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

```
data = pd.read_csv('/content/Amazon Sales data.csv')
```

```
data.head()
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	To Pro
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	224598

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

Generate a slider using jupyter widgets Close

```
data.shape
```

```
(100, 14)
```

```
data.size
```

```
1400
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Region          100 non-null    object
 1   Country         100 non-null    object
 2   Item Type       100 non-null    object
 3   Sales Channel   100 non-null    object
 4   Order Priority  100 non-null    object
 5   Order Date      100 non-null    object
 6   Order ID        100 non-null    int64
 7   Ship Date       100 non-null    object
 8   Units Sold      100 non-null    int64
 9   Unit Price      100 non-null    float64
 10  Unit Cost       100 non-null    float64
 11  Total Revenue   100 non-null    float64
 12  Total Cost      100 non-null    float64
 13  Total Profit    100 non-null    float64
dtypes: float64(5), int64(2), object(7)
memory usage: 11.1+ KB
```

```
data.describe()
```

	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	1.000000e+02	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.000000e+02
mean	5.550204e+08	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.416820e+05
std	2.606153e+08	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4.385379e+05
min	1.146066e+08	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.258020e+03
25%	3.389225e+08	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.214436e+05
50%	5.577086e+08	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.907680e+05
75%	7.907551e+08	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6.358288e+05
max	9.940222e+08	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1.719922e+06

```
data.isnull()
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	False	False	False	False	False	False	False	False	False	False	False	False	False	False
96	False	False	False	False	False	False	False	False	False	False	False	False	False	False
97	False	False	False	False	False	False	False	False	False	False	False	False	False	False
98	False	False	False	False	False	False	False	False	False	False	False	False	False	False
99	False	False	False	False	False	False	False	False	False	False	False	False	False	False

```
data.isnull().sum()
```

	0
Region	0
Country	0
Item Type	0
Sales Channel	0
Order Priority	0
Order Date	0
Order ID	0
Ship Date	0
Units Sold	0
Unit Price	0
Unit Cost	0
Total Revenue	0
Total Cost	0
Total Profit	0

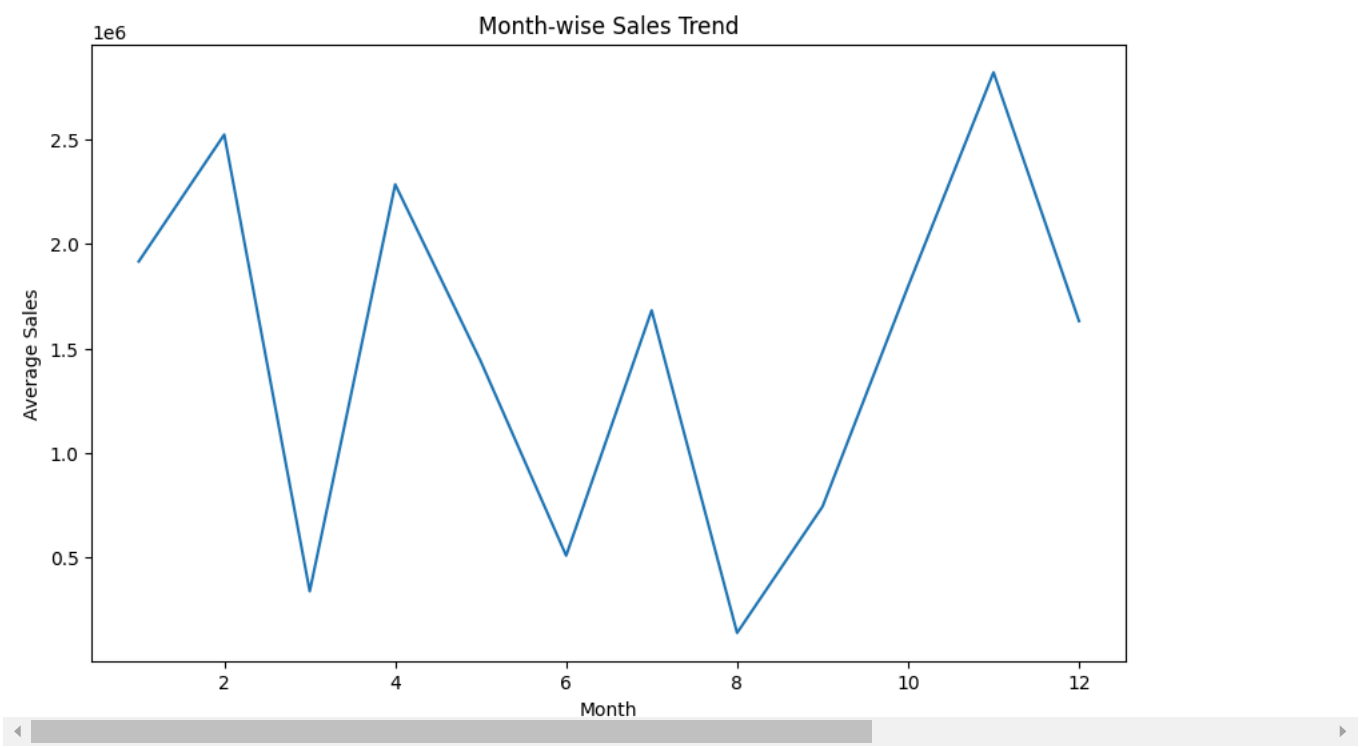
```
data.isnull().any()
```

	0
Region	False
Country	False
Item Type	False
Sales Channel	False
Order Priority	False
Order Date	False
Order ID	False
Ship Date	False
Units Sold	False
Unit Price	False
Unit Cost	False
Total Revenue	False
Total Cost	False
Total Profit	False

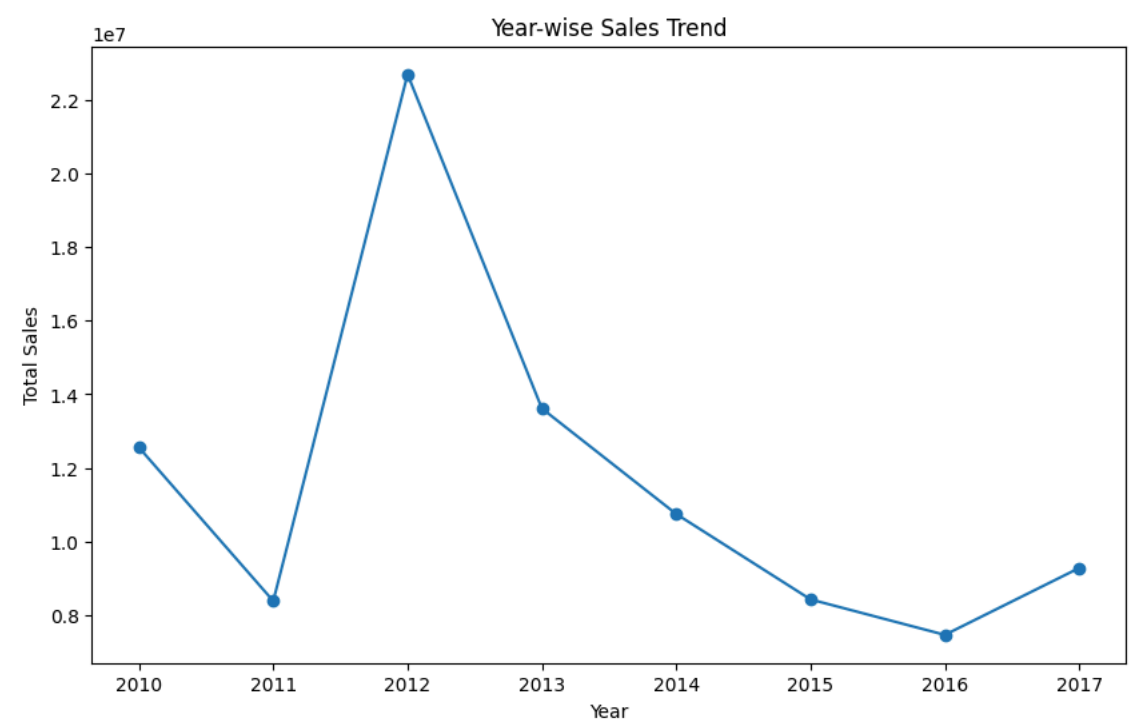
```
data[data.duplicated()==True]
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit

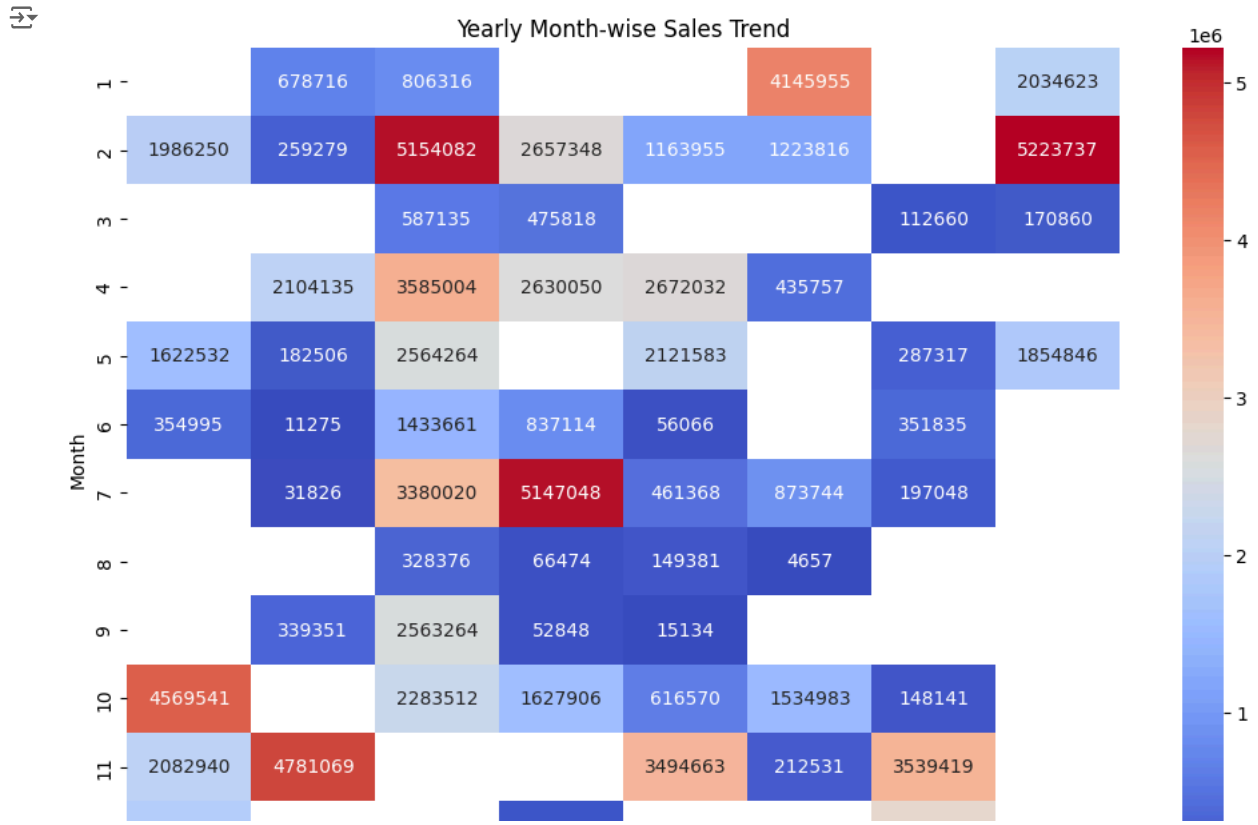
```
data.drop_duplicates(inplace=True)


data['Order Date'] = pd.to_datetime(data['Order Date'])


# Extract month and year from the date
data['month'] = data['Order Date'].dt.month
data['year'] = data['Order Date'].dt.year


# Aggregate data for month-wise, year-wise, and yearly_month-wise analysis
monthly_sales = data.groupby(['year', 'month']).agg({'Total Cost': 'sum'}).reset_index()
yearly_sales = data.groupby('year').agg({'Total Cost': 'sum'}).reset_index()
yearly_monthly_sales = data.pivot_table(index='month', columns='year', values='Total Cost', aggfunc='sum').reset_index()


# Month-wise Sales Trend
plt.figure(figsize=(10, 6))
monthly_sales.groupby('month')['Total Cost'].mean().plot(kind='line')
plt.title('Month-wise Sales Trend')
plt.xlabel('Month')
plt.ylabel('Average Sales')
plt.show()
```




```
# Year-wise Sales Trend
plt.figure(figsize=(10, 6))
plt.plot(yearly_sales['year'], yearly_sales['Total Cost'], marker='o')
plt.title('Year-wise Sales Trend')
plt.xlabel('Year')
plt.ylabel('Total Sales')
plt.show()
```




```
# Yearly Month-wise Sales Heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(yearly_monthly_sales.set_index('month'), annot=True, fmt='.0f', cmap='coolwarm')
plt.title('Yearly Month-wise Sales Trend')
plt.xlabel('Year')
plt.ylabel('Month')
plt.show()
```




```
# Total Sales
total_sales = data['Total Cost'].sum()
print(f"Total Sales: ${total_sales:,.2f}")
```

```
Total Sales: $93,180,569.91
```

```
# Average Sales per Month/Year
avg_sales_per_month = data.groupby('month')['Total Cost'].mean()
avg_sales_per_year = data.groupby('year')['Total Cost'].mean()
print("Average Sales per Month:")
print(avg_sales_per_month)
print("\nAverage Sales per Year:")
print(avg_sales_per_year)
```

```
Average Sales per Month:
month
1     1.095087e+06
2     1.359113e+06
3     3.366182e+05
4     1.269664e+06
5     7.848225e+05
6     3.044946e+05
7     8.409213e+05
8     1.372221e+05
9     5.941193e+05
10    9.800594e+05
11    1.567847e+06
12    9.786464e+05
Name: Total Cost, dtype: float64

Average Sales per Year:
year
2010    1.255646e+06
2011    6.990132e+05
2012    1.031165e+06
2013    1.134586e+06
2014    7.167168e+05
2015    7.664949e+05
2016    7.469029e+05
2017    1.160508e+06
Name: Total Cost, dtype: float64
```