Restaurant Sales report

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

loading data

```
rdf=pd.read_csv(r"C:\Users\Bhoomika.G\OneDrive\Documents\Balaji Fast
Food Sales.csv")

rdf.head()

   order_id        date        item_name  item_type  item_price
quantity  \
0         1  07-03-2022         Aalopuri   Fastfood          20
13
1         2   8/23/2022          Vadapav   Fastfood          20
15
2         3  11/20/2022          Vadapav   Fastfood          20
1
3         4  02-03-2023  Sugarcane juice  Beverages          25
6
4         5  10-02-2022  Sugarcane juice  Beverages          25
8

   transaction_amount transaction_type received_by time_of_sale
0                 260              NaN         Mr.        Night
1                 300             Cash         Mr.    Afternoon
2                  20             Cash         Mr.    Afternoon
3                 150           Online         Mr.        Night
4                 200           Online         Mr.      Evening
```

EDA Techniques

1.checking basic information

```
rdf.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
```

```
---  ------              --------------  -----
 0   order_id            1000 non-null   int64
 1   date                1000 non-null   object
 2   item_name           1000 non-null   object
 3   item_type           1000 non-null   object
 4   item_price          1000 non-null   int64
 5   quantity            1000 non-null   int64
 6   transaction_amount  1000 non-null   int64
 7   transaction_type    893 non-null    object
 8   received_by         1000 non-null   object
 9   time_of_sale        1000 non-null   object
dtypes: int64(4), object(6)
memory usage: 78.3+ KB
```

```
Conclusion:
-here order_id ,date ,item_price, quantity, transection amount thes
are all in the int datatype
-item_name ,item_type,transiction_type ,recived_by,time_of_sale
-null value are exit and non-null value are also exit
-10 features  and 1000 entites
```

```
rdf.isnull().sum()
```

```
order_id                0
date                    0
item_name               0
item_type               0
item_price              0
quantity                0
transaction_amount      0
transaction_type      107
received_by             0
time_of_sale            0
dtype: int64
```

```
rdf.dropna(inplace=True)
```

```
rdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 893 entries, 1 to 999
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   order_id            893 non-null    int64
 1   date                893 non-null    object
 2   item_name           893 non-null    object
 3   item_type           893 non-null    object
 4   item_price          893 non-null    int64
 5   quantity            893 non-null    int64
 6   transaction_amount  893 non-null    int64
 7   transaction_type    893 non-null    object
 8   received_by         893 non-null    object
 9   time_of_sale        893 non-null    object
dtypes: int64(4), object(6)
memory usage: 76.7+ KB
```

```
rdf.isnull().sum()
```

```
order_id              0
date                  0
item_name             0
item_type             0
item_price            0
quantity              0
transaction_amount    0
transaction_type      0
received_by           0
time_of_sale          0
dtype: int64
```

```
Conclusion:
-all null values are droped
```

# Summary Statistics

```
rdf.describe(include='all')
```

```
           order_id        date    item_name item_type  item_price  \
count    893.000000         893          893       893  893.000000
unique          NaN         339            7         2         NaN
top             NaN  02-03-2023  Cold coffee  Fastfood         NaN
freq            NaN           7          138       619         NaN
mean     506.166853         NaN          NaN       NaN   33.359462
std      287.605949         NaN          NaN       NaN   15.000808
min        2.000000         NaN          NaN       NaN   20.000000
25%      260.000000         NaN          NaN       NaN   20.000000
50%      510.000000         NaN          NaN       NaN   25.000000
75%      755.000000         NaN          NaN       NaN   50.000000
max     1000.000000         NaN          NaN       NaN   60.000000

          quantity  transaction_amount transaction_type received_by  \
count   893.000000          893.000000              893         893
unique         NaN                 NaN                2           2
top            NaN                 NaN             Cash         Mr.
freq           NaN                 NaN              476         459
mean      8.100784          272.603583              NaN         NaN
std       4.390283          202.457635              NaN         NaN
min       1.000000           20.000000              NaN         NaN
25%       4.000000          120.000000              NaN         NaN
50%       8.000000          240.000000              NaN         NaN
```

```
75%      12.000000          360.000000              NaN          NaN
max      15.000000          900.000000              NaN          NaN

       time_of_sale
count           893
unique            5
top         Evening
freq            185
mean            NaN
std             NaN
min             NaN
25%             NaN
50%             NaN
75%             NaN
max             NaN
```

```
Conclusion:
1.order_id:
    -the minimum order_id is 2,maximum order_id is 10000,the average
order_id is 506.166
    -The majority of order_id b/w 260 to 755
2.Date:
    -here
3.item_name:
    -So we can say item are sligtly dominating
    -they are only 7 items are unique out of 893
4.item_type:
    -here item_type is dominating by 619 outof 893
    -there are 2 unique values
    -in this top one item_type is FASTFOOD
5.item_price:
    -the minimum item_price is 20,the maximum price is 60,the average
item_price is 33.35
    -The majority item_price  b/w 20 and 50
6.quantity:
    -the minmum quantity is 1,the maxmum quantity is 15,the avarage
quantity is 8.1
    -The majority of quantity b/w 4 and 12
7.transaction_amount:
    -the minmum transaction_amount is 20,the maximum
transaction_amount is 900,avaragetransaction_amount is 272.6
    -The majority of transaction_amount b/w 120 to 360
8.transiction_type:
    -transiction_type is dominating by 476 outof 893
    -only 2 unique values
    -most of transiction_type used is CASH
9.received_by:
    -received_by is dominating by 459
    -only 2 unique values
    -most of received_by Mr
```

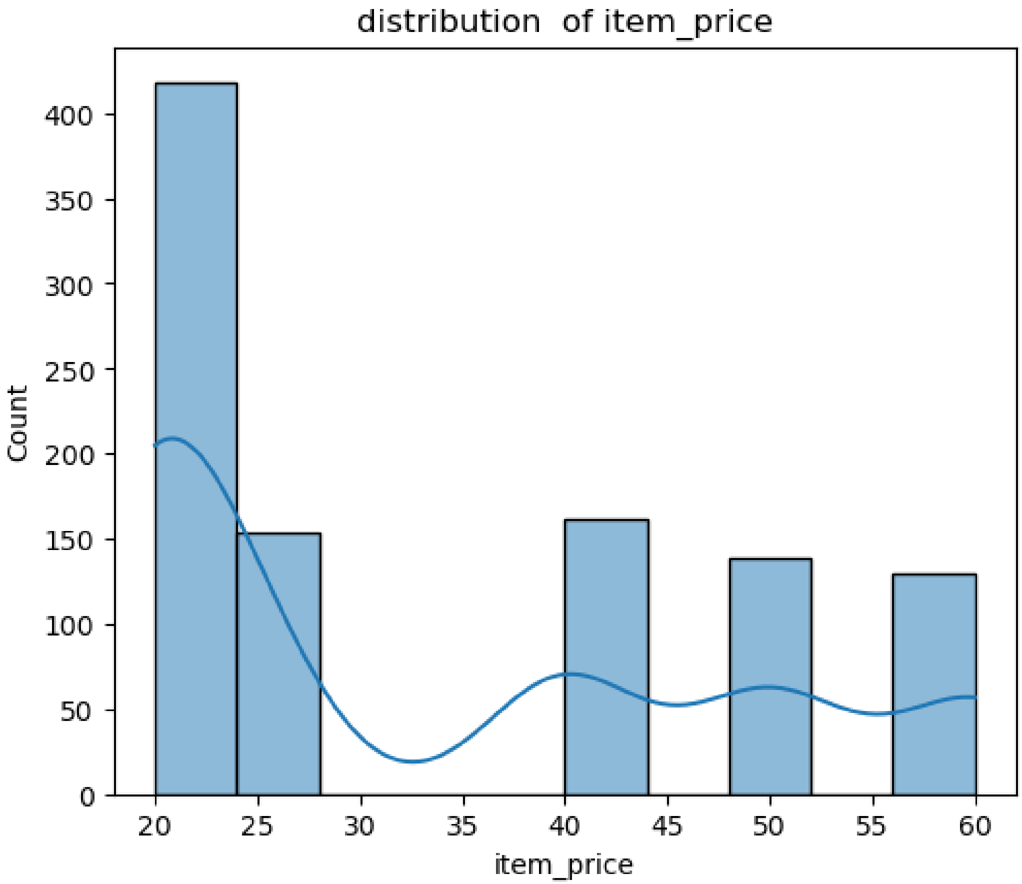
```
10.time_of_sale:
    -only 5 unique values
    -the top sales time is evining
```

visulalization

1.Histogram

Analysis of order_id

```
plt.figure(figsize =(6,5))
sns.histplot(rdf['item_price'],kde=True,bins=10)
plt.title('distribution  of item_price')
plt.show()
```



```
Conclusion:
-the price range is 20 to 60
-the order_price majority b/w 20 to23
-no outline occurs
```

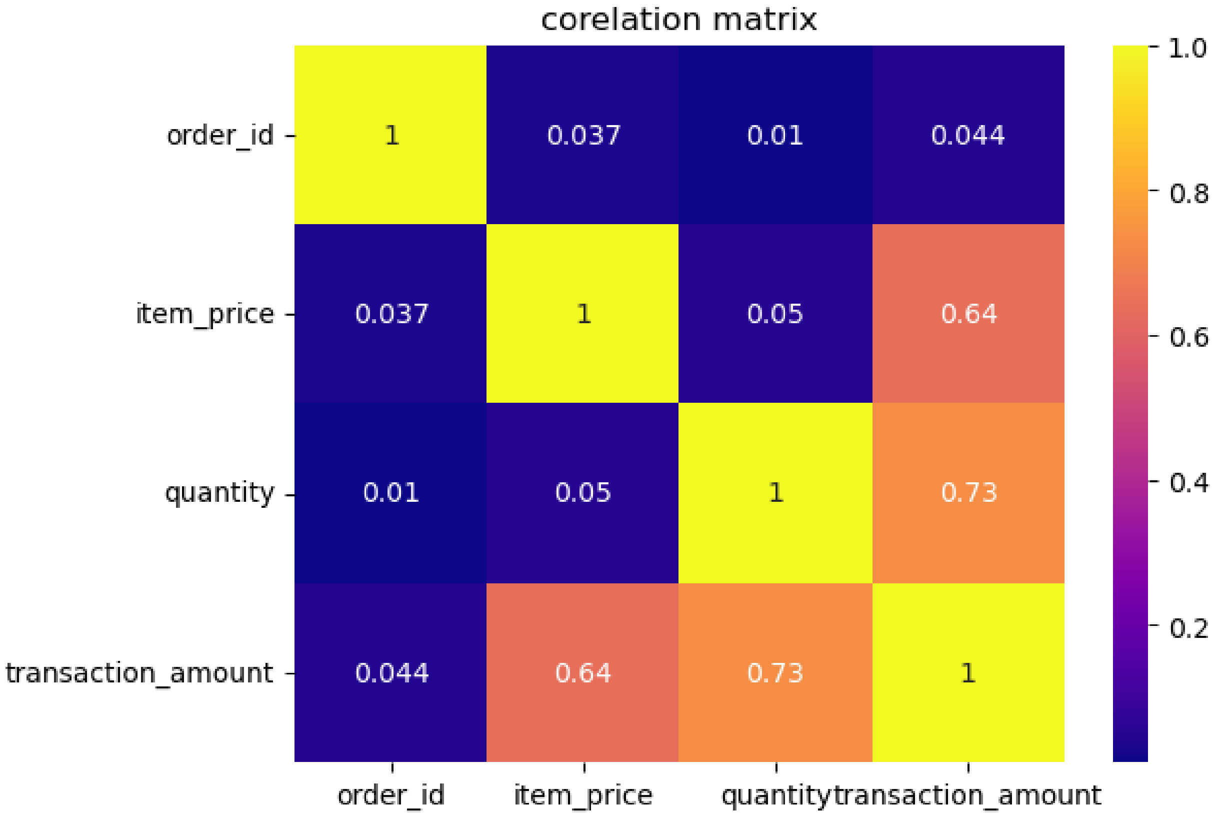
```
-the average occus in 22
-there is a right skew
```

2.Heatmap

```
rdf1=rdf1.select_dtypes(include=['number'])
rdf1.head()
```

```
   order_id  item_price  quantity  transaction_amount
0         1          20        13                 260
1         2          20        15                 300
2         3          20         1                  20
3         4          25         6                 150
4         5          25         8                 200
```

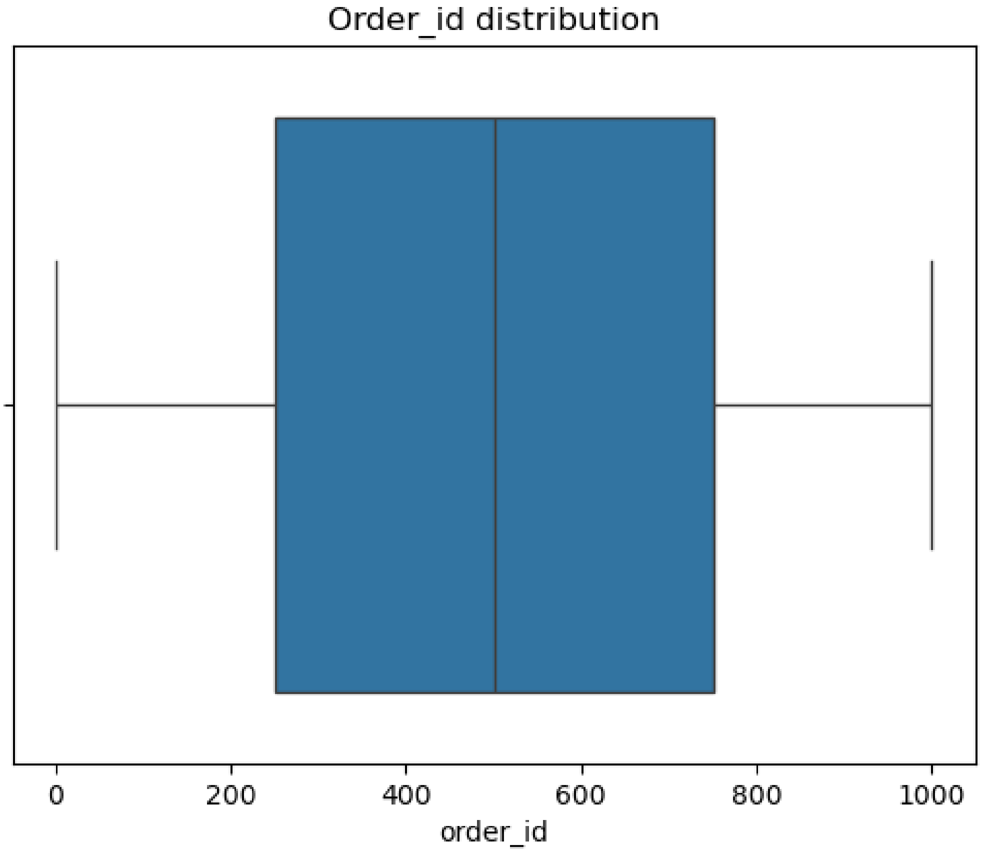
```
plt.figure(figure=(6,5))
sns.heatmap(rdf1.corr(),cmap='plasma',annot=True)
plt.title('corelation matrix')
plt.show()
```



```
conclusion:
-quantity and transiction amount have a good corelation
-order_id and quantity have a lowest corelation
```

3.Boxplot

```
plt.figure(figure=(6,5))
sns.boxplot(x=rdf['order_id'])
plt.title('Order_id distribution')
plt.show()
```

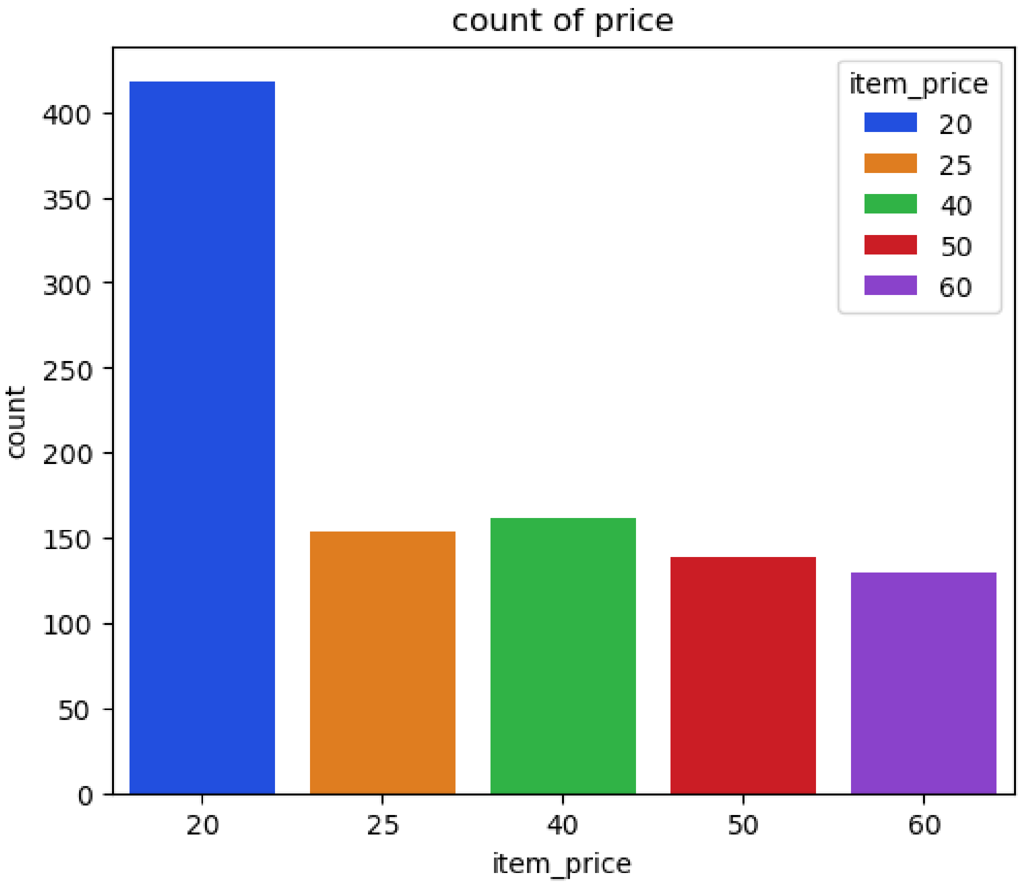


```
conclusion:
-no outline
-the boxplot is normal
--the majority order_id is the 250 to 780
-the average  is 500
```

4.countplot

```
plt.figure(figsize=(6,5))
sns.countplot(x=rdf['item_price'],palette='bright',hue=rdf['item_price
'])
plt.title('count of price')

Text(0.5, 1.0, 'count of price')
```

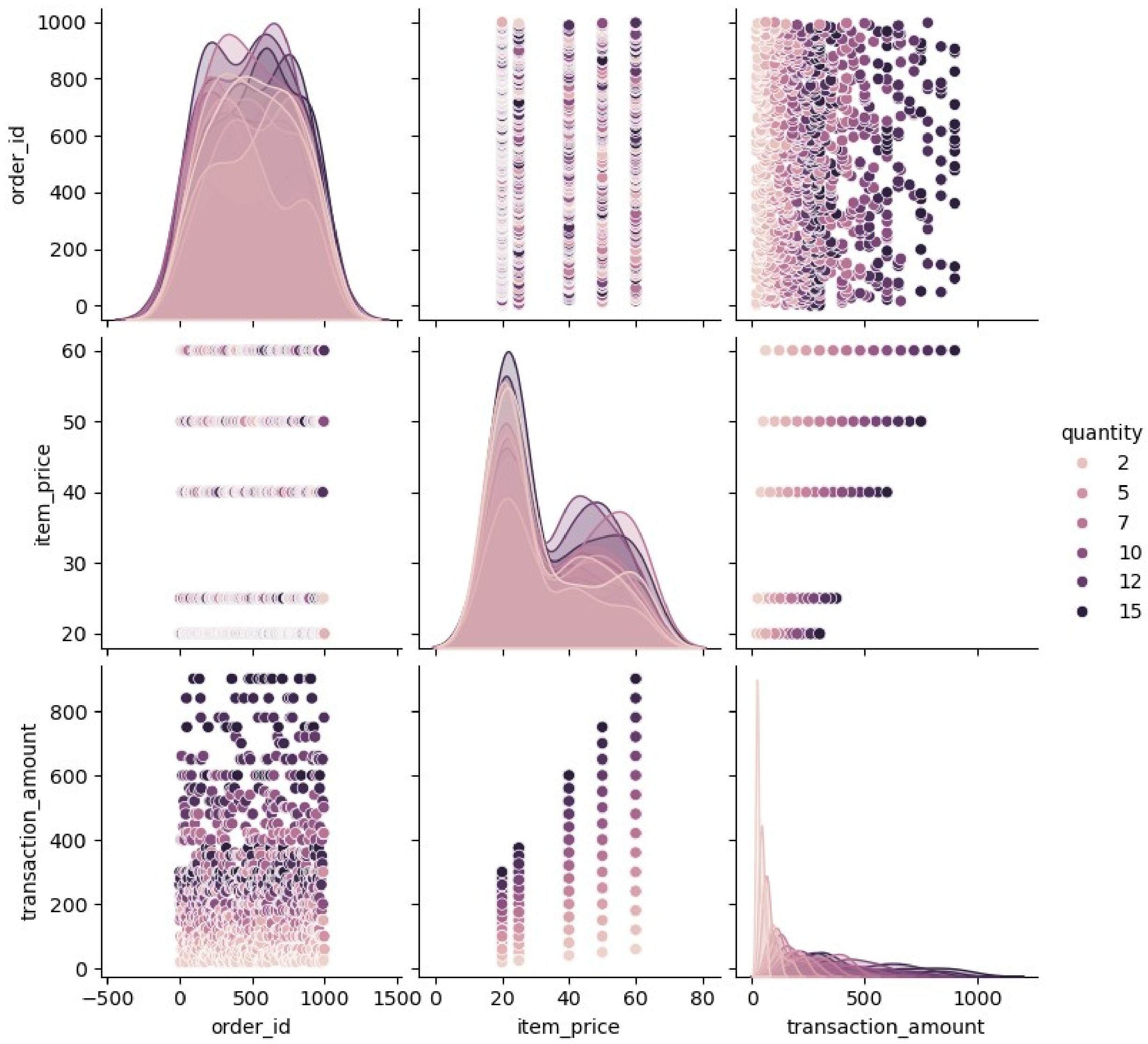


```
conclusion:
-majority food is avalible in 20 price
-the average item price is 40
```

5.pairplot

```
sns.pairplot(rdf,hue='quantity')
```

```
<seaborn.axisgrid.PairGrid at 0x1af8ee2fbf0>
```



```
conclusion:
-the peak quantity are given to order_id
-coustomers  are conistences in transection amount

rdf.groupby("time_of_sale").agg({'order_id':['count','mean']})
```

```
             order_id
                count        mean
time_of_sale
Afternoon         205  505.331707
Evening           201  505.721393
Midnight          199  502.723618
Morning           190  481.684211
Night             205  505.829268
```