In [1]:
```
import pandas as pd
```

In [2]:
```
file_path =r"C:\Users\krupa\OneDrive\Documents\Superstore.csv"
```

In [3]:
```
df=pd.read_csv(file_path, encoding='latin1')
```

In [4]:
```
df
```

Out[4]:

	ï»¿Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country
0	1	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2016-138688	12-06-2016	16-06-2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2015-108966	11-10-2015	18-10-2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
...	...	...	...	...	...	...	...	...	...
94	95	CA-2015-149587	31-01-2015	05-02-2015	Second Class	KB-16315	Karl Braun	Consumer	United States
95	96	US-2017-109484	06-11-2017	12-11-2017	Standard Class	RB-19705	Roger Barcio	Home Office	United States
96	97	CA-2017-161018	09-11-2017	11-11-2017	Second Class	PN-18775	Parhena Norris	Home Office	United States
97	98	CA-2017-157833	17-06-2017	20-06-2017	First Class	KD-16345	Katherine Ducich	Consumer	United States
98	99	CA-2016-149223	06-09-2016	11-09-2016	Standard Class	ER-13855	Elpida Rittenbach	Corporate	United States

99 rows × 21 columns

In [5]:
```
print("Columns:", df.columns.tolist())
```

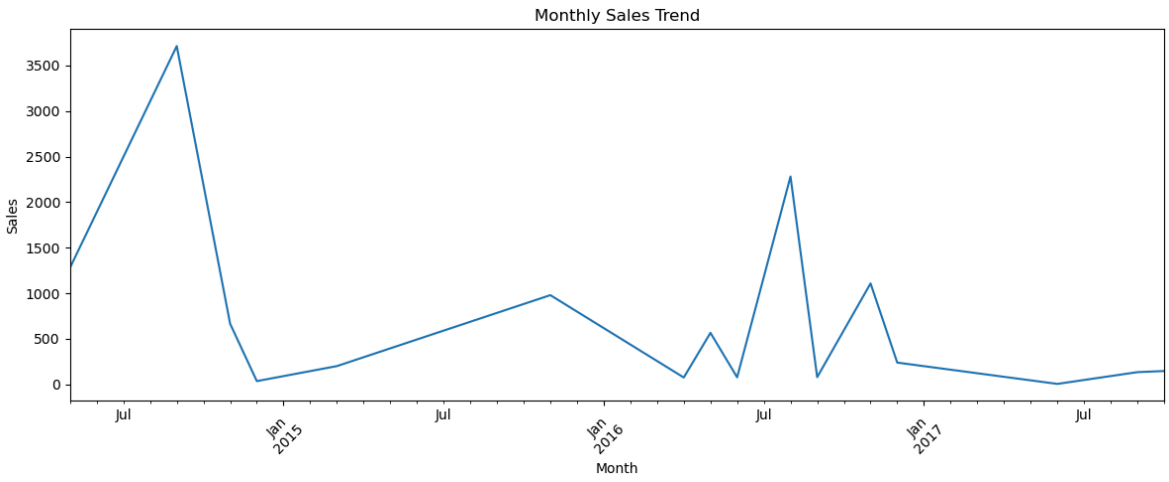
```
Columns: ['ï»¿Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Custo
mer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal Code',
'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name', 'Sales', 'Qua
ntity', 'Discount', 'Profit']
```

In [6]:
```
df.columns = df.columns.str.strip()
```

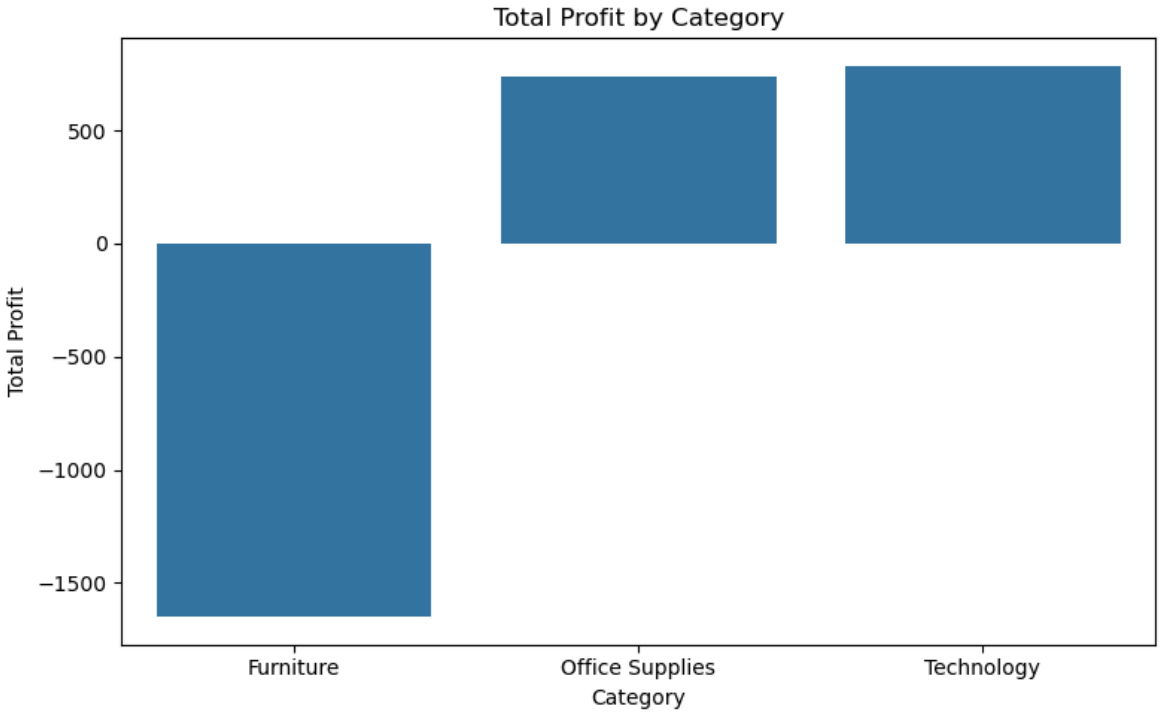
In [7]:
```
# Convert dates
df['Order Date'] = pd.to_datetime(df['Order Date'], errors='coerce')
df['Ship Date'] = pd.to_datetime(df['Ship Date'], errors='coerce')
```

In [8]:
```
# Add Month for time-based grouping
df['Month'] = df['Order Date'].dt.to_period('M')
```

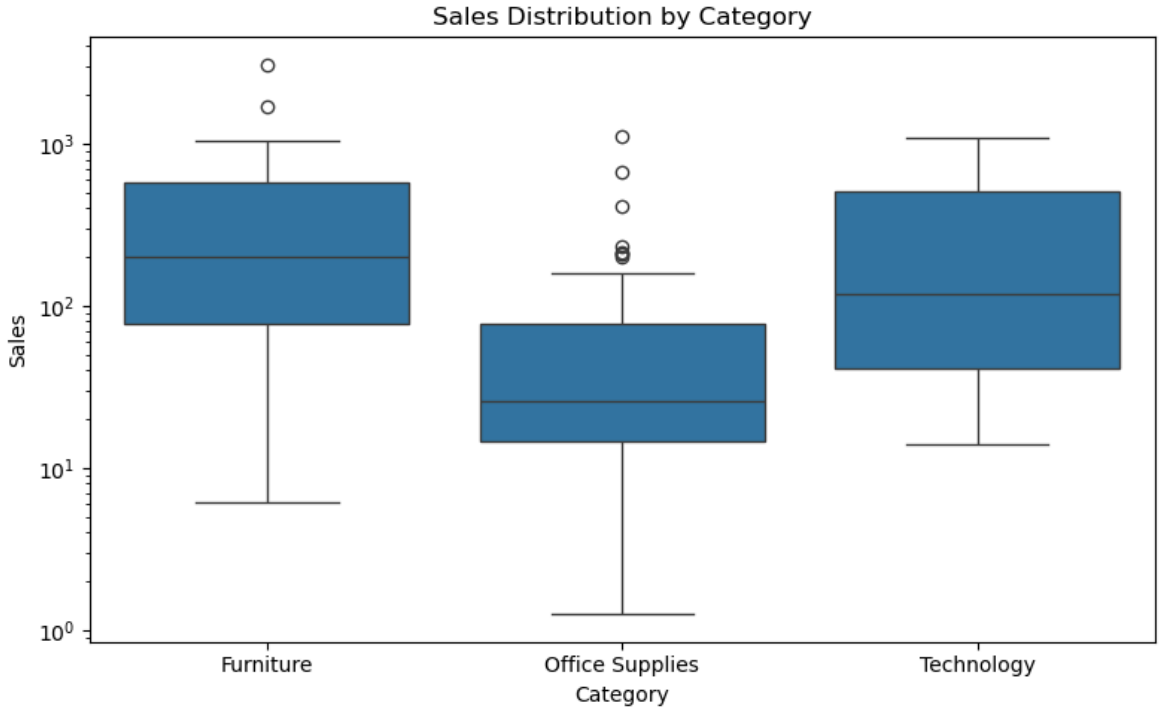
In [9]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# 1 Monthly Sales Trend
monthly_sales = df.groupby('Month')['Sales'].sum()

plt.figure(figsize=(12, 5))
monthly_sales.plot()
plt.title("Monthly Sales Trend")
plt.ylabel("Sales")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



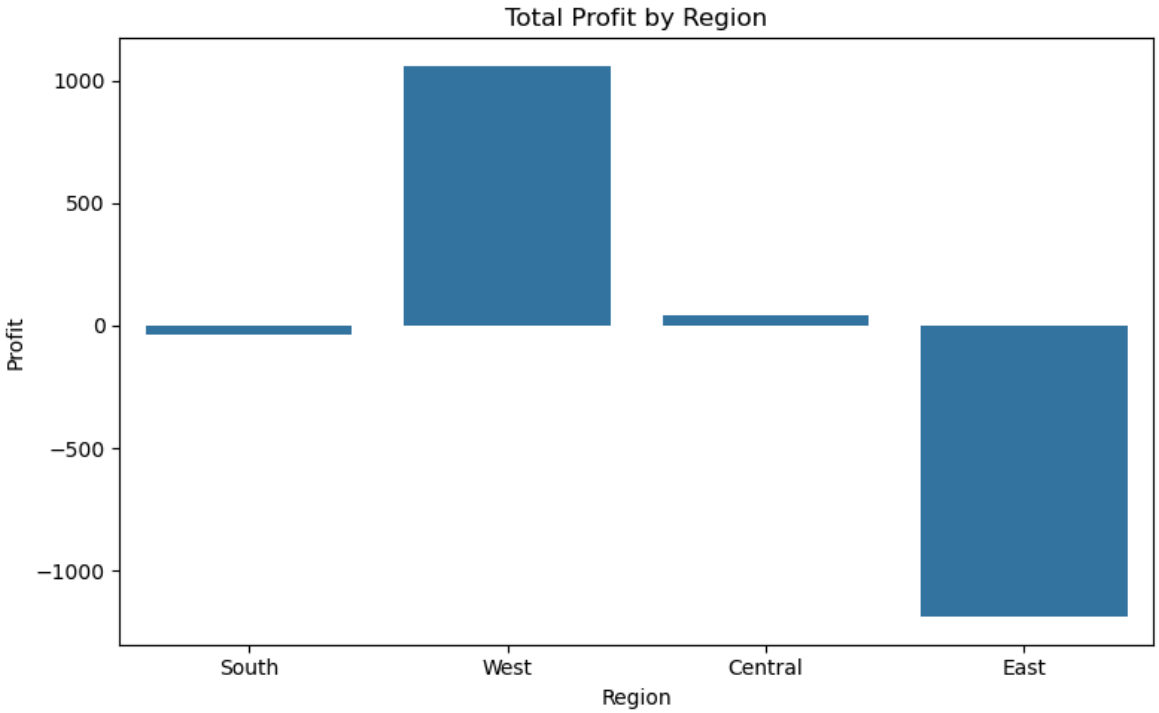
In [12]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Total Profit by Category
plt.figure(figsize=(8, 5))
sns.barplot(x='Category', y='Profit', data=df, estimator=sum, errorbar=None)
plt.title("Total Profit by Category")
plt.ylabel("Total Profit")
plt.tight_layout()
plt.show()
```



In [12]:
```
# 3 Sales Distribution by Category - Boxplot
plt.figure(figsize=(8, 5))
sns.boxplot(x='Category', y='Sales', data=df)
plt.title("Sales Distribution by Category")
plt.yscale('log')  # Optional: Use log scale if values vary widely
plt.tight_layout()
plt.show()
```

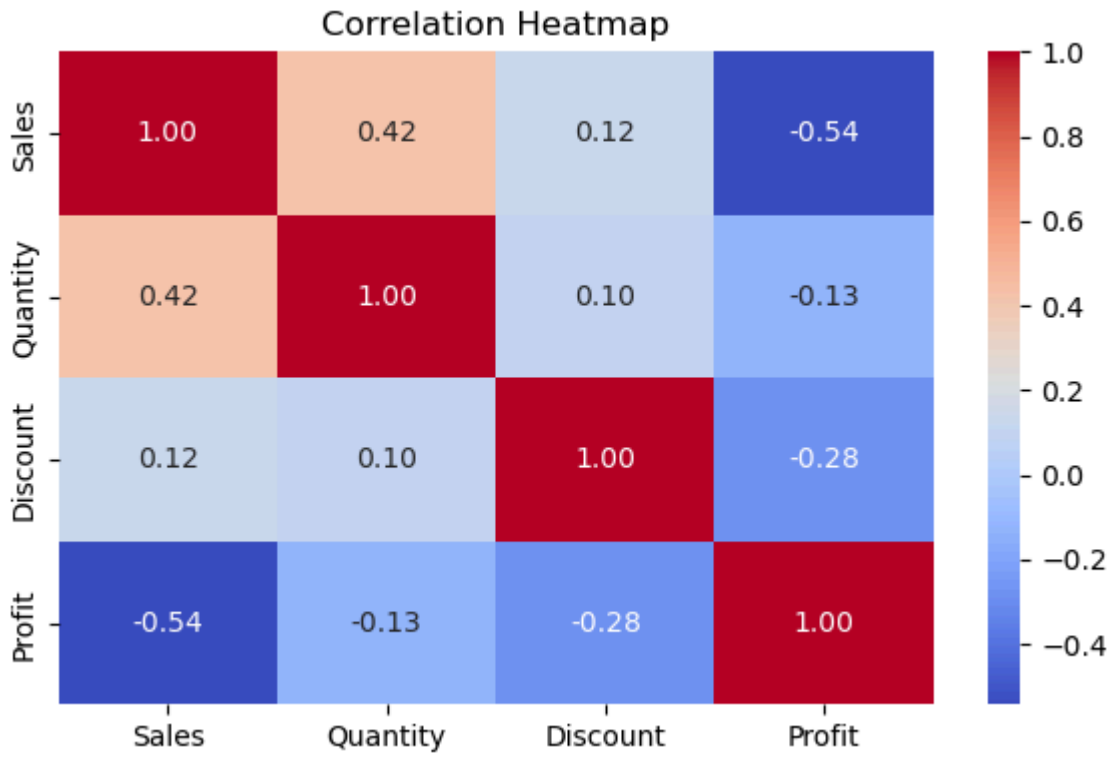


In [13]:
```
# Profit by Region - BarPlot
plt.figure(figsize=(8, 5))
sns.barplot(x='Region', y='Profit', data=df, estimator=sum, errorbar=None)
plt.title("Total Profit by Region")
plt.tight_layout()
plt.show()
```



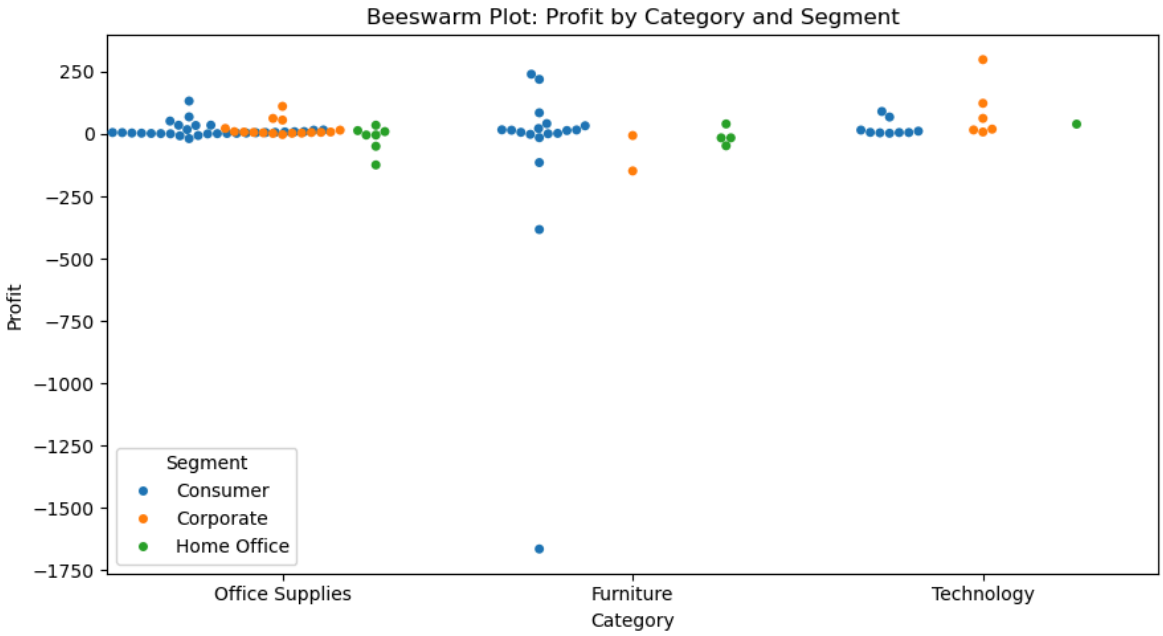
In [14]:
```
# 5 Sales vs Profit - Scatter Plot
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Sales', y='Profit', hue='Category', data=df)
plt.title("Sales vs Profit by Category")
plt.tight_layout()
plt.show()
```

In [15]:
```
# 6 Correlation Heatmap
plt.figure(figsize=(6, 4))
numeric_df = df[['Sales', 'Quantity', 'Discount', 'Profit']]
corr = numeric_df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()
```



In [18]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Get a safe sample size (max 500 or actual number of rows)
sample_size = min(len(df), 500)
df_sample = df.sample(n=sample_size, random_state=1)

plt.figure(figsize=(9, 5))
sns.swarmplot(x='Category', y='Profit', data=df_sample, hue='Segment', dodge=Tru
plt.title("Beeswarm Plot: Profit by Category and Segment")
plt.xlabel("Category")
plt.ylabel("Profit")
plt.legend(title="Segment")
plt.tight_layout()
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\categorical.py:3399: UserWarni
ng: 8.3% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
```


Beeswarm Plot: Profit by Category and Segment
Profit
250
0
−250
−500
−750
−1000
−1250
−1500
−1750
Segment
Consumer
Corporate
Home Office
Office Supplies
Furniture
Technology
Category

In [ ]: