Generate | create a dataframe with 2 columns and 10 rows | Close

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


df = pd.read_csv('/content/SuperMarket Analysis.csv')  # Adjust path if needed
df.head()
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payment	cogs	pe
0	750-67-8428	Alex	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	1:08:00 PM	Ewallet	522.83	
1	226-31-3081	Giza	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29:00 AM	Cash	76.40	
2	631-41-3108	Alex	Yangon	Normal	Female	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	1:23:00 PM	Credit card	324.31	
3	123-19-1176	Alex	Yangon	Member	Female	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	8:33:00 PM	Ewallet	465.76	
4	373-73-7910	Alex	Yangon	Member	Female	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37:00 AM	Ewallet	604.17	

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Check shape and info
print("Shape of dataset:", df.shape)
df.info()

# Check for missing values
print("\nMissing values:\n", df.isnull().sum())
```

```
Shape of dataset: (1000, 17)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Invoice ID               1000 non-null   object
 1   Branch                   1000 non-null   object
 2   City                     1000 non-null   object
 3   Customer type            1000 non-null   object
 4   Gender                   1000 non-null   object
 5   Product line             1000 non-null   object
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64
 8   Tax 5%                   1000 non-null   float64
 9   Sales                    1000 non-null   float64
 10  Date                     1000 non-null   object
 11  Time                     1000 non-null   object
 12  Payment                  1000 non-null   object
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB

Missing values:
 Invoice ID                0
Branch                     0
City                       0
Customer type              0
Gender                     0
Product line               0
Unit price                 0
Quantity                   0
Tax 5%                     0
Sales                      0
Date                       0
Time                       0
Payment                    0
cogs
gross margin percentage
gross income               0
Rating                     0
dtype: int64
```

What can I help you build?

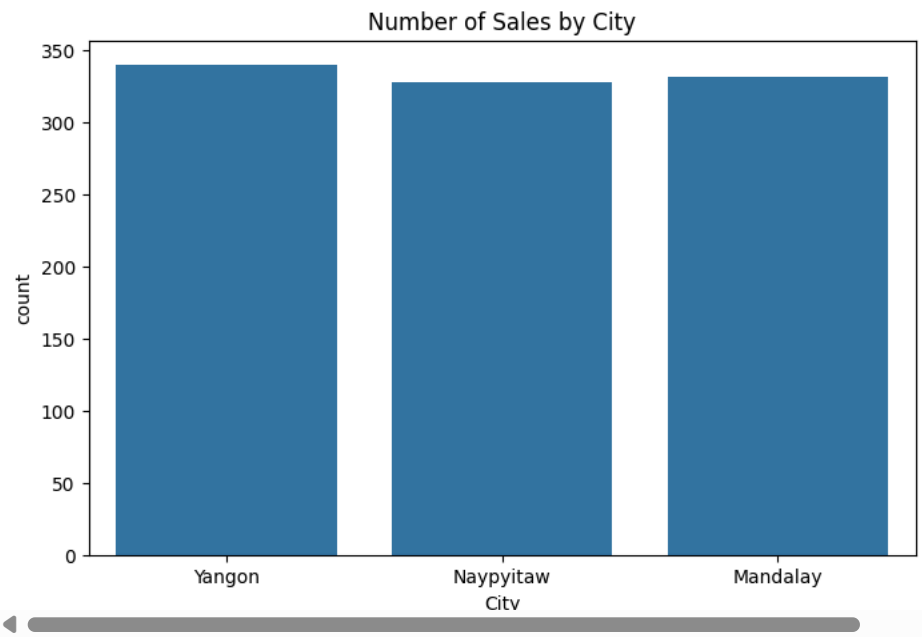
```
df.describe()
```

	Unit price	Quantity	Tax 5%	Sales	cogs	gross margin percentage	gross income	Rating
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1.000000e+03	1000.000000	1000.00000
mean	55.672130	5.510000	15.379369	322.966749	307.58738	4.761905e+00	15.379369	6.97270
std	26.494628	2.923431	11.708825	245.885335	234.17651	6.131498e-14	11.708825	1.71858
min	10.080000	1.000000	0.508500	10.678500	10.17000	4.761905e+00	0.508500	4.00000
25%	32.875000	3.000000	5.924875	124.422375	118.49750	4.761905e+00	5.924875	5.50000
50%	55.230000	5.000000	12.088000	253.848000	241.76000	4.761905e+00	12.088000	7.00000
75%	77.935000	8.000000	22.445250	471.350250	448.90500	4.761905e+00	22.445250	8.50000
max	99.960000	10.000000	49.650000	1042.650000	993.00000	4.761905e+00	49.650000	10.00000

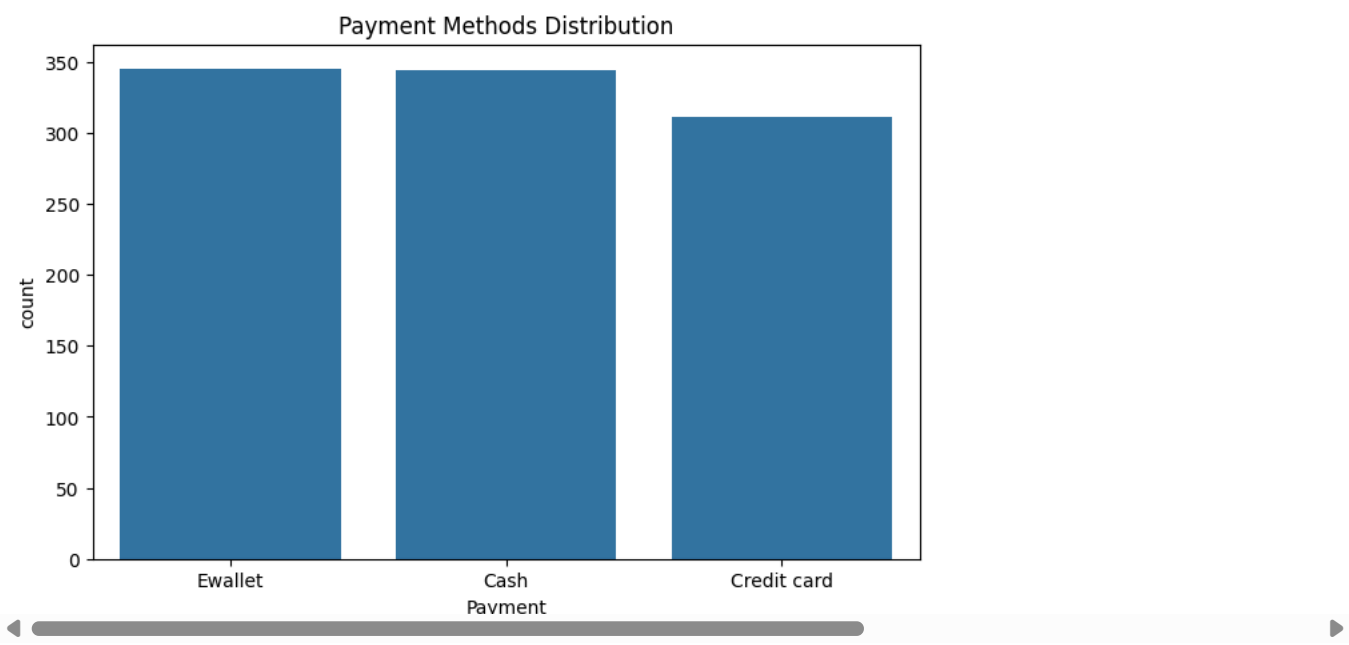
```
print(df['City'].unique())
print(df['Payment'].unique())
print(df['Product line'].unique())
```

```
['Yangon' 'Naypyitaw' 'Mandalay']
['Ewallet' 'Cash' 'Credit card']
['Health and beauty' 'Electronic accessories' 'Home and lifestyle'
 'Sports and travel' 'Food and beverages' 'Fashion accessories']
```

```
plt.figure(figsize=(8,5))
sns.countplot(data=df, x='City')
plt.title('Number of Sales by City')
plt.show()
```



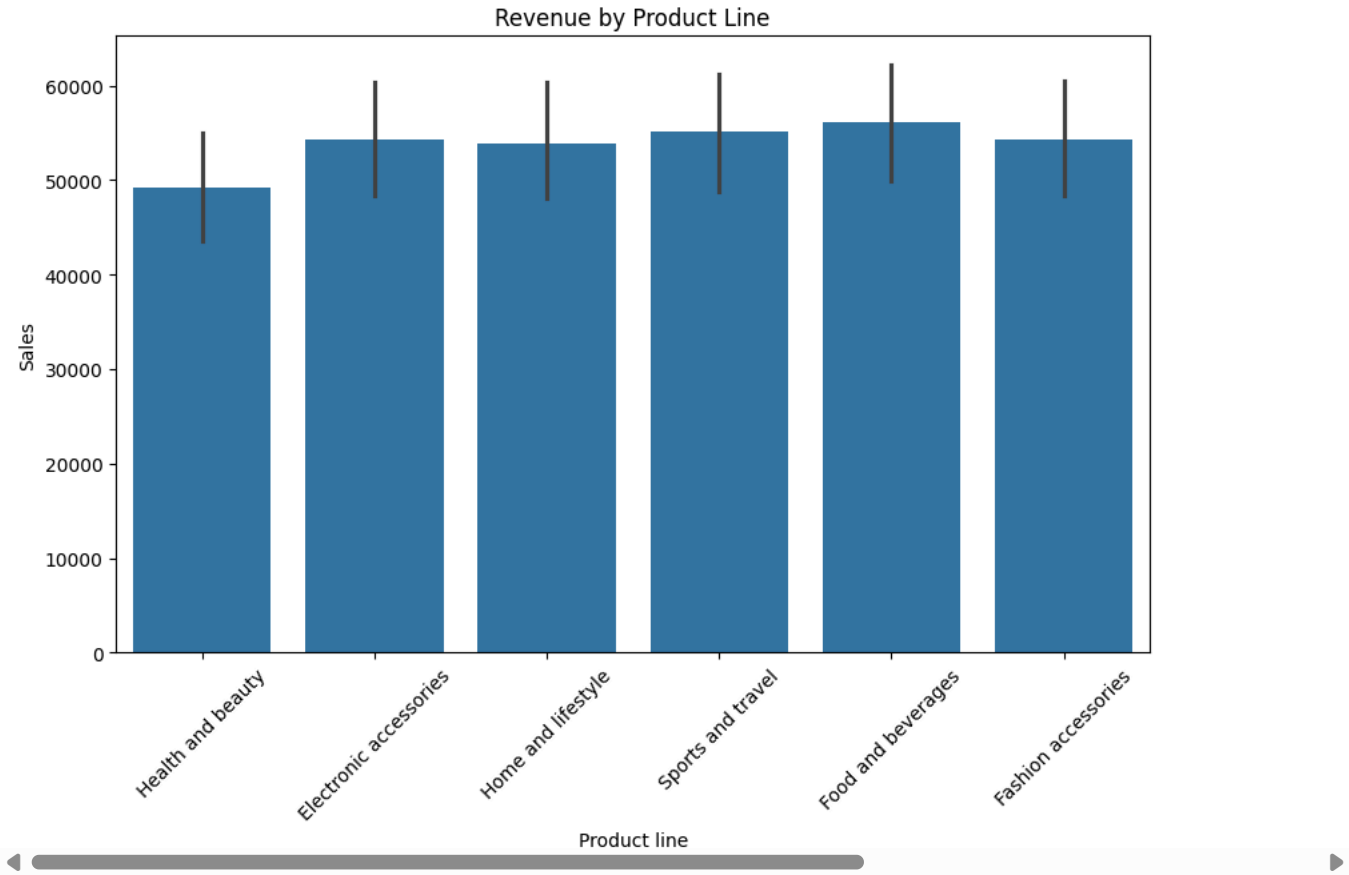

```
plt.figure(figsize=(8,5))
sns.countplot(data=df, x='Payment')
plt.title('Payment Methods Distribution')
plt.show()
```




```
print(df.columns)
```

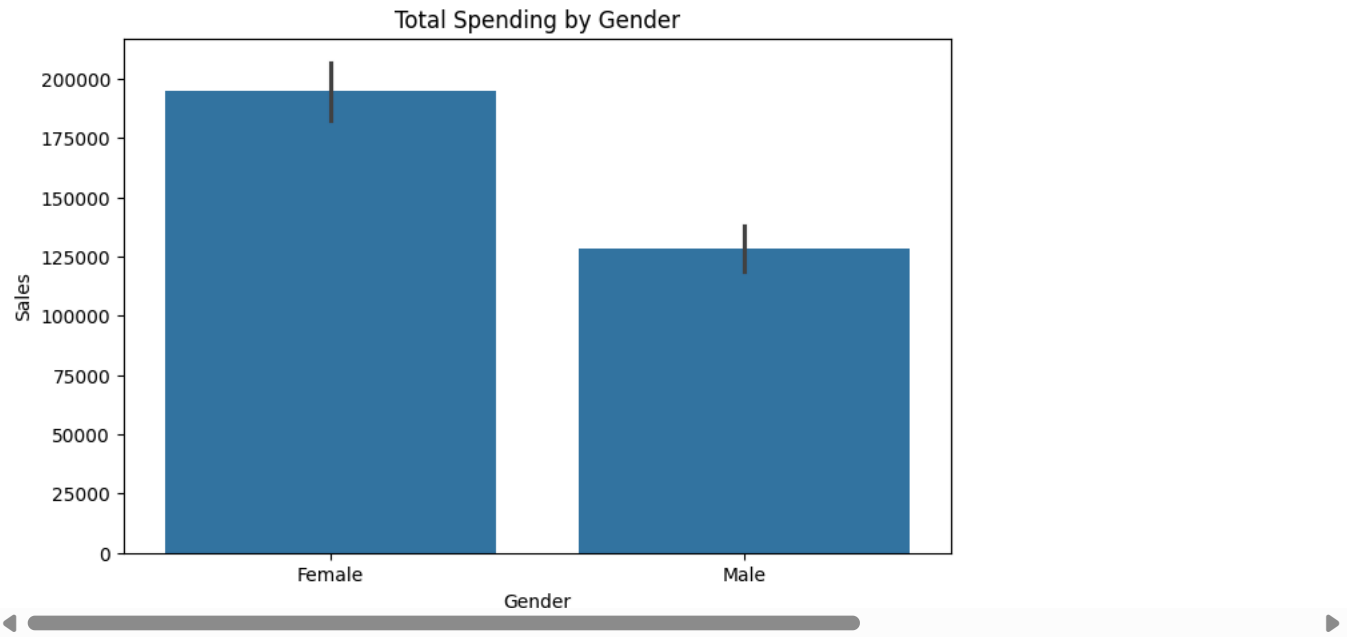
```
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Sales', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating'],
      dtype='object')
```

```
plt.figure(figsize=(10,6))
sns.barplot(data=df, x='Product line', y='Sales', estimator=sum)
plt.xticks(rotation=45)
plt.title('Revenue by Product Line')
plt.show()
```




```
plt.figure(figsize=(8,5))
sns.barplot(data=df, x='Gender', y='Sales', estimator=sum)
plt.title('Total Spending by Gender')
```

```
plt.show()
```




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

# Plot heatmap with only numeric columns
plt.figure(figsize=(10,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='Blues')
plt.title('Correlation Heatmap')
plt.show()
```