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

# Set a nice style for plots
sns.set(style="whitegrid")
```

```
from google.colab import files

uploaded = files.upload()
```

Choose Files Sample - Superstore.csv

- **Sample - Superstore.csv**(text/csv) - 2287806 bytes, last modified: 5/20/2025 - 100% done

```
Saving Sample - Superstore.csv to Sample - Superstore (1).csv
```

```
import os
print(os.listdir())
```

```
['.config', '.ipynb_checkpoints', 'Sample - Superstore.csv', 'sample_data']
```

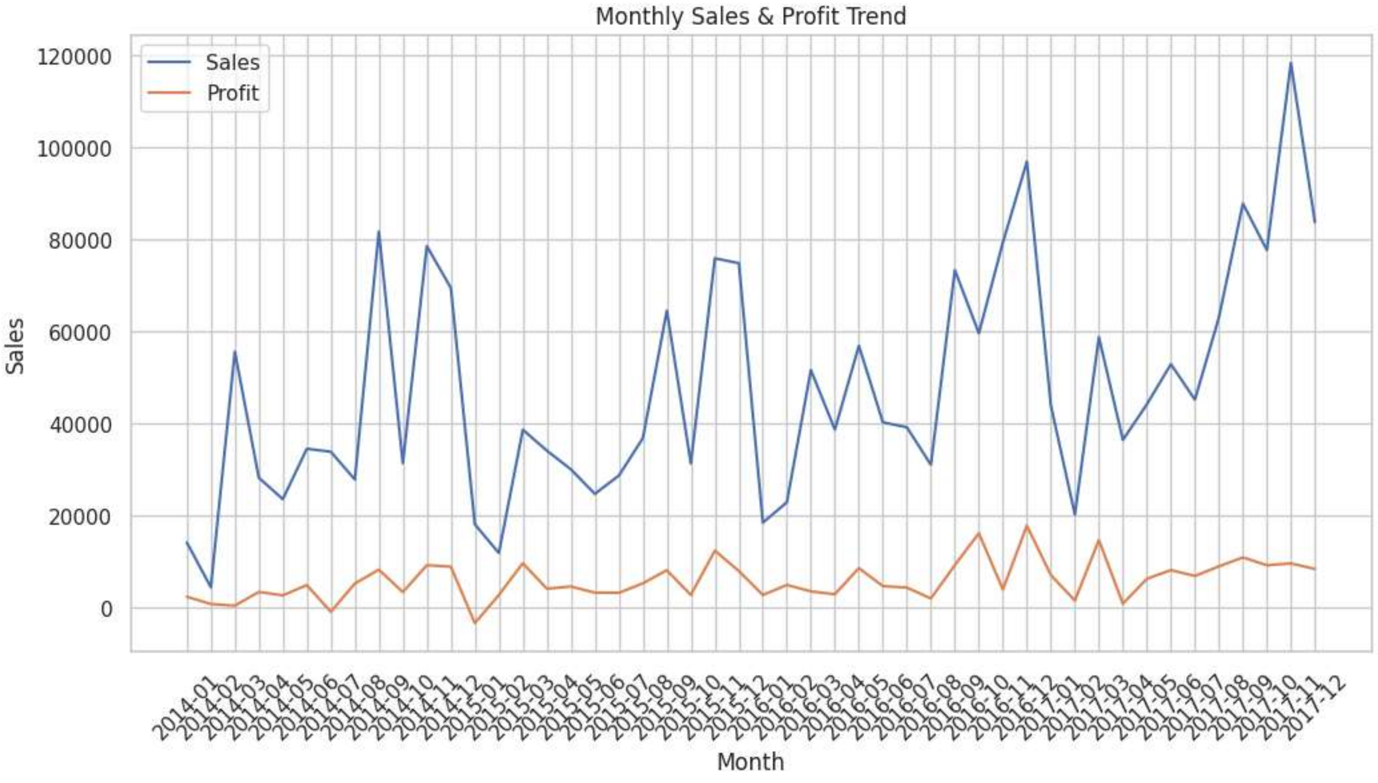
```
import pandas as pd

df = pd.read_csv("Sample - Superstore.csv", encoding='latin1')
print(df.head())
print(df.info())
```

```
   Row ID        Order ID  Order Date   Ship Date       Ship Mode Customer ID  \
0       1  CA-2016-152156   11/8/2016  11/11/2016    Second Class    CG-12520
1       2  CA-2016-152156   11/8/2016  11/11/2016    Second Class    CG-12520
2       3  CA-2016-138688   6/12/2016   6/16/2016    Second Class    DV-13045
3       4  US-2015-108966  10/11/2015  10/18/2015  Standard Class    SO-20335
4       5  US-2015-108966  10/11/2015  10/18/2015  Standard Class    SO-20335

     Customer Name    Segment        Country             City  ...  \
0      Claire Gute   Consumer  United States        Henderson  ...
1      Claire Gute   Consumer  United States        Henderson  ...
2  Darrin Van Huff  Corporate  United States      Los Angeles  ...
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  ...

  Postal Code  Region       Product ID         Category Sub-Category  \
0       42420   South  FUR-BO-10001798        Furniture    Bookcases
1       42420   South  FUR-CH-10000454        Furniture       Chairs
2       90036    West  OFF-LA-10000240  Office Supplies       Labels
3       33311   South  FUR-TA-10000577        Furniture       Tables
4       33311   South  OFF-ST-10000760  Office Supplies      Storage

                                        Product Name     Sales  Quantity  \
0                  Bush Somerset Collection Bookcase  261.9600         2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,...  731.9400         3
2  Self-Adhesive Address Labels for Typewriters b...   14.6200         2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                     Eldon Fold 'N Roll Cart System   22.3680         2

   Discount    Profit
0      0.00   41.9136
1      0.00  219.5820
2      0.00    6.8714
3      0.45 -383.0310
4      0.20    2.5164

[5 rows x 21 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
```

```
# Drop Postal Code column (not needed)
df.drop(['Postal Code'], axis=1, inplace=True)

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

# Create Month column for monthly aggregation
df['Month'] = df['Order Date'].dt.to_period('M')
```

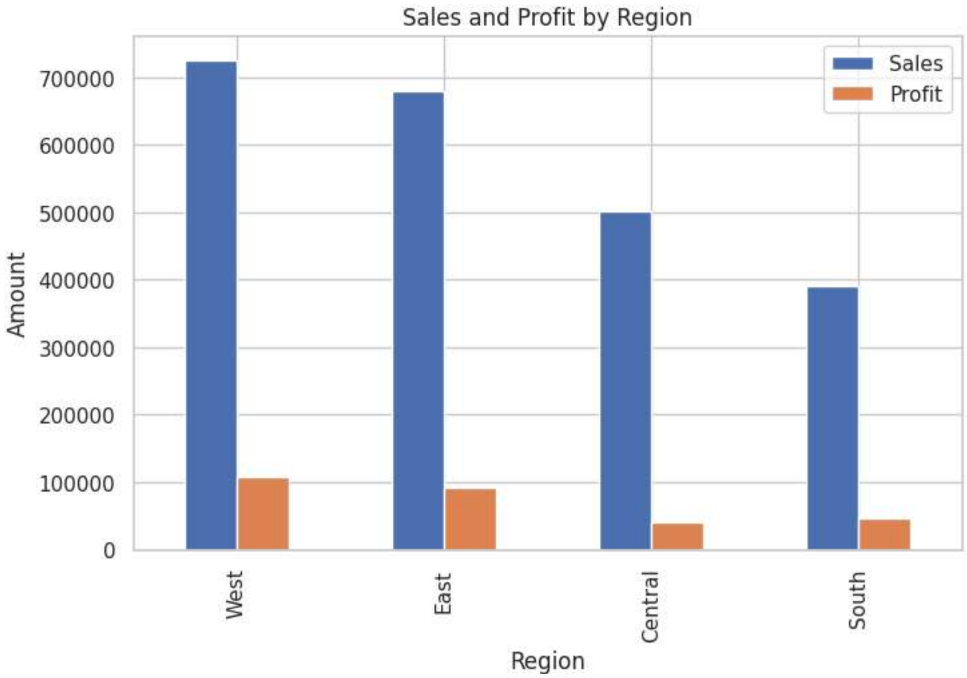
```
monthly['Month'] = monthly['Month'].astype(str)
```

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

plt.figure(figsize=(12,6))
sns.lineplot(data=monthly, x='Month', y='Sales', label='Sales')
sns.lineplot(data=monthly, x='Month', y='Profit', label='Profit')
plt.title("Monthly Sales & Profit Trend")
plt.xticks(rotation=45)
plt.show()
```

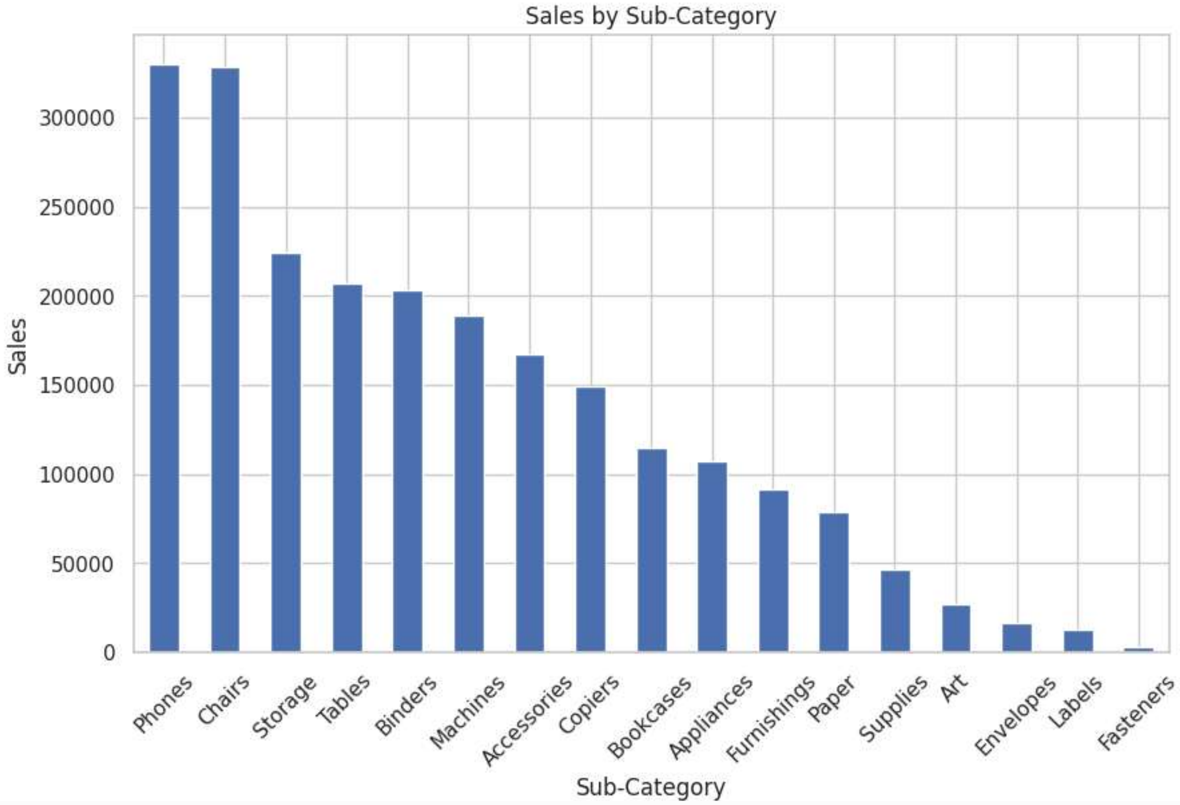


```
region_sales = df.groupby('Region')[['Sales', 'Profit']].sum().sort_values(by='Sales', ascending=False)
```

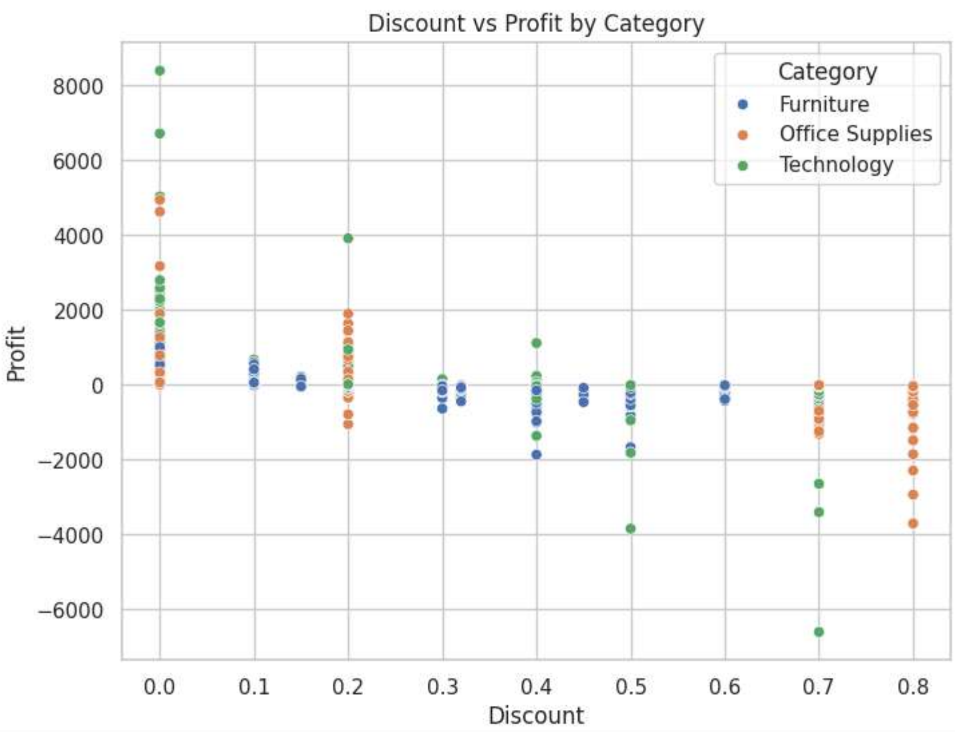
```
region_sales.plot(kind='bar', figsize=(8,5), title='Sales and Profit by Region')
plt.ylabel('Amount')
plt.show()
```



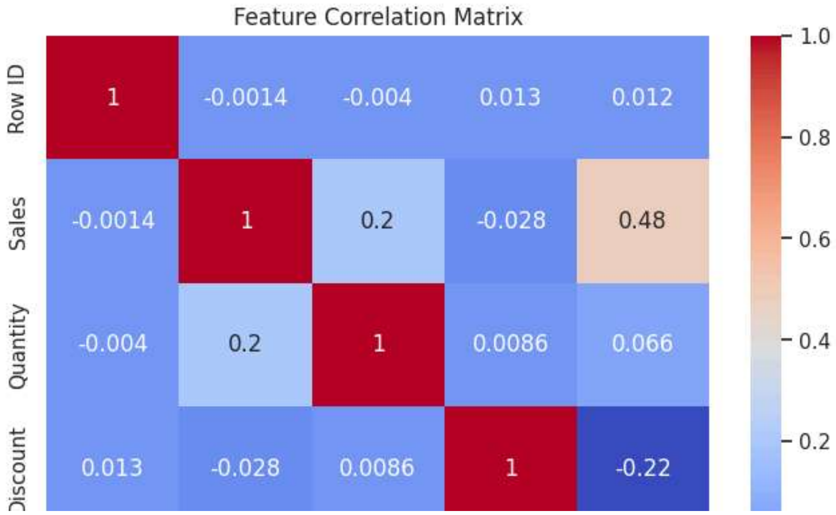
```
subcat_sales = df.groupby('Sub-Category')[['Sales', 'Profit']].sum().sort_values(by='Sales', ascending=False)

plt.figure(figsize=(10,6))
subcat_sales['Sales'].plot(kind='bar')
plt.title('Sales by Sub-Category')
plt.xticks(rotation=45)
plt.ylabel('Sales')
plt.show()
```



```
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='Discount', y='Profit', hue='Category')
plt.title('Discount vs Profit by Category')
plt.show()
```



```
plt.figure(figsize=(8,6))

# Select only numeric columns
numeric_df = df.select_dtypes(include=['float64', 'int64'])

sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Feature Correlation Matrix')
plt.show()
```


Feature Correlation Matrix
Row ID
Sales
Quantity
Discount
1
-0.0014
-0.004
0.013
0.012
-0.0014
1
0.2
-0.028
0.48
-0.004
0.2
1
0.0086
0.066
0.013
-0.028
0.0086
1
-0.22
1.0
0.8
0.6
0.4
0.2