In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv("C:\\Users\\Hp\\OneDrive\\Desktop\\e commerce sales\\raw data s
df
```

Out[3]:

	customer_id	city	state	registration_date	order_date	order_id	ord
0	CUST0001	Lake Amyshire	Maryland	11-06-2025	05-10-2024	ORD03260	
1	CUST0001	Lake Amyshire	Maryland	11-06-2025	06-12-2024	ORD02997	
2	CUST0001	Lake Amyshire	Maryland	11-06-2025	06-12-2024	ORD02997	
3	CUST0001	Lake Amyshire	Maryland	11-06-2025	06-12-2024	ORD02997	
4	CUST0001	Lake Amyshire	Maryland	11-06-2025	06-12-2024	ORD02997	
...	...	...	...	...	...	...	...
10442	CUST0150	Port Bethview	Mississippi	02-03-2025	31-05-2025	ORD00816	
10443	CUST0150	Port Bethview	Mississippi	02-03-2025	31-05-2025	ORD00816	
10444	CUST0150	Port Bethview	Mississippi	02-03-2025	31-10-2024	ORD01554	
10445	CUST0150	Port Bethview	Mississippi	02-03-2025	31-10-2024	ORD01554	
10446	CUST0150	Port Bethview	Mississippi	02-03-2025	31-10-2024	ORD01554	

10447 rows × 26 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10447 entries, 0 to 10446
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   customer_id        10447 non-null  object 
 1   city               10447 non-null  object 
 2   state              10447 non-null  object 
 3   registration_date  10447 non-null  object 
 4   order_date         10447 non-null  object 
 5   order_id           10447 non-null  object 
 6   order_item_id      10447 non-null  object 
 7   product_id         10447 non-null  object 
 8   quantity           10447 non-null  int64  
 9   unit_price         10447 non-null  float64
 10  total_price        10447 non-null  float64
 11  total_amount       10447 non-null  float64
 12  status             10447 non-null  object 
 13  delivery_date      10447 non-null  object 
 14  name               10447 non-null  object 
 15  category           10447 non-null  object 
 16  brand              10447 non-null  object 
 17  price              10447 non-null  float64
 18  stock_quantity     10447 non-null  int64  
 19  added_date         10447 non-null  object 
 20  rating             10447 non-null  int64  
 21  payment_id         10447 non-null  object 
 22  payment_date       10447 non-null  object 
 23  payment_method     10447 non-null  object 
 24  payment_status     10447 non-null  object 
 25  amount_paid        10447 non-null  float64
dtypes: float64(5), int64(3), object(18)
memory usage: 2.1+ MB
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
customer_id          0
city                 0
state                0
registration_date    0
order_date           0
order_id             0
order_item_id        0
product_id           0
quantity             0
unit_price           0
total_price          0
total_amount         0
status               0
delivery_date        0
name                 0
category             0
brand                0
price                0
stock_quantity       0
added_date           0
rating               0
payment_id           0
payment_date         0
payment_method       0
payment_status       0
amount_paid          0
dtype: int64
```

In [6]:
```
df.duplicated()
```

Out[6]:
```
0        False
1        False
2        False
3        False
4        False
         ...
10442    False
10443    False
10444    False
10445    False
10446    False
Length: 10447, dtype: bool
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['customer_id', 'city', 'state', 'registration_date', 'order_date',
       'order_id', 'order_item_id', 'product_id', 'quantity', 'unit_price',
       'total_price', 'total_amount', 'status', 'delivery_date', 'name',
       'category', 'brand', 'price', 'stock_quantity', 'added_date', 'rating',
       'payment_id', 'payment_date', 'payment_method', 'payment_status',
       'amount_paid'],
      dtype='object')
```

In [8]:
```
df1 = df.groupby([ 'category', 'brand'] , as_index = False)['total_price'].sum()
df1
```

Out[8]:

	category	brand	total_price
93	Home	Waller-Murray	1817956.03
52	Electronics	Cummings, Thomas and Sanchez	1817121.88
112	Sports	Reed PLC	1807864.24
31	Beauty	Joseph, Davis and Blake	1704673.25
47	Electronics	Boyd-Peterson	1681778.39
85	Home	Olson-Aguilar	1675906.40
96	Home	Williams, Sutton and Lopez	1659782.87
0	Apparel	Andrews-Robinson	1645324.00
97	Home	Young-Coleman	1574986.00
28	Beauty	George Ltd	1562844.78
19	Apparel	Thompson-Smith	1530447.27
27	Beauty	Gates, Hayden and Hines	1484856.50
89	Home	Ruiz Inc	1478290.80
30	Beauty	Johnson-Jones	1455222.08
62	Electronics	Lee, Roberts and Young	1438491.45
45	Electronics	Benson and Sons	1421897.30
12	Apparel	Li, Phillips and Hudson	1325827.98
34	Beauty	Martin-Dennis	1309441.32
1	Apparel	Banks, Brown and Cardenas	1274157.82
115	Sports	Stephenson, Johnson and Garcia	1252126.88

In [9]:
```
df2 = df.groupby('city', as_index = False)['total_price'].sum().sort_values(by =
df2
```

Out[9]:

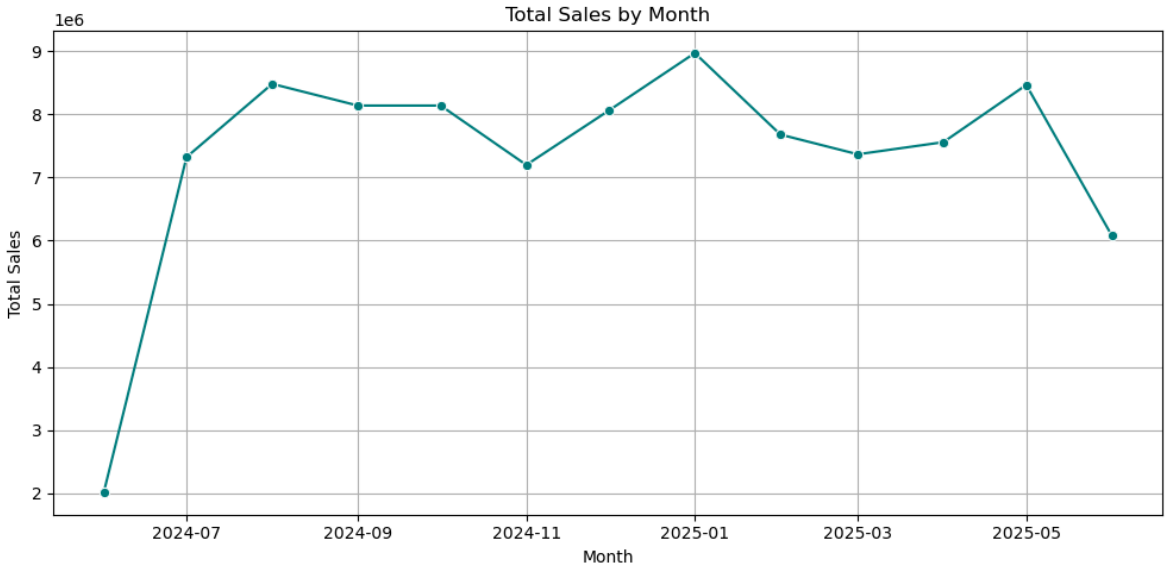
	city	total_price
138	West Christopher	1548450.38
5	Barbarafurt	1260092.92
116	South Mitchellport	1081561.44
148	Woodsview	1075556.18
3	Andreastad	991714.24
17	Davisshire	975257.42
65	Maryton	941892.71
60	Lake Ryan	905994.08
43	Jonathanfort	902004.50
123	Sydneyberg	895076.51
93	Port Loriport	891061.78
142	West Johnton	888452.93
76	New Jeffrey	876994.40
26	Estradaside	866646.09
56	Lake Emilyport	865695.80
94	Port Martin	865023.21
140	West Connie	856159.17
77	New Nicholas	844365.05
12	Carmenshire	816739.10
20	East Michaelview	804615.70

Total Sales by Month

In [10]:
```
df['order_date'] = pd.to_datetime(df['order_date'] , dayfirst=True)
df['month'] = df['order_date'].dt.to_period('M')


gf = df.groupby('month', as_index=False)['total_price'].sum()
gf['month'] = gf['month'].dt.to_timestamp()

plt.figure(figsize=(10, 5))
sns.lineplot( x= 'month', y = 'total_price', data = gf , marker = "o" , color='t
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.title("Total Sales by Month")
plt.grid()
plt.tight_layout()
plt.show()
```



In [11]:
```
df['total_price'].sum()
```

Out[11]: 95446459.42000002

top 10 customers

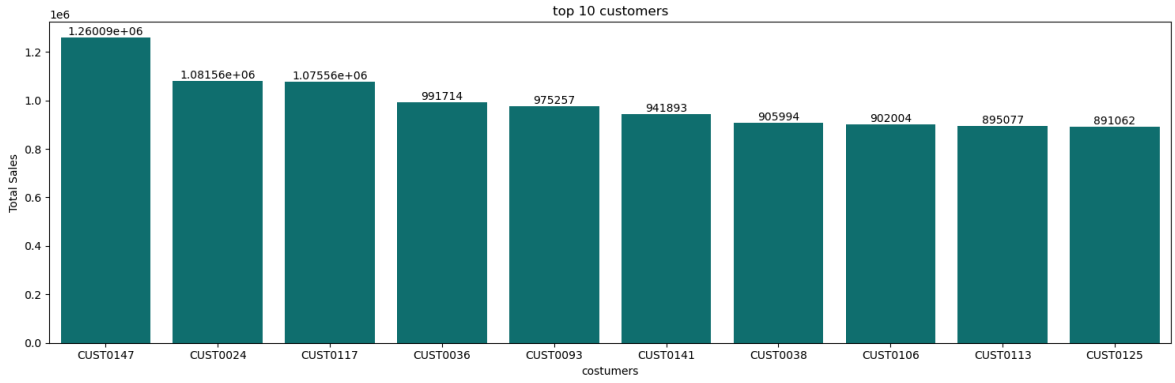
e+06 = 1000000

1e6 = 1000000

In [12]:
```
gf = df.groupby('customer_id', as_index=False)['total_price'].sum().sort_values(

plt.figure(figsize=(15, 5))
ax = sns.barplot( x = 'customer_id', y = 'total_price', data = gf , color='teal'
ax.bar_label(ax.containers[0])

plt.xlabel("costumers")
plt.ylabel("Total Sales")
plt.title("top 10 customers")
plt.tight_layout()
plt.show()
```

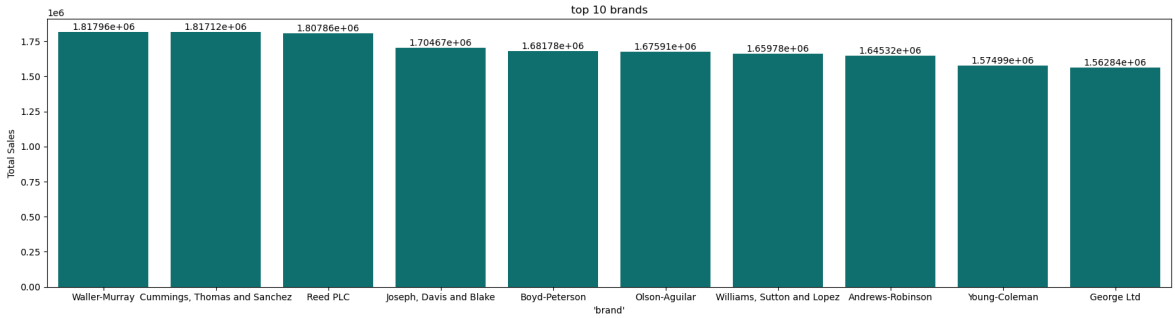


"top 10 brands"

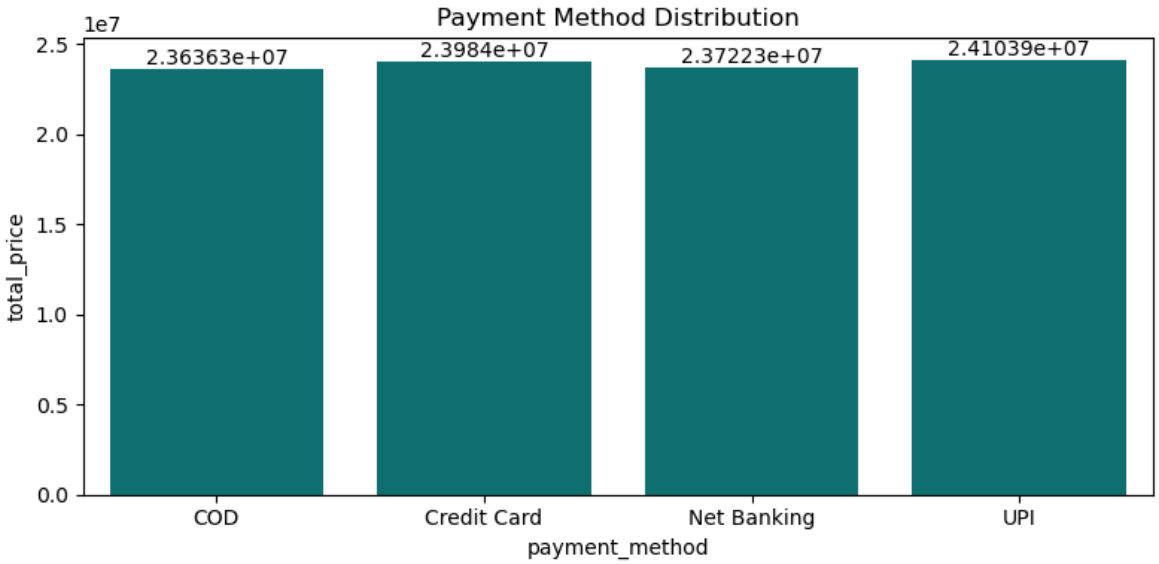
In [13]:
```
gf = df.groupby('brand', as_index=False)['total_price'].sum().sort_values(by='to

plt.figure(figsize=(18, 5))
ax = sns.barplot( x = 'brand', y = 'total_price', data = gf , color='teal')
ax.bar_label(ax.containers[0])

plt.xlabel("'brand'")
plt.ylabel("Total Sales")
plt.title("top 10 brands")
plt.tight_layout()
plt.show()
```



In [14]:
```
gf = df.groupby('payment_method' , as_index = False)['total_price'].sum()
# plt.pie(x = 'payment_method' , data= gf)
plt.figure(figsize=(8,4))
ax = sns.barplot(x  = 'payment_method' , y = 'total_price' , data = gf , color =
ax.bar_label(ax.containers[0])
plt.title('Payment Method Distribution')
plt.tight_layout()
plt.show()
```



'Payment Method Distribution'

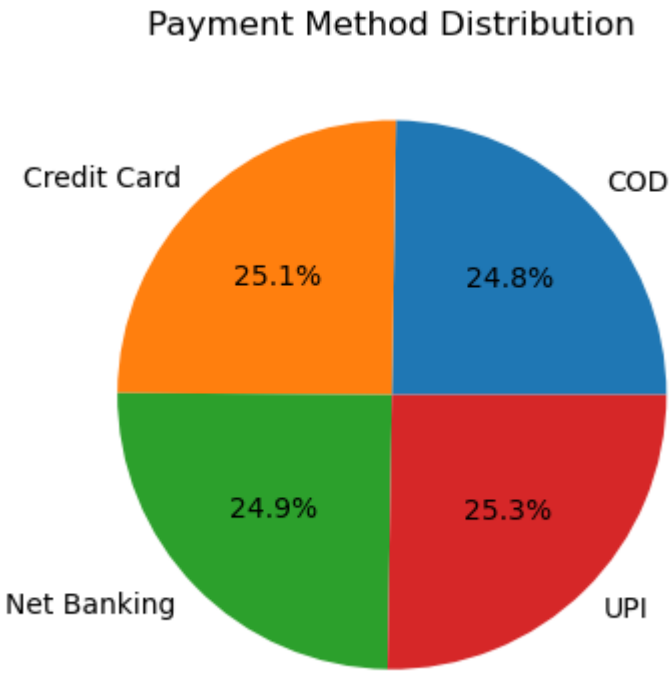
In [15]:
```
gf = df.groupby('payment_method' , as_index = False)['total_price'].sum().sort_v
gf
```

Out[15]:

	payment_method	total_price
0	COD	23636273.17
2	Net Banking	23722342.36
1	Credit Card	23983958.72
3	UPI	24103885.17

In [16]:
```
gf = df.groupby('payment_method' , as_index = False)['total_price'].sum()

plt.figure(figsize=(4,4))
plt.pie(gf['total_price'], labels=gf['payment_method'], autopct='%1.1f%%'  )
plt.title('Payment Method Distribution')
plt.tight_layout()
plt.show()
```



category Distribution

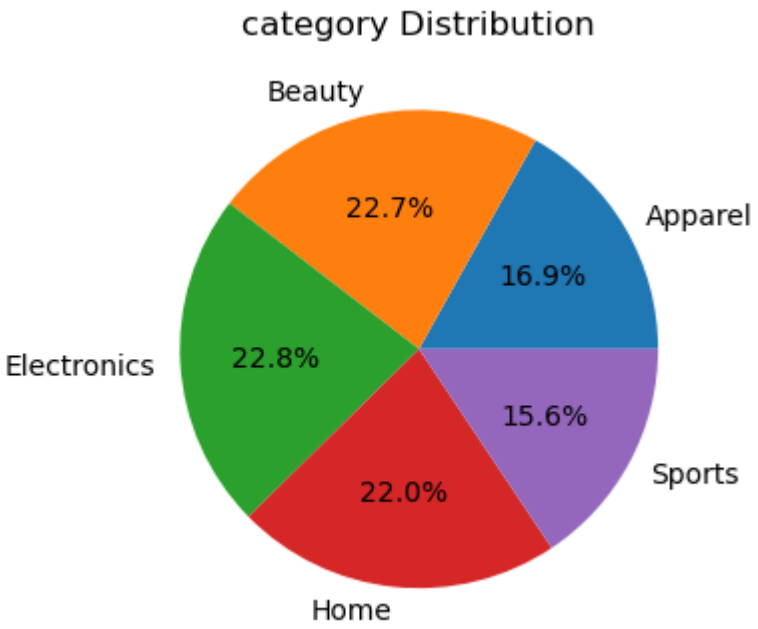
In [17]:
```
gf = df.groupby('category' , as_index = False)['total_price'].sum().sort_values(
gf
```

Out[17]:

	category	total_price
4	Sports	14923429.34
0	Apparel	16128403.98
3	Home	21003862.67
1	Beauty	21636417.40
2	Electronics	21754346.03

In [18]:
```
gf = df.groupby('category' , as_index = False)['total_price'].sum()

plt.figure(figsize=(4,4))
plt.pie(gf['total_price'], labels=gf['category'], autopct='%1.1f%%'  )
plt.title('category Distribution')
plt.tight_layout()
plt.show()
```



'payment_status Distribution'

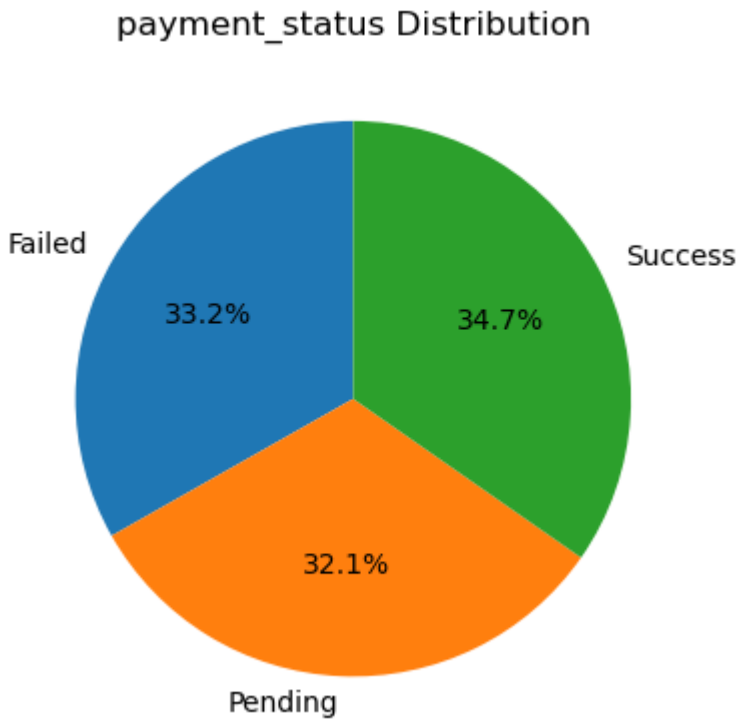
In [19]:
```
gf = df.groupby('payment_status' , as_index = False)['total_price'].sum().sort_v
gf
```

Out[19]:

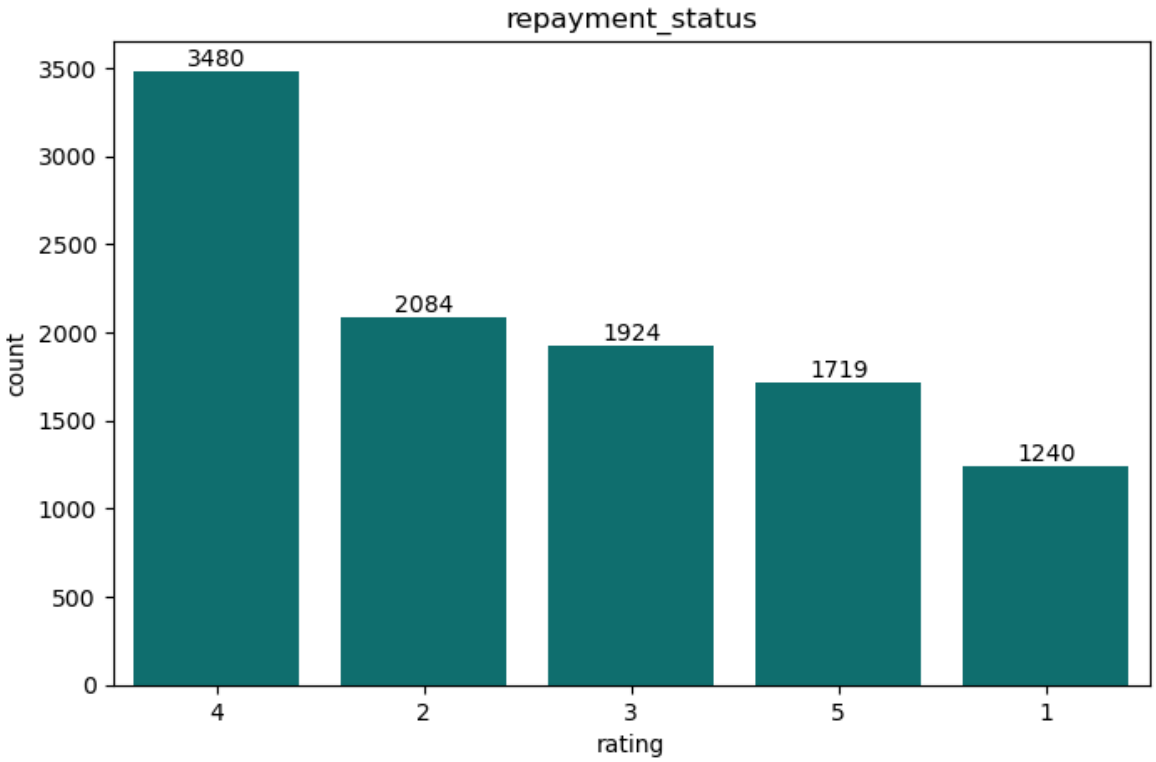
	payment_status	total_price
1	Pending	30636828.97
0	Failed	31699738.39
2	Success	33109892.06

In [20]:
```
gf = df.groupby('payment_status' , as_index = False)['total_price'].sum()

plt.figure(figsize=(4,4))
plt.pie(gf['total_price'], labels=gf['payment_status'], autopct='%1.1f%%' , star
plt.title('payment_status Distribution')
plt.tight_layout()
plt.show()
```



In [21]:
```
plt.figure(figsize=(8, 5))
ax = sns.countplot(data=df, x= 'rating' , order=df[ 'rating'].value_counts().ind
ax.bar_label(ax.containers[0])
plt.title('repayment_status')
plt.show()
```



In [23]:
```
df.to_excel("C:\\Users\\Hp\\OneDrive\\Desktop\\e commerce sales\\amite.xlsx")
```

In [ ]:

In [ ]:

In [ ]: