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

In [2]:
```
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 5)
```

In [3]:
```
# Load dataset
df = pd.read_csv(r"C:\Users\HP\Downloads\archive (2)\train.csv")
df
```

Out[3]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segme
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consum
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consum
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corpora
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consum
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consum
...	...	...	...	...	...	...	...	
9795	9796	CA-2017-125920	21/05/2017	28/05/2017	Standard Class	SH-19975	Sally Hughsby	Corpora
9796	9797	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corpora
9797	9798	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corpora
9798	9799	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corpora
9799	9800	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corpora

9800 rows × 18 columns

In [4]:
```
# Drop duplicates
df.drop_duplicates(inplace=True)
```

In [5]:
```
# Clean the dataset
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
df['order_date'] = pd.to_datetime(df['order_date'], dayfirst=True)
df['ship_date'] = pd.to_datetime(df['ship_date'], dayfirst=True)
```

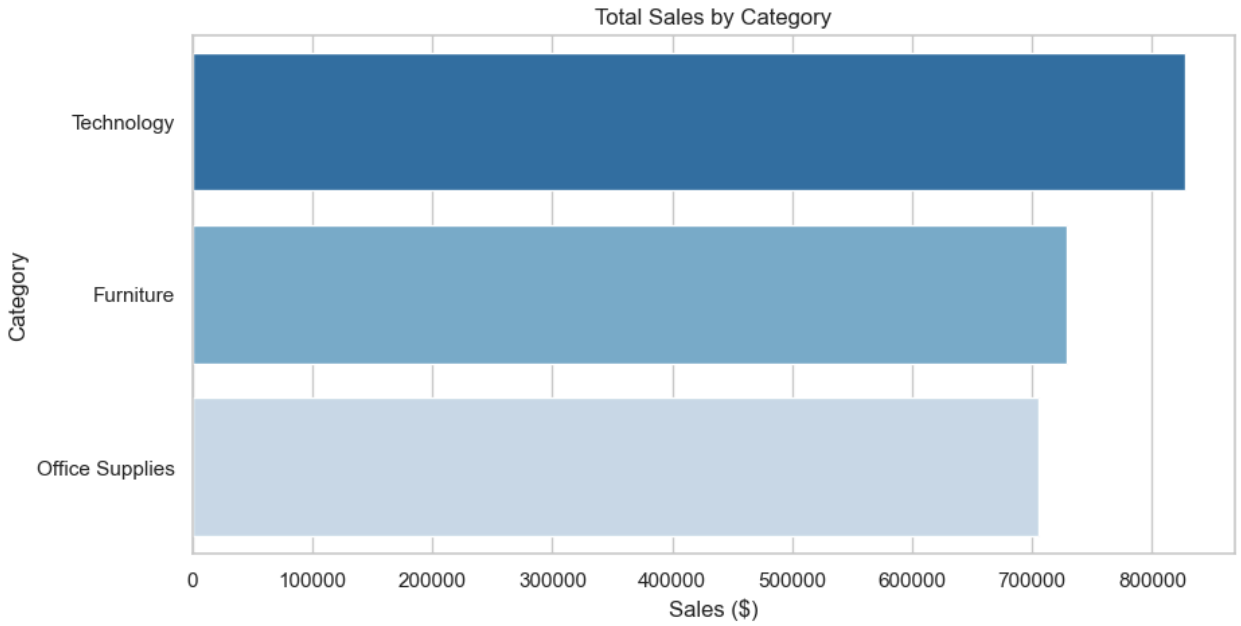
In [6]:
```
# Check for missing values
df.isnull().sum()
```

Out[6]:
```
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      11
region            0
product_id        0
category          0
sub-category      0
product_name      0
sales             0
dtype: int64
```

In [7]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [10]:
```
#sales by category
category_sales = df.groupby('category')['sales'].sum().sort_values(ascending=F

sns.barplot(x=category_sales.values, y=category_sales.index, palette='Blues_r'
plt.title("Total Sales by Category")
plt.xlabel("Sales ($)")
plt.ylabel("Category")
plt.show()
```

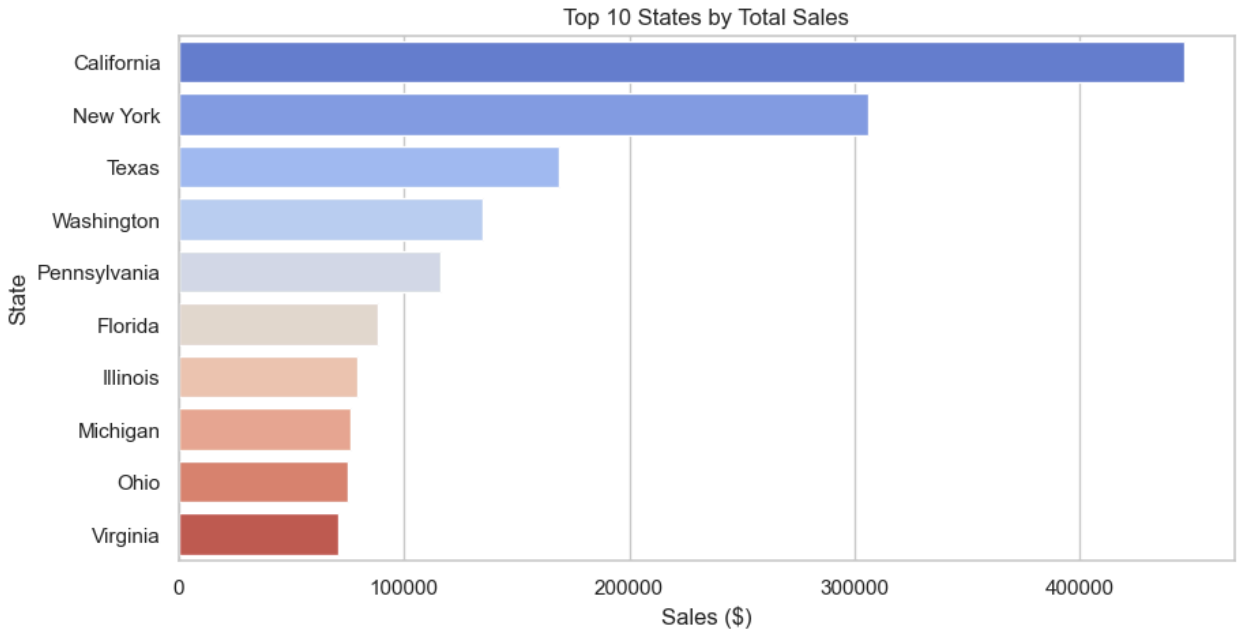


Sales by Category – Key Insights

- Technology leads in sales – high demand and high-value items.
- Furniture has moderate sales – room for growth.
- Office Supplies has lowest sales – frequent but low-value purchases.

In [11]:
```
#"Top 10 States by Total Sales
top_states = df.groupby('state')['sales'].sum().sort_values(ascending=False).h

sns.barplot(x=top_states.values, y=top_states.index, palette='coolwarm')
plt.title("Top 10 States by Total Sales")
plt.xlabel("Sales ($)")
plt.ylabel("State")
plt.show()
```

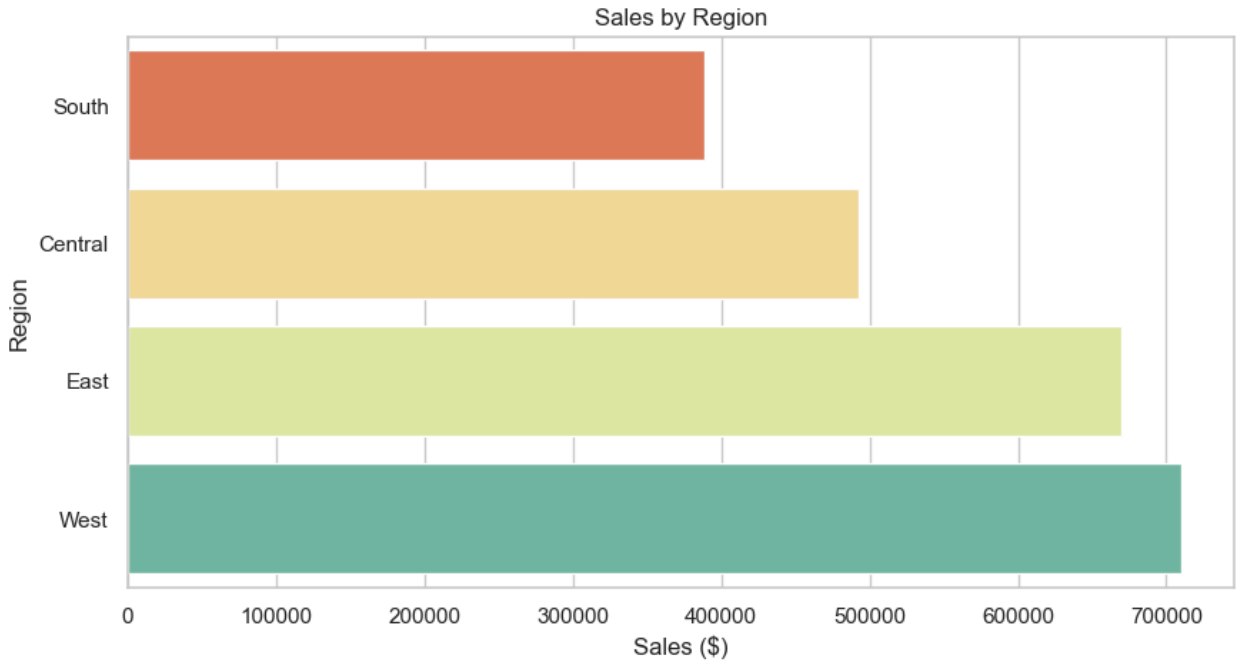


Top 10 States by Sales – Key Insights

- California dominates sales by a large margin — it's the primary revenue driver.
- New York, Texas, and Washington follow with strong performance.
- Other top states (e.g., Michigan, Pennsylvania) contribute steadily but with lower volume.

In [13]:
```
#Sales by Region
region_sales = df.groupby('region')['sales'].sum().sort_values()

sns.barplot(x=region_sales.values, y=region_sales.index, palette='Spectral')
plt.title("Sales by Region")
plt.xlabel("Sales ($)")
plt.ylabel("Region")
plt.show()
```



Sales by Region – Key Insights

- West region leads in total sales, followed by East.
- Central and South regions have lower sales comparatively.
- Strengthen marketing in Central and South to balance regional performance.

🔷 **Summary:**

- Technology is the top-selling category.
- California leads all states in total sales.
- West region outperforms others in revenue.

Recommendation: Boost marketing in underperforming regions and expand high-profit categories.

In [ ]: