In [3]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import missingno as msno
```

In [5]:
```
data = pd.read_csv("file:///C:/Users/Vasu%20Prasad/OneDrive/Documents/MCA/Internships/Oasis%20Infobyte/Explora
data
```

Out[5]:

	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
...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

1000 rows × 9 columns

In [6]:
```
pd.concat([data.head(),data.tail()])
```

Out[6]:

	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
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120

In [7]:
```
data.info()
```

```
<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
```

In [8]:
```
data.describe()
```

Out[8]:

	Transaction ID	Age	Quantity	Price per Unit	Total Amount
count	1000.000000	1000.00000	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

In [9]:
```
data.isnull().sum().reset_index().rename(columns = {0:"count"})
```

Out[9]:

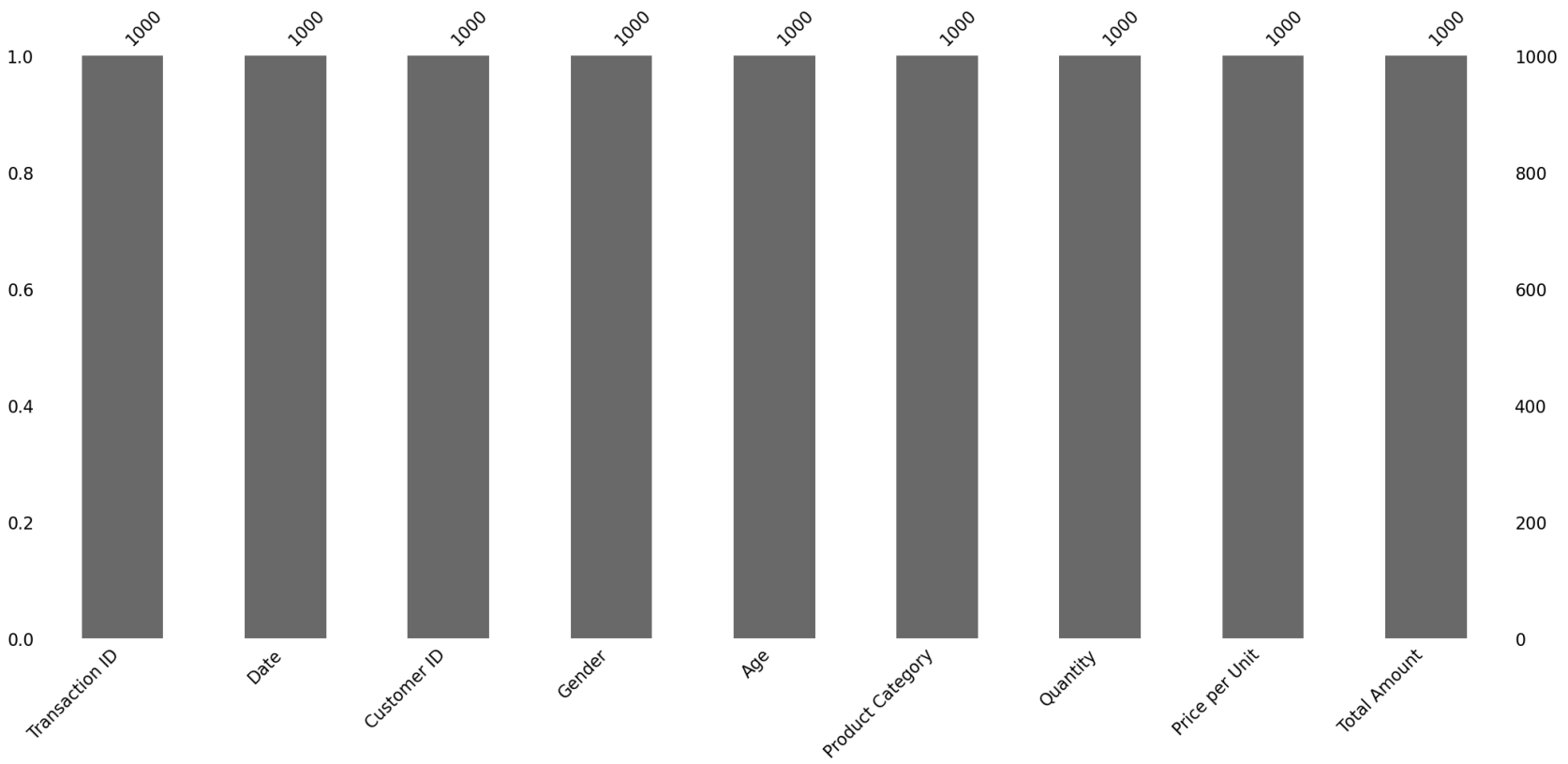
	index	count
0	Transaction ID	0
1	Date	0
2	Customer ID	0
3	Gender	0
4	Age	0
5	Product Category	0
6	Quantity	0
7	Price per Unit	0
8	Total Amount	0

In [10]:
```
data.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
msno.bar(data)
```

Out[11]: <Axes: >



In [12]:
```
data.columns
```

Out[12]:
```
Index(['Transaction ID', 'Date', 'Customer ID', 'Gender', 'Age',
       'Product Category', 'Quantity', 'Price per Unit', 'Total Amount'],
      dtype='object')
```

In [13]:
```
data.dtypes
```

Out[13]:
```
Transaction ID       int64
Date                object
Customer ID         object
Gender              object
Age                  int64
Product Category    object
Quantity             int64
Price per Unit       int64
Total Amount         int64
dtype: object
```

In [14]:
```
data['Date'] = pd.to_datetime(data["Date"])
data['Product Category'] = data['Product Category'].astype("category")
```

In [15]:
```
data.dtypes
```

Out[15]:
```
Transaction ID                int64
Date                 datetime64[ns]
Customer ID                  object
Gender                       object
Age                           int64
Product Category           category
Quantity                      int64
Price per Unit                int64
Total Amount                  int64
dtype: object
```

In [16]:
```
data['Date'].describe(datetime_is_numeric=True)
```

Out[16]:
```
count                             1000
mean     2023-07-03 00:25:55.200000256
min                2023-01-01 00:00:00
25%                2023-04-08 00:00:00
50%                2023-06-29 12:00:00
75%                2023-10-04 00:00:00
max                2024-01-01 00:00:00
Name: Date, dtype: object
```

In [17]:
```
data[["Gender","Age"]].groupby("Gender").min().reset_index()
```

Out[17]:

	Gender	Age
0	Female	18
1	Male	18

In [18]:
```
data[["Gender","Age"]].groupby("Gender").max().reset_index()
```

Out[18]:

	Gender	Age
0	Female	64
1	Male	64

In [19]:
```
Gender_counts = data[["Gender"]].value_counts().reset_index().rename(columns = {0:"Total number of peoples"})
Gender_counts
```

Out[19]:

	Gender	Total number of peoples
0	Female	510
1	Male	490

In [20]:
```
plt.figure(figsize = (6,5))
sns.barplot(x = Gender_counts['Gender'],y = Gender_counts['Total number of peoples'],
            data = Gender_counts,palette = 'rocket',width = 0.3,hue = "Gender")
plt.title("Total number of peoples come to shop by gender")

plt.show()
```

In [21]:
```
pd.unique(data['Product Category'])
```

Out[21]:
```
['Beauty', 'Clothing', 'Electronics']
Categories (3, object): ['Beauty', 'Clothing', 'Electronics']
```

In [22]:
```
data.drop("Transaction ID",inplace = True,axis = 1)
```

In [23]:
```
Total_sales = data[["Product Category","Total Amount"]]
Total_sales
```

Out[23]:

	Product Category	Total Amount
0	Beauty	150
1	Clothing	1000
2	Electronics	30
3	Clothing	500
4	Beauty	100
...	...	...
995	Clothing	50
996	Beauty	90
997	Beauty	100
998	Electronics	150
999	Electronics	120

1000 rows × 2 columns

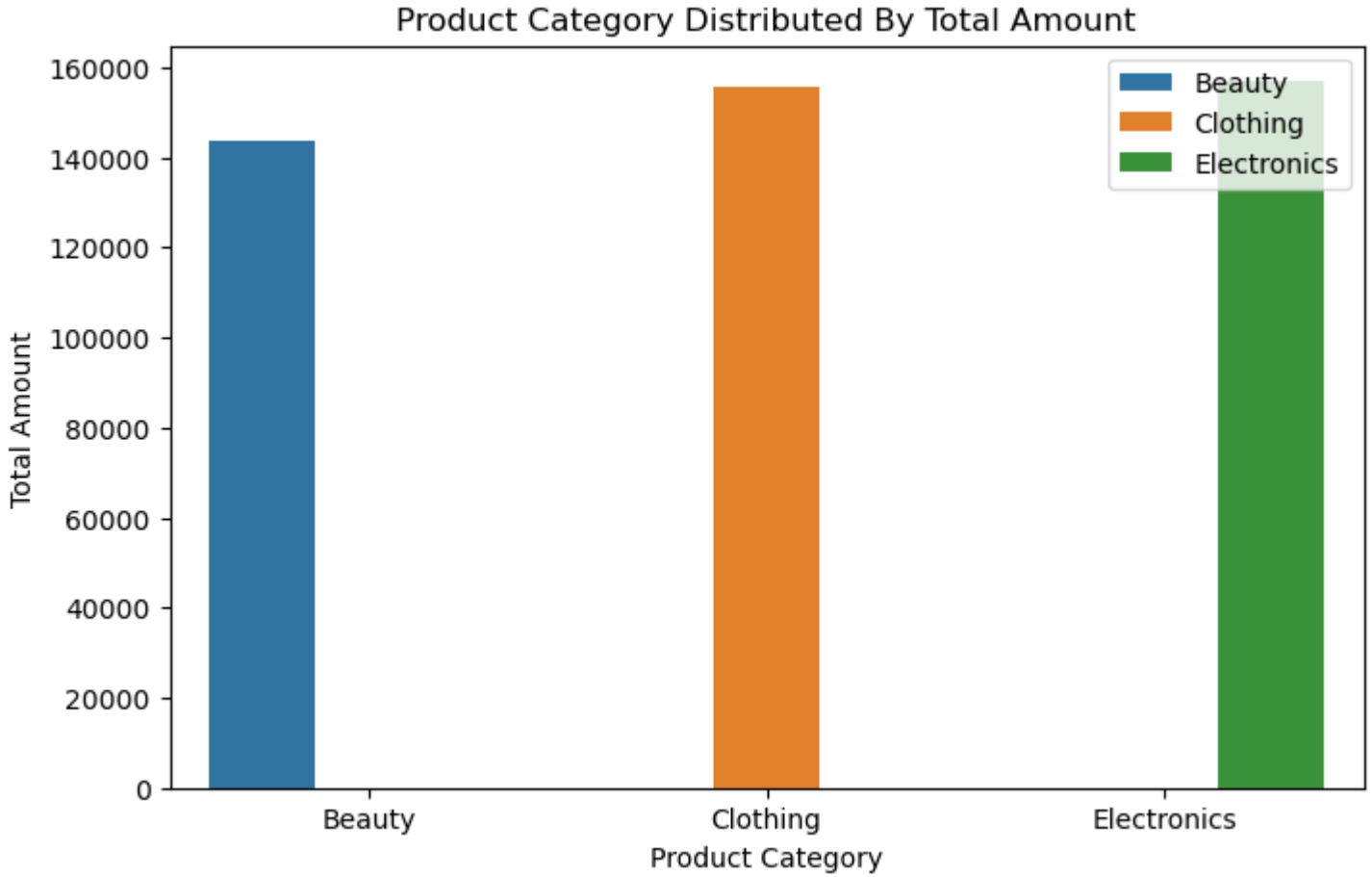
In [24]:
```
Total_sales = Total_sales.groupby(by = ["Product Category"]).sum().reset_index()
```

In [25]:
```
Total_sales
```

Out[25]:

	Product Category	Total Amount
0	Beauty	143515
1	Clothing	155580
2	Electronics	156905

In [26]:
```
plt.figure(figsize = (8,5))
sns.barplot(x = Total_sales["Product Category"],y = Total_sales["Total Amount"],
            data = Total_sales,hue = "Product Category",width = 0.8)
plt.legend(loc=1)
plt.title("Product Category Distributed By Total Amount")
plt.show()
```



In [27]:
```
Product_quantity = data[["Product Category","Quantity"]]
Product_quantity
```

Out[27]:

	Product Category	Quantity
0	Beauty	3
1	Clothing	2
2	Electronics	1
3	Clothing	1
4	Beauty	2
...	...	...
995	Clothing	1
996	Beauty	3
997	Beauty	4
998	Electronics	3
999	Electronics	4

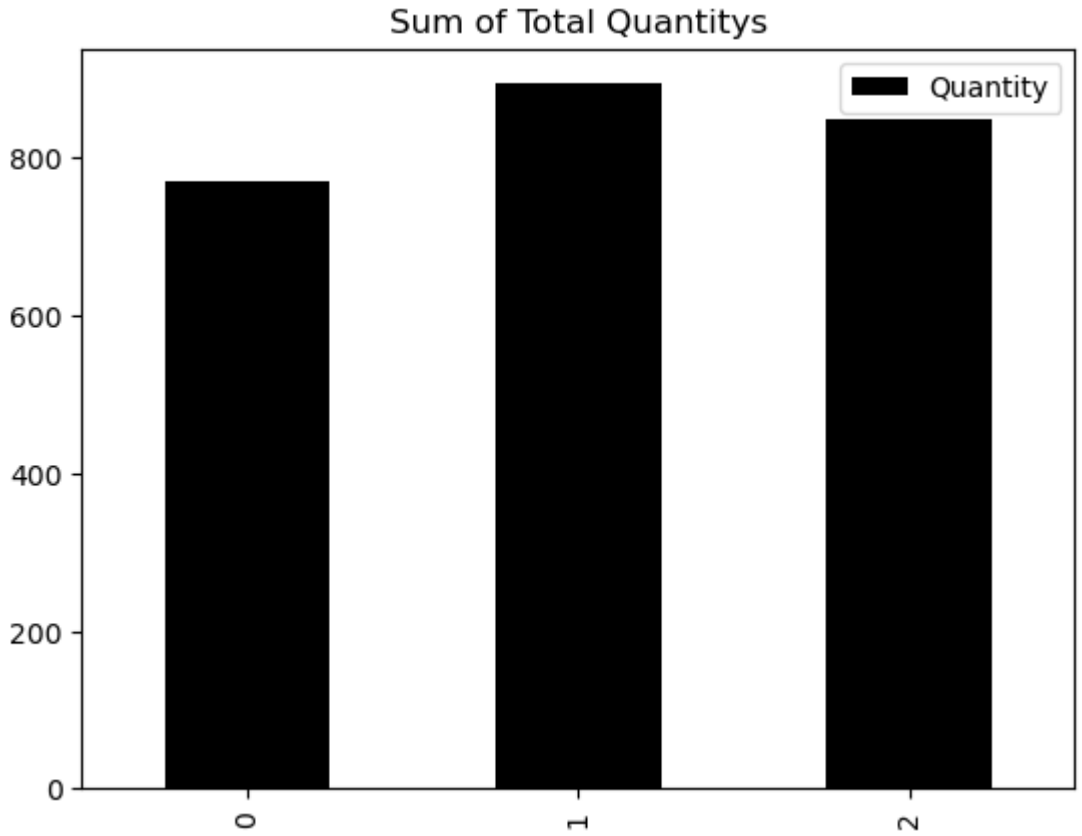
1000 rows × 2 columns

In [28]:
```
Total_quantity = Product_quantity.groupby(by = ["Product Category"]).sum().reset_index()
Total_quantity
```

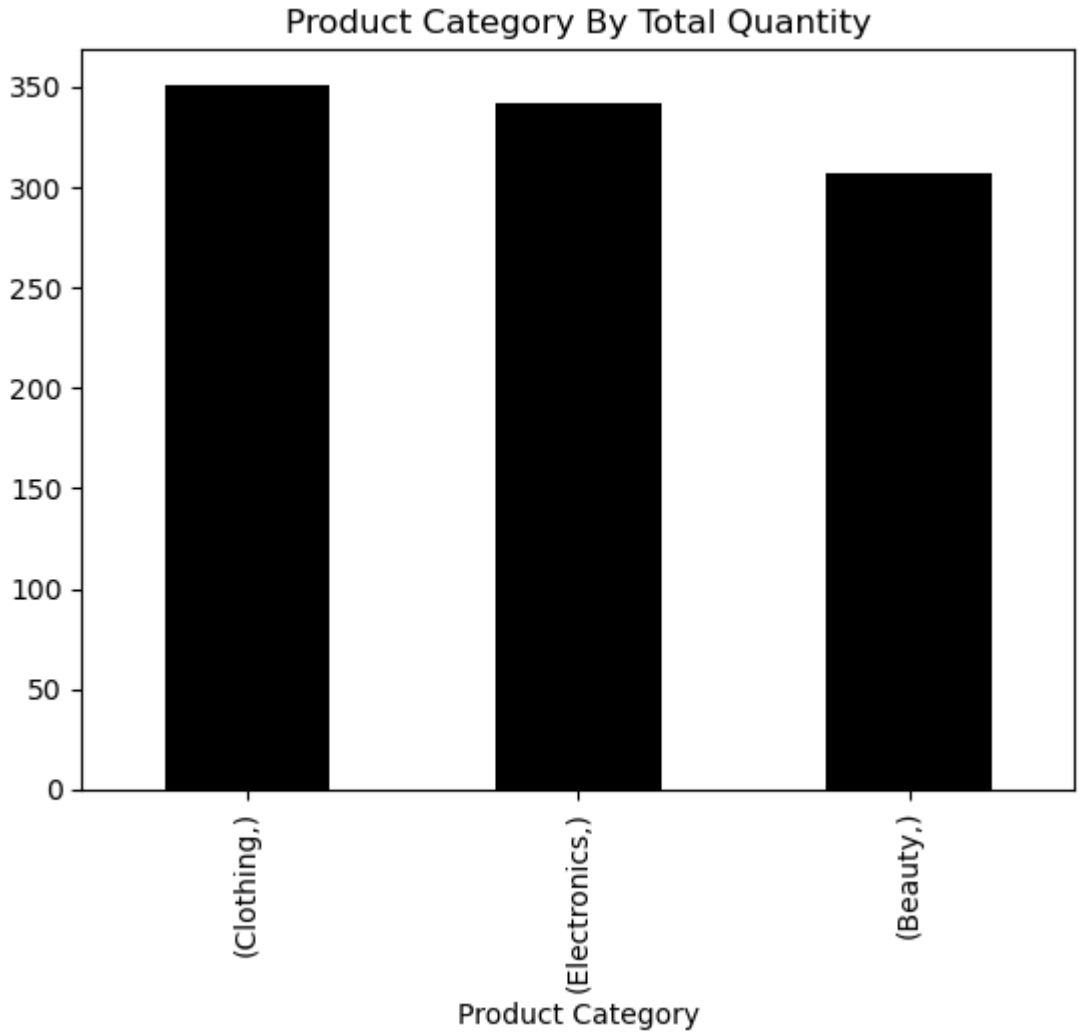
Out[28]:

	Product Category	Quantity
0	Beauty	771
1	Clothing	894
2	Electronics	849

In [29]:
```
Total_quantity.plot(kind  = 'bar',color = 'black')
plt.title("Sum of Total Quantitys")
plt.show()
```



In [30]:
```
data[["Product Category"]].value_counts().plot(kind = 'bar',color = 'black')
plt.title("Product Category By Total Quantity")
plt.show()
```

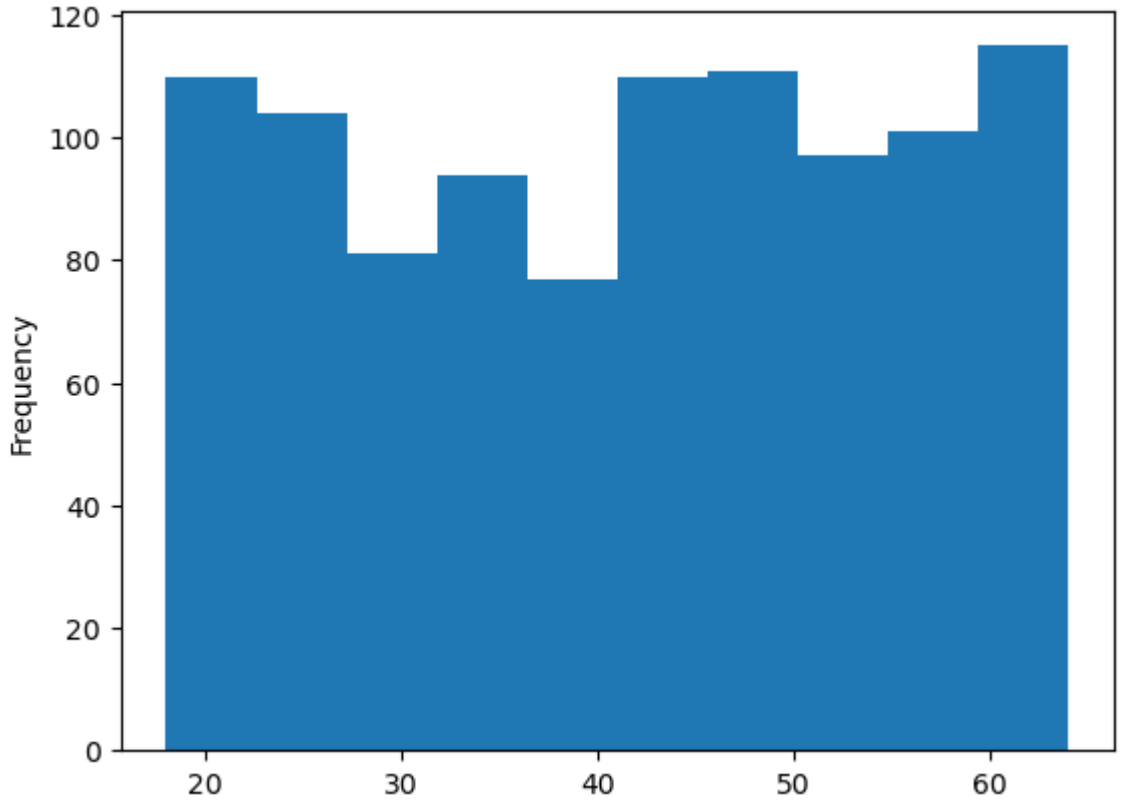


In [31]:
```
data["Age"].describe()
```

Out[31]:
```
count    1000.00000
mean       41.39200
std        13.68143
min        18.00000
25%        29.00000
50%        42.00000
75%        53.00000
max        64.00000
Name: Age, dtype: float64
```

In [32]:
```
data['Age'].plot(kind = 'hist')
```

Out[32]:
```
<Axes: ylabel='Frequency'>
```

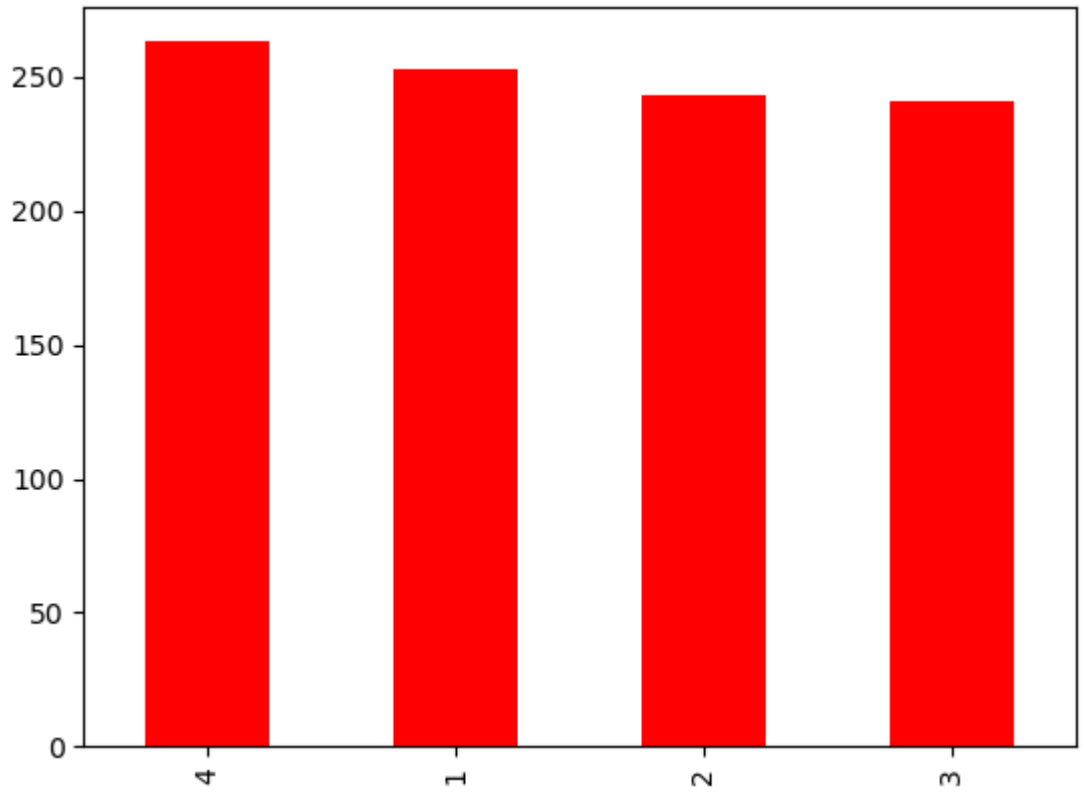


In [33]:
```
data["Quantity"].describe()
```

Out[33]:
```
count    1000.000000
mean        2.514000
std         1.132734
min         1.000000
25%         1.000000
50%         3.000000
75%         4.000000
max         4.000000
Name: Quantity, dtype: float64
```

In [34]:
```
data['Quantity'].value_counts().plot(kind = 'bar',color = 'red')
```

Out[34]:
```
<Axes: >
```



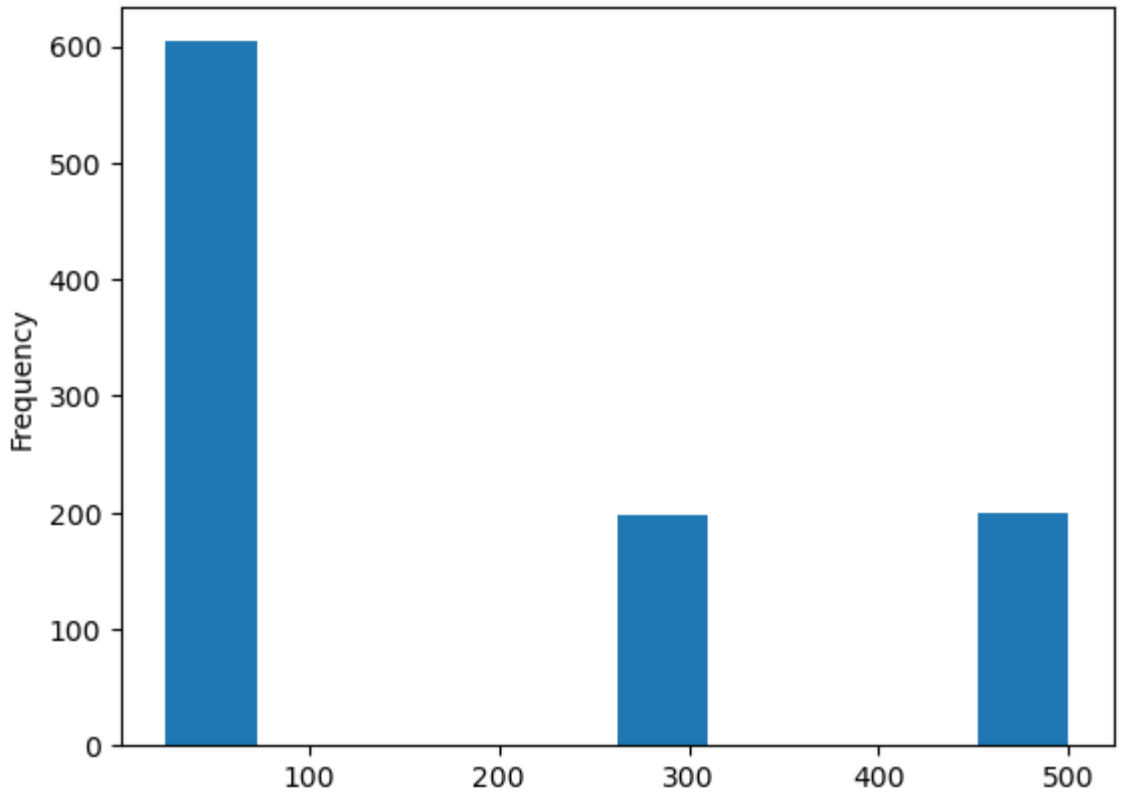
In [35]:
```
data[["Price per Unit"]].describe()
```

Out[35]:

	Price per Unit
count	1000.000000
mean	179.890000
std	189.681356
min	25.000000
25%	30.000000
50%	50.000000
75%	300.000000
max	500.000000

In [36]:
```
data['Price per Unit'].plot(kind = 'hist')
```

Out[36]: <Axes: ylabel='Frequency'>

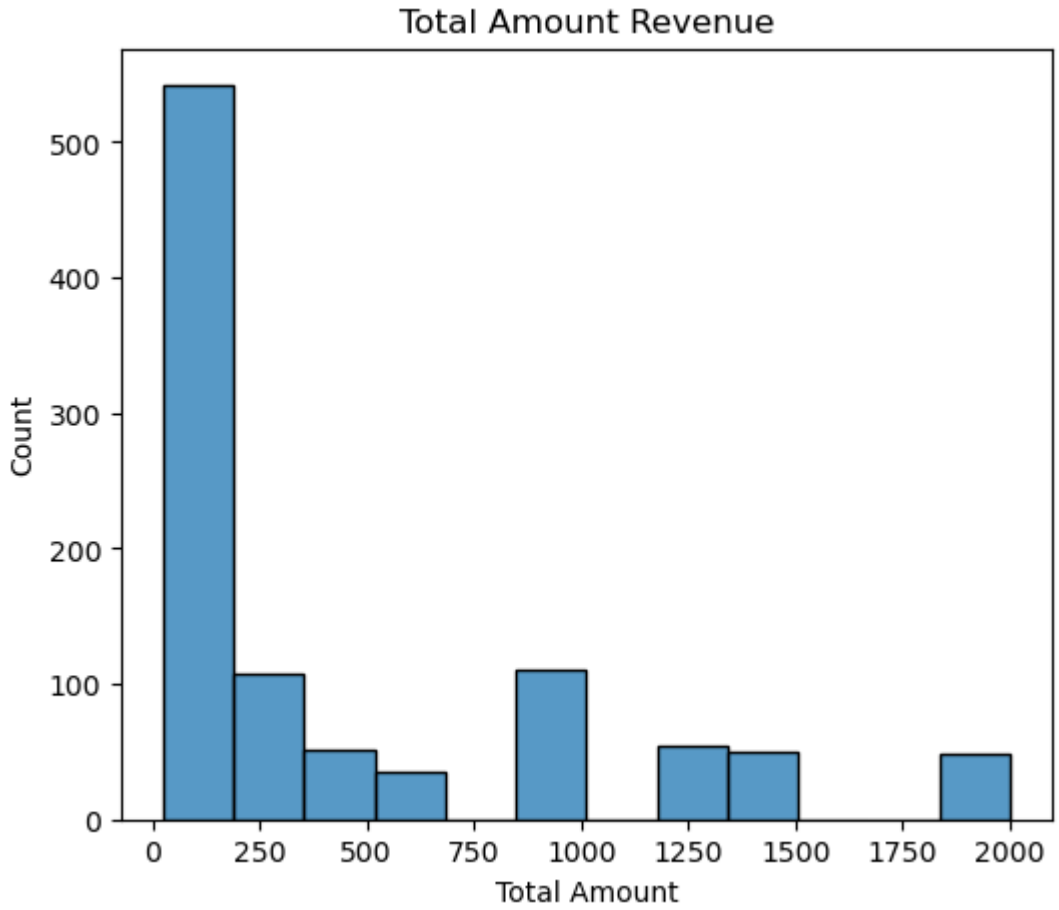


In [37]:
```
data[["Total Amount"]].describe()
```

Out[37]:

	Total Amount
count	1000.000000
mean	456.000000
std	559.997632
min	25.000000
25%	60.000000
50%	135.000000
75%	900.000000
max	2000.000000

In [38]:
```
plt.figure(figsize = (6,5))
sns.histplot(data = data['Total Amount'])
plt.title("Total Amount Revenue")
plt.show()
```

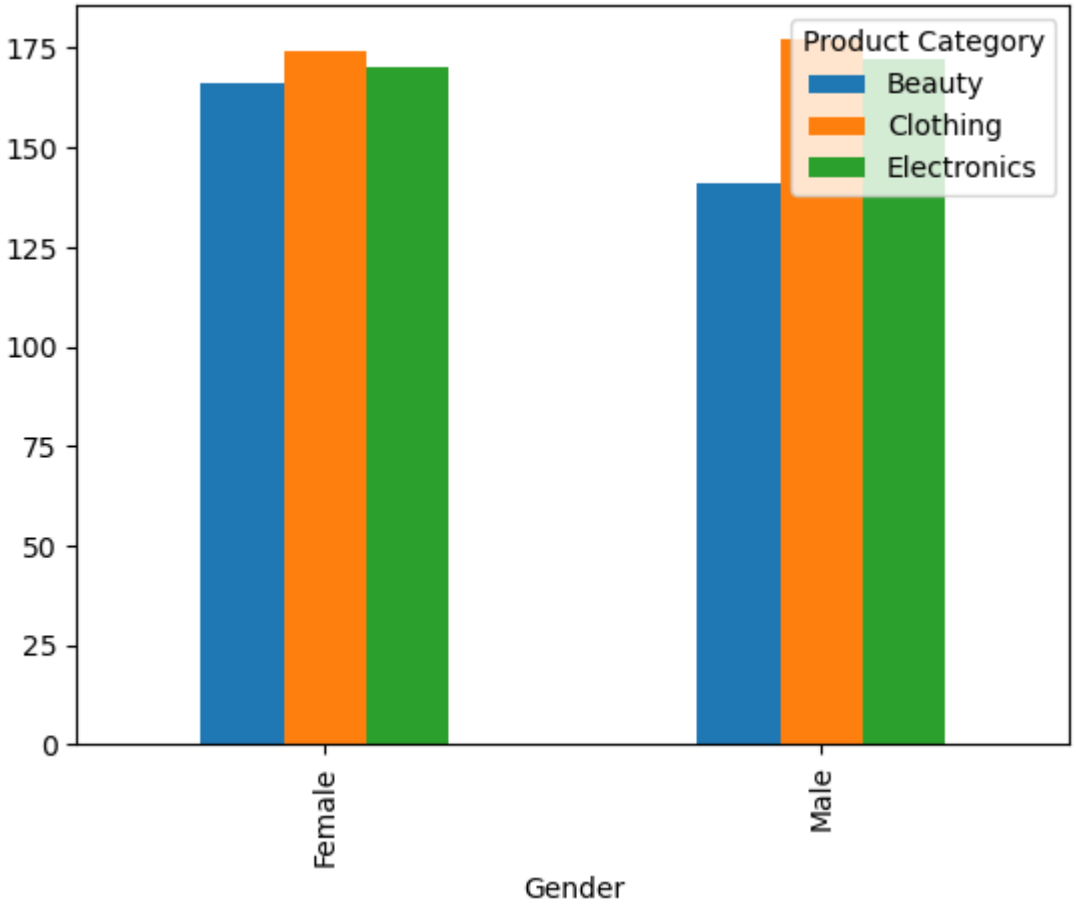


In [39]:
```
pd.crosstab(data['Gender'],data['Product Category'])
```

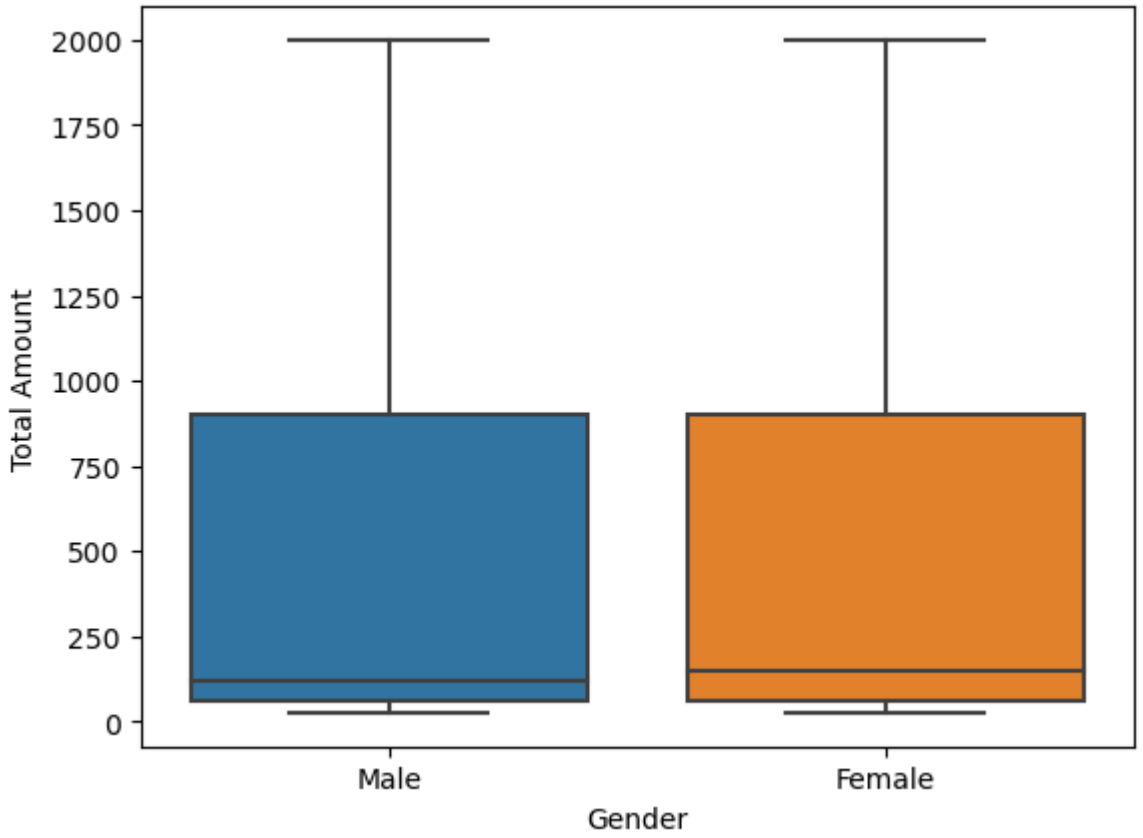
Out[39]:

Product Category	Beauty	Clothing	Electronics
Gender			
Female	166	174	170
Male	141	177	172

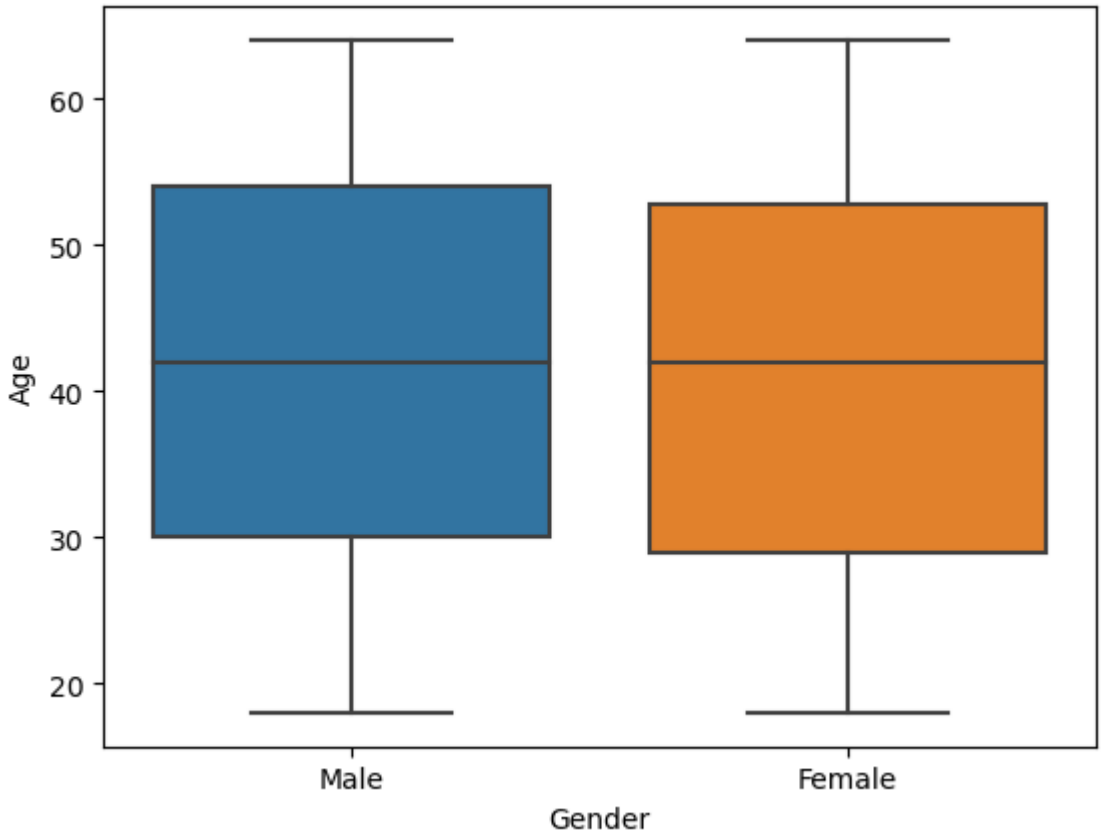
In [40]:
```
pd.crosstab(data['Gender'],data['Product Category']).plot(kind = 'bar')
plt.show()
```



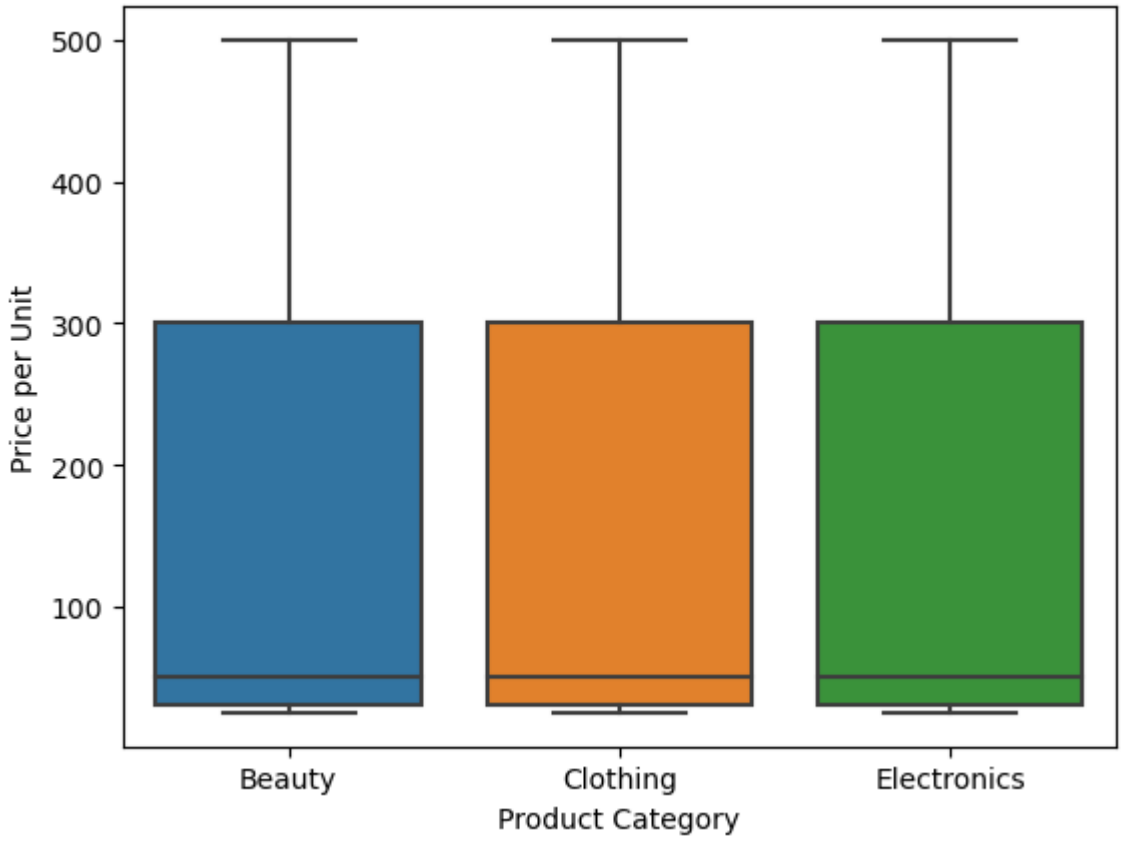
In [41]:
```
sns.boxplot(data = data,x = 'Gender',y = 'Total Amount')
plt.show()
```



In [42]:
```
sns.boxplot(data = data,x = 'Gender', y = 'Age' )
plt.show()
```



In [43]:
```
sns.boxplot(data = data,x = 'Product Category',y = 'Price per Unit' )
plt.show()
```



In [ ]: