

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

```
df = pd.read_excel('Amazon Sales data.csv.xlsx')
df.shape
```

(100, 14)

```
df.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	Total Profit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	1158502.59	933903.84	224598.75
3	Sub-Saharan Africa	Sao Tome	Food	Online	C	2014-05-02	514001700	2014-05-08	8100	200.00	100.00	1620000.00	810000.00	810000.00

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
df.describe()
```

	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	100	1.000000e+02	100	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.000000e+02
mean	2013-09-16 14:09:36	5.550204e+08	2013-10-09 22:48:00	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.416820e+05
min	2010-02-02 00:00:00	1.146066e+08	2010-02-25 00:00:00	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.258020e+03
25%	2012-02-14 12:00:00	3.389225e+08	2012-02-24 18:00:00	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.214436e+05
50%	2013-07-12 12:00:00	5.577086e+08	2013-08-11 12:00:00	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.907680e+05

```
df.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   datetime64[ns]
6   Order ID            100 non-null   int64
7   Ship Date           100 non-null   datetime64[ns]
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: datetime64[ns](2), float64(5), int64(2), object(5)
memory usage: 11.1+ KB
```

```
df.tail()
```

	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
95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-09-03	888	109.28	35.84	97040.64	31825.92	65214.7
96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	58471.11	43367.64	15103.4
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-06-01	728815257	2016-06-29	1485	154.06	90.93	228779.10	135031.05	93748.0
98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	471336.91	326815.89	144521.0
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-02-10	665095412	2012-02-15	5367	668.27	502.54	3586605.09	2697132.18	889472.9

df.head(4)

	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	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	1158502.59	933903.84	224598.75

Next steps:

Generate code with df

View recommended plots

New interactive sheet

df.index

RangeIndex(start=0, stop=100, step=1)

df.columns

Index(['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'], dtype='object')

df.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

dtype: int64

```
df['Total Cost'].sum()
```

```
↗ 93180569.91000001
```

```
df['Total Revenue'].mean()
```

```
↗ 1373487.6831
```

```
df['Total Revenue'].memory_usage()
```

```
↗ 928
```

```
df['Total Revenue'].median()
```

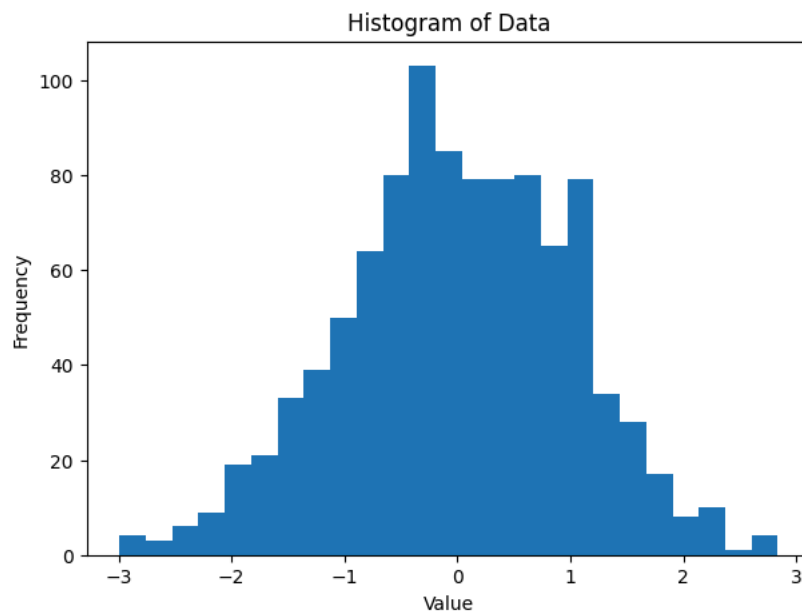
```
↗ 752314.36
```

```
df['Total Revenue'].count()
```

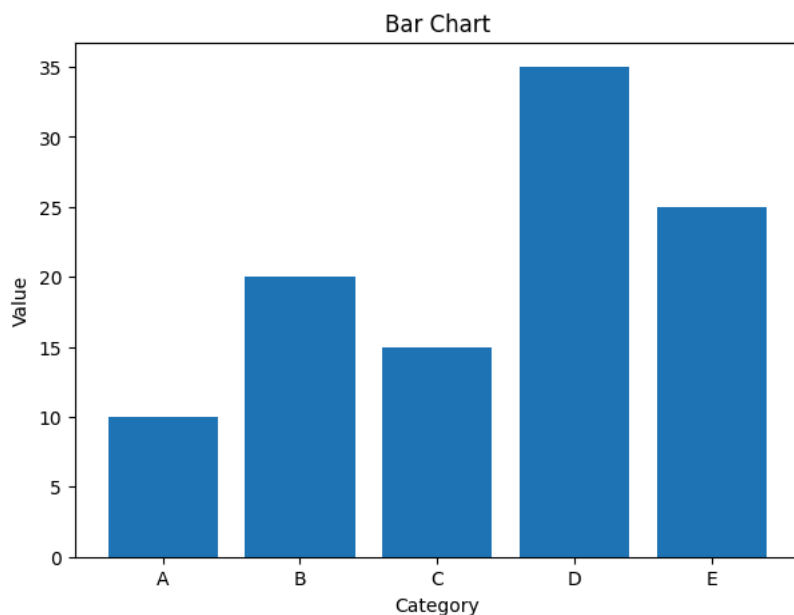
```
↗ 100
```

```
import numpy as np
data = np.random.randn(1000)
plt.figure(figsize=(7,5))
plt.hist(data,bins=25)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Data')
plt.show()
```

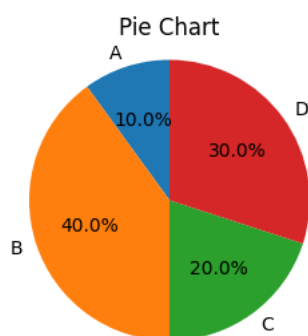
```
↗
```



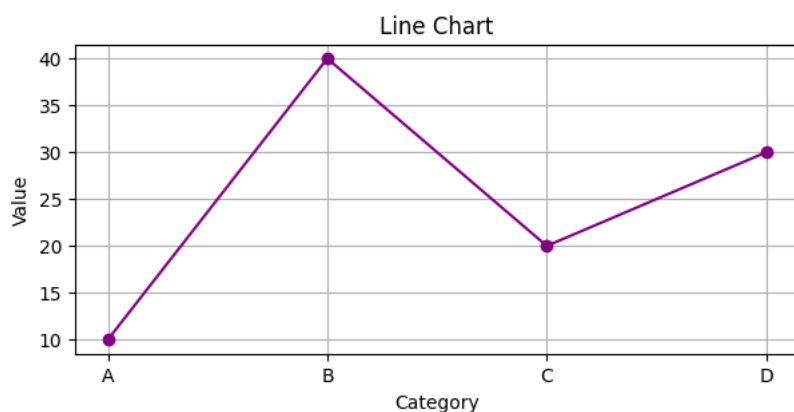
```
import matplotlib.pyplot as plt
categories = ['A', 'B', 'C', 'D', 'E']
values = [10, 20, 15, 35, 25]
plt.figure(figsize=(7,5))
plt.bar(categories, values)
plt.xlabel('Category')
plt.ylabel('Value')
plt.title('Bar Chart')
plt.show()
```



```
import matplotlib.pyplot as plt
values = [10,40,20,30]
categories = ['A','B','C','D']
plt.figure(figsize=(4,3))
plt.pie(values, labels=categories, autopct='%1.1f%%', startangle=90)
plt.axis('equal')
plt.title('Pie Chart')
plt.show()
```

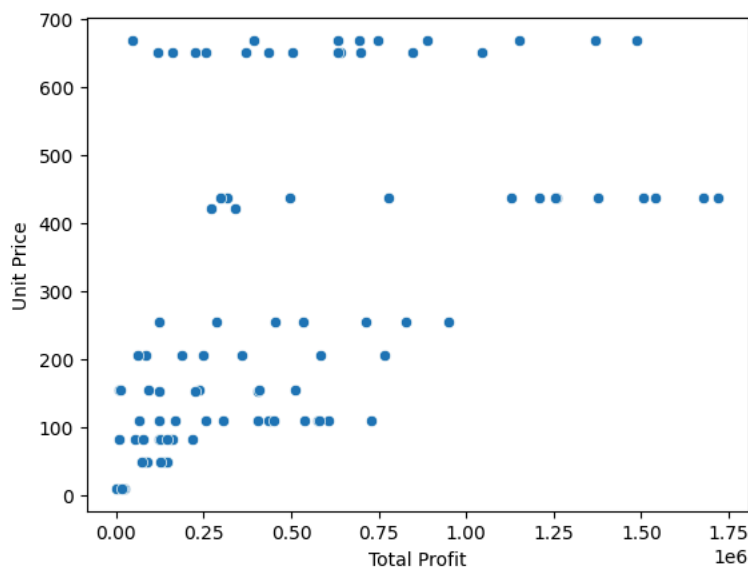


```
plt.figure(figsize=(7,3))
plt.plot(categories, values, marker='o', linestyle='-', color='purple')
plt.xlabel('Category')
plt.ylabel('Value')
plt.title('Line Chart')
plt.grid(True)
plt.show()
```



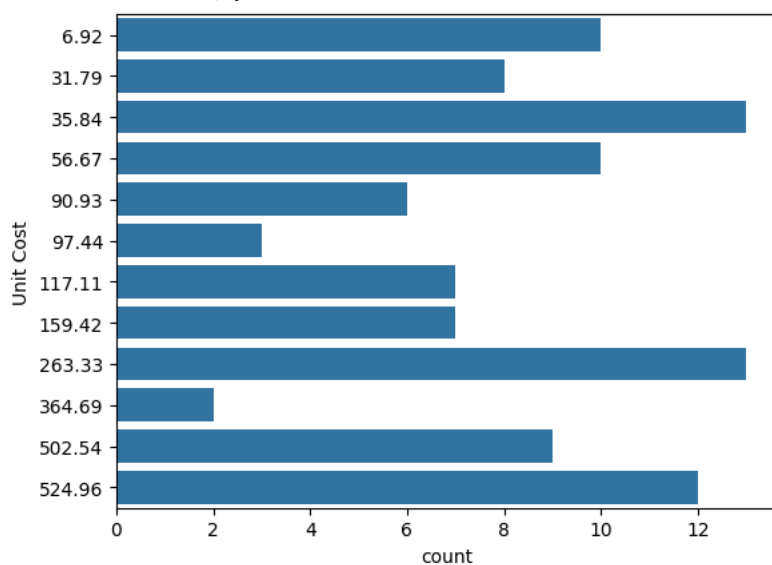
```
sns.scatterplot(x='Total Profit', y='Unit Price', data=df)
```

<Axes: xlabel='Total Profit', ylabel='Unit Price'>



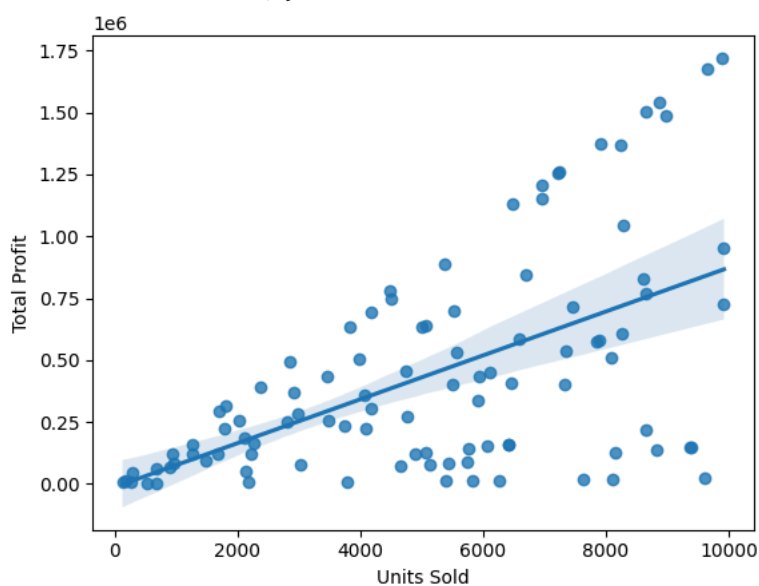
```
sns.countplot(y='Unit Cost', data=df)
```

<Axes: xlabel='count', ylabel='Unit Cost'>



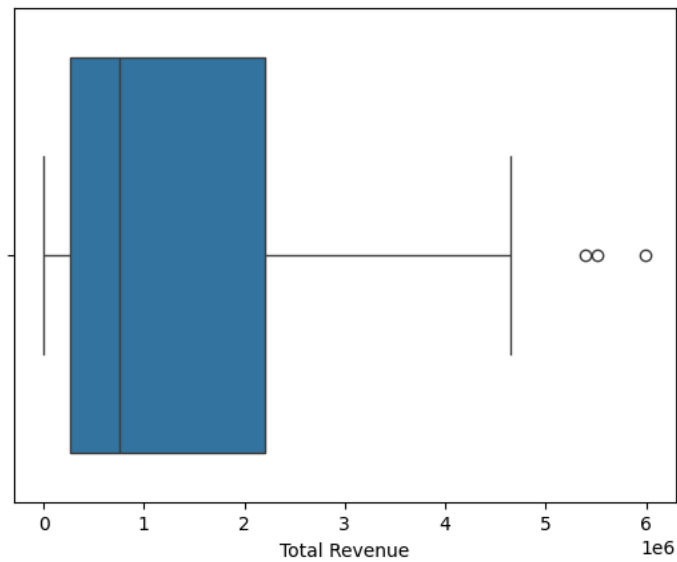
```
sns.regplot(x='Units Sold', y='Total Profit', data=df)
```

<Axes: xlabel='Units Sold', ylabel='Total Profit'>



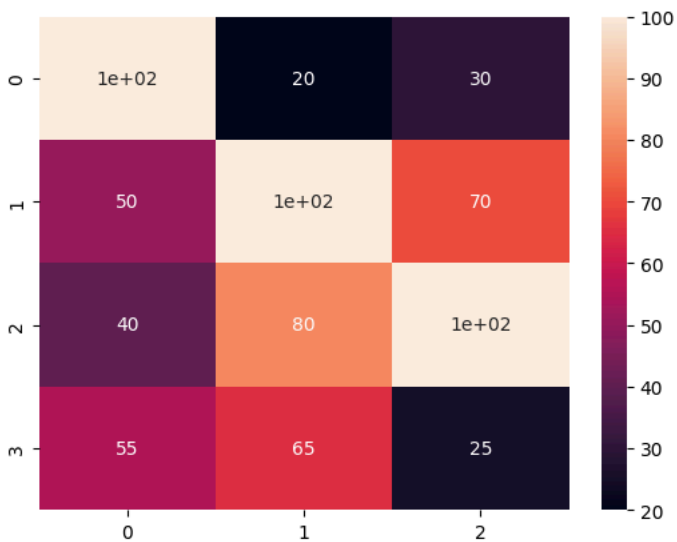
```
sns.boxplot(x='Total Revenue', data=df)
```

<Axes: xlabel='Total Revenue'>



```
sns.heatmap([[100,20,30],
             [50,100,70],
             [40,80,100],
             [55,65,25]], annot=True)
```

<Axes: >

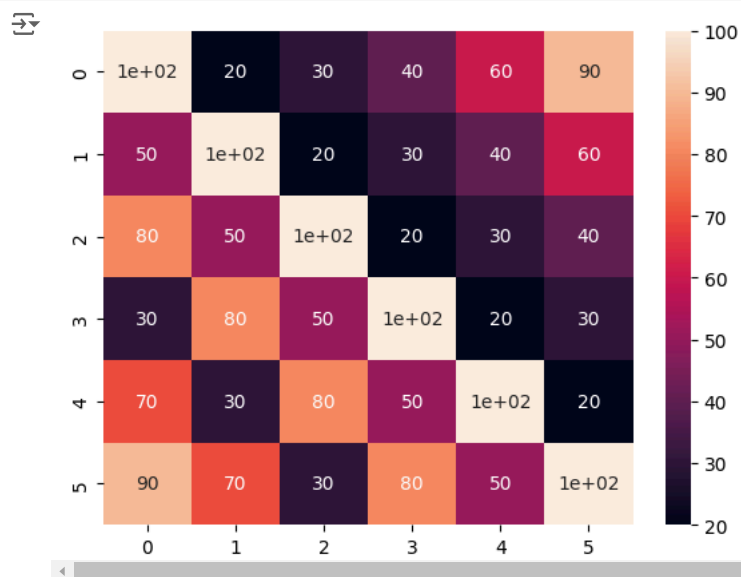


```
sns.boxplot(x='Total Profit', data=df)
```

```
df['year'].label='Total Profit'
```

```
data= [[100,20,30,40,60,90],
        [50,100,20,30,40,60],
        [80,50,100,20,30,40],
        [30,80,50,100,20,30],
        [70,30,80,50,100,20],
        [90,70,30,80,50,100]]
```

```
sns.heatmap(data, annot=True)
plt.show()
```



```
years = np.arange(1990,2061)
data = np.random.randint(0, years.size, size=(years.size, 3))
df = pd.DataFrame(data, columns=list('ABC'))
df['year'] = years
df.plot.area(x='year', y=['A','B','C'], stacked=True)
plt.show()
```

