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

sales=pd.read_excel("C:\\Users\\Dell\\Documents\\Anaconda\\data\\
Coffee Shop Sales.xlsx")
print(sales.shape)
```

```
(149116, 11)
```

```
#For month full name
#sales.insert(2,"month",sales["transaction_date"].dt.strftime("%B"))
sales.head()
```

```
   transaction_id transaction_date transaction_time
transaction_qty  \
0               1       2023-01-01         07:06:11                2

1               2       2023-01-01         07:08:56                2

2               3       2023-01-01         07:14:04                2

3               4       2023-01-01         07:20:24                1

4               5       2023-01-01         07:22:41                2


   store_id   store_location  product_id  unit_price
product_category  \
0         5  Lower Manhattan          32         3.0
Coffee
1         5  Lower Manhattan          57         3.1
Tea
2         5  Lower Manhattan          59         4.5  Drinking
Chocolate
3         5  Lower Manhattan          22         2.0
Coffee
4         5  Lower Manhattan          57         3.1
Tea

            product_type               product_detail
0  Gourmet brewed coffee                  Ethiopia Rg
1        Brewed Chai tea     Spicy Eye Opener Chai Lg
2          Hot chocolate            Dark chocolate Lg
3            Drip coffee  Our Old Time Diner Blend Sm
4        Brewed Chai tea     Spicy Eye Opener Chai Lg
```

```
#for days
sales.insert(3,"Day",sales["transaction_date"].dt.strftime("%A"))
sales.head()
```

```
   transaction_id transaction_date transaction_time     Day
transaction_qty  \
0               1       2023-01-01         07:06:11  Sunday
2
1               2       2023-01-01         07:08:56  Sunday
2
2               3       2023-01-01         07:14:04  Sunday
2
3               4       2023-01-01         07:20:24  Sunday
1
4               5       2023-01-01         07:22:41  Sunday
2

   store_id   store_location  product_id  unit_price
product_category  \
0         5  Lower Manhattan          32         3.0
Coffee
1         5  Lower Manhattan          57         3.1
Tea
2         5  Lower Manhattan          59         4.5  Drinking
Chocolate
3         5  Lower Manhattan          22         2.0
Coffee
4         5  Lower Manhattan          57         3.1
Tea

            product_type               product_detail
0  Gourmet brewed coffee                  Ethiopia Rg
1        Brewed Chai tea     Spicy Eye Opener Chai Lg
2          Hot chocolate            Dark chocolate Lg
3            Drip coffee  Our Old Time Diner Blend Sm
4        Brewed Chai tea     Spicy Eye Opener Chai Lg

#for time
#sales.insert(5,"time1",sales["transaction_time"].dt.time)
sales.head()
#sales = sales.drop(columns=["time"])

   transaction_id transaction_date transaction_time     Day
transaction_qty  \
0               1       2023-01-01         07:06:11  Sunday
2
1               2       2023-01-01         07:08:56  Sunday
2
2               3       2023-01-01         07:14:04  Sunday
2
3               4       2023-01-01         07:20:24  Sunday
1
4               5       2023-01-01         07:22:41  Sunday
2
```

```
   store_id   store_location  product_id  unit_price
product_category  \
0         5  Lower Manhattan          32         3.0
Coffee
1         5  Lower Manhattan          57         3.1
Tea
2         5  Lower Manhattan          59         4.5  Drinking
Chocolate
3         5  Lower Manhattan          22         2.0
Coffee
4         5  Lower Manhattan          57         3.1
Tea

            product_type               product_detail
0  Gourmet brewed coffee                  Ethiopia Rg
1        Brewed Chai tea     Spicy Eye Opener Chai Lg
2         Hot chocolate            Dark chocolate Lg
3           Drip coffee  Our Old Time Diner Blend Sm
4        Brewed Chai tea     Spicy Eye Opener Chai Lg
```

```
print(f"number of rows:{sales.shape[0]} and number of columns:
{sales.shape[1]}")
print(f"number of duplicated rows:{sales.duplicated().sum()}")
```

```
number of rows:149116 and number of columns:12
number of duplicated rows:0
```

```
print(f"Names of columns:{sales.columns}")
```

```
Names of columns:Index(['transaction_id', 'transaction_date',
'transaction_time', 'Day',
       'transaction_qty', 'store_id', 'store_location', 'product_id',
       'unit_price', 'product_category', 'product_type',
'product_detail'],
      dtype='object')
```

```
sales.head(10)
```

```
   transaction_id transaction_date transaction_time     Day
transaction_qty  \
0               1       2023-01-01         07:06:11  Sunday
2
1               2       2023-01-01         07:08:56  Sunday
2
2               3       2023-01-01         07:14:04  Sunday
2
3               4       2023-01-01         07:20:24  Sunday
1
4               5       2023-01-01         07:22:41  Sunday
2
```

```
5               6      2023-01-01        07:22:41  Sunday
1
6               7      2023-01-01        07:25:49  Sunday
1
7               8      2023-01-01        07:33:34  Sunday
2
8               9      2023-01-01        07:39:13  Sunday
1
9              10      2023-01-01        07:39:34  Sunday
2

   store_id   store_location  product_id  unit_price
product_category  \
0         5  Lower Manhattan          32        3.00
Coffee
1         5  Lower Manhattan          57        3.10
Tea
2         5  Lower Manhattan          59        4.50  Drinking
Chocolate
3         5  Lower Manhattan          22        2.00
Coffee
4         5  Lower Manhattan          57        3.10
Tea
5         5  Lower Manhattan          77        3.00
Bakery
6         5  Lower Manhattan          22        2.00
Coffee
7         5  Lower Manhattan          28        2.00
Coffee
8         5  Lower Manhattan          39        4.25
Coffee
9         5  Lower Manhattan          58        3.50  Drinking
Chocolate

            product_type               product_detail
0  Gourmet brewed coffee                  Ethiopia Rg
1        Brewed Chai tea     Spicy Eye Opener Chai Lg
2          Hot chocolate           Dark chocolate Lg
3            Drip coffee  Our Old Time Diner Blend Sm
4        Brewed Chai tea     Spicy Eye Opener Chai Lg
5                  Scone                Oatmeal Scone
6            Drip coffee  Our Old Time Diner Blend Sm
7  Gourmet brewed coffee    Columbian Medium Roast Sm
8       Barista Espresso                     Latte Rg
9          Hot chocolate           Dark chocolate Rg

sales.dtypes

transaction_id               int64
transaction_date    datetime64[ns]
```

```
transaction_time            object
Day                         object
transaction_qty              int64
store_id                     int64
store_location              object
product_id                   int64
unit_price                 float64
product_category            object
product_type                object
product_detail              object
dtype: object

sales["transaction_time"]=pd.to_datetime(sales["transaction_time"],for
mat="%H:%M:%S")
sales

        transaction_id transaction_date    transaction_time     Day  \
0                    1       2023-01-01 1900-01-01 07:06:11  Sunday
1                    2       2023-01-01 1900-01-01 07:08:56  Sunday
2                    3       2023-01-01 1900-01-01 07:14:04  Sunday
3                    4       2023-01-01 1900-01-01 07:20:24  Sunday
4                    5       2023-01-01 1900-01-01 07:22:41  Sunday
...                ...              ...                 ...     ...
149111          149452       2023-06-30 1900-01-01 20:18:41  Friday
149112          149453       2023-06-30 1900-01-01 20:25:10  Friday
149113          149454       2023-06-30 1900-01-01 20:31:34  Friday
149114          149455       2023-06-30 1900-01-01 20:57:19  Friday
149115          149456       2023-06-30 1900-01-01 20:57:19  Friday

        transaction_qty  store_id   store_location  product_id
unit_price  \
0                     2         5  Lower Manhattan          32
3.00
1                     2         5  Lower Manhattan          57
3.10
2                     2         5  Lower Manhattan          59
4.50
3                     1         5  Lower Manhattan          22
2.00
4                     2         5  Lower Manhattan          57
3.10
...                 ...       ...              ...         ...
...
149111                2         8   Hell's Kitchen          44
2.50
149112                2         8   Hell's Kitchen          49
3.00
149113                1         8   Hell's Kitchen          45
3.00
149114                1         8   Hell's Kitchen          40
```

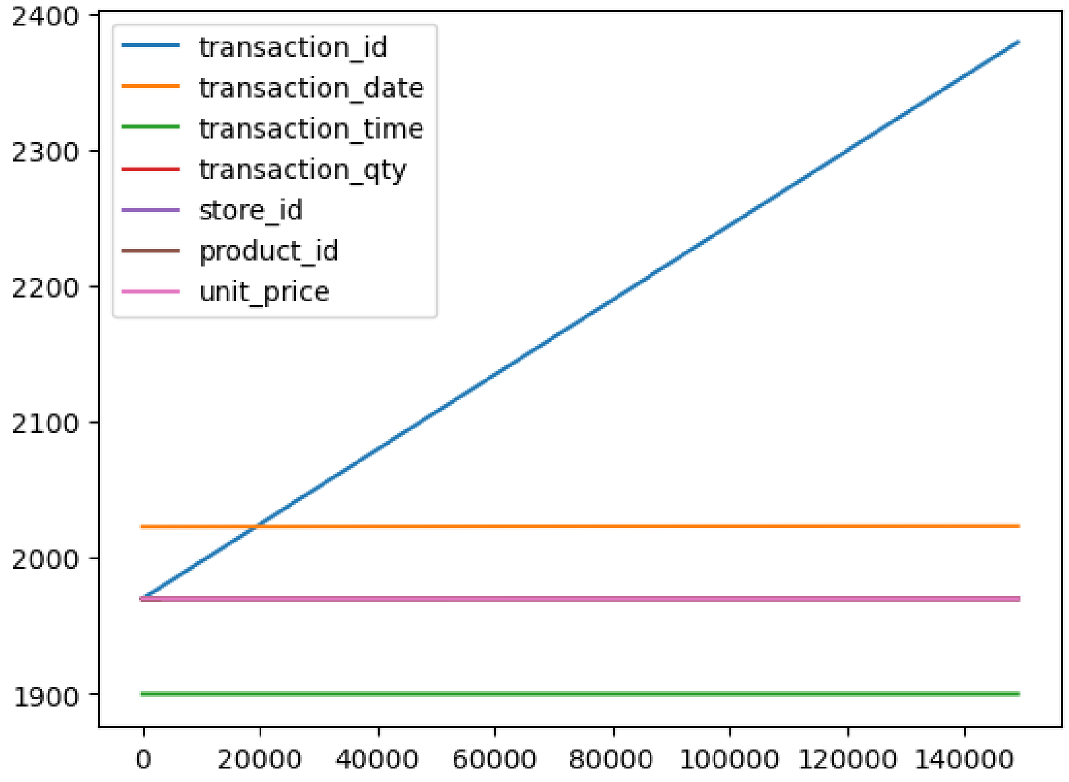
```
3.75
149115                  2         8   Hell's Kitchen          64
0.80

          product_category           product_type
product_detail
0                   Coffee  Gourmet brewed coffee
Ethiopia Rg
1                      Tea        Brewed Chai tea     Spicy Eye Opener
Chai Lg
2       Drinking Chocolate          Hot chocolate            Dark
chocolate Lg
3                   Coffee            Drip coffee  Our Old Time Diner
Blend Sm
4                      Tea        Brewed Chai tea     Spicy Eye Opener
Chai Lg
...                    ...                    ...
...
149111                 Tea      Brewed herbal tea
Peppermint Rg
149112                 Tea       Brewed Black tea         English
Breakfast Lg
149113                 Tea      Brewed herbal tea
Peppermint Lg
149114              Coffee       Barista Espresso
Cappuccino
149115            Flavours          Regular syrup
Hazelnut syrup

[149116 rows x 12 columns]
```

```
sales.dtypes
```

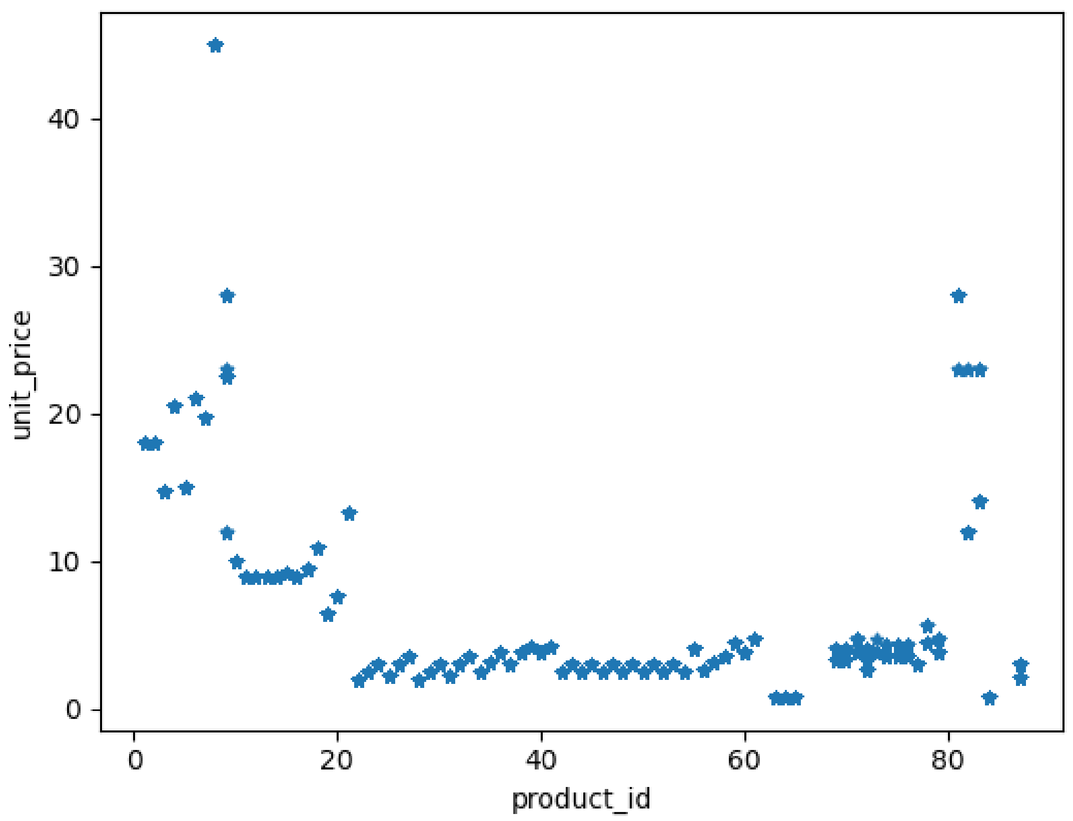
```
transaction_id               int64
transaction_date    datetime64[ns]
transaction_time    datetime64[ns]
Day                         object
transaction_qty              int64
store_id                     int64
store_location              object
product_id                   int64
unit_price                 float64
product_category            object
product_type                object
product_detail              object
dtype: object
```

Practice

```
sales.plot()
plt.show()
```

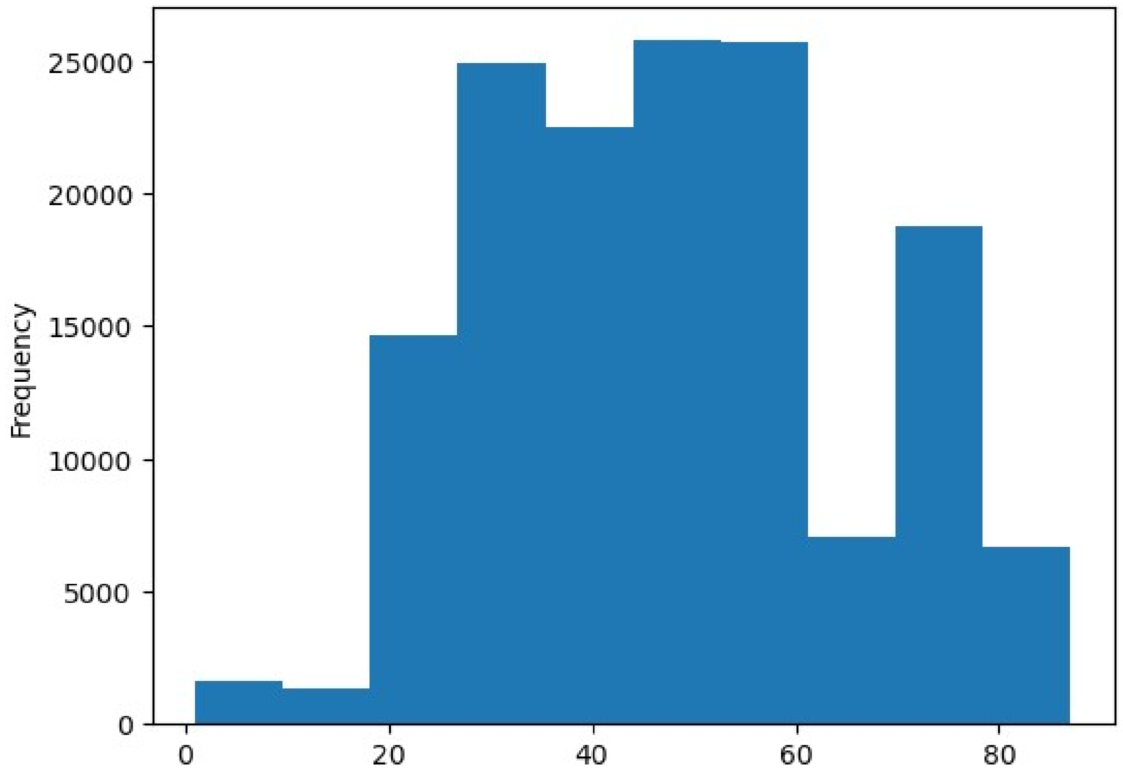


```
sales.plot(kind="scatter",x="product_id",y="unit_price",marker="*")
plt.show()
```



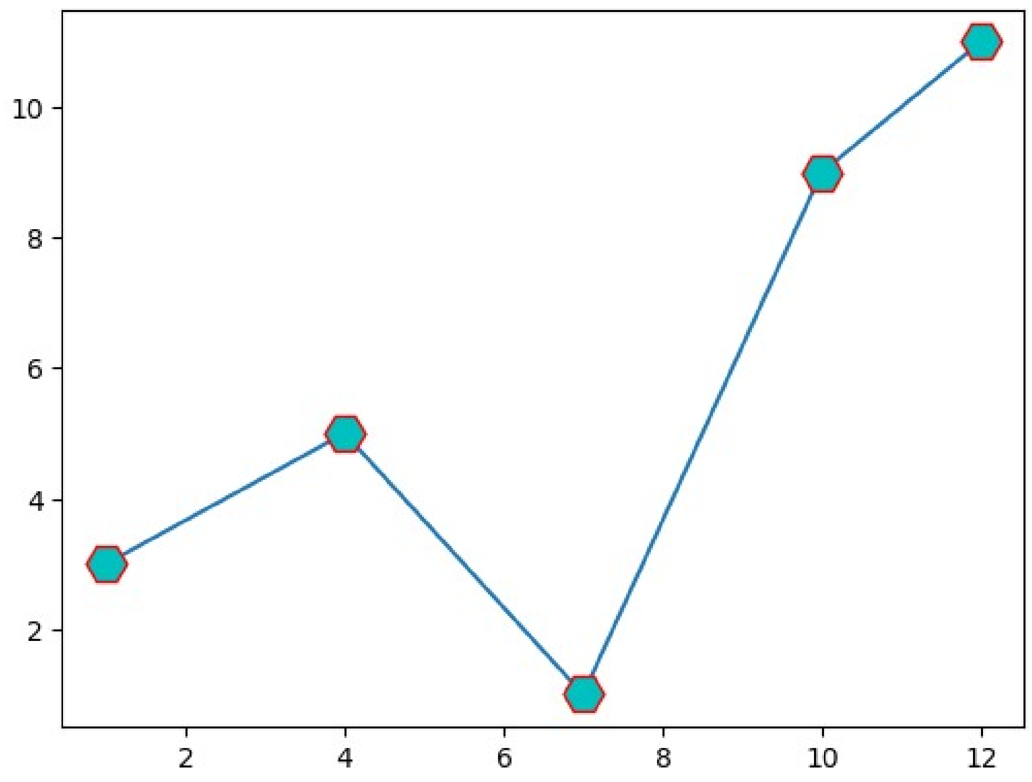
```
sales["product_id"].plot(kind="hist")
```

```
<Axes: ylabel='Frequency'>
```



```
a=np.array([1,4,7,10,12])
b=np.array([3,5,1,9,11])
#plt.plot(a,b,'o:c')
#plt.plot(a,b,marker="o",ms=15,mec='r')
plt.plot(a,b,marker="H",ms=15,mfc='c',mec='r')
```

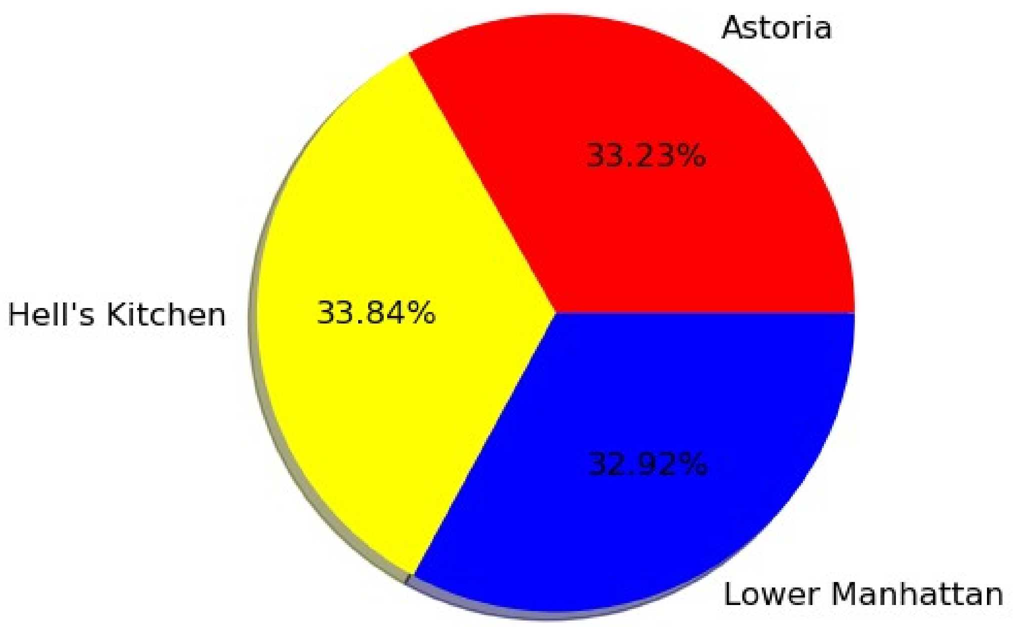
```
[<matplotlib.lines.Line2D at 0x21c97ec5eb0>]
```



Classwork Resume

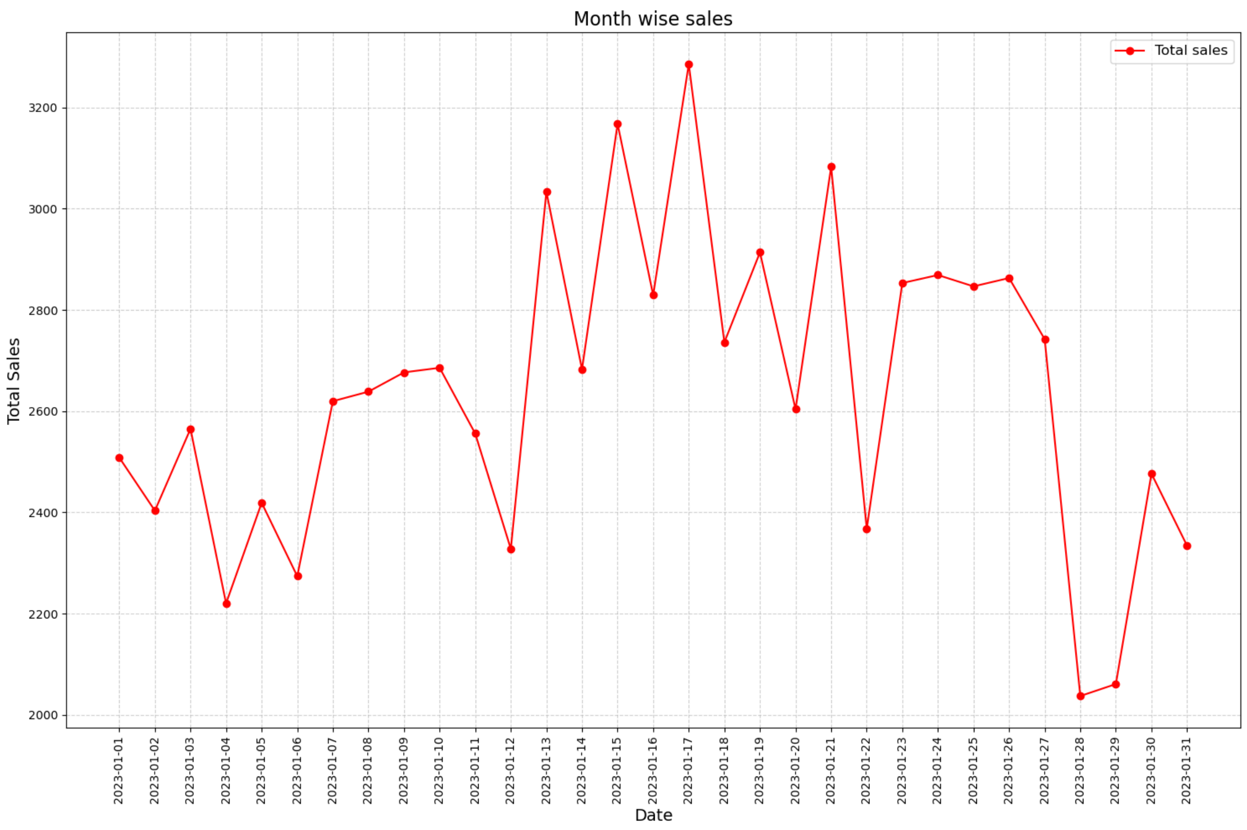
```
sales["total_sales"]=sales["unit_price"]*sales["transaction_qty"]
sales.head(5)

   transaction_id transaction_date    transaction_time     Day  \
0               1       2023-01-01 1900-01-01 07:06:11  Sunday
1               2       2023-01-01 1900-01-01 07:08:56  Sunday
2               3       2023-01-01 1900-01-01 07:14:04  Sunday
3               4       2023-01-01 1900-01-01 07:20:24  Sunday
4               5       2023-01-01 1900-01-01 07:22:41  Sunday

   transaction_qty  store_id   store_location  product_id  unit_price
\
0                2         5  Lower Manhattan          32         3.0

1                2         5  Lower Manhattan          57         3.1

2                2         5  Lower Manhattan          59         4.5

3                1         5  Lower Manhattan          22         2.0

4                2         5  Lower Manhattan          57         3.1
```

```
     product_category           product_type 
product_detail  \
0              Coffee  Gourmet brewed coffee                        Ethiopia
Rg
1                 Tea        Brewed Chai tea      Spicy Eye Opener Chai
Lg
2  Drinking Chocolate          Hot chocolate            Dark chocolate
Lg
3              Coffee            Drip coffee  Our Old Time Diner Blend
Sm
4                 Tea        Brewed Chai tea      Spicy Eye Opener Chai
Lg

   total_sales
0          6.0
1          6.2
2          9.0
3          2.0
4          6.2

#finding the total sales as per store location
store_sales=pd.DataFrame(sales.groupby("store_location")
["total_sales"].sum()).reset_index()
#plotting the pie chart with the help of matplotlib
plt.figure(figsize=(10,5))
plt.pie(store_sales["total_sales"],
        labels=store_sales["store_location"],
        autopct='%1.2f%%',
        colors=['red','yellow','blue'],
        textprops={'fontsize':12},shadow=True)
plt.show()
```



```
#print(sales["product_category"[.unique())
#date wise sales using input user for month year
month=input("Enter the month:")
#trend_sales=sales[sales["transaction_date"].dt.month==int(month)]
date_trend_sale=pd.DataFrame(sales[sales["transaction_date"].dt.month=
=int(month)].groupby(sales["transaction_date"].dt.date)
["total_sales"].sum()).reset_index()
date_trend_sale["transaction_date"]=pd.to_datetime(date_trend_sale["tr
ansaction_date"])
print(date_trend_sale)
#plotting line chart
plt.figure(figsize=(15,10))
plt.plot(date_trend_sale['transaction_date'],date_trend_sale['total_sa
les'],
         marker='o',linestyle='-',color='red',label='Total sales')
plt.title('Month wise sales',fontsize=16)
plt.xlabel('Date',fontsize=14)
plt.ylabel('Total Sales',fontsize=14)
plt.grid(True, linestyle='--',alpha=0.6)
plt.xticks(date_trend_sale['transaction_date'],labels=date_trend_sale[
'transaction_date'].dt.strftime('%Y-%m-%d'),rotation=90)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()

Enter the month: 1
```

```
   transaction_date  total_sales
0        2023-01-01      2508.20
1        2023-01-02      2403.35
2        2023-01-03      2565.00
3        2023-01-04      2220.10
4        2023-01-05      2418.85
5        2023-01-06      2273.85
6        2023-01-07      2619.65
7        2023-01-08      2638.53
8        2023-01-09      2676.61
9        2023-01-10      2685.65
10       2023-01-11      2555.75
11       2023-01-12      2327.70
12       2023-01-13      3033.60
13       2023-01-14      2682.51
14       2023-01-15      3167.71
15       2023-01-16      2829.16
16       2023-01-17      3285.80
17       2023-01-18      2735.96
18       2023-01-19      2913.68
19       2023-01-20      2603.73
20       2023-01-21      3082.85
21       2023-01-22      2367.33
22       2023-01-23      2853.15
23       2023-01-24      2868.95
24       2023-01-25      2846.55
25       2023-01-26      2863.03
26       2023-01-27      2742.10
27       2023-01-28      2037.10
28       2023-01-29      2060.75
29       2023-01-30      2476.41
30       2023-01-31      2334.13
```



```
sales["store_location"].unique()
```

```
array(['Lower Manhattan', "Hell's Kitchen", 'Astoria'], dtype=object)
```

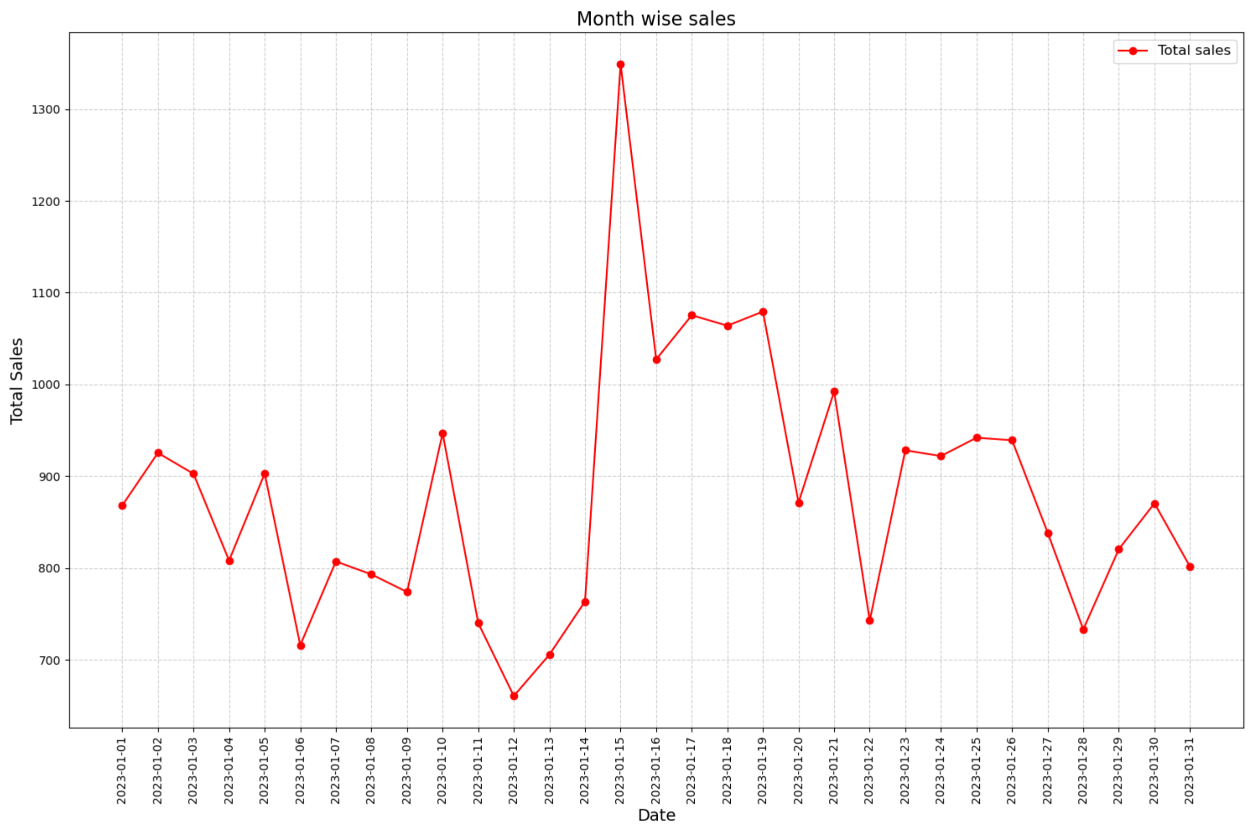
```
sales["product_category"].unique()
```

```
array(['Coffee', 'Tea', 'Drinking Chocolate', 'Bakery', 'Flavours',
       'Loose Tea', 'Coffee beans', 'Packaged Chocolate', 'Branded'],
      dtype=object)
```

```
#print(sales["product_category"[.unique())
#date wise sales using input user for month year
month=input("Enter the month:")
store=input("Enter the store name=")
#trend_sales=sales[sales["transaction_date"].dt.month==int(month)]
date_trend_sale=pd.DataFrame(sales[(sales["transaction_date"].dt.month
==int(month))&(sales["store_location"]==store)].groupby(sales["transac
tion_date"].dt.date)["total_sales"].sum()).reset_index()
date_trend_sale["transaction_date"]=pd.to_datetime(date_trend_sale["tr
ansaction_date"])
print(date_trend_sale)
#plotting line chart
plt.figure(figsize=(15,10))
plt.plot(date_trend_sale['transaction_date'],date_trend_sale['total_sa
les'],
         marker='o',linestyle='-',color='red',label='Total sales')
plt.title('Month wise sales',fontsize=16)
plt.xlabel('Date',fontsize=14)
plt.ylabel('Total Sales',fontsize=14)
plt.grid(True, linestyle='--',alpha=0.6)
plt.xticks(date_trend_sale['transaction_date'],labels=date_trend_sale[
'transaction_date'].dt.strftime('%Y-%m-%d'),rotation=90)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

```
Enter the month: 1
Enter the store name= Astoria
```

```
   transaction_date  total_sales
0        2023-01-01       868.40
1        2023-01-02       925.50
2        2023-01-03       902.75
3        2023-01-04       808.25
4        2023-01-05       903.05
5        2023-01-06       716.05
6        2023-01-07       807.30
7        2023-01-08       793.15
8        2023-01-09       774.01
9        2023-01-10       947.00
10       2023-01-11       740.15
11       2023-01-12       660.70
12       2023-01-13       705.80
13       2023-01-14       763.70
14       2023-01-15      1348.93
15       2023-01-16      1027.33
16       2023-01-17      1075.40
17       2023-01-18      1064.03
18       2023-01-19      1079.38
19       2023-01-20       871.43
20       2023-01-21       992.35
21       2023-01-22       742.80
22       2023-01-23       928.30
23       2023-01-24       922.05
24       2023-01-25       942.00
25       2023-01-26       939.25
26       2023-01-27       838.55
27       2023-01-28       733.25
28       2023-01-29       820.70
29       2023-01-30       870.60
30       2023-01-31       801.50
```



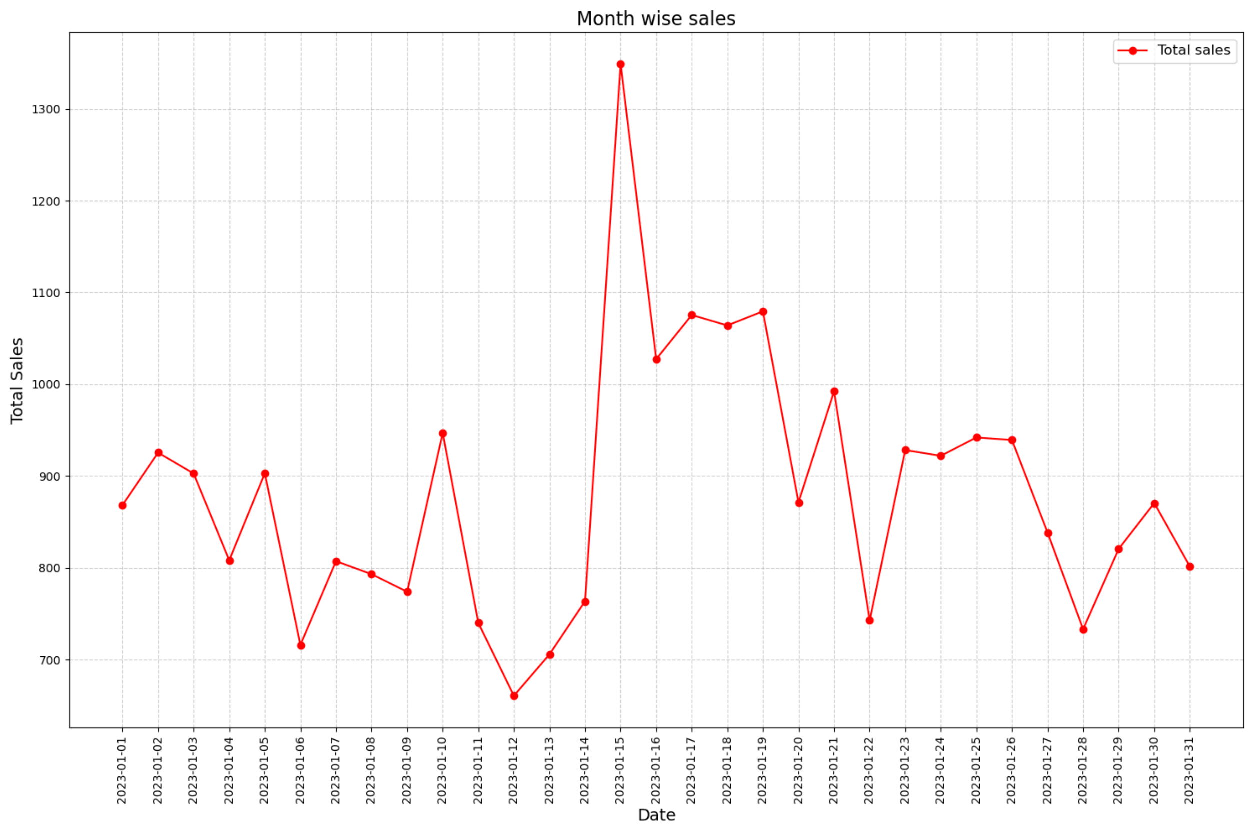
```
sales["product_category"].unique()
```

```
array(['Coffee', 'Tea', 'Drinking Chocolate', 'Bakery', 'Flavours',
       'Loose Tea', 'Coffee beans', 'Packaged Chocolate', 'Branded'],
      dtype=object)
```

```
#One more way usin
#print(sales["product_category"[.unique())
#date wise sales using input user for month year
month=input("Enter the month:")
store=input("Enter the store name=")
#trend_sales=sales[sales["transaction_date"].dt.month==int(month)]
date_trend_sale=pd.DataFrame(sales[(sales["transaction_date"].dt.strft
ime("%B")==month)&(sales["store_location"]==store)].groupby(sales["tra
nsaction_date"].dt.date)["total_sales"].sum()).reset_index()
date_trend_sale["transaction_date"]=pd.to_datetime(date_trend_sale["tr
ansaction_date"])
print(date_trend_sale)
#plotting line chart
plt.figure(figsize=(15,10))
plt.plot(date_trend_sale['transaction_date'],date_trend_sale['total_sa
les'],
         marker='o',linestyle='-',color='red',label='Total sales')
plt.title('Month wise sales',fontsize=16)
plt.xlabel('Date',fontsize=14)
plt.ylabel('Total Sales',fontsize=14)
plt.grid(True, linestyle='--',alpha=0.6)
plt.xticks(date_trend_sale['transaction_date'],labels=date_trend_sale[
'transaction_date'].dt.strftime('%Y-%m-%d'),rotation=90)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

```
Enter the month: January
Enter the store name= Astoria
```

```
   transaction_date  total_sales
0        2023-01-01       868.40
1        2023-01-02       925.50
2        2023-01-03       902.75
3        2023-01-04       808.25
4        2023-01-05       903.05
5        2023-01-06       716.05
6        2023-01-07       807.30
7        2023-01-08       793.15
8        2023-01-09       774.01
9        2023-01-10       947.00
10       2023-01-11       740.15
11       2023-01-12       660.70
12       2023-01-13       705.80
13       2023-01-14       763.70
14       2023-01-15      1348.93
15       2023-01-16      1027.33
16       2023-01-17      1075.40
17       2023-01-18      1064.03
18       2023-01-19      1079.38
19       2023-01-20       871.43
20       2023-01-21       992.35
21       2023-01-22       742.80
22       2023-01-23       928.30
23       2023-01-24       922.05
24       2023-01-25       942.00
25       2023-01-26       939.25
26       2023-01-27       838.55
27       2023-01-28       733.25
28       2023-01-29       820.70
29       2023-01-30       870.60
30       2023-01-31       801.50
```



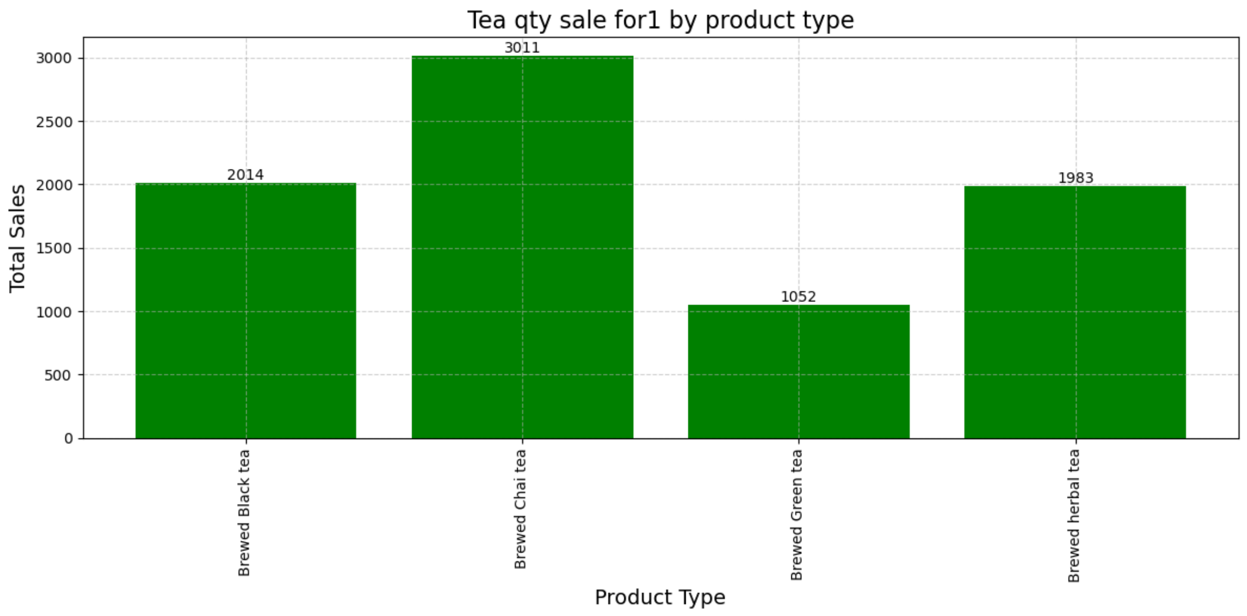
```
sales.head()

   transaction_id transaction_date    month      Day     time1  \
0               1       2023-01-01  January   Sunday  07:06:11
1               2       2023-01-01  January   Sunday  07:08:56
2               3       2023-01-01  January   Sunday  07:14:04
3               4       2023-01-01  January   Sunday  07:20:24
4               5       2023-01-01  January   Sunday  07:22:41

     transaction_time  transaction_qty  store_id   store_location
product_id  \
0 1900-01-01 07:06:11                2         5  Lower Manhattan
32
1 1900-01-01 07:08:56                2         5  Lower Manhattan
57
2 1900-01-01 07:14:04                2         5  Lower Manhattan
59
3 1900-01-01 07:20:24                1         5  Lower Manhattan
22
4 1900-01-01 07:22:41                2         5  Lower Manhattan
57

   unit_price    product_category           product_type  \
0         3.0              Coffee  Gourmet brewed coffee
1         3.1                 Tea        Brewed Chai tea
```

```
2         4.5  Drinking Chocolate         Hot chocolate
3         2.0              Coffee          Drip coffee
4         3.1                 Tea       Brewed Chai tea
```

```
                product_detail  total_sales
0                 Ethiopia Rg          6.0
1     Spicy Eye Opener Chai Lg          6.2
2            Dark chocolate Lg          9.0
3  Our Old Time Diner Blend Sm          2.0
4     Spicy Eye Opener Chai Lg          6.2
```

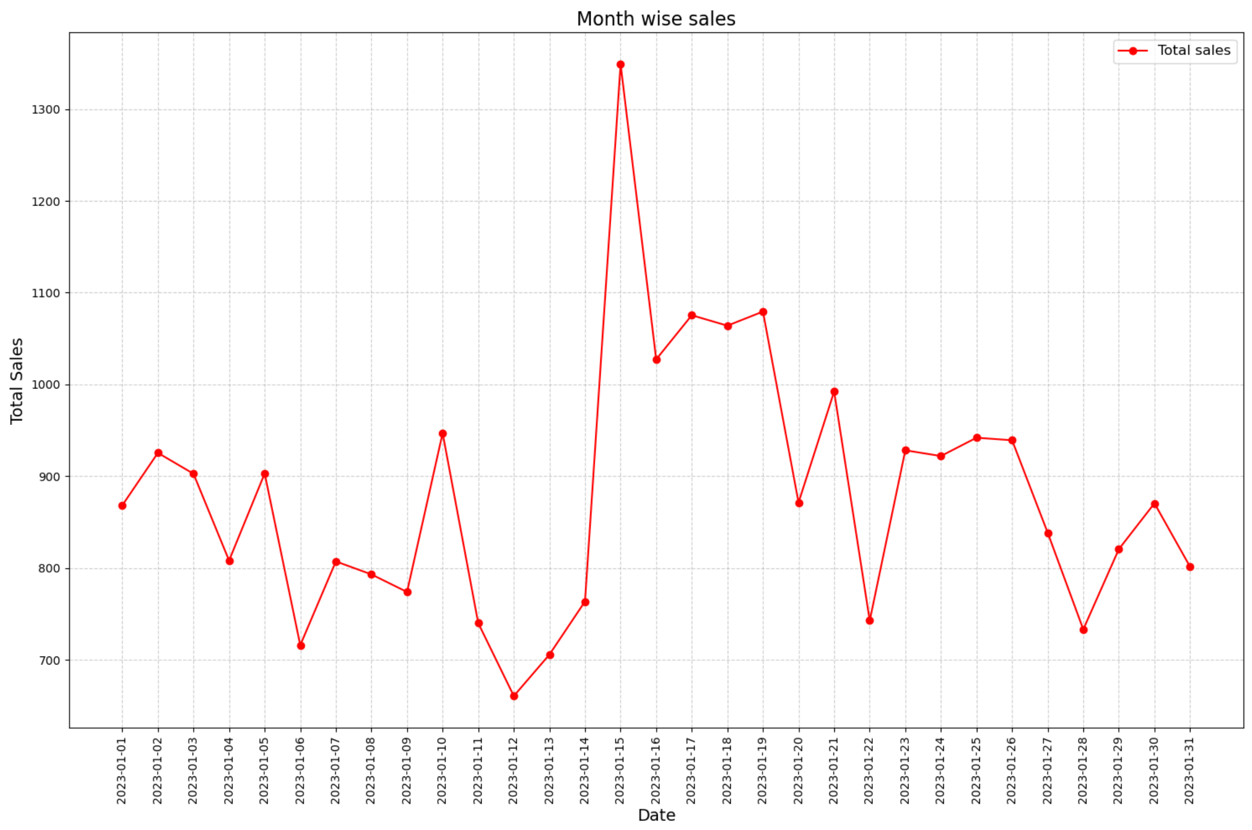
```python
# One more way by using month column
#date_trend_sale=pd.DataFrame(sales[(sales["month"]==month)&(sales["st
ore_location"]==store)].groupby(sales["transaction_date"].dt.date)
["total_sales"].sum()).reset_index()

m1=input("Enter the month=")
p1=input("Enter the product category=")
d1=pd.DataFrame(sales[(sales["transaction_date"].dt.month==int(m1))&(s
ales["product_category"]==p1)].groupby("product_type")
["transaction_qty"].sum()).reset_index()
d1
#plotting the bar chart
plt.figure(figsize=(12,6))
plt.bar_label(plt.bar(d1["product_type"],d1["transaction_qty"],color='
Green'))
plt.title(f'{p1} qty sale for{m1} by product type',fontsize=16)
plt.xlabel('Product Type',fontsize=14)
plt.ylabel('Total Sales',fontsize=14)
plt.xticks(rotation=90)
plt.grid(True,linestyle='--',alpha=0.6)
plt.tight_layout()
plt.show()
```

```
Enter the month= 1
Enter the product category= Tea
```



```
#One more way usin
#print(sales["product_category"[.unique())
#date wise sales using input user for month year
month=input("Enter the :")
store=input("Enter the store name=")
#trend_sales=sales[sales["transaction_date"].dt.month==int(month)]
date_trend_sale=pd.DataFrame(sales[(sales["month"]==month)&(sales["sto
re_location"]==store)].groupby(sales["transaction_date"].dt.date)
["total_sales"].sum()).reset_index()
date_trend_sale["transaction_date"]=pd.to_datetime(date_trend_sale["tr
ansaction_date"])
print(date_trend_sale)
#plotting line chart
plt.figure(figsize=(15,10))
plt.plot(date_trend_sale['transaction_date'],date_trend_sale['total_sa
les'],
         marker='o',linestyle='-',color='red',label='Total sales')
plt.title('Month wise sales',fontsize=16)
plt.xlabel('Date',fontsize=14)
plt.ylabel('Total Sales',fontsize=14)
plt.grid(True, linestyle='--',alpha=0.6)
plt.xticks(date_trend_sale['transaction_date'],labels=date_trend_sale[
'transaction_date'].dt.strftime('%Y-%m-%d'),rotation=90)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()

Enter the : January
Enter the store name= Astoria
```

```
   transaction_date  total_sales
0        2023-01-01       868.40
1        2023-01-02       925.50
2        2023-01-03       902.75
3        2023-01-04       808.25
4        2023-01-05       903.05
5        2023-01-06       716.05
6        2023-01-07       807.30
7        2023-01-08       793.15
8        2023-01-09       774.01
9        2023-01-10       947.00
10       2023-01-11       740.15
11       2023-01-12       660.70
12       2023-01-13       705.80
13       2023-01-14       763.70
14       2023-01-15      1348.93
15       2023-01-16      1027.33
16       2023-01-17      1075.40
17       2023-01-18      1064.03
18       2023-01-19      1079.38
19       2023-01-20       871.43
20       2023-01-21       992.35
21       2023-01-22       742.80
22       2023-01-23       928.30
23       2023-01-24       922.05
24       2023-01-25       942.00
25       2023-01-26       939.25
26       2023-01-27       838.55
27       2023-01-28       733.25
28       2023-01-29       820.70
29       2023-01-30       870.60
30       2023-01-31       801.50
```

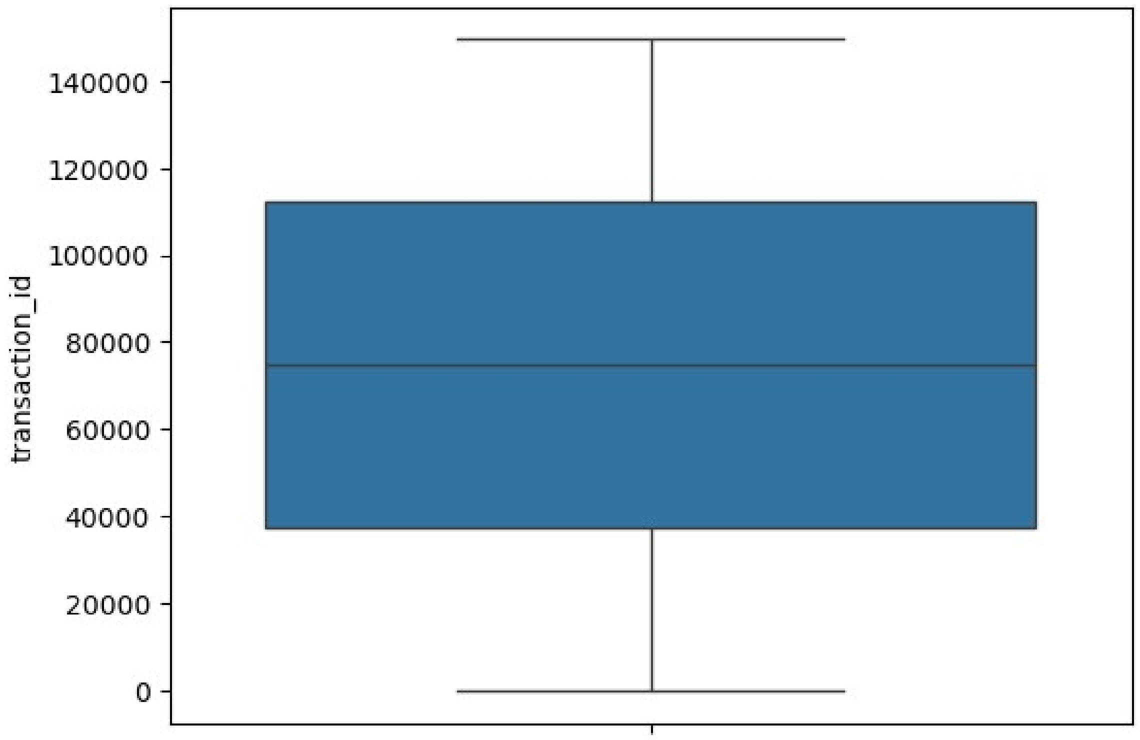


```
sales["product_type"].unique()

array(['Gourmet brewed coffee', 'Brewed Chai tea', 'Hot chocolate',
       'Drip coffee', 'Scone', 'Barista Espresso', 'Brewed Black tea',
       'Brewed Green tea', 'Brewed herbal tea', 'Biscotti', 'Pastry',
       'Organic brewed coffee', 'Premium brewed coffee', 'Regular
syrup',
       'Herbal tea', 'Gourmet Beans', 'Organic Beans', 'Sugar free
syrup',
       'Drinking Chocolate', 'Premium Beans', 'Chai tea', 'Green
beans',
       'Espresso Beans', 'Green tea', 'Organic Chocolate',
'Housewares',
       'Black tea', 'House blend Beans', 'Clothing'], dtype=object)

sales["product_category"].unique()

array(['Coffee', 'Tea', 'Drinking Chocolate', 'Bakery', 'Flavours',
       'Loose Tea', 'Coffee beans', 'Packaged Chocolate', 'Branded'],
      dtype=object)
```

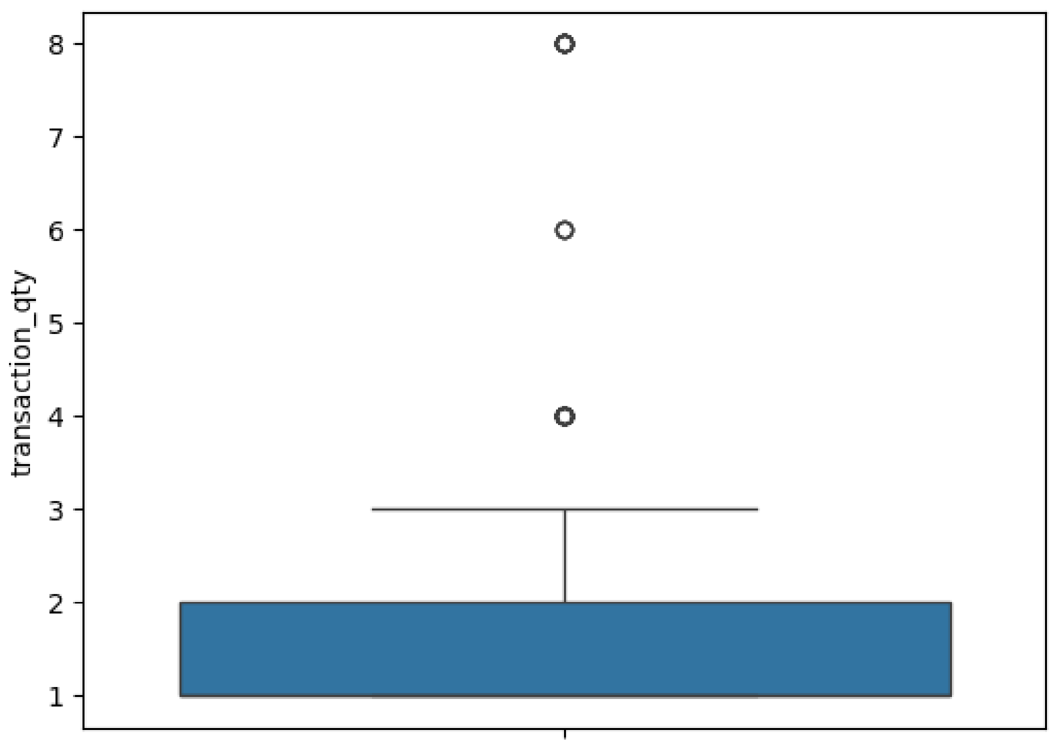
Practice

```
for i in sales.select_dtypes(include=np.number):
    print(i)
    sns.boxplot([i])
    plt.show()
```

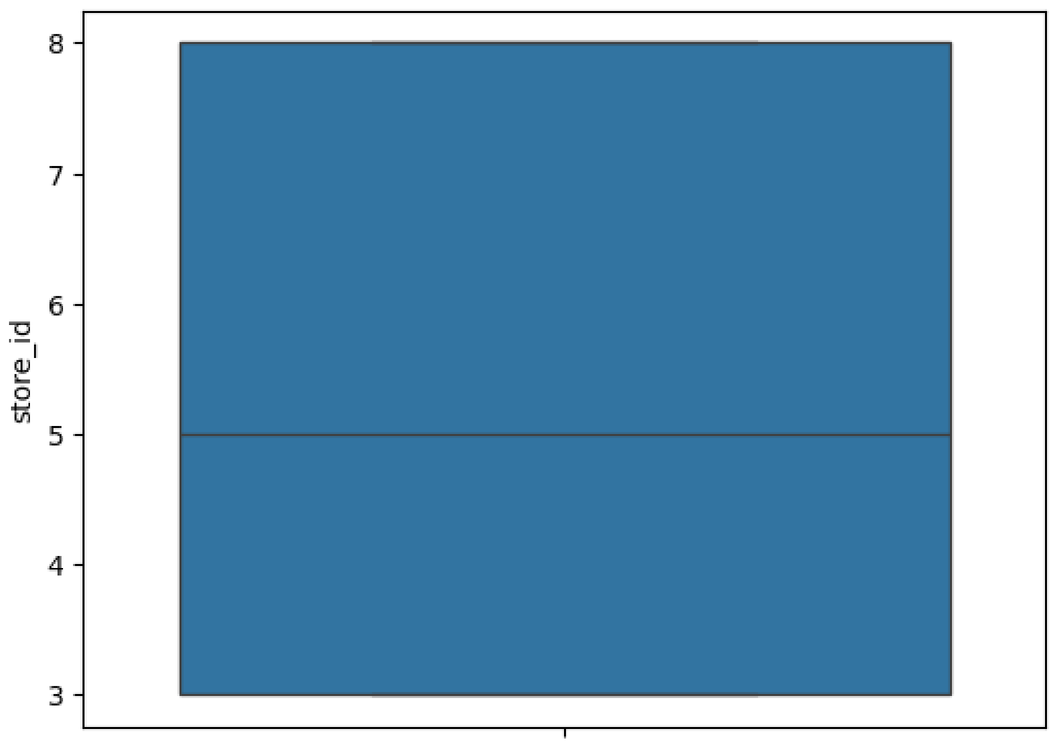
```
transaction_id
```



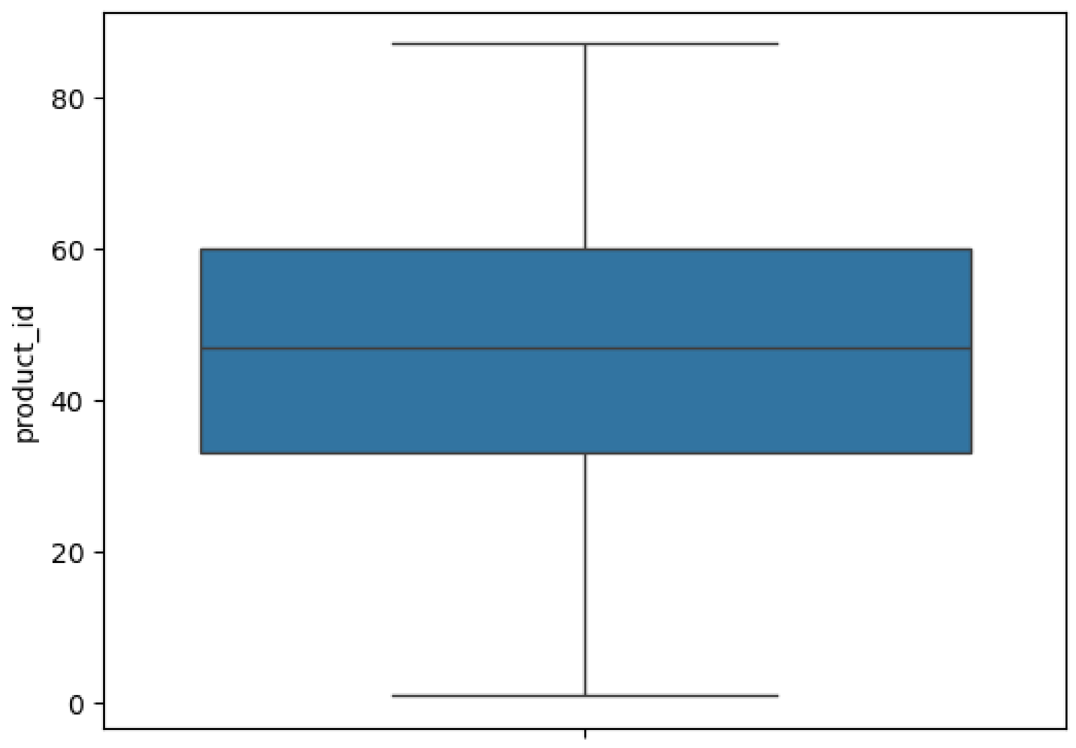
```
transaction_qty
```



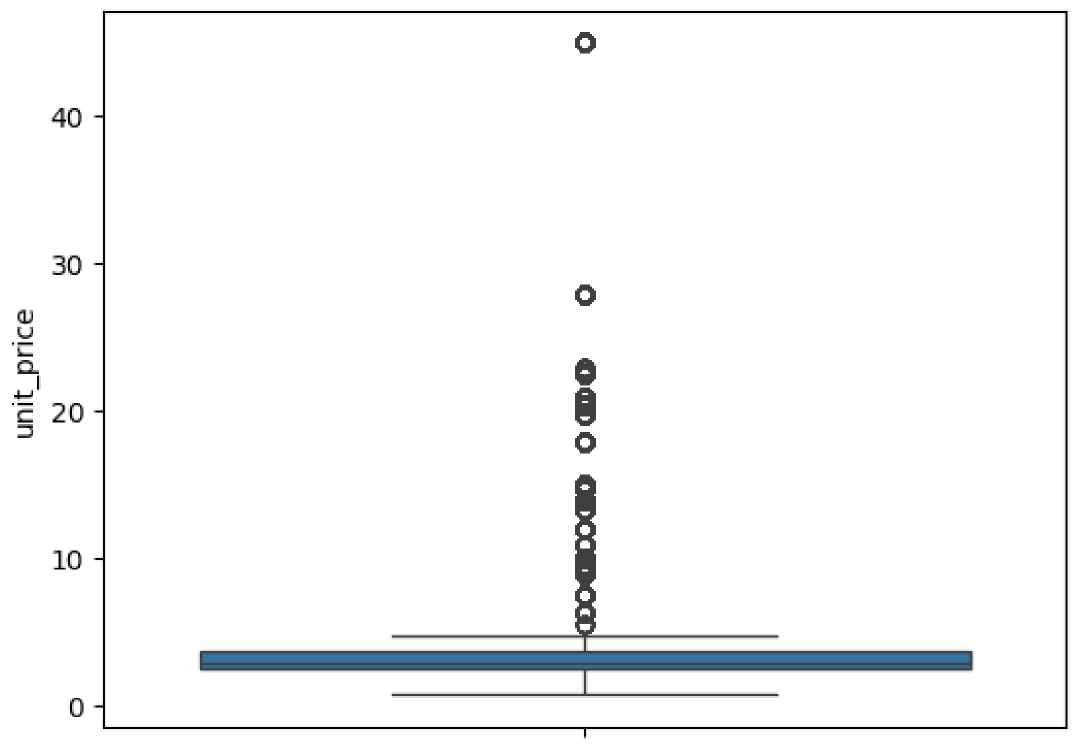
```
store_id
```



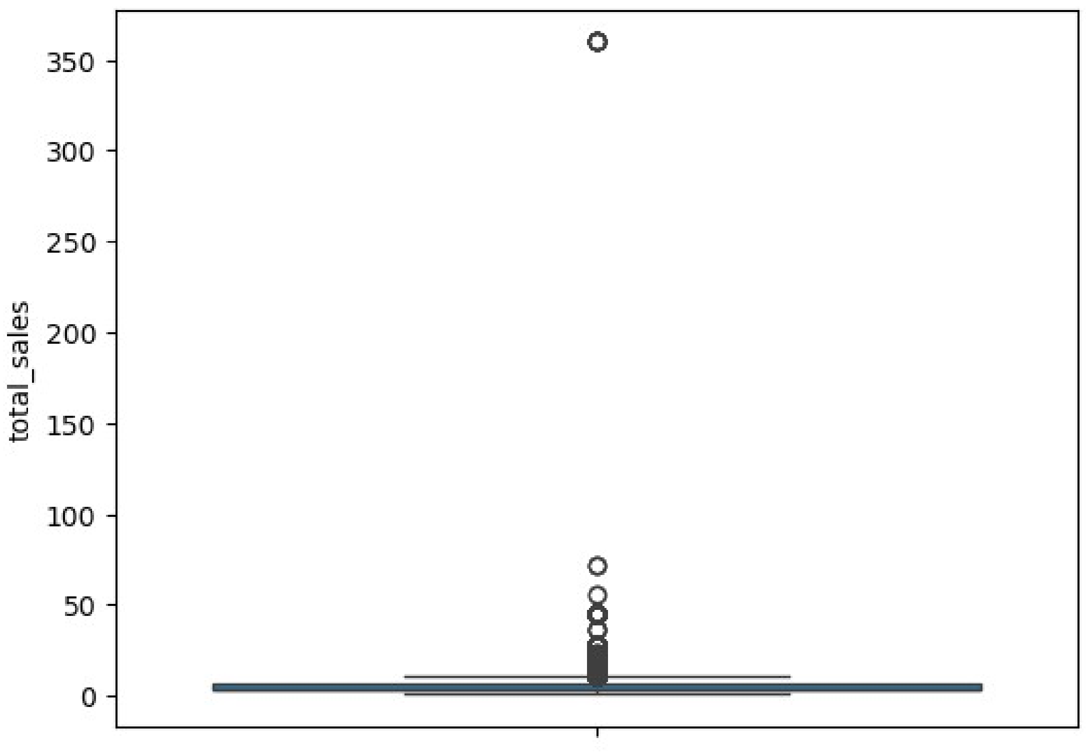
```
product_id
```



```
unit_price
```



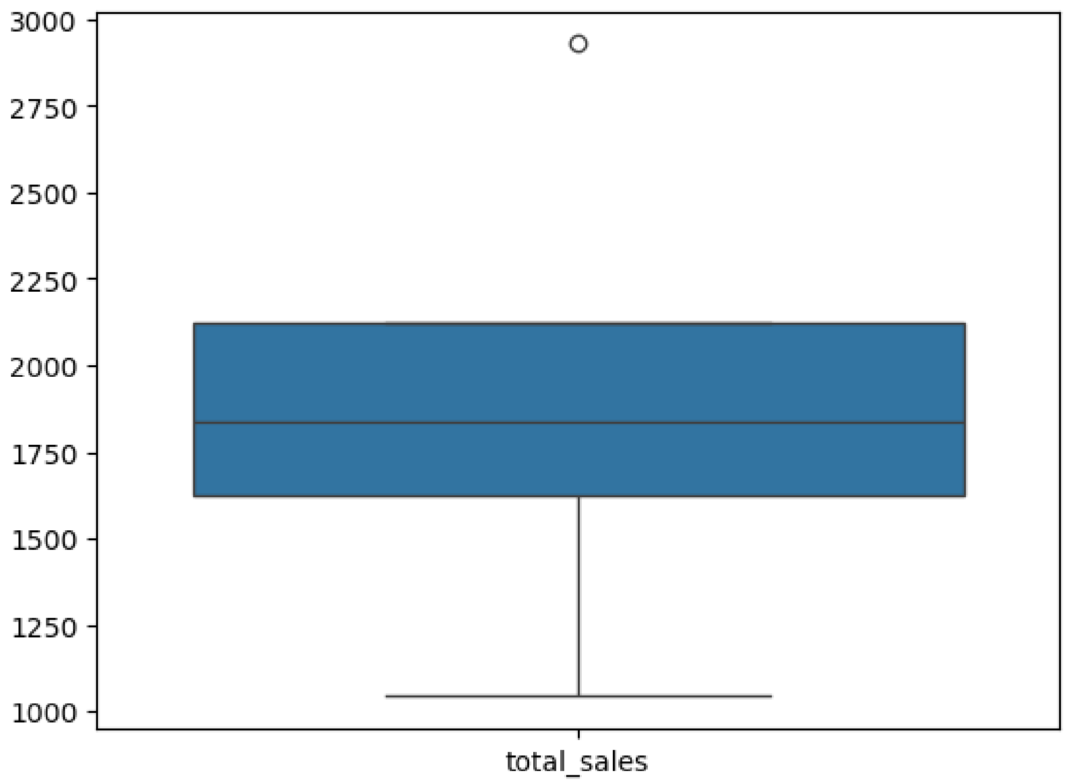
```
total_sales
```



```
m1=input("Enter the month=")
p1=input("Enter the product category=")
s1=input("Enter the store name=")
d1=pd.DataFrame(sales[(sales["transaction_date"].dt.month==int(m1))&(s
ales["product_category"]==p1)&(sales["store_location"]==s1)].groupby("
product_type")["total_sales"].sum()).reset_index()
d1
sns.boxplot(d1)
```

```
Enter the month= 1
Enter the product category= Tea
Enter the store name= Hell's Kitchen
```

```
<Axes: >
```



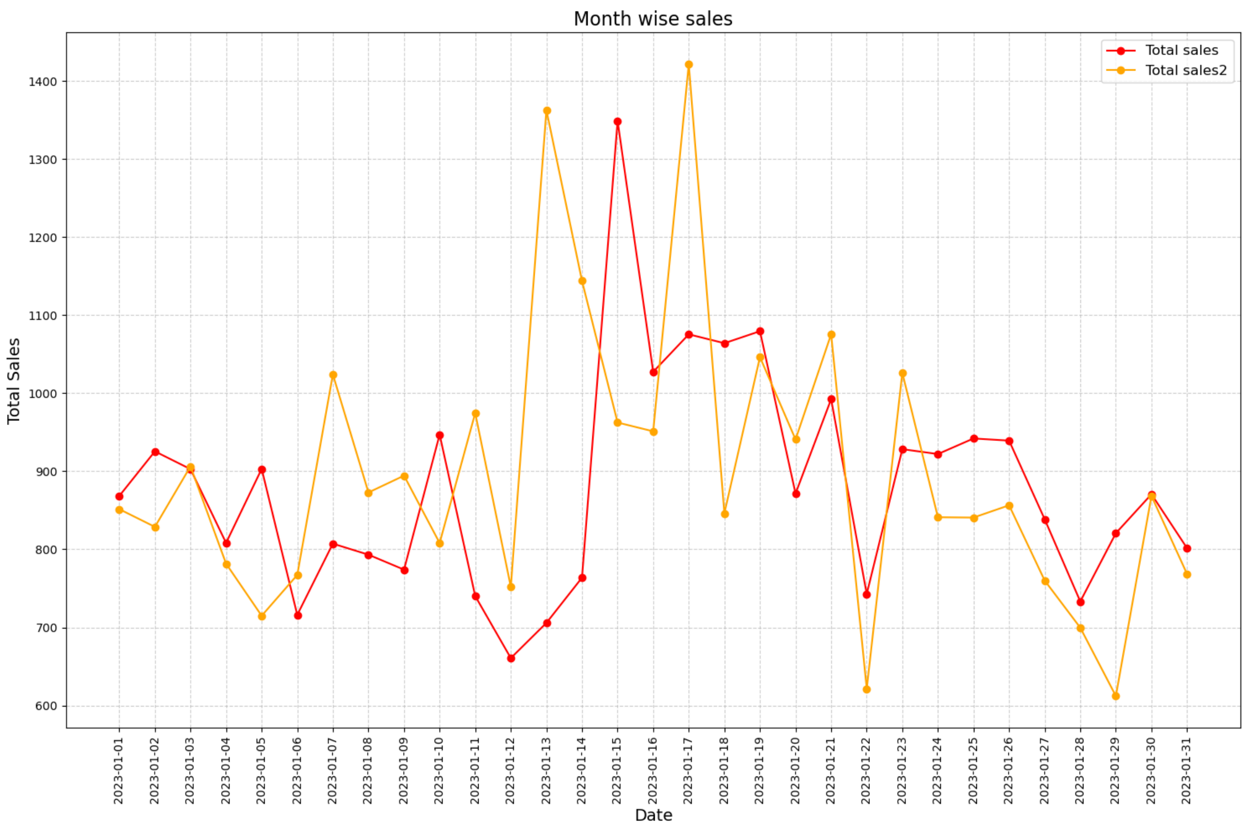
```
#print(sales["product_category"[.unique())
#date wise sales using input user for month year
month=input("Enter the month:")
store=input("Enter the store name=")
s2=input("Enter the store")
#trend_sales=sales[sales["transaction_date"].dt.month==int(month)]
date_trend_sale=pd.DataFrame(sales[(sales["transaction_date"].dt.month
==int(month))&(sales["store_location"]==store)].groupby(sales["transac
tion_date"].dt.date)["total_sales"].sum()).reset_index()
date_trend_sale["transaction_date"]=pd.to_datetime(date_trend_sale["tr
ansaction_date"])
d2=pd.DataFrame(sales[(sales["transaction_date"].dt.month==int(month))
&(sales["store_location"]==s2)].groupby(sales["transaction_date"].dt.d
ate)["total_sales"].sum()).reset_index()
print(date_trend_sale)
print(d2)
#plotting line chart
plt.figure(figsize=(15,10))
plt.plot(date_trend_sale['transaction_date'],date_trend_sale['total_sa
les'],
         marker='o',linestyle='-',color='red',label='Total sales')
plt.plot(d2['transaction_date'],d2['total_sales'],
         marker='o',linestyle='-',color='orange',label='Total sales2')
plt.title('Month wise sales',fontsize=16)
plt.xlabel('Date',fontsize=14)
```

```python
plt.ylabel('Total Sales',fontsize=14)
plt.grid(True, linestyle='--',alpha=0.6)
plt.xticks(date_trend_sale['transaction_date'],labels=date_trend_sale[
'transaction_date'].dt.strftime('%Y-%m-%d'),rotation=90)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

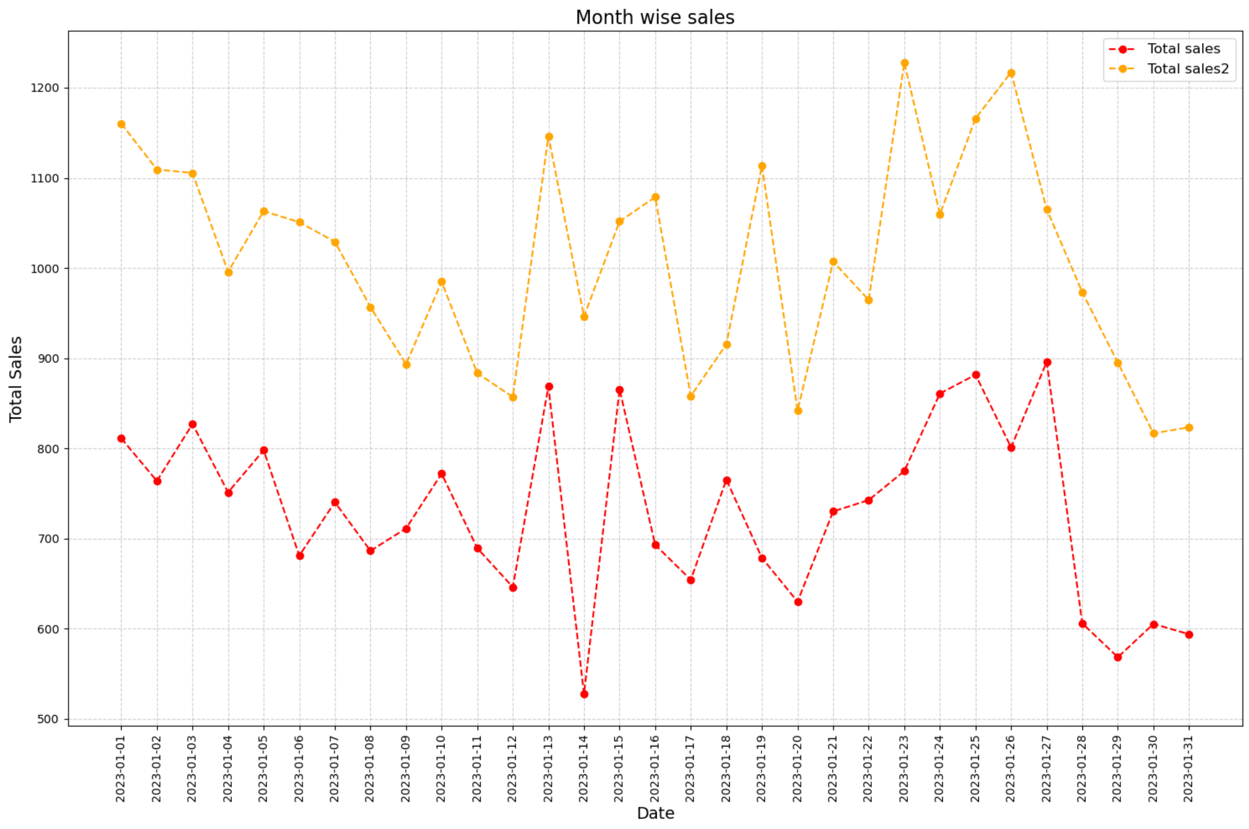
```
Enter the month: 1
Enter the store name= Astoria
Enter the store Hell's Kitchen
```

```
   transaction_date  total_sales
0        2023-01-01       868.40
1        2023-01-02       925.50
2        2023-01-03       902.75
3        2023-01-04       808.25
4        2023-01-05       903.05
5        2023-01-06       716.05
6        2023-01-07       807.30
7        2023-01-08       793.15
8        2023-01-09       774.01
9        2023-01-10       947.00
10       2023-01-11       740.15
11       2023-01-12       660.70
12       2023-01-13       705.80
13       2023-01-14       763.70
14       2023-01-15      1348.93
15       2023-01-16      1027.33
16       2023-01-17      1075.40
17       2023-01-18      1064.03
18       2023-01-19      1079.38
19       2023-01-20       871.43
20       2023-01-21       992.35
21       2023-01-22       742.80
22       2023-01-23       928.30
23       2023-01-24       922.05
24       2023-01-25       942.00
25       2023-01-26       939.25
26       2023-01-27       838.55
27       2023-01-28       733.25
28       2023-01-29       820.70
29       2023-01-30       870.60
30       2023-01-31       801.50
   transaction_date  total_sales
0        2023-01-01       851.45
1        2023-01-02       828.80
2        2023-01-03       906.25
3        2023-01-04       781.65
4        2023-01-05       714.90
```

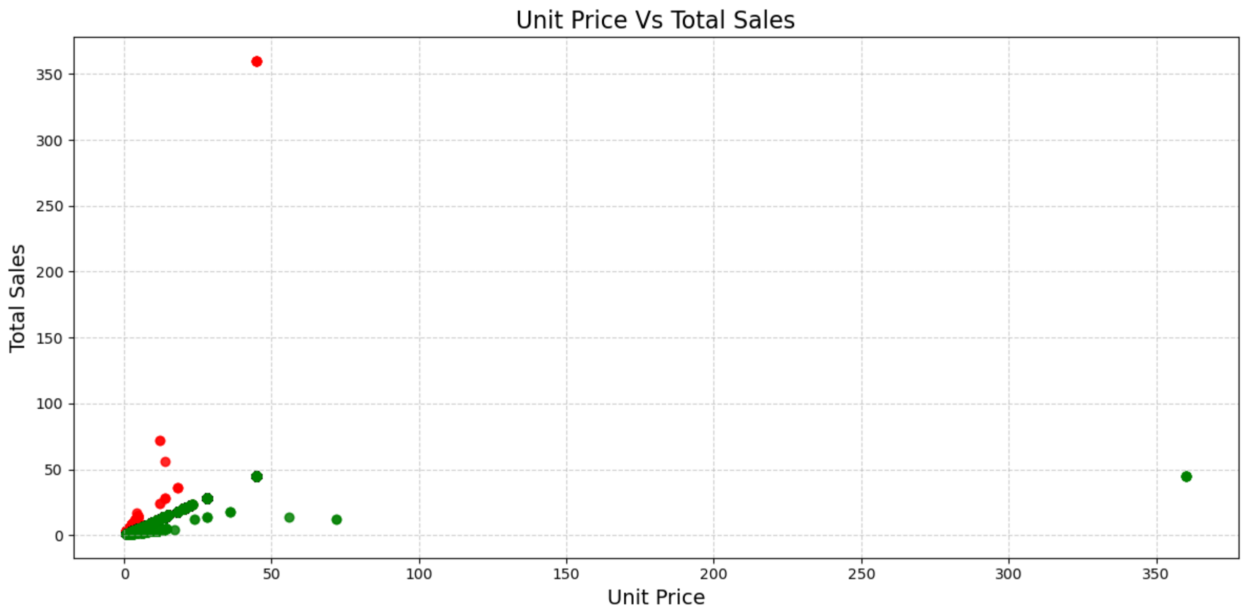
```
5         2023-01-06      767.20
6         2023-01-07     1024.10
7         2023-01-08      872.83
8         2023-01-09      894.40
9         2023-01-10      808.10
10        2023-01-11      974.55
11        2023-01-12      751.90
12        2023-01-13     1362.60
13        2023-01-14     1143.81
14        2023-01-15      962.43
15        2023-01-16      951.15
16        2023-01-17     1421.50
17        2023-01-18      846.10
18        2023-01-19     1046.25
19        2023-01-20      940.75
20        2023-01-21     1075.50
21        2023-01-22      621.45
22        2023-01-23     1026.10
23        2023-01-24      841.10
24        2023-01-25      840.70
25        2023-01-26      856.43
26        2023-01-27      760.05
27        2023-01-28      699.65
28        2023-01-29      612.15
29        2023-01-30      868.40
30        2023-01-31      768.40
```



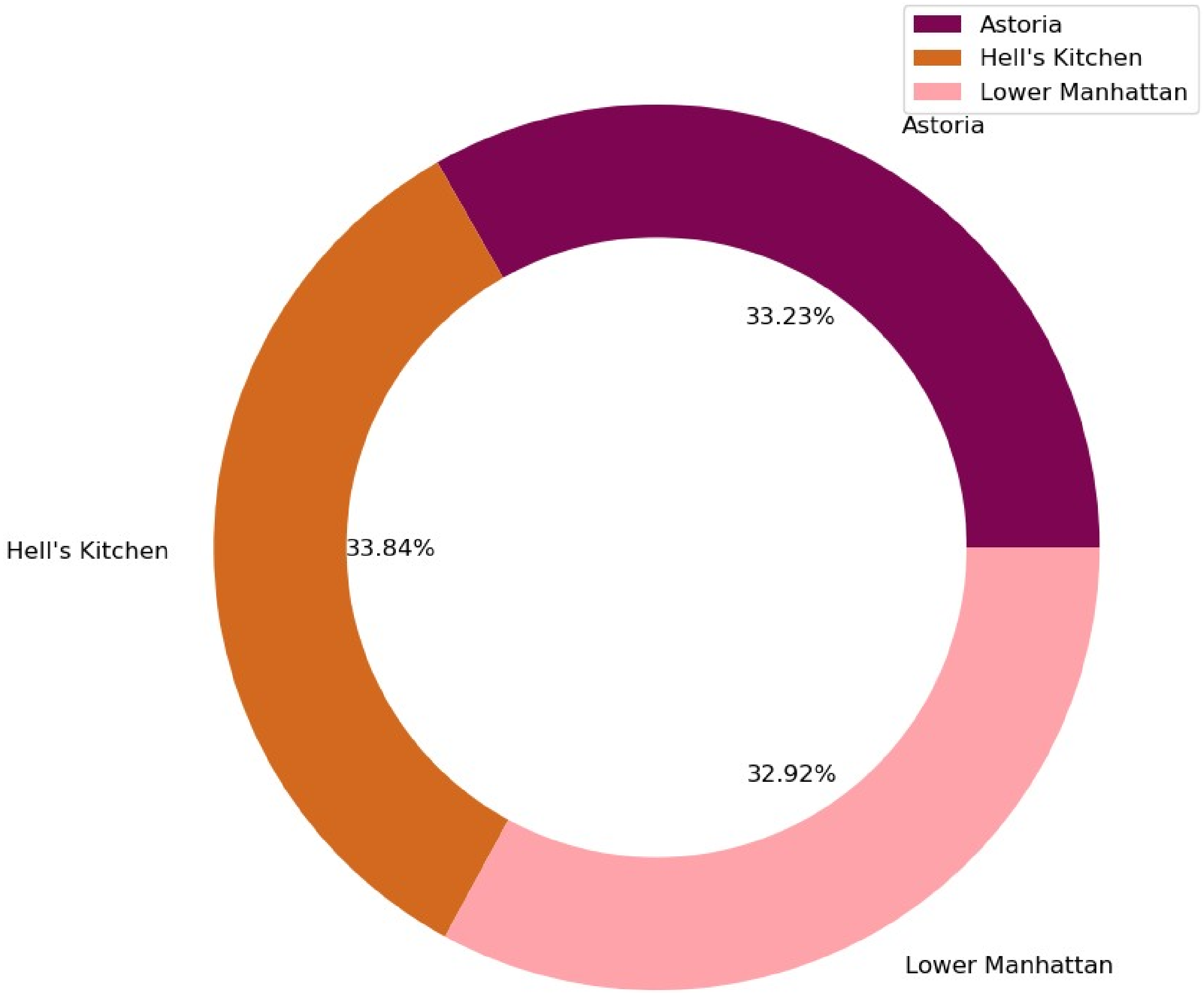
```
m1=input("Enter the month=")
p1=input("Enter the product category=")
p2=input("Enter the product category=")
d1=pd.DataFrame(sales[(sales['transaction_date'].dt.month==int(m1))&(s
ales["product_category"]==p1)].groupby("transaction_date")
["total_sales"].sum()).reset_index()
d1["transaction_date"]=pd.to_datetime(date_trend_sale["transaction_dat
e"])
d2=pd.DataFrame(sales[(sales['transaction_date'].dt.month==int(m1))&(s
ales["product_category"]==p2)].groupby("transaction_date")
["total_sales"].sum()).reset_index()
d2["transaction_date"]=pd.to_datetime(date_trend_sale["transaction_dat
e"])
plt.figure(figsize=(15,10))
plt.plot(d1['transaction_date'],d1['total_sales'],
         marker='o',linestyle='--',color='red',label='Total sales')
plt.plot(d2['transaction_date'],d2['total_sales'],
         marker='o',linestyle='--',color='orange',label='Total
sales2')
plt.title('Month wise sales',fontsize=16)
plt.xlabel('Date',fontsize=14)
plt.ylabel('Total Sales',fontsize=14)
plt.grid(True, linestyle='--',alpha=0.6)
plt.xticks(date_trend_sale['transaction_date'],labels=date_trend_sale[
'transaction_date'].dt.strftime('%Y-%m-%d'),rotation=90)
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()

Enter the month= 1
Enter the product category= Tea
Enter the product category= Coffee
```



```
plt.figure(figsize=(12,6))
plt.scatter(sales["unit_price"], sales["total_sales"], color="red",
alpha=0.6, label="Unit Price")
plt.scatter(sales["total_sales"], sales["unit_price"], color="green",
alpha=0.6, label="Total Sales")
plt.title("Unit Price Vs Total Sales", fontsize=16)
plt.xlabel("Unit Price", fontsize=14)
plt.ylabel("Total Sales", fontsize=14)
plt.grid(True, linestyle="--", alpha=0.6)
plt.tight_layout()
plt.show()
```



```python
# Donut Chart
store_sales = pd.DataFrame(sales.groupby("store_location")
["total_sales"].sum()).reset_index()
plt.figure(figsize=(10,10))
plt.pie(store_sales["total_sales"],
       labels=store_sales["store_location"],
       autopct="%1.2f%%",
       colors=["#7D0552","#D2691E","#FEA3AA"],
       textprops={"fontsize":12})
plt.gca().add_artist(plt.Circle((0,0), 0.7, fc="white"))
plt.legend(fontsize=12, loc="upper right")
plt.show()
```


Astoria
Hell's Kitchen
Lower Manhattan
Astoria
33.23%
Hell's Kitchen
33.84%
32.92%
Lower Manhattan