In [1]:

```
# Step 1: Imported libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
%matplotlib inline
```

```
C:\Users\pramo\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:60: Use
rWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.
3.5' currently installed).
  from pandas.core import (
```

In [2]:

```
# Step 2: Loaded dataset
df = pd.read_csv('Sample_Superstore.csv', encoding='latin1')
df.head()
```

Out[2]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	C
0	1	CA-2016-152156	2016/08/11	2016/11/11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hender
1	2	CA-2016-152156	2016/11/08	2016/11/11	Second Class	CG-12520	Claire Gute	Consumer	United States	Hender
2	3	CA-2016-138688	2016/06/12	2016/06/16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Ang
3	4	US-2015-108966	2015/10/11	2015/10/18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder
4	5	US-2015-108966	2015/10/11	2015/10/18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder

5 rows × 21 columns

In [3]:

```
# Step 3: Checked data
df.info()
df.describe()
```

```
<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[3]:

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

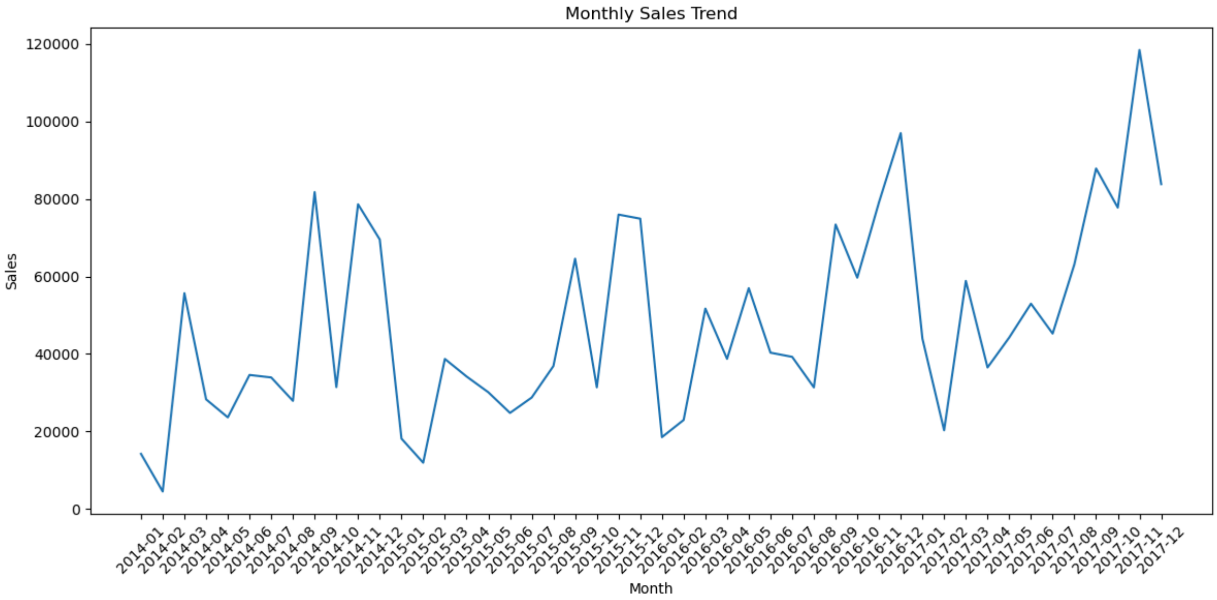
In [4]:

```
# Step 4: Fixed Order Date column and extract Month
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Month'] = df['Order Date'].dt.to_period('M').astype(str)
```

In [5]:

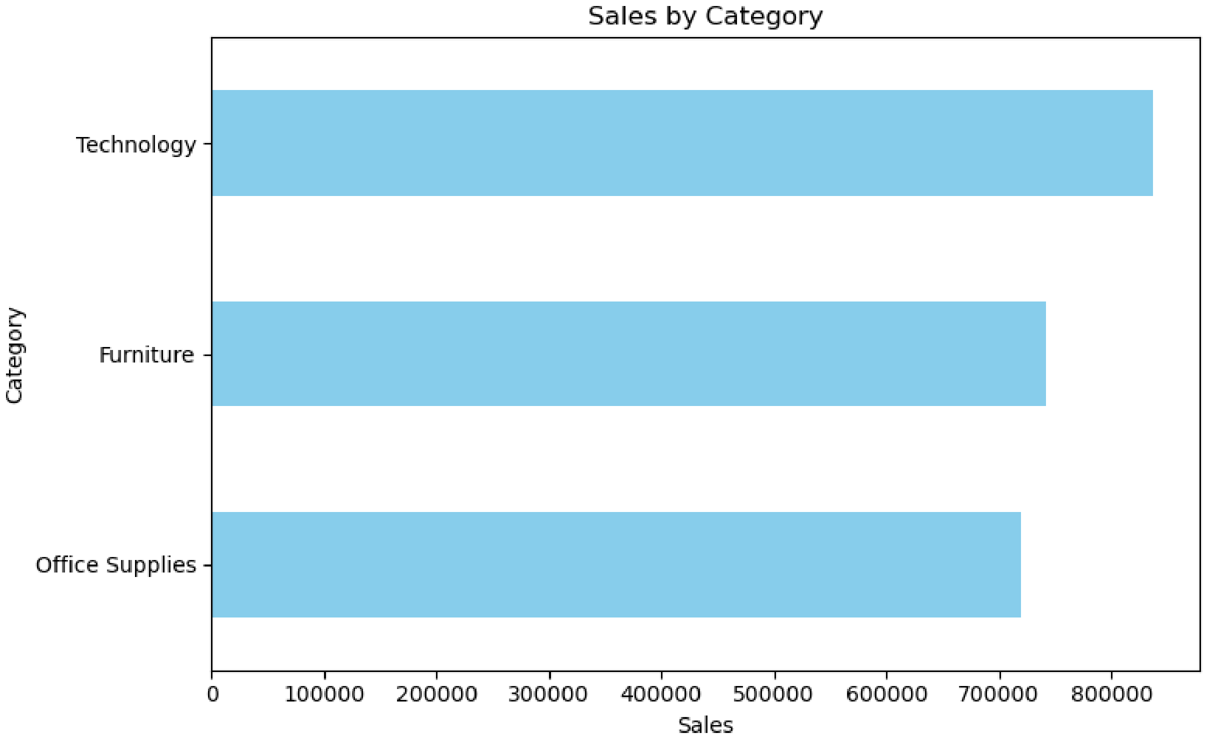
```python
# Step 5: Sales over time
monthly_sales = df.groupby('Month')['Sales'].sum().reset_index()
plt.figure(figsize=(12, 6))
sns.lineplot(data=monthly_sales, x='Month', y='Sales')
plt.title('Monthly Sales Trend')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\pramo\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future versio
n. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\pramo\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future versio
n. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



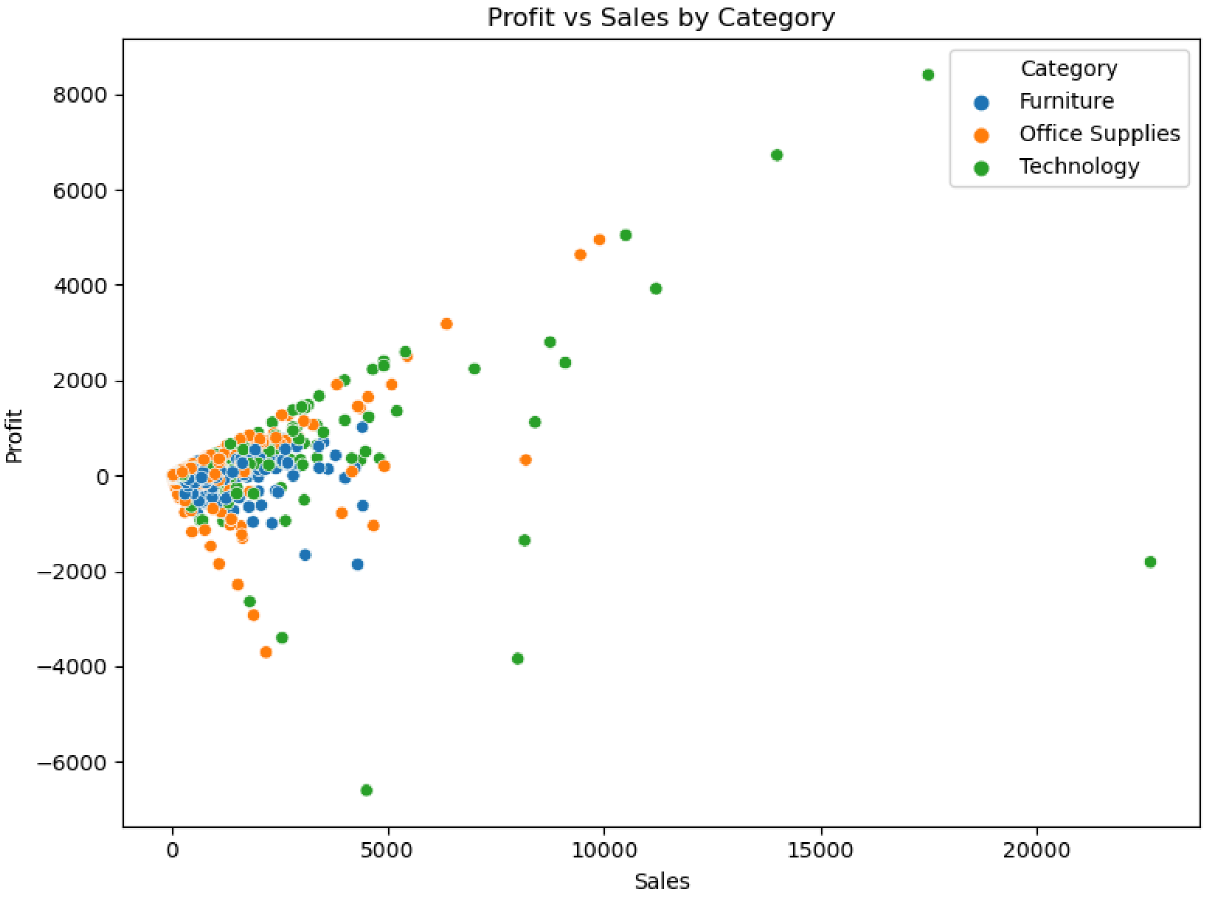
In [6]:

```
# Step 6: Sales by Category
category_sales = df.groupby('Category')['Sales'].sum().sort_values()
category_sales.plot(kind='barh', color='skyblue', figsize=(8, 5))
plt.title('Sales by Category')
plt.xlabel('Sales')
plt.ylabel('Category')
plt.tight_layout()
plt.show()
```



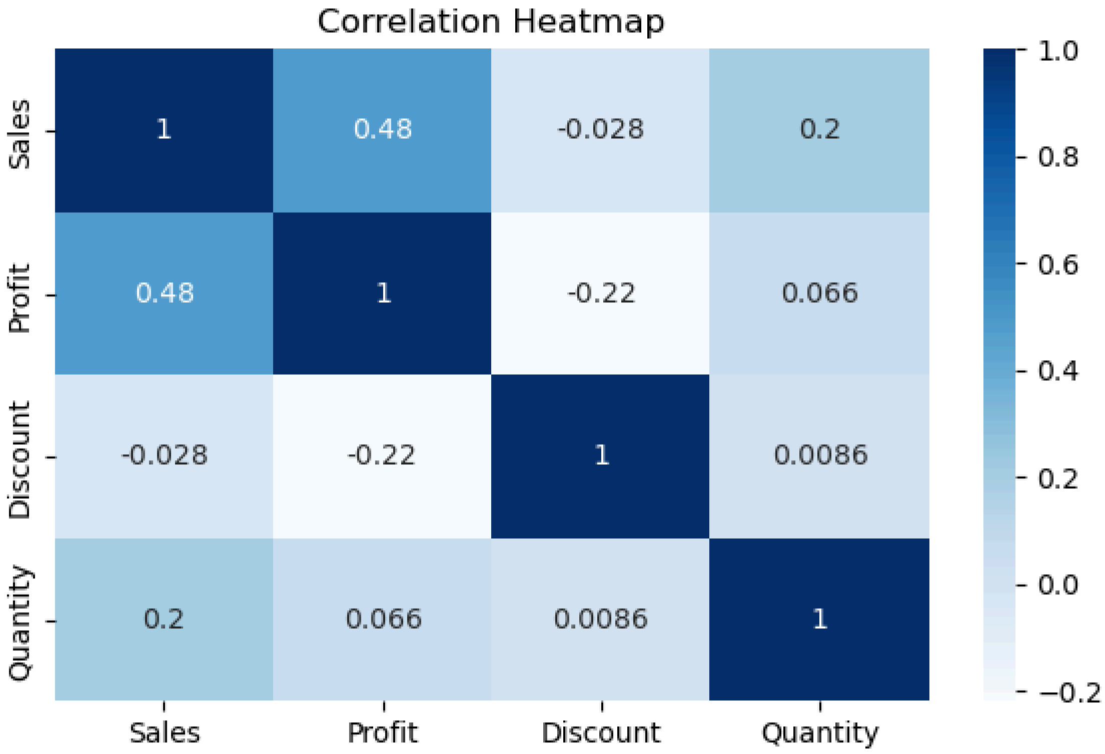
In [7]:

```
# Step 7: Profit vs Sales Scatter Plot
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Sales', y='Profit', hue='Category')
plt.title('Profit vs Sales by Category')
plt.tight_layout()
plt.show()
```



In [8]:

```
# Step 8: Correlation Heatmap
corr = df[['Sales', 'Profit', 'Discount', 'Quantity']].corr()
plt.figure(figsize=(6, 4))
sns.heatmap(corr, annot=True, cmap='Blues')
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```


Correlation Heatmap
Sales
Profit
Discount
Quantity
Sales
Profit
Discount
Quantity