In [1]:
```
# Task-2:
# Create EDA using Test Data file(Yoshops.com Sale Order file) :
# Input Value for genrate Graph chart:
# Enter 1 to see the analysis of Reviews given by Customers
# Enter 2 to see the analysis of different payment methods used by the Custome
# Enter 3 to see the analysis of Top Consumer States of India
# Enter 4 to see the analysis of Top Consumer Cities of India
# Enter 5 to see the analysis of Top Selling Product Categories
# Enter 6 to see the analysis of Reviews for All Product Categories
# Enter 7 to see the analysis of Number of Orders Per Month Per Year
# Enter 8 to see the analysis of Reviews for Number of Orders Per Month Per Ye
# Enter 9 to see the analysis of Number of Orders Across Parts of a Day
# Enter 10 to see the Full Report

# Enter the number to see the analysis of your choice: 1

# OutPut:Genrate analysis report in format PDF and Excel file
```

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

In [3]:
```
df1 = pd.read_csv("orders_2016-2020_Dataset.csv")
df2 = pd.read_csv("review_dataset.csv")
```

In [4]:
```
df1.describe()
```

Out[4]:

	Gift Cards	Special Instructions	LineItem Qty
count	0.0	0.0	2297.000000
mean	NaN	NaN	3.740531
std	NaN	NaN	46.748117
min	NaN	NaN	1.000000
25%	NaN	NaN	1.000000
50%	NaN	NaN	1.000000
75%	NaN	NaN	1.000000
max	NaN	NaN	999.000000

In [5]:
```
df1.shape
```

Out[5]:
```
(2297, 41)
```

In [6]:
```
df2.describe()
```

Out[6]:

	product_name	product_url	category	status	stars
count	1861	1861	1861	606	606
unique	523	524	62	1	12
top	Hammer Sting 2.0 Wireless Bluetooth Neckband E...	https://yoshops.com/products/hammer-sting-2-0-...	Mobiles	Reviewd	5.0 star rating
freq	18	18	163	606	499

In [7]:
```
df2.shape
```

Out[7]:
```
(1861, 5)
```

In [8]:
```
df1.info()
```

```
 23  Shipping Name                 2297 non-null   object
 24  Shipping Country              2297 non-null   object
 25  Shipping Street Address       2279 non-null   object
 26  Shipping Street Address 2     1526 non-null   object
 27  Shipping City                 2279 non-null   object
 28  Shipping State                2276 non-null   object
 29  Shipping Zip                  2276 non-null   object
 30  Gift Cards                    0 non-null      float64
 31  Payment Method                240 non-null    object
 32  Tracking #                    83 non-null     object
 33  Special Instructions          0 non-null      float64
 34  LineItem Name                 2297 non-null   object
 35  LineItem SKU                  2208 non-null   object
 36  LineItem Options              169 non-null    object
 37  LineItem Add-ons              91 non-null     object
 38  LineItem Qty                  2297 non-null   int64
 39  LineItem Sale Price           2297 non-null   object
 40  LineItem Type                 2297 non-null   object
dtypes: float64(2), int64(1), object(38)
memory usage: 735.9+ KB
```

In [9]:
```
count_nan = df1.isna().sum().sum()
count_nan
```

Out[9]:
```
38748
```

In [10]:
```
df1["Billing Name"].isna().sum()
```

Out[10]:
```
1967
```

In [11]:
```
df1["Billing Country"].isna().sum()
```

Out[11]:
```
1967
```

In [12]:
```
df1["Shipping Name"].isna().sum()
```

Out[12]:
```
0
```

In [13]:
```
print(df1['Shipping Street Address'].isna().sum())
df1['Shipping Street Address 2'].isna().sum()
```

```
18
```

Out[13]:
```
771
```

In [14]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1861 entries, 0 to 1860
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   product_name  1861 non-null   object
 1   product_url   1861 non-null   object
 2   category      1861 non-null   object
 3   status        606 non-null    object
 4   stars         606 non-null    object
dtypes: object(5)
memory usage: 72.8+ KB
```

In [15]:
```
df1.head()
```

Out[15]:

	Order #	Order Date and Time Stamp	Fulfillment Status	Payment Status	Payment Date and Time Stamp	Fulfillment Date and Time Stamp	Currency	Subtotal	Shippir Methc
0	R929392577	09-11-2020 20:36:26 +0530	Unfulfilled	Unpaid	NaN	NaN	INR	₹ 799.00	Shi Fre
1	R653462960	09-11-2020 20:18:26 +0530	Unfulfilled	Unpaid	NaN	NaN	INR	₹ 699.00	Shi Fre
2	R226302759	09-11-2020 19:56:21 +0530	Unfulfilled	Unpaid	NaN	NaN	INR	₹ 799.00	Shi Fre
3	R390235057	09-11-2020 19:37:40 +0530	Unfulfilled	Unpaid	NaN	NaN	INR	₹ 599.00	Shi Fre
4	R813855117	09-11-2020 18:35:47 +0530	Cancelled	Paid	NaN	NaN	INR	₹ 699.00	Shi Fre

5 rows × 41 columns

```
In [16]: df2.head(14)
```

Out[16]:

	product_name	product_url	category	status	stars
0	Sony PlayStation PS2 Gaming Console 150 GB Har...	https://yoshops.com/products/sony-playstation-...	Toys & Games	Reviewd	5.0 star rating
1	Vmax HX 750 Quadcopter Drone (No Camera)	https://yoshops.com/products/hx-750-remote-con...	Toys & Games	Reviewd	5.0 star rating
2	Yoshops VR BOX Virtual Reality Glasses Headset...	https://yoshops.com/products/yoshops-vr-box-vi...	Toys & Games	Reviewd	5.0 star rating
3	Sony PlayStation PS3 Console Slim 320 GB (Black)	https://yoshops.com/products/sony-playstation-...	Toys & Games	Reviewd	4.9 star rating
4	Barbie Doll (pink)	https://yoshops.com/products/barbie-doll	Toys & Games	Reviewd	4.9 star rating
5	HX-713 Remote Control Helicopter	https://yoshops.com/products/hx-713-remote-con...	Toys & Games	Reviewd	4.9 star rating
6	Puppy House Coin Piggy Bank	https://yoshops.com/products/puppy-house-coin-...	Toys & Games	Reviewd	5.0 star rating
7	The Amazing Spider Man Micro Drone Q Series Hy...	https://yoshops.com/products/the-amazing-spide...	Toys & Games	Reviewd	5.0 star rating
8	Super Power JCB Truck Construction Loader Exca...	https://yoshops.com/products/super-power-jcb-t...	Toys & Games	Reviewd	5.0 star rating
9	Falcon Drone Four Axis Aircraft with 2.4 GHz R...	https://yoshops.com/products/falcon-drone-four...	Toys & Games	Reviewd	5.0 star rating
10	Kids Drone Quadcopter 2.4G 6-Channel Without C...	https://yoshops.com/products/kids-drone-quadco...	Toys & Games	Reviewd	4.6 star rating
11	Sony PlayStation PS1 with in-built DVD Player ...	https://yoshops.com/products/sony-playstation-...	Toys & Games	Reviewd	5.0 star rating
12	VMax HX763 Vision Drone 2.4GHz RC Quad-copter ...	https://yoshops.com/products/vmax-vision-hx763...	Toys & Games	Reviewd	5.0 star rating
13	HX770 V-Max Aircraft Drone	https://yoshops.com/products/hx770-v-max-aircr...	Toys & Games	Reviewd	4.9 star rating

In [17]:

```
df3 = df1[["Order #","Order Date and Time Stamp","Fulfillment Status","Payment
df3.head(200)
```

Out[17]:

	Order #	Order Date and Time Stamp	Fulfillment Status	Payment Status	Total	Shipping Street Address	Shipping Name	Ship
0	R929392577	09-11-2020 20:36:26 +0530	Unfulfilled	Unpaid	₹ 799.00	Sec-86 nawada fatehpur, postoffice-Sikanderpur...	Neetu Yadav	12
1	R653462960	09-11-2020 20:18:26 +0530	Unfulfilled	Unpaid	₹ 699.00	Nashik	Lucky Koli	42
2	R226302759	09-11-2020 19:56:21 +0530	Unfulfilled	Unpaid	₹ 799.00	Madhuranagar 2nd stage hostel	Raghu A	56
3	R390235057	09-11-2020 19:37:40 +0530	Unfulfilled	Unpaid	₹ 599.00	Civil line near lic office	Hemant Vaishnav	Gfj d Hald us
4	R813855117	09-11-2020 18:35:47 +0530	Cancelled	Paid	₹ 699.00	Nps thakur sthan Rajgir	Munna mumar Munna	
...	...	...	...	...	...	...	...	
195	R718754077	30-10-2020 08:05:01 +0530	Unfulfilled	Unpaid	₹ 1,199.00	102budwa	Shivam Bais	48
196	R075519011	30-10-2020 08:00:22 +0530	Unfulfilled	Unpaid	₹ 2,299.00	102budwa	Shivam Bais	48
197	R431135392	30-10-2020 05:49:02 +0530	Unfulfilled	Unpaid	₹ 799.00	Bari kewai	Rohit Raj	I
198	R129726220	30-10-2020 00:55:58 +0530	Unfulfilled	Unpaid	₹ 799.00	Anaj mandi	Lokesh Agfarwal	12

	Order #	Order Date and Time Stamp	Fulfillment Status	Payment Status	Total	Shipping Street Address	Shipping Name	Ship
199	R875418116	29-10-2020 22:42:35 +0530	Unfulfilled	Unpaid	₹ 999.00	Bsk 2nd stage kaveri nagar Bangalore560070near...	Javeed miraj	56

200 rows × 11 columns

In [18]:
```
df1['Payment Method'].value_counts()
```

Out[18]:
```
Offline Payment ₹1,499.00     18
Offline Payment ₹1,999.00     10
Offline Payment ₹799.00       10
Offline Payment ₹300.00        9
Offline Payment ₹1,399.00      8
                              ..
Offline Payment ₹19,176.00     1
Offline Payment ₹9,950.00      1
Offline Payment ₹13,990.00     1
Offline Payment ₹18,995.00     1
Offline Payment ₹2,000.00      1
Name: Payment Method, Length: 96, dtype: int64
```

In [26]:
```
df1 = df1.rename(columns={'LineItem Name': 'product_name'})
```

In [27]:
```
df1 = pd.merge(df1,df2, on='product_name')
df1
```

4	R344945254	2020-07-11 14:24:09+05:30	Unfulfilled	Unpaid	NaN	NaN	INR	₹ 69
...	...	...	...	...	...	...	...	
4083	R968858875	2016-06-10 16:22:38+05:30	Cancelled	Unpaid	NaN	NaN	INR	9,00
4084	R326096945	2016-06-10 15:02:12+05:30	Cancelled	Unpaid	NaN	NaN	INR	₹ 65
4085	R326096945	2016-06-10 15:02:12+05:30	Cancelled	Unpaid	NaN	NaN	INR	₹ 65
4086	R378835168	2016-06-10 14:45:39+05:30	Cancelled	Unpaid	NaN	NaN	INR	2,70
4087	R378835168	2016-06-10 14:45:39+05:30	Cancelled	Unpaid	NaN	NaN	INR	2,70

In [28]:

```
df1['stars'] = df1['stars'].str.extract('(\d+)')

# convert the extracted values to numeric data type using pd.to_numeric()
df1['stars'] = pd.to_numeric(df1['stars'], errors='coerce')

# replace the NaN values with 0
df1['stars'] = df1['stars'].fillna(0).astype(int)

# print the updated dataframe
print(df1)
```

```
         Order # Order Date and Time Stamp Fulfillment Status Payment Status  
\
0     R653462960 2020-09-11 20:18:26+05:30        Unfulfilled         Unpaid  
1     R653462960 2020-09-11 20:18:26+05:30        Unfulfilled         Unpaid  
2     R926799219 2020-09-11 12:33:30+05:30          Cancelled           Paid  
3     R926799219 2020-09-11 12:33:30+05:30          Cancelled           Paid  
4     R344945254 2020-07-11 14:24:09+05:30        Unfulfilled         Unpaid  
...          ...                       ...                ...            ...  
4083  R968858875 2016-06-10 16:22:38+05:30          Cancelled         Unpaid  
4084  R326096945 2016-06-10 15:02:12+05:30          Cancelled         Unpaid  
4085  R326096945 2016-06-10 15:02:12+05:30          Cancelled         Unpaid  
4086  R378835168 2016-06-10 14:45:39+05:30          Cancelled         Unpaid  
4087  R378835168 2016-06-10 14:45:39+05:30          Cancelled         Unpaid  

     Payment Date and Time Stamp Fulfillment Date and Time Stamp Currency  \
0                            NaN                             NaN      INR   
1                            NaN                             NaN      INR   
2                            NaN                             NaN      INR   
3                            NaN                             NaN      INR   
4                            NaN                             NaN      INR   
...                          ...                             ...      ...   
4083                         NaN                             NaN      INR   
4084                         NaN                             NaN      INR   
4085                         NaN                             NaN      INR   
4086                         NaN                             NaN      INR   
4087                         NaN                             NaN      INR   

        Subtotal Shipping Method Shipping Cost  ... LineItem Qty  \
0       ₹ 699.00      Ships Free        ₹ 0.00  ...            1   
1       ₹ 699.00      Ships Free        ₹ 0.00  ...            1   
2       ₹ 699.00      Ships Free        ₹ 0.00  ...            1   
3       ₹ 699.00      Ships Free        ₹ 0.00  ...            1   
4       ₹ 699.00      Ships Free        ₹ 0.00  ...            1   
...          ...             ...           ...  ...          ...   
4083  ₹ 9,000.00   Free Shipping        ₹ 0.00  ...            1   
4084    ₹ 650.00      Ships Free        ₹ 0.00  ...            1   
4085    ₹ 650.00      Ships Free        ₹ 0.00  ...            1   
4086  ₹ 2,700.00      Ships Free        ₹ 0.00  ...            1   
4087  ₹ 2,700.00      Ships Free        ₹ 0.00  ...            1   

     LineItem Sale Price LineItem Type  Year      Month Hour  \
0               ₹ 699.00      physical  2020  September   20   
1               ₹ 699.00      physical  2020  September   20   
2               ₹ 699.00      physical  2020  September   12   
3               ₹ 699.00      physical  2020  September   12   
4               ₹ 699.00      physical  2020       July   14   
...                  ...           ...   ...        ...  ...   
4083          ₹ 9,000.00      physical  2016       June   16   
4084            ₹ 650.00      physical  2016       June   15   
4085            ₹ 650.00      physical  2016       June   15   
4086          ₹ 2,700.00      physical  2016       June   14   
4087          ₹ 2,700.00      physical  2016       June   14   

                                            product_url  \
0     https://yoshops.com/products/samsung-u-flex-wi... (https://yoshops.com/
products/samsung-u-flex-wi...)
1     https://yoshops.com/products/samsung-u-flex-wi... (https://yoshops.com/
products/samsung-u-flex-wi...)
2     https://yoshops.com/products/samsung-u-flex-wi... (https://yoshops.com/
products/samsung-u-flex-wi...)
3     https://yoshops.com/products/samsung-u-flex-wi... (https://yoshops.com/
products/samsung-u-flex-wi...)
4     https://yoshops.com/products/samsung-u-flex-wi... (https://yoshops.com/
products/samsung-u-flex-wi...)
...                                                  ...
4083  https://yoshops.com/products/iball-excelance-c... (https://yoshops.com/
products/iball-excelance-c...)
4084  https://yoshops.com/products/ambranepowerbank-... (https://yoshops.com/
products/ambranepowerbank-...)
4085  https://yoshops.com/products/ambranepowerbank-... (https://yoshops.com/
products/ambranepowerbank-...)
4086  https://yoshops.com/products/samsung-metro-350... (https://yoshops.com/
products/samsung-metro-350...)
4087  https://yoshops.com/products/samsung-metro-350... (https://yoshops.com/
products/samsung-metro-350...)

                    category   status stars
0                    Mobiles  Reviewd     5
1                 Headphones  Reviewd     5
2                    Mobiles  Reviewd     5
3                 Headphones  Reviewd     5
4                    Mobiles  Reviewd     5
...                      ...      ...   ...
4083                 Tablets  Reviewd     5
4084                 Mobiles      NaN     0
4085      Mobiles Accessories      NaN     0
4086                 Mobiles      NaN     0
4087  Feature Keypad Mobiles      NaN     0

[4088 rows x 48 columns]
```

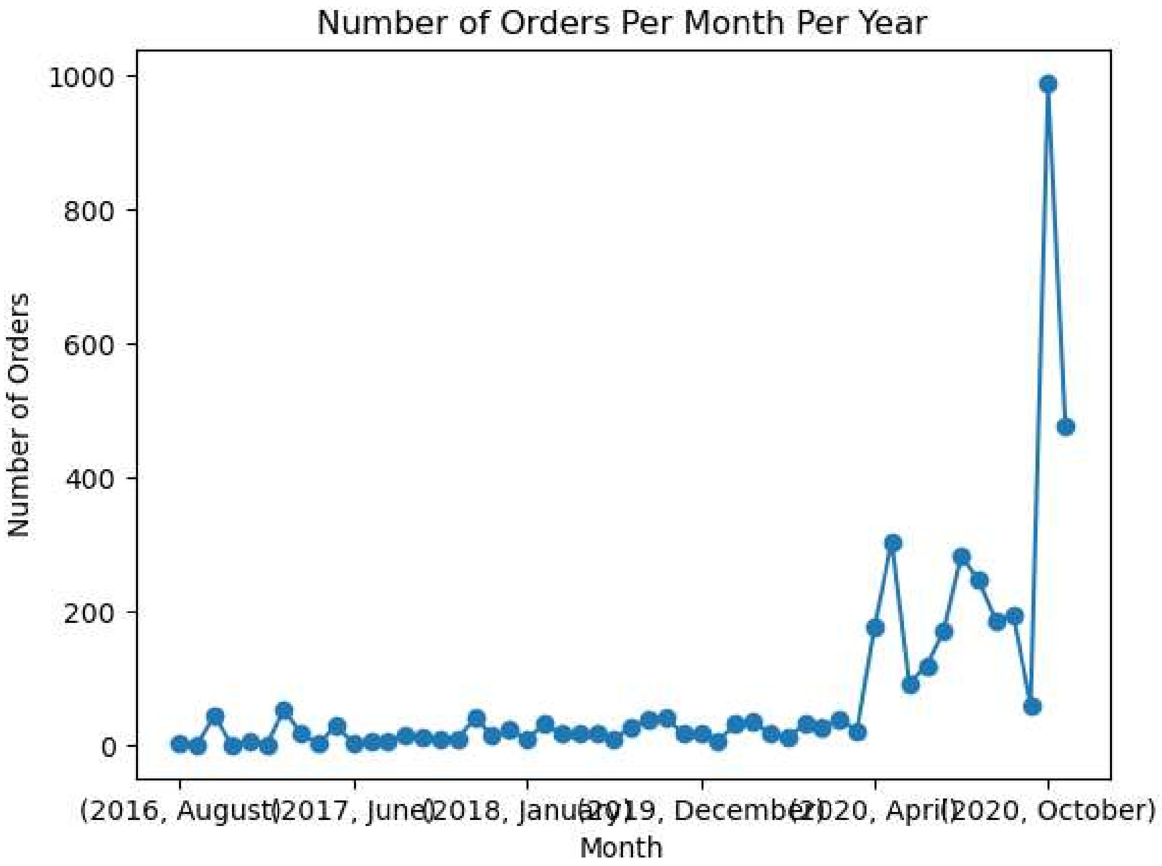
In [29]:
```
df1["stars"].dtypes
```

Out[29]:
```
dtype('int32')
```

In [61]:

```
print("Enter the number to see the analysis of your choice:")
af = int(input())
if af == 1:
    print("analysis of Reviews given by Customers")
    gkk = df2.groupby(['stars','product_name'])
    print(gkk.first())
elif af == 2:
    count1 = 0
    count2 = 0
    list1 = df1['Payment Method'].values.tolist()
    cleanedList = [x for x in list1 if str(x) != 'nan']
    for i in cleanedList:
        list2 = i.split()
        if "CCAvenue" in list2:
            count1 += 1
        if "Offline" in list2:
            count2 += 1
    print("CCAvenue = ",count1)
    print("Offline payment = ",count2)
    list1 = ["CCAvenue", "Offline payment"]
    list2 = [count1, count2]
    plt.bar(list1, list2, color ='maroon',
        width = 0.4)
elif af == 3:
    print("Top consumer states in India :\n",df1['Shipping State'].value_count
elif af ==4:
    print("Top consumer cities In India: \n",df1['Shipping City'].value_counts
elif af == 5:
    print("Top selling products In India: ",df1['product_name'].value_counts()
elif af == 6:
    gkk = df2.groupby(['stars','product_name'])
    print(gkk.first())
elif af ==7:
    df1["Order Date and Time Stamp"] = pd.to_datetime(df1["Order Date and Time
    df1["Year"] = df1["Order Date and Time Stamp"].dt.year
    df1["Month"] = df1["Order Date and Time Stamp"].dt.month_name()

    # group the orders by year and month and count the number of orders in eac
    orders_per_month_per_year = df1.groupby(["Year", "Month"])["Order #"].coun

    # plot the results using a line chart or bar chart
    orders_per_month_per_year.plot(kind="line", marker="o")
    plt.xlabel("Month")
    plt.ylabel("Number of Orders")
    plt.title("Number of Orders Per Month Per Year")
    plt.show()
elif af == 8:
    df1["Order Date and Time Stamp"] = pd.to_datetime(df1["Order Date and Time
    df1["Hour"] = df1["Order Date and Time Stamp"].dt.hour

    # group the orders by hour and count the number of orders in each hour gro
    orders_by_hour = df1.groupby("Hour")["Order #"].count()

    # plot the results using a bar chart or histogram
    orders_by_hour.plot(kind="bar")
    plt.xlabel("Hour of the Day")
    plt.ylabel("Number of Orders")
    plt.title("Number of Orders Across Parts of a Day")
    plt.show()
elif af == 9 :
    df1["Order Date and Time Stamp"] = pd.to_datetime(df1["Order Date and Time
    df1["Year"] = df1["Order Date and Time Stamp"].dt.year
    df1["Month"] = df1["Order Date and Time Stamp"].dt.month

    # group the orders by year and month and count the number of orders in eac
    orders_by_hour = df1.groupby(["Year", "Month"])["stars"].mean()

    # plot the results using a bar chart or histogram
    orders_by_hour.plot(kind="bar")
    plt.xlabel("Month")
    plt.ylabel("reviews")
    plt.title("analysis of Reviews for Number of Orders Per Month Per Year")
    plt.show()
elif af == 10:
    print("FULL REPORT ==")
    gkk = df2.groupby(['stars','product_name'])
    print(gkk.first())
    count1 = 0
    count2 = 0
    list1 = df1['Payment Method'].values.tolist()
    cleanedList = [x for x in list1 if str(x) != 'nan']
    for i in cleanedList:
        list2 = i.split()
        if "CCAvenue" in list2:
            count1 += 1
        if "Offline" in list2:
            count2 += 1
    print("CCAvenue = ",count1)
    print("Offline payment = ",count2)
    print("Top consumer states in India :\n",df1['Shipping State'].value_count
    print("Top consumer cities In India: \n",df1['Shipping City'].value_counts
    print("Top selling products In India: ",df1['product_name'].value_counts()
    print("analysis of Reviews given by Customers")
    gkk = df2.groupby(['stars','product_name'])
    print(gkk.first())
    df1["Order Date and Time Stamp"] = pd.to_datetime(df1["Order Date and Time
    df1["Year"] = df1["Order Date and Time Stamp"].dt.year
    df1["Month"] = df1["Order Date and Time Stamp"].dt.month_name()

    # group the orders by year and month and count the number of orders in eac
    orders_per_month_per_year = df1.groupby(["Year", "Month"])["Order #"].coun

    # plot the results using a line chart or bar chart
    orders_per_month_per_year.plot(kind="line", marker="o")
    plt.xlabel("Month")
    plt.ylabel("Number of Orders")
    plt.title("Number of Orders Per Month Per Year")
    plt.show()
    df1["Order Date and Time Stamp"] = pd.to_datetime(df1["Order Date and Time
    df1["Hour"] = df1["Order Date and Time Stamp"].dt.hour

    # group the orders by hour and count the number of orders in each hour gro
    orders_by_hour = df1.groupby("Hour")["Order #"].count()
    # plot the results using a bar chart or histogram
    orders_by_hour.plot(kind="bar")
    plt.xlabel("Hour of the Day")
    plt.ylabel("Number of Orders")
    plt.title("Number of Orders Across Parts of a Day")
    plt.show()
    df1["Order Date and Time Stamp"] = pd.to_datetime(df1["Order Date and Time
    df1["Year"] = df1["Order Date and Time Stamp"].dt.year
    df1["Month"] = df1["Order Date and Time Stamp"].dt.month

    # group the orders by year and month and count the number of orders in eac
    orders_by_hour = df1.groupby(["Year", "Month"])["stars"].mean()

    # plot the results using a bar chart or histogram
    orders_by_hour.plot(kind="bar")
    plt.xlabel("Month")
    plt.ylabel("reviews")
    plt.title("analysis of Reviews for Number of Orders Per Month Per Year")
    plt.show()
```

```
Enter the number to see the analysis of your choice:
7
```



In [ ]: