit_products.index, palette='summer')
```



In [2]:

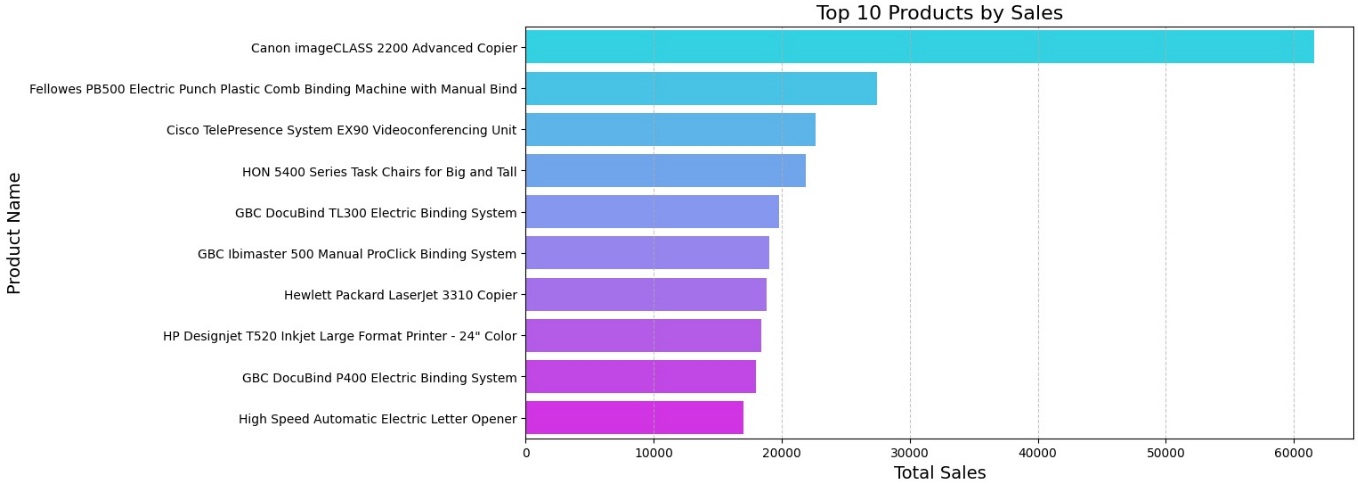
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = r"Sample Superstore\Sample - Superstore.csv"
df = pd.read_csv(file_path, encoding='latin1')

# Convert Order Date and Ship Date to datetime
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])
```

In [3]:

```
# Top 10 Products by Sales
top_sales_products = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
```

```
sns.barplot(x=top_sales_products.values, y=top_sales_products.index, palette='cool')

plt.title('Top 10 Products by Sales', fontsize=16)
plt.xlabel('Total Sales', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)

plt.show()
```

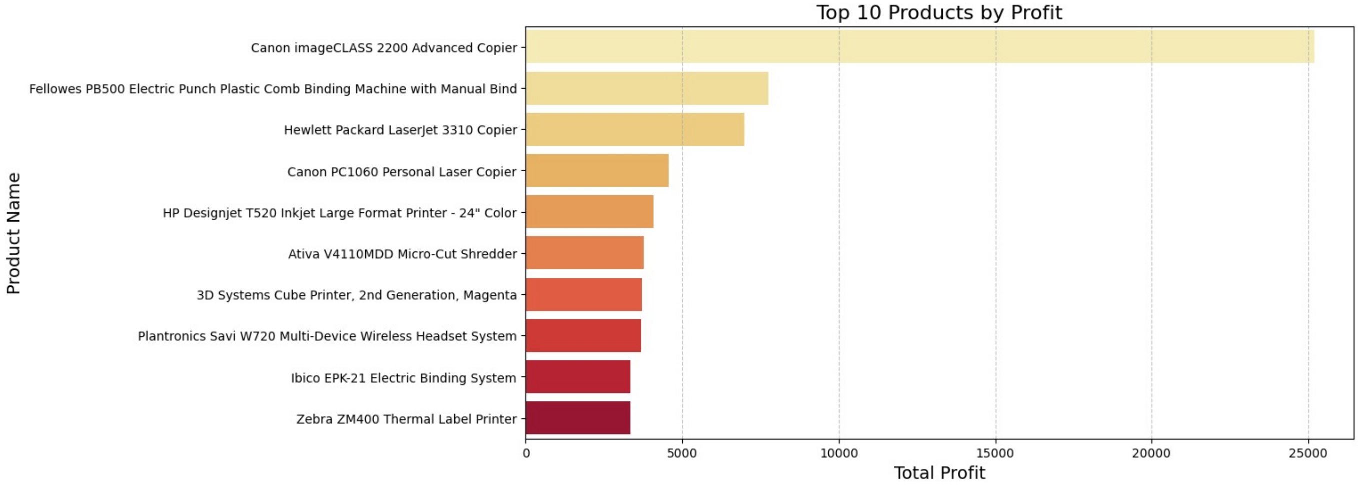
```
C:\Users\anand\AppData\Local\Temp\ipykernel_15532\1541101452.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_sales_products.values, y=top_sales_products.index, palette='cool')
```



In [4]:

```
# Top 10 Products by Profit
top_profit_products = df.groupby('Product Name')['Profit'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_profit_products.values, y=top_profit_products.index, palette='YlOrRd')

plt.title('Top 10 Products by Profit', fontsize=16)
plt.xlabel('Total Profit', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)

plt.show()
```

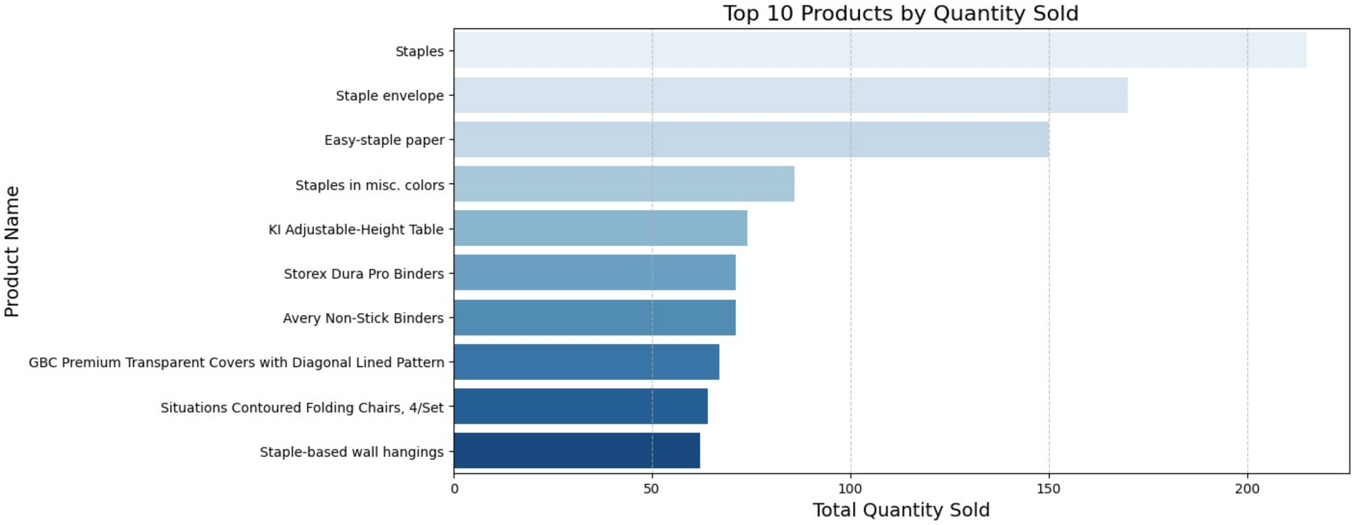
```
C:\Users\anand\AppData\Local\Temp\ipykernel_15532\3972120076.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_profit_products.values, y=top_profit_products.index, palette='YlOrRd')
```



In [5]:

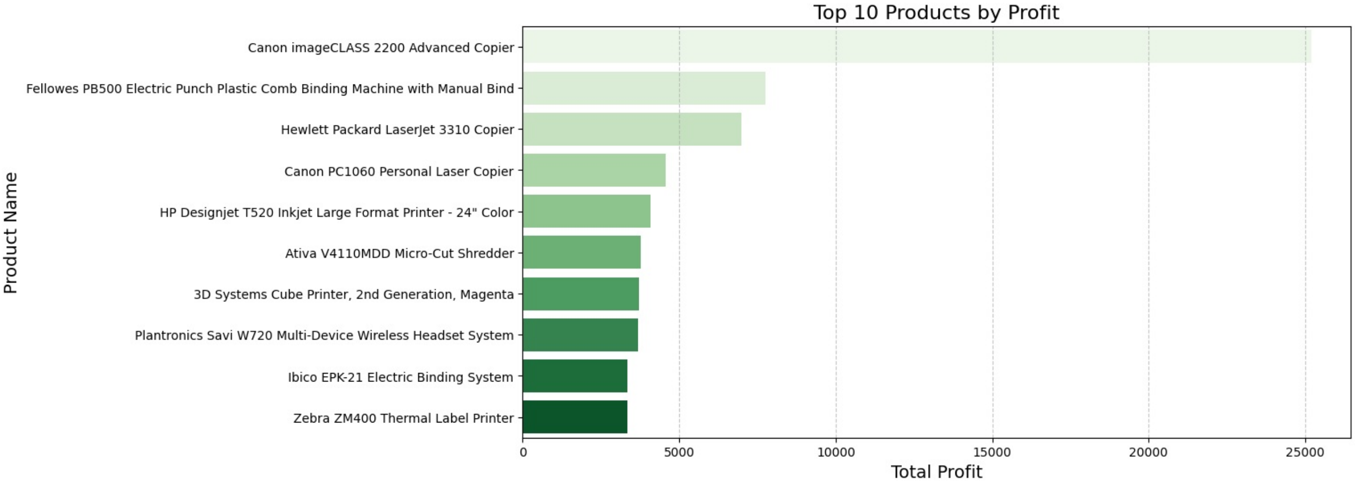
```
# Top 10 Products by Quantity Sold
top_quantity_products = df.groupby('Product Name')['Quantity'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_quantity_products.values, y=top_quantity_products.index, palette='Blues')

plt.title('Top 10 Products by Quantity Sold', fontsize=16)
plt.xlabel('Total Quantity Sold', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_15532\4187374170.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_quantity_products.values, y=top_quantity_products.index, palette='Blues')
```

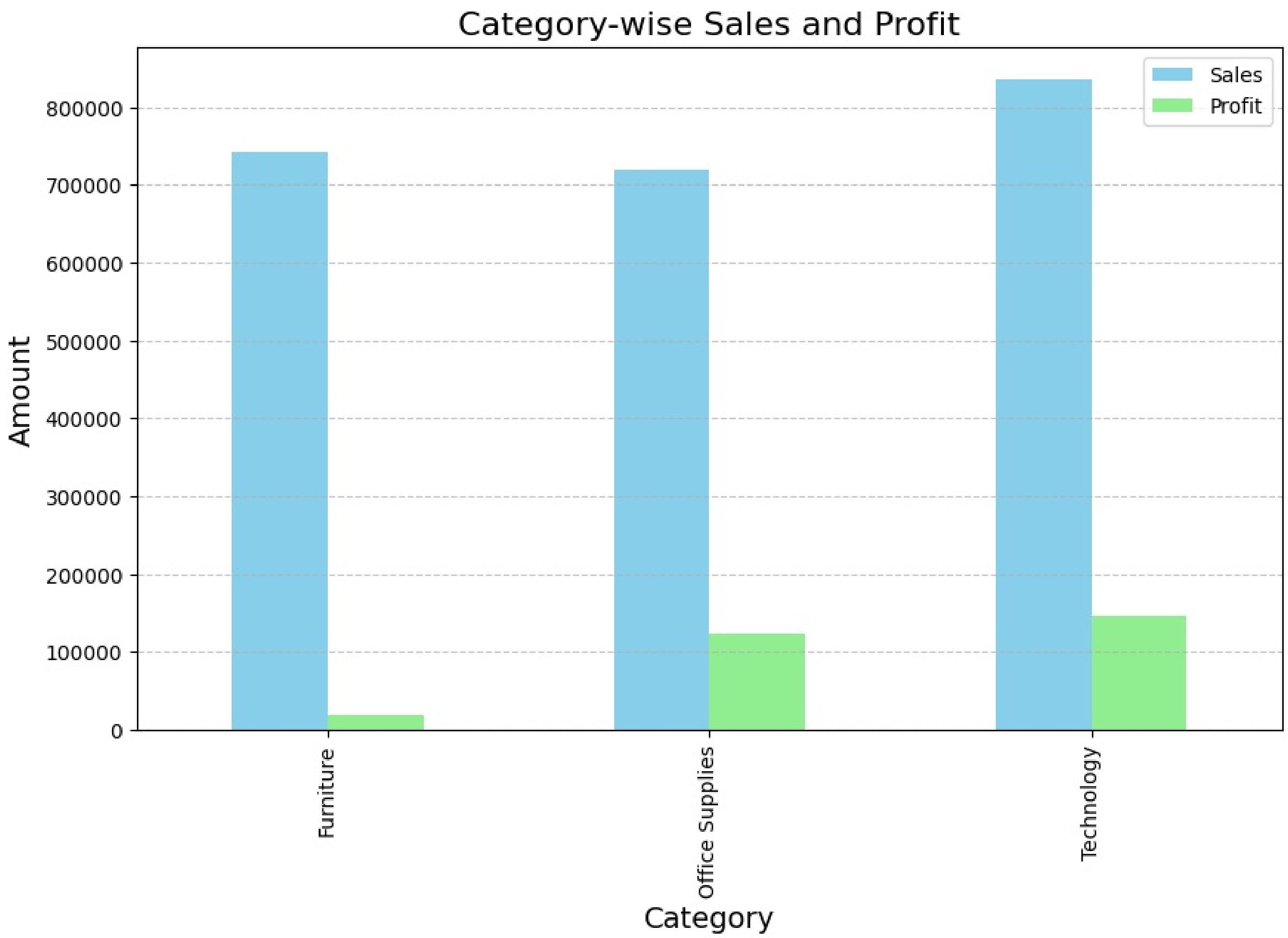


In [6]:
```
# Top 10 Products by Profit
top_profit_products = df.groupby('Product Name')['Profit'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,6))
sns.barplot(x=top_profit_products.values, y=top_profit_products.index, palette='Greens')

plt.title('Top 10 Products by Profit', fontsize=16)
plt.xlabel('Total Profit', fontsize=14)
plt.ylabel('Product Name', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)

plt.show()
```

```
C:\Users\anand\AppData\Local\Temp\ipykernel_15532\4057181923.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_profit_products.values, y=top_profit_products.index, palette='Greens')
```

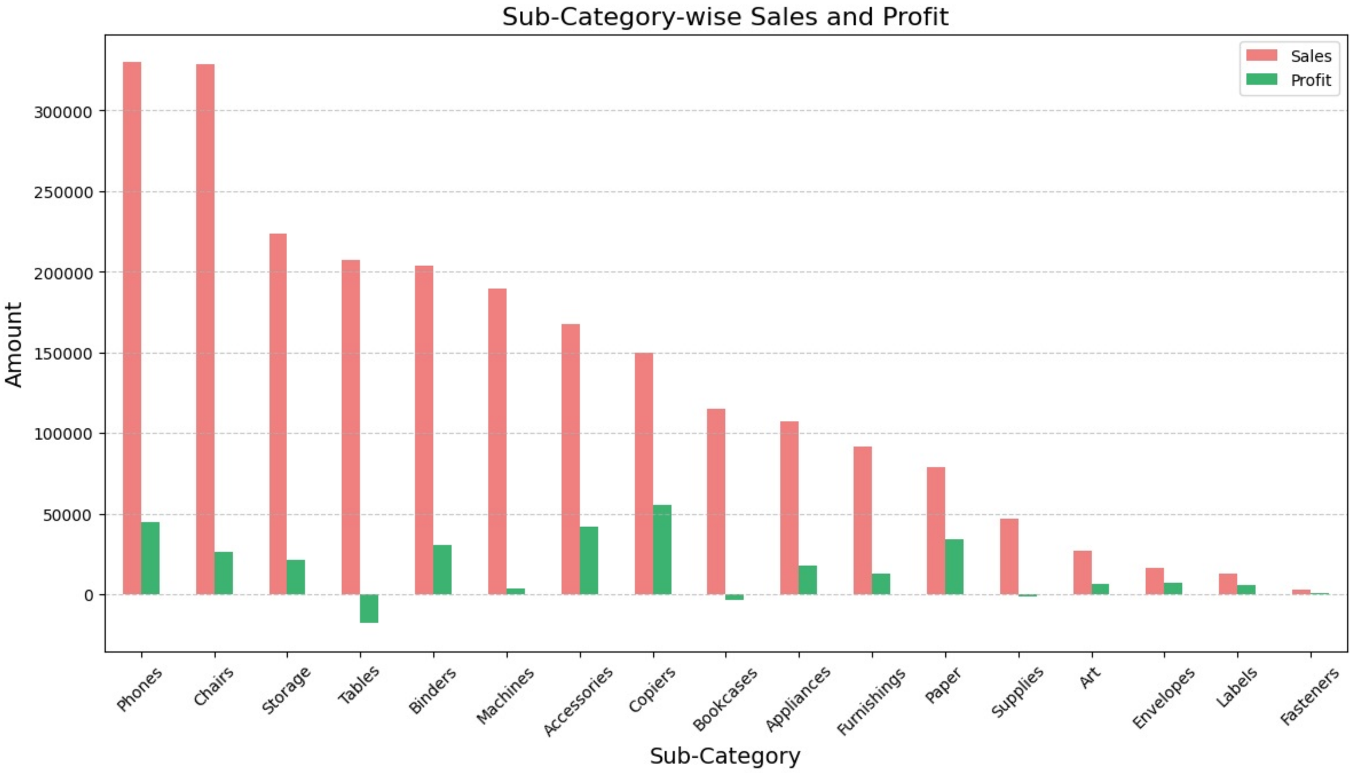


In [7]:
```
# Category-wise Sales and Profit
category_sales_profit = df.groupby('Category')[['Sales', 'Profit']].sum().reset_index()

category_sales_profit.plot(x='Category', y=['Sales', 'Profit'], kind='bar', figsize=(10,6), color=['skyblue', '

plt.title('Category-wise Sales and Profit', fontsize=16)
plt.xlabel('Category', fontsize=14)
plt.ylabel('Amount', fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



In [8]:

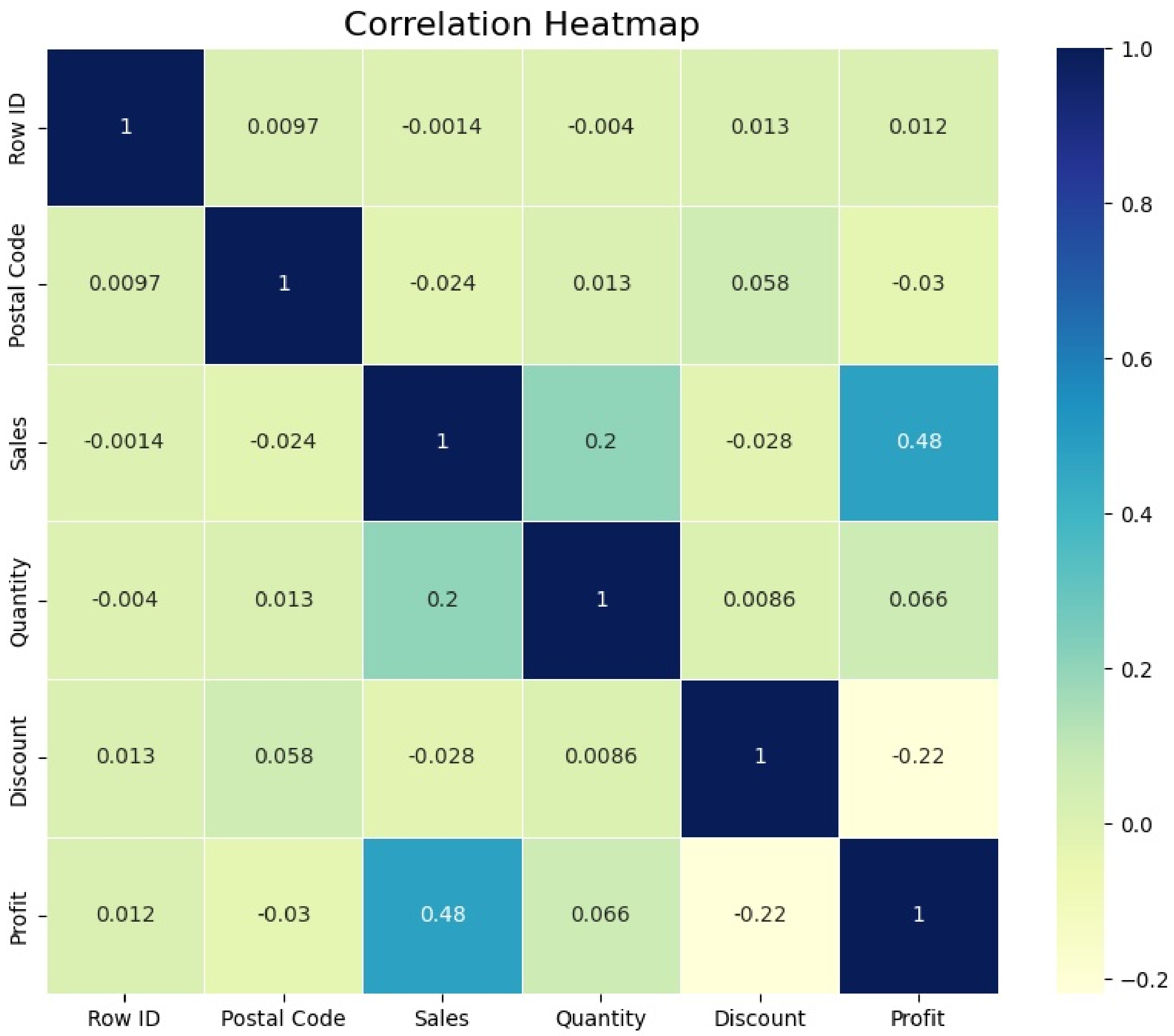
```
# Sub-Category-wise Sales and Profit
subcategory_sales_profit = df.groupby('Sub-Category')[['Sales', 'Profit']].sum().sort_values(by='Sales', ascendi

subcategory_sales_profit.plot(x='Sub-Category', y=['Sales', 'Profit'], kind='bar', figsize=(14,7), color=['ligh

plt.title('Sub-Category-wise Sales and Profit', fontsize=16)
plt.xlabel('Sub-Category', fontsize=14)
plt.ylabel('Amount', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



In [10]:

```
# Correlation Heatmap (Only Numeric Columns)
```

```
plt.figure(figsize=(10,8))
corr = df.select_dtypes(include=['float64', 'int64']).corr()

sns.heatmap(corr, annot=True, cmap='YlGnBu', linewidths=0.5)
plt.title('Correlation Heatmap', fontsize=16)

plt.show()
```



Correlation Heatmap

|  | Row ID | Postal Code | Sales | Quantity | Discount | Profit |
|---|---|---|---|---|---|---|
| Row ID | 1 | 0.0097 | -0.0014 | -0.004 | 0.013 | 0.012 |
| Postal Code | 0.0097 | 1 | -0.024 | 0.013 | 0.058 | -0.03 |
| Sales | -0.0014 | -0.024 | 1 | 0.2 | -0.028 | 0.48 |
| Quantity | -0.004 | 0.013 | 0.2 | 1 | 0.0086 | 0.066 |
| Discount | 0.013 | 0.058 | -0.028 | 0.0086 | 1 | -0.22 |
| Profit | 0.012 | -0.03 | 0.48 | 0.066 | -0.22 | 1 |

```
In [11]: # Pivot Table: Sales and Profit by Region and Category
pivot_table = pd.pivot_table(df, values=['Sales', 'Profit'],
                             index='Region',
                             columns='Category',
                             aggfunc='sum',
                             fill_value=0)

print(pivot_table)
```

```
              Profit                                    Sales                  \
Category   Furniture Office Supplies  Technology    Furniture Office Supplies
Region
Central   -2871.0494       8879.9799  33697.4320  163797.1638      167026.415
East       3046.1658      41014.5791  47462.0351  208291.2040      205516.055
South      6771.2061      19986.3928  19991.8314  117298.6840      125651.313
West      11504.9503      52609.8490  44303.6496  252612.7435      220853.249


Category  Technology
Region
Central   170416.312
East      264973.981
South     148771.908
West      251991.832
```

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = r"Sample Superstore\Sample - Superstore.csv"
df = pd.read_csv(file_path, encoding='latin1')

# Convert Order Date and Ship Date to datetime
df['Order Date'] = pd.to_datetime(df['Order Date'])
```

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

In [3]:

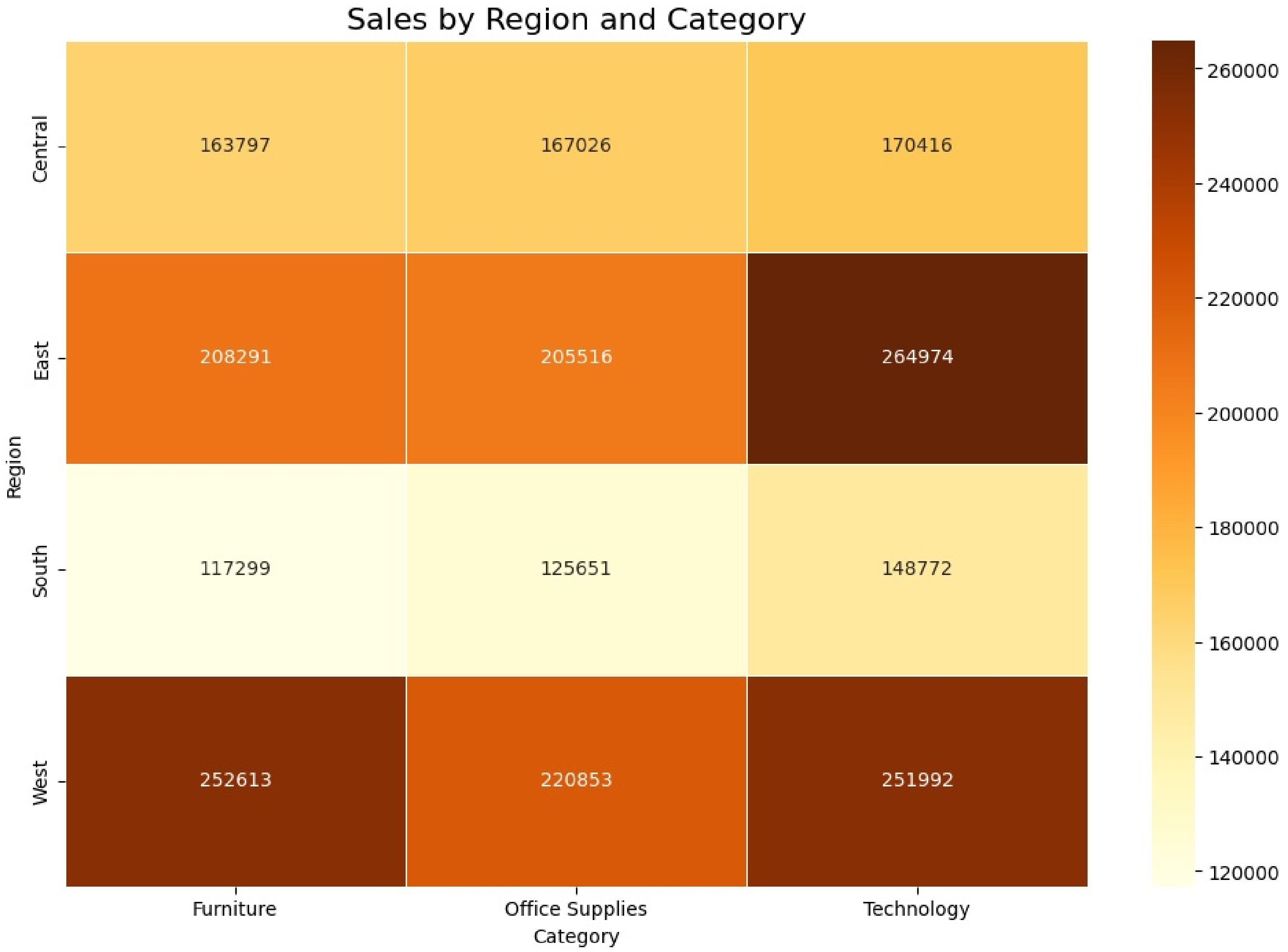
```
pivot_table = df.pivot_table(values='Sales', index='Region', columns='Category', aggfunc='sum')
pivot_table
```

Out[3]:

| Category | Furniture | Office Supplies | Technology |
|---|---|---|---|
| **Region** | | | |
| **Central** | 163797.1638 | 167026.415 | 170416.312 |
| **East** | 208291.2040 | 205516.055 | 264973.981 |
| **South** | 117298.6840 | 125651.313 | 148771.908 |
| **West** | 252612.7435 | 220853.249 | 251991.832 |

In [4]:

```
plt.figure(figsize=(12,8))
sns.heatmap(pivot_table, annot=True, fmt=".0f", cmap='YlOrBr', linewidths=0.5)
plt.title('Sales by Region and Category', fontsize=16)
plt.show()
```



In [ ]: