```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
sales = pd.read_csv('/content/sales.csv')
```

```
sales.head()
```

	CT	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Payment	gross income	Rating
0	CAI	Cairo	Member	Female	Health and beauty	$74.69	7	26.1415	$548.97	Ewallet	$26.14	9.1
1	ALX	Alexandria	Normal	Female	Electronic accessories	$15.28	5	3.8200	$80.22	Cash	$3.82	9.6
2	CAI	Cairo	Normal	Male	Home and lifestyle	$46.33	7	16.2155	$340.53	Credit card	$16.22	7.4
3	CAI	Cairo	Member	Male	Health and beauty	$58.22	8	23.2880	$489.05	Ewallet	$23.29	8.4
4	CAI	Cairo	Normal	Male	Sports and travel	$86.31	7	30.2085	$634.38	Ewallet	$30.21	5.3

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

```
sales.columns
```

```
Index(['CT', 'City', 'Customer type', 'Gender', 'Product line', ' Unit price ',
       'Quantity', 'Tax 5%', ' Total ', 'Payment', ' gross income ', 'Rating'],
      dtype='object')
```

Double-click (or enter) to edit

```
sales.rename(columns={'CT': 'CityID'}, inplace=True)
```

rename column CT to CityID

```
sales.columns
```

```
Index(['CityID', 'City', 'Customer type', 'Gender', 'Product line',
       ' Unit price ', 'Quantity', 'Tax 5%', ' Total ', 'Payment',
       ' gross income ', 'Rating'],
      dtype='object')
```

```
sales.dtypes
```

```
                    0
CityID         object
City           object
Customer type  object
Gender         object
Product line   object
Unit price     object
Quantity        int64
Tax 5%        float64
Total          object
Payment        object
gross income   object
Rating        float64
```

```
sales = sales.astype({'CityID': 'string'})
```

```
sales = sales.astype({ 'Customer type': 'string'})
```

```
sales = sales.astype({'Product line': 'string'})
```

```
sales = sales.astype({'Gender': 'string'})
```

```
sales = sales.astype({'Quantity': 'int'})
```

```
sales = sales.astype({'Payment': 'string'})
```

```
sales = sales.astype({'City': 'string'})
```

```
sales.dtypes
```

	0
CityID	string[python]
City	string[python]
Customer type	string[python]
Gender	string[python]
Product line	string[python]
Unit price	object
Quantity	int64
Tax 5%	float64
Total	object
Payment	string[python]
gross income	object
Rating	float64

```
sales[' Total '] = sales[' Total '].str.strip().replace({'\$':'', ',': ''}, regex=True).astype(float)
```

sales[' gross income '] = sales[' gross income '].replace({'$': ''}, regex=True).astype(float)

print(sales[[' Total ', ' gross income ']].head())

sales[' Total ']

```
sales.dtypes
```

```
                         0
CityID         string[python]
City           string[python]
Customer type  string[python]
Gender         string[python]
Product line   string[python]
Unit price             object
Quantity                int64
Tax 5%                float64
Total                 float64
Payment        string[python]
gross income           object
Rating                float64
```

```
sales.describe()
```

```
          Quantity        Tax 5%        Total       Rating
count  1000.000000  1000.000000  1000.000000  1000.00000
mean      5.510000    15.379369   322.967430     6.97270
std       2.923431    11.708825   245.885557     1.71858
min       1.000000     0.508500    10.680000     4.00000
25%       3.000000     5.924875   124.425000     5.50000
50%       5.000000    12.088000   253.850000     7.00000
75%       8.000000    22.445250   471.350000     8.50000
max      10.000000    49.650000  1042.650000    10.00000
```

```
sales.isnull().sum()
```

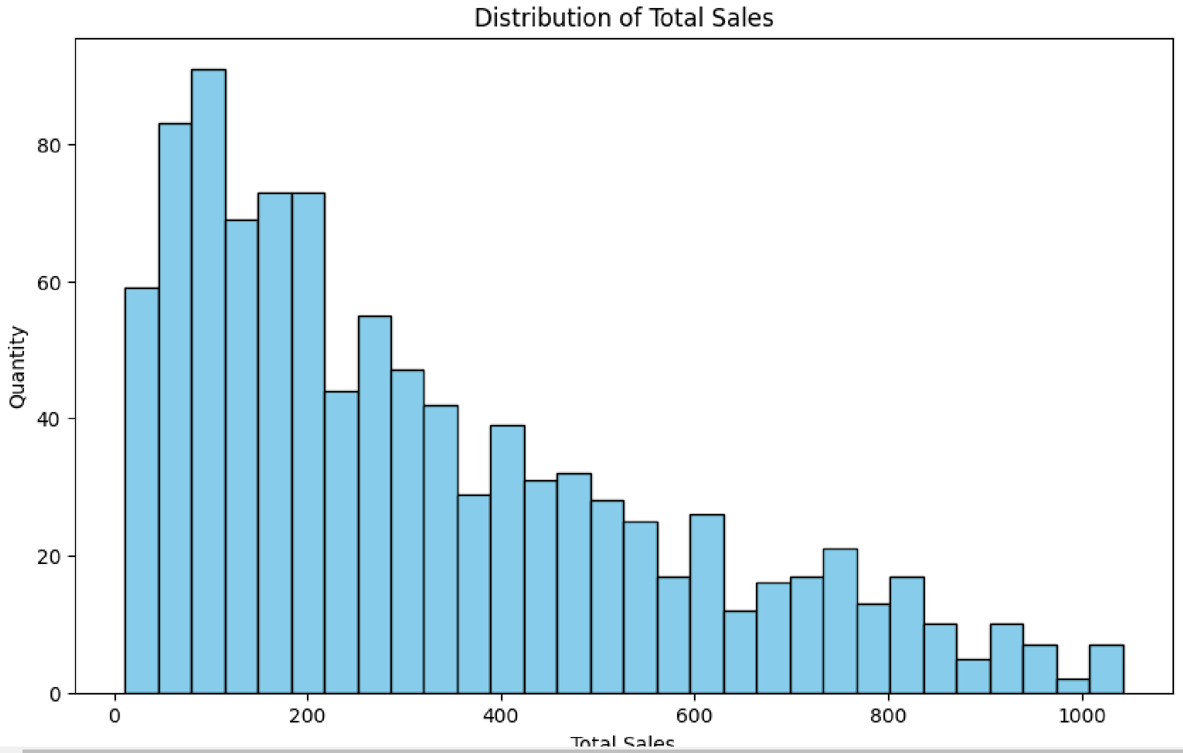
```
                0
CityID          0
City            0
Customer type   0
Gender          0
Product line    0
Unit price      0
Quantity        0
Tax 5%          0
Total           0
Payment         0
gross income    0
Rating          0
```

```
duplicates = sales.duplicated()


duplicate_rows = sales[duplicates]
duplicate_rows
```

```
CityID  City  Customer type  Gender  Product line  Unit price  Quantity  Tax 5%  Total  Payment  gross income  Rating
```

```
plt.figure(figsize=(10, 6))
plt.hist(sales[' Total '], bins=30, color='skyblue', edgecolor='black')
plt.title('Distribution of Total Sales')
plt.xlabel('Total Sales')
plt.ylabel('Quantity')
plt.show()
```



```
sales.head()
```

	CityID	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Payment	gross income	Rating
0	CAI	Cairo	Member	Female	Health and beauty	$74.69	7	26.1415	548.97	Ewallet	$26.14	9.1
1	ALX	Alexandria	Normal	Female	Electronic accessories	$15.28	5	3.8200	80.22	Cash	$3.82	9.6
2	CAI	Cairo	Normal	Male	Home and lifestyle	$46.33	7	16.2155	340.53	Credit card	$16.22	7.4
3	CAI	Cairo	Member	Male	Health and beauty	$58.22	8	23.2880	489.05	Ewallet	$23.29	8.4
4	CAI	Cairo	Normal	Male	Sports and travel	$86.31	7	30.2085	634.38	Ewallet	$30.21	5.3

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

```
sales.columns
```

```
Index(['CityID', 'City', 'Customer type', 'Gender', 'Product line',
       ' Unit price ', 'Quantity', 'Tax 5%', ' Total ', 'Payment',
       ' gross income ', 'Rating'],
      dtype='object')
```

```
sales = sales.rename(columns={' Total ': 'Total',' Unit price ': 'Unit price', ' gross income ': 'Gross Income'})
```

```
sales.columns
```

```
Index(['CityID', 'City', 'Customer type', 'Gender', 'Product line',
       'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Payment', 'Gross Income',
       'Rating'],
      dtype='object')
```

```
sales
```

	CityID	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Payment	Gross Income	Rating
0	CAI	Cairo	Member	Female	Health and beauty	$74.69	7	26.1415	548.97	Ewallet	$26.14	9.1
1	ALX	Alexandria	Normal	Female	Electronic accessories	$15.28	5	3.8200	80.22	Cash	$3.82	9.6
2	CAI	Cairo	Normal	Male	Home and lifestyle	$46.33	7	16.2155	340.53	Credit card	$16.22	7.4
3	CAI	Cairo	Member	Male	Health and beauty	$58.22	8	23.2880	489.05	Ewallet	$23.29	8.4
4	CAI	Cairo	Normal	Male	Sports and travel	$86.31	7	30.2085	634.38	Ewallet	$30.21	5.3
...	...	...	...	...	...	...	...	...	...	...	...	...
995	ALX	Alexandria	Normal	Male	Health and beauty	$40.35	1	2.0175	42.37	Ewallet	$2.02	6.2
996	PS	Port said	Normal	Female	Home and lifestyle	$97.38	10	48.6900	1022.49	Ewallet	$48.69	4.4
997	CAI	Cairo	Member	Male	Food and beverages	$31.84	1	1.5920	33.43	Cash	$1.59	7.7
998	CAI	Cairo	Normal	Male	Home and lifestyle	$65.82	1	3.2910	69.11	Cash	$3.29	4.1
999	CAI	Cairo	Member	Female	Fashion accessories	$88.34	7	30.9190	649.30	Cash	$30.92	6.6

1000 rows × 12 columns

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

```
sales.describe()
```

	Quantity	Tax 5%	Total	Rating
count	1000.000000	1000.000000	1000.000000	1000.00000
mean	5.510000	15.379369	322.967430	6.97270
std	2.923431	11.708825	245.885557	1.71858
25%	3.000000	5.924875	124.422000	5.50000

```
sales.to_csv('cleaned.csv')
```

```
sales['Unit price'] = sales['Unit price'].str.strip().replace({'\$':'', ',': ''}, regex=True).astype(float)
```

```
sales.to_csv('cleaned2.csv')
```

```
sales.dtypes
```

	0
CityID	string[python]
City	string[python]
Customer type	string[python]
Gender	string[python]
Product line	string[python]
Unit price	float64
Quantity	int64
Tax 5%	float64