```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files


# Load the dataset with the correct encoding
df = pd.read_csv('/content/superstore_final_dataset.csv',

# View first few rows
df.head()
```

	Row_ID	Order_ID	Order_Date	Ship_Date	Ship_Mod
0	1	CA-2017-152156	8/11/2017	11/11/2017	Secor Clas
1	2	CA-2017-152156	8/11/2017	11/11/2017	Secor Clas
2	3	CA-2017-138688	12/6/2017	16/06/2017	Secor Clas
3	4	US-2016-108966	11/10/2016	18/10/2016	Standar Clas
4	5	US-2016-108966	11/10/2016	18/10/2016	Standar Clas

Next steps: (Generate code with df)   (⦿ View recommended plots)

```python
# Load the dataset with the correct encoding
df = pd.read_csv('/content/superstore_final_dataset.csv'

# View first few rows
df.head()
```

superstore_final_dataset (1).csv › •••

1 to 10 of 9800 entries   Filter   ⎘

Row_ID	Order_ID	Order_Date	Ship_
1	CA-2017-152156	8/11/2017	11/11/2
2	CA-2017-152156	8/11/2017	11/11/2
3	CA-2017-138688	12/6/2017	16/06/
4	US-2016-108966	11/10/2016	18/10/
5	US-2016-108966	11/10/2016	18/10/
6	CA-2015-115812	9/6/2015	14/06/
7	CA-2015-115812	9/6/2015	14/06/
8	CA-2015-115812	9/6/2015	14/06/
9	CA-2015-115812	9/6/2015	14/06/
10	CA-2015-115812	9/6/2015	14/06/

Show [10 ▾] per page

[1]   2   10   100   900   970

                                        980

	Row_ID	Order_ID	Order_Date	Ship_Date	Ship_Mod
0	1	CA-2017-152156	8/11/2017	11/11/2017	Secor Clas
1	2	CA-2017-152156	8/11/2017	11/11/2017	Secor Clas
2	3	CA-2017-138688	12/6/2017	16/06/2017	Secor Clas
3	4	US-2016-108966	11/10/2016	18/10/2016	Standar Clas
4	5	US-2016-108966	11/10/2016	18/10/2016	Standar Clas

Next steps: (Generate code with df) (⬤ View recommended plots)

```
# Check shape and data types
print("Shape:", df.shape)
print("\nInfo:")
print(df.info())

# Summary statistics
df.describe()
```

⊡▼   Shape: (9800, 18)

      Info:
      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 9800 entries, 0 to 9799
      Data columns (total 18 columns):
       #   Column   Non-Null Count   Dtype
      ---   ------   --------------   -----
       0   Row_ID   9800 non-null   int64
       1   Order_ID   9800 non-null   object
       2   Order_Date   9800 non-null   object
       3   Ship_Date   9800 non-null   object
       4   Ship_Mode   9800 non-null   object
       5   Customer_ID   9800 non-null   object
       6   Customer_Name   9800 non-null   object
       7   Segment   9800 non-null   object
       8   Country   9800 non-null   object
       9   City   9800 non-null   object
       10   State   9800 non-null   object
       11   Postal_Code   9789 non-null   float64
       12   Region   9800 non-null   object
       13   Product_ID   9800 non-null   object
       14   Category   9800 non-null   object
       15   Sub_Category   9800 non-null   object
       16   Product_Name   9800 non-null   object
       17   Sales   9800 non-null   float64
      dtypes: float64(2), int64(1), object(15)
      memory usage: 1.3+ MB
      None

	Row_ID	Postal_Code	Sales
count	9800.000000	9789.000000	9800.000000
mean	4900.500000	55273.322403	230.769059
std	2829.160653	32041.223413	626.651875
min	1.000000	1040.000000	0.444000
25%	2450.750000	23223.000000	17.248000
50%	4900.500000	58103.000000	54.490000
75%	7350.250000	90008.000000	210.605000
max	9800.000000	99301.000000	22638.480000

```python
# Total missing values
df.isnull().sum()

# Optionally drop or fill missing values if needed
# df = df.dropna()
# df = df.fillna(method='ffill')
```

	0
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

```
# Clean column names
df.columns = df.columns.str.strip().str.replace(" ", "_"
```

```
plt.figure(figsize=(8,5))
sns.barplot(data=df, x='Category', y='Sales', estimator=
plt.title('Total Sales by Category')
plt.xticks(rotation=45)
plt.show()
```
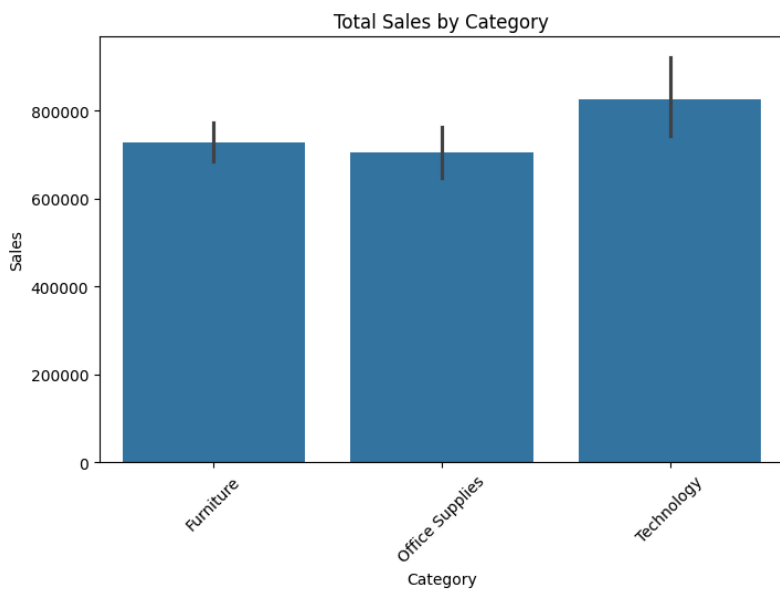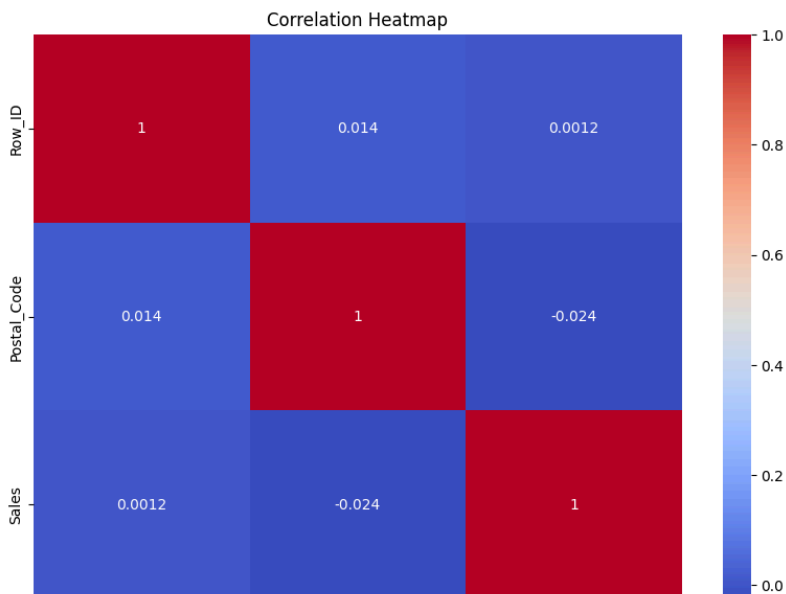
```
print(df.columns.tolist())
```

```
['Row_ID', 'Order_ID', 'Order_Date', 'Ship_Date', 'S
```

```
plt.figure(figsize=(10,7))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap
plt.title("Correlation Heatmap")
plt.show()
```

Correlation Heatmap

```python
df.to_csv('cleaned_superstore_data.csv', index=False)
files.download('cleaned_superstore_data.csv')
```

```python
df.to_csv('cleaned_superstore_data.csv', index=False)
```

```python
from google.colab import files
files.download('cleaned_superstore_data.csv')
```