



```
from google.colab import files
uploaded = files.upload()
```

 Choose files retail\_sales\_dataset.csv

- **retail\_sales\_dataset.csv**(text/csv) - 51673 bytes, last modified: 14/03/2025 - 100% done

Saving retail\_sales\_dataset.csv to retail\_sales\_dataset.csv

```
import pandas as pd
df = pd.read_csv('retail_sales_dataset.csv')
df.head()
```



	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
df.info()
df.isnull().sum()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Transaction ID         1000 non-null   int64
1   Date                   1000 non-null   object
2   Customer ID            1000 non-null   object
3   Gender                  1000 non-null   object
4   Age                     1000 non-null   int64
5   Product Category       1000 non-null   object
6   Quantity                1000 non-null   int64
7   Price per Unit          1000 non-null   int64
8   Total Amount           1000 non-null   int64
dtypes: int64(5), object(4)
memory usage: 70.4+ KB
```

	0
Transaction ID	0
Date	0
Customer ID	0
Gender	0
Age	0
Product Category	0
Quantity	0
Price per Unit	0
Total Amount	0

dtype: int64

```
import pandas as pd

print(df.describe())
print(df['Total Amount'].mean())
print(df['Total Amount'].median())
print(df['Total Amount'].std())
```

```
➡
```

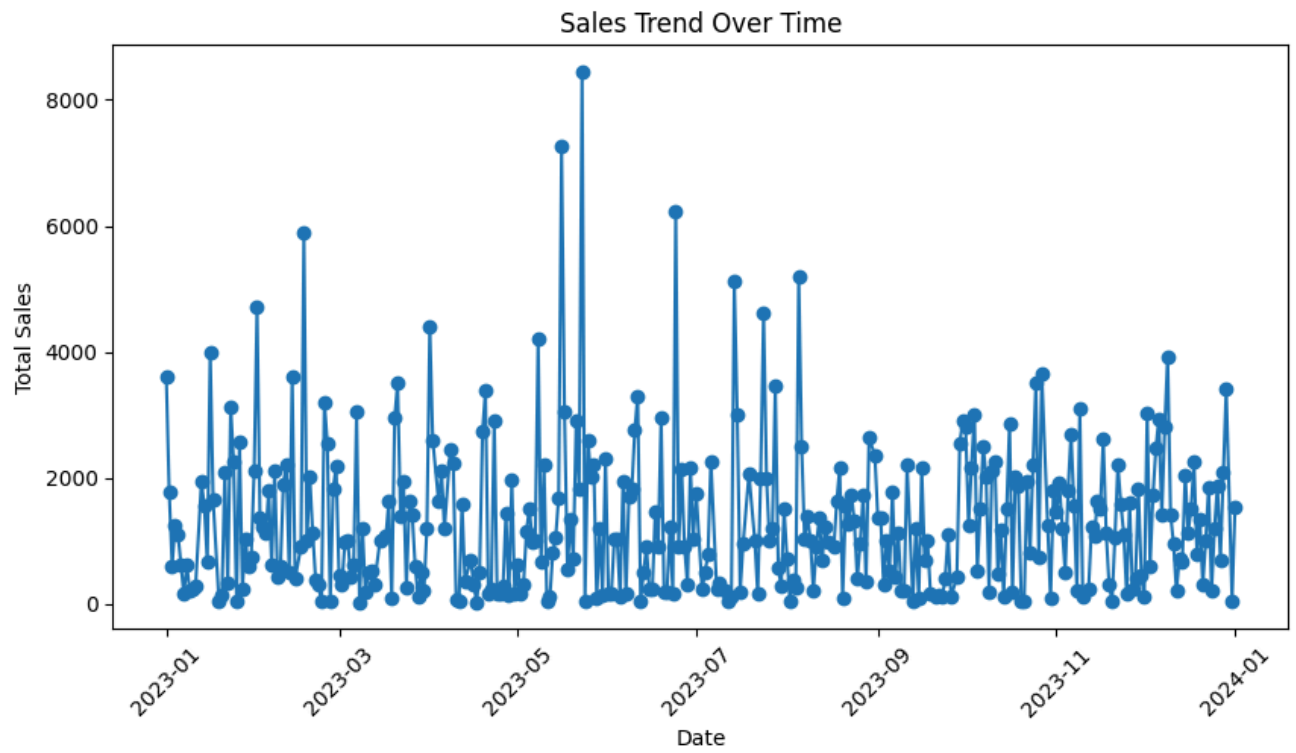
	Transaction ID	Age	Quantity	Price per Unit	Total Amount
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	41.39200	2.514000	179.890000	456.000000
std	288.819436	13.68143	1.132734	189.681356	559.997632
min	1.000000	18.00000	1.000000	25.000000	25.000000
25%	250.750000	29.00000	1.000000	30.000000	60.000000
50%	500.500000	42.00000	3.000000	50.000000	135.000000
75%	750.250000	53.00000	4.000000	300.000000	900.000000
max	1000.000000	64.00000	4.000000	500.000000	2000.000000
456.0					
135.0					
559.997631555123					

```
import matplotlib.pyplot as plt

df['Date'] = pd.to_datetime(df['Date'])

sales_trend = df.groupby('Date')['Total Amount'].sum()

plt.figure(figsize=(10,5))
plt.plot(sales_trend, marker='o', linestyle='-')
plt.title('Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.show()
```

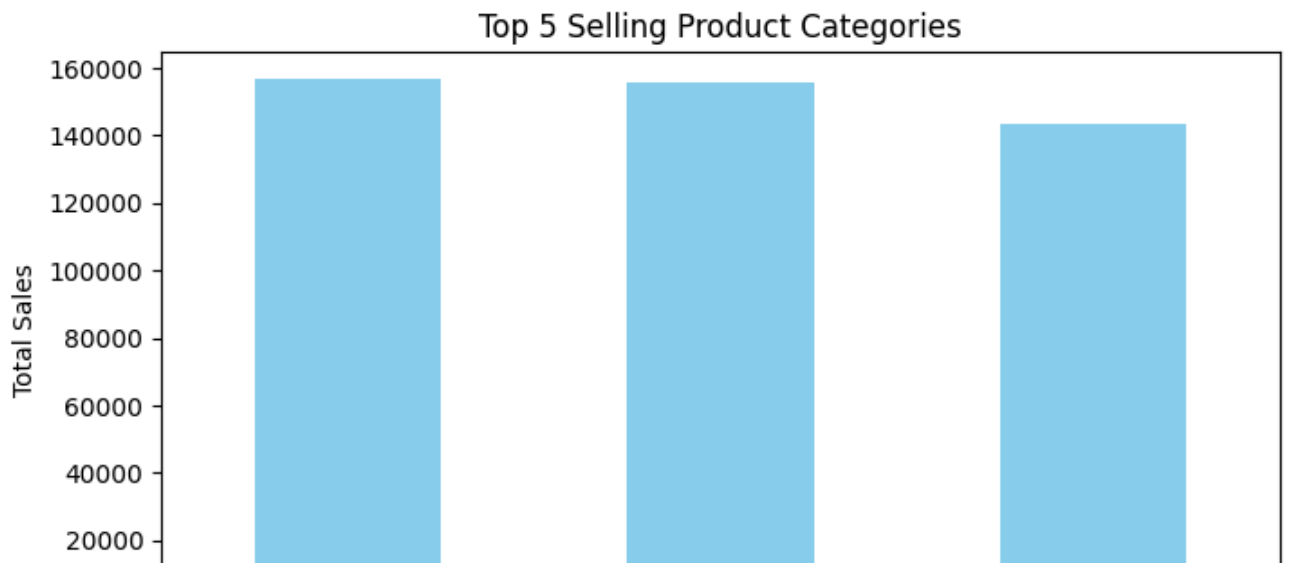
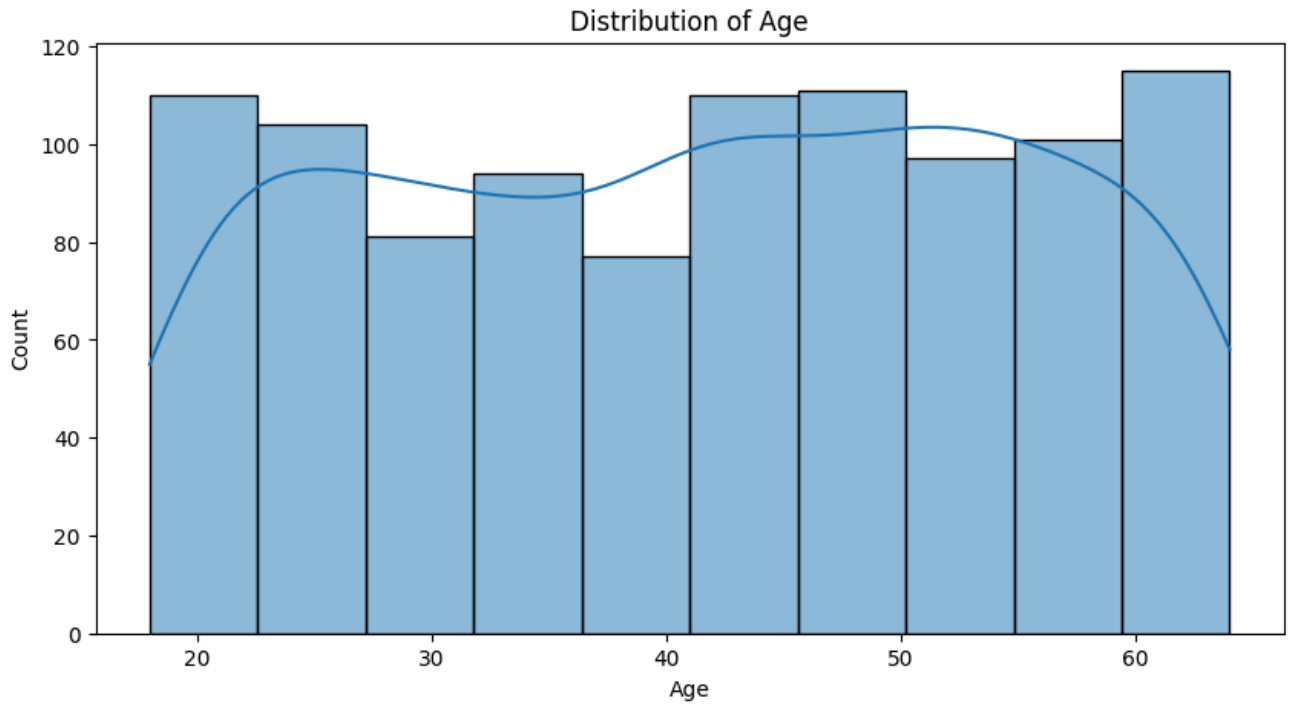
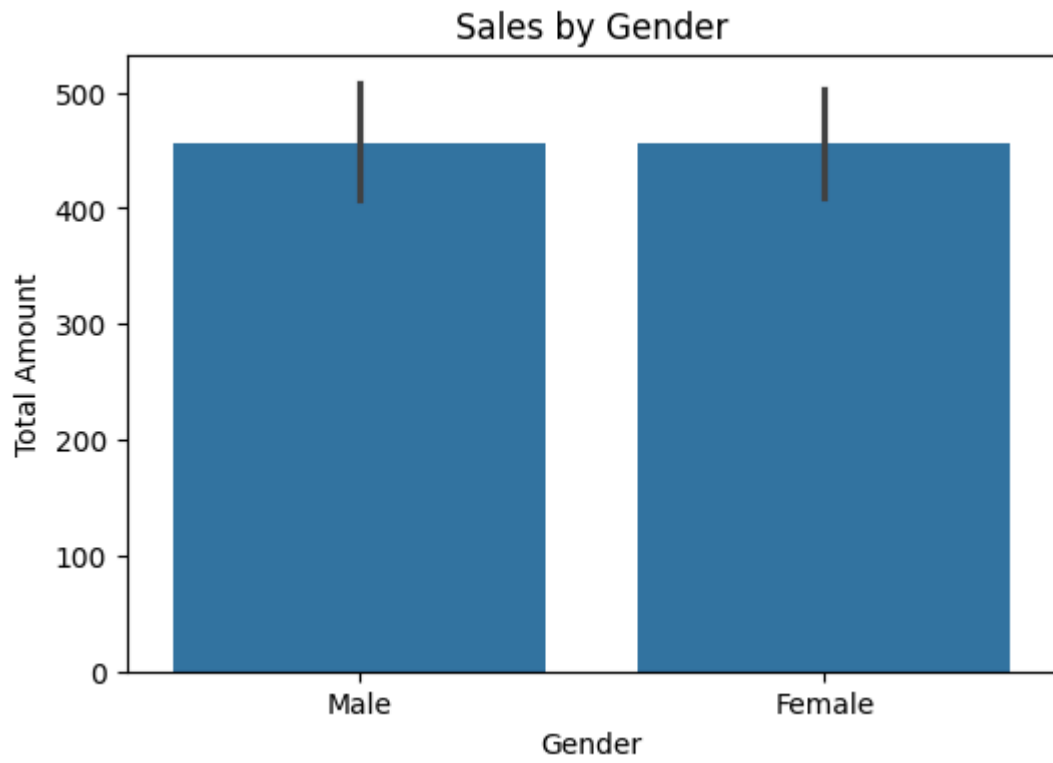


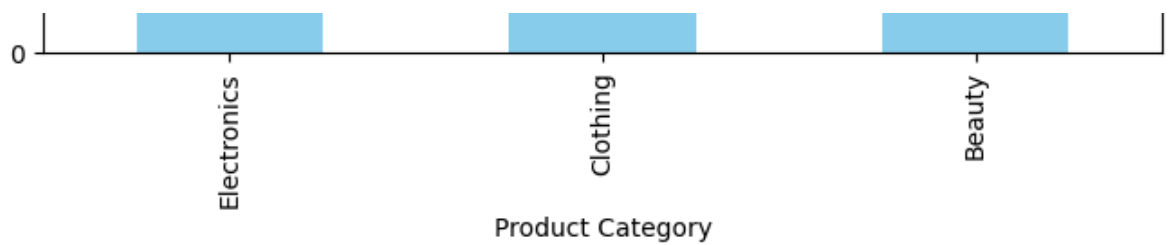
```
import seaborn as sns
```

```
plt.figure(figsize=(6,4))
sns.barplot(x=df['Gender'], y=df['Total Amount'])
plt.title('Sales by Gender')
plt.show()
```

```
plt.figure(figsize=(10,5))
sns.histplot(df['Age'], bins=10, kde=True)
plt.title('Distribution of Age')
plt.show()
```

```
top_products = df.groupby('Product Category')['Total Amount'].sum().sort_values(ascending=False)
top_products.plot(kind='bar', color='skyblue', figsize=(8,4))
plt.title('Top 5 Selling Product Categories')
plt.ylabel('Total Sales')
plt.show()
```





```
print(df.dtypes) # Check data types
```

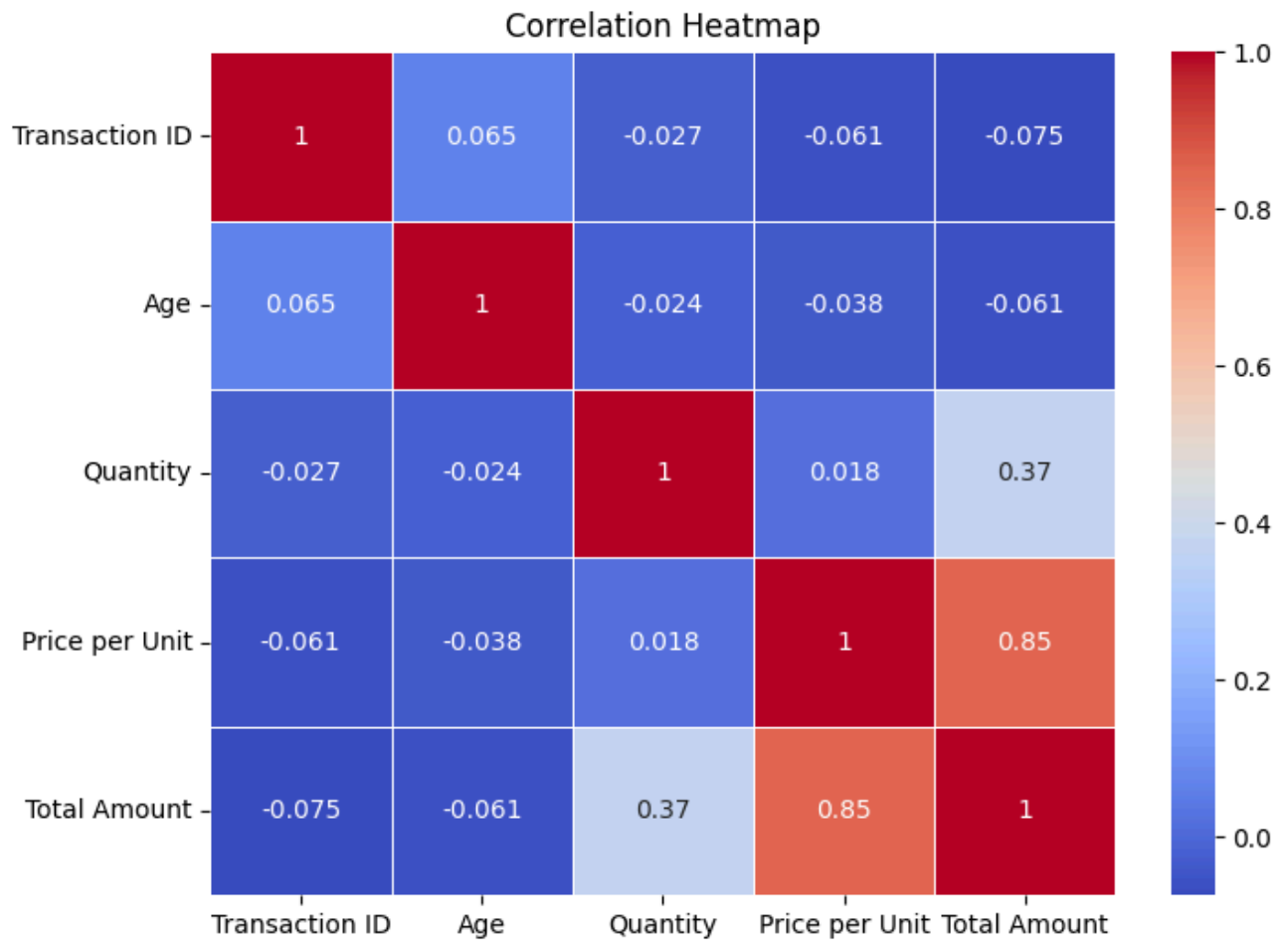
```
Transaction ID      int64
Date                datetime64[ns]
Customer ID         object
Gender              object
Age                 int64
Product Category    object
Quantity            int64
Price per Unit       int64
Total Amount        int64
dtype: object
```

```
df_numeric = df.select_dtypes(include=['int64', 'float64']) # Sirf numerical columns lo
print(df_numeric.head())
df_numeric = df_numeric.fillna(0)
```

```
Transaction ID  Age  Quantity  Price per Unit  Total Amount
0              1   34         3             50          150
1              2   26         2            500         1000
2              3   50         1             30           30
3              4   37         1            500          500
4              5   30         2             50          100
```

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

```
plt.figure(figsize=(8,6))
sns.heatmap(df_numeric.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



```
from IPython.display import display, Markdown
```

```
# Analysis and Insights
```

```
display(Markdown("### 1. **Sales Trend Over Time:**"))
```

```
display(Markdown("The code groups total sales by date and plots them, showing sales fluctuations over time."))  
display(Markdown("**Insight:** There are noticeable peaks, suggesting occasional spikes in sales activity."))
```

```
display(Markdown("### 2. **Sales by Gender:**"))
```

```
display(Markdown("A bar plot compares sales between genders."))
```

```
display(Markdown("**Insight:** Sales are almost equal for both genders, implying no significant gender-based sales difference."))
```

```
display(Markdown("### 3. **Age Distribution:**"))
```

```
display(Markdown("The histogram shows the age distribution of customers."))
```

```
display(Markdown("**Insight:** The majority of customers are in the 30-40 age group, making it a key demographic for targeted marketing."))
```

```
display(Markdown("### 4. **Top 5 Product Categories:**"))
```

```
display(Markdown("Bar chart of total sales per product category."))
```

```
display(Markdown("**Insight:** A few categories contribute the most to sales. Focusing on these top categories could optimize inventory and marketing efforts."))
```

```
display(Markdown("### 5. **Correlation Heatmap:**"))
```

```
display(Markdown("Shows the relationship between numerical variables."))
```

```
display(Markdown("**Insight:** Total Amount is strongly correlated with Quantity and Price per Unit, indicating these factors are major drivers of sales value."))
```

```
# Recommendations
```

```
display(Markdown("## **Recommendations:**"))
```

```
display(Markdown("1. **Run targeted promotions** during peak sales periods to boost revenue."))
```

```
display(Markdown("2. **Focus marketing efforts** on the 30-40 age group."))
```

```
display(Markdown("3. **Prioritize top-selling product categories** in inventory management."))
```

```
display(Markdown("4. **Maintain a balanced product mix** to cater to diverse customer preferences."))
```