Customer Engagement & Offer Effectiveness Analysis

Problem Statement

- The Cafe Rewards program runs marketing campaigns that send promotional offers to customers over a 30-day period.
- Offers can be of three types: informational, discount, or buy one, get one (BOGO).
- Customers receive different mixes of offers and have a limited time to redeem them.
- Customer engagement is recorded through events — offers received, viewed, completed — and purchases (transactions).
- The company wants to understand which offers and customer segments generate the highest engagement and sales to improve targeting in future campaigns.

Project Objectives

- Determine which offer types (BOGO, discount, informational) are most effective in driving completions.
- how many informational offers were followed by transactions.
- Measure the impact of offers on purchase behavior.
- Identify patterns in offer completions based on demographic factors.
- Evaluate which communication channels (email, web, mobile, social) lead to higher completion rates.
- Measure average time taken from offer receipt to completion.
- Provide data-driven recommendations to improve future marketing campaigns.

Importing File & Required Liabraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
Customers=pd.read_csv(r'C:\Users\Aarav Computer\Downloads\Caffe Rewards\customers.csv')
```

In [3]:
```
Offers=pd.read_csv(r'C:\Users\Aarav Computer\Downloads\Caffe Rewards\offers.csv')
```

Exploratory Data Analysis

Checking Data Types,Missing Values

In [4]:
```
Customers.head(10)
```

Out[4]:

	customer_id	became_member_on	gender	age	income
0	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN	118	NaN
1	0610b486422d4921ae7d2bf64640c50b	20170715	F	55	112000.0
2	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN	118	NaN
3	78afa995795e4d85b5d9ceeca43f5fef	20170509	F	75	100000.0
4	a03223e636434f42ac4c3df47e8bac43	20170804	NaN	118	NaN
5	e2127556f4f64592b11af22de27a7932	20180426	M	68	70000.0
6	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN	118	NaN
7	68617ca6246f4fbc85e91a2a49552598	20171002	NaN	118	NaN
8	389bc3fa690240e798340f5a15918d5c	20180209	M	65	53000.0
9	8974fc5686fe429db53ddde067b88302	20161122	NaN	118	NaN

In [6]:
```
Customers.dtypes
```

```
Out[6]: customer_id          object
        became_member_on      int64
        gender               object
        age                   int64
        income              float64
        dtype: object
```

```
In [7]: Customers.isnull().sum()
```

```
Out[7]: customer_id            0
        became_member_on       0
        gender              2175
        age                    0
        income              2175
        dtype: int64
```

Adding columns, Changing data types ,Renaming columns,Filling Missing/Null values

```
In [5]: def Age_Category(age):
                if age <= 20:
                    return '0-20'
                elif age <= 40:
                     return '21-40'
                elif age <= 60:
                    return '41-60'
                else:
                    return 'Greater than 60'

        Customers['Age_Category']=Customers['age'].apply(Age_Category)
```

```
In [141…  def Income_Category(income):
                  if income <= 40000:
                      return '0-40000'
                  elif income <= 80000:
                       return '41000-80000'
                  elif income <= 100000:
                      return '81000-100000'
                  else:
                      return 'higher than 100000'

          Customers['Income_Category']=Customers['income'].apply(Income_Category)
```

```
In [6]: Customers.rename(columns={'became_member_on':'Membership'},inplace = True)
```

```
In [7]: Customers['income'].isna().mean()*100
```

```
Out[7]: np.float64(12.794117647058822)
```

```
In [8]: Customers['income'].interpolate(method='linear',inplace=True)
```

```
In [9]: Customers['gender']=Customers['gender'].fillna('unknown')
```

```
In [10]: Customers['Membership']=pd.to_datetime(Customers['Membership'],format='%Y%m%d')
```

```
In [142…  Customers.head(5)
```

Out[142…

	customer_id	Membership	gender	age	income	Age_Category	Year	Income_Category
0	68be06ca386d4c31939f3a4f0e3dd783	2017-02-12	unknown	118	NaN	Greater than 60	2017	higher than 100000
1	0610b486422d4921ae7d2bf64640c50b	2017-07-15	F	55	112000.0	41-60	2017	higher than 100000
2	38fe809add3b4fcf9315a9694bb96ff5	2018-07-12	unknown	118	106000.0	Greater than 60	2018	higher than 100000
3	78afa995795e4d85b5d9ceeca43f5fef	2017-05-09	F	75	100000.0	Greater than 60	2017	81000-100000
4	a03223e636434f42ac4c3df47e8bac43	2017-08-04	unknown	118	85000.0	Greater than 60	2017	81000-100000

```
In [12]: # Adding Columns
         import datetime as dt
         Customers['Year']=Customers['Membership'].dt.year
```

```
In [13]: Events=pd.read_csv(r'C:\Users\Aarav Computer\Downloads\Caffe Rewards\events.csv')
```

```
In [53]: Events.head(10)
```

Out[53]:

	customer_id	event	offer_id	time	day
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1
1	a03223e636434f42ac4c3df47e8bac43	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1
2	e2127556f4f64592b11af22de27a7932	offer received	2906b810c7d4411798c6938adc9daaa5	0	1
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	fafdcd668e3743c1bb461111dcafc2a4	0	1
4	68617ca6246f4fbc85e91a2a49552598	offer received	4d5c57ea9a6940dd891ad53e9dbe8da0	0	1
5	389bc3fa690240e798340f5a15918d5c	offer received	f19421c1d4aa40978ebb69ca19b0e20d	0	1
6	c4863c7985cf408faee930f111475da3	offer received	2298d6c36e964ae4a3e7e9706d1fb8c2	0	1
7	2eeac8d8feae4a8cad5a6af0499a211d	offer received	3f207df678b143eea3cee63160fa8bed	0	1
8	aa4862eba776480b8bb9c68455b8c2e1	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1
9	31dda685af34476cad5bc968bdb01c53	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1

In [37]:
```
Events['day']=(Events['time']//24)+1
```

In [19]:
```
Events.dtypes
```

Out[19]:
```
customer_id    object
event          object
value          object
time            int64
day             int64
dtype: object
```

In [20]:
```
Events.isnull().sum()
```

Out[20]:
```
customer_id    0
event          0
value          0
time           0
day            0
dtype: int64
```

Data cleaning

In [15]:
```
Events.rename(columns={'value':'offer_id'},inplace=True)
```

In [16]:
```
Events['offer_id']=Events['offer_id'].str.replace(r"[{'offer_id':']","",regex=True)
```

In [17]:
```
Events['offer_id']=Events['offer_id'].str.replace(r"[{'offer_id':']","",regex=True)
```

In [18]:
```
Events['offer_id']=Events['offer_id'].str.replace(r"[amunt]","",regex=True)
```

In [19]:
```
Events['offer_id']=Events['offer_id'].str.replace(r"[}]","",regex=True)
```

In [20]:
```
Events['offer_id']=Events['offer_id'].replace({"9b98b8c733c4b65b9b679969":"9b98b8c7a33c4b65b9aebfe6a799e6d9",
                                                "0b115392cc45b7b97c272217":"0b1e1539f2cc45b7b9fa7c272da2e1d7",
                                                "2906b810c74411798c6938c95":"2906b810c7d4411798c6938adc9daaa5",
                                                "c6683743c1bb461111cc24":"fafdcd668e3743c1bb461111dcafc2a4",
                                                "3207678b1433c631608b":"3f207df678b143eea3cee63160fa8bed",
                                                "45c5796940891539b80":"4d5c57ea9a6940dd891ad53e9dbe8da0",
                                                "22986c3696443797061b8c2":"2298d6c36e964ae4a3e7e9706d1fb8c2",
                                                "19421c1440978bb69c19b020":"f19421c1d4aa40978ebb69ca19b0e20d",
                                                "26436372046b9bb56bc8210":"ae264e3637204a6fb9bb56bc8210ddfd",
                                                "58bc65990b2455138643c4b9837":"5a8bc65990b245e5a138643cd4eb9837"},regex=Tru
```

In [21]:
```
Events.head(10)
```

Out[21]:

	customer_id	event	offer_id	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	9b98b8c7a33c4b65b9aebfe6a799e6d9	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0
2	e2127556f4f64592b11af22de27a7932	offer received	2906b810c7d4411798c6938adc9daaa5	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	fafdcd668e3743c1bb461111dcafc2a4	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	4d5c57ea9a6940dd891ad53e9dbe8da0	0
5	389bc3fa690240e798340f5a15918d5c	offer received	f19421c1d4aa40978ebb69ca19b0e20d	0
6	c4863c7985cf408faee930f111475da3	offer received	2298d6c36e964ae4a3e7e9706d1fb8c2	0
7	2eeac8d8feae4a8cad5a6af0499a211d	offer received	3f207df678b143eea3cee63160fa8bed	0
8	aa4862eba776480b8bb9c68455b8c2e1	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0
9	31dda685af34476cad5bc968bdb01c53	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0

In [40]:

```
Offers
```

Out[40]:

	offer_id	offer_type	difficulty	reward	duration	channels
0	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10	10	7	['email', 'mobile', 'social']
1	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10	10	5	['web', 'email', 'mobile', 'social']
2	3f207df678b143eea3cee63160fa8bed	informational	0	0	4	['web', 'email', 'mobile']
3	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5	5	7	['web', 'email', 'mobile']
4	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	20	5	10	['web', 'email']
5	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	7	3	7	['web', 'email', 'mobile', 'social']
6	fafdcd668e3743c1bb461111dcafc2a4	discount	10	2	10	['web', 'email', 'mobile', 'social']
7	5a8bc65990b245e5a138643cd4eb9837	informational	0	0	3	['email', 'mobile', 'social']
8	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5	5	5	['web', 'email', 'mobile', 'social']
9	2906b810c7d4411798c6938adc9daaa5	discount	10	2	7	['web', 'email', 'mobile']

Correcting Values

In [57]:

```
Offerscompleted=Events[Events['event']=='offer completed']
Offerscompleted
```

Out[57]:

	customer_id	event	offer_id	time	day
12658	9fa9ae8f57894cc9a3b8a9bbe0fc1b2f	offer completed	2906b810c7d4411798c6938adc9daaa5, w 2	0	1
12672	fe97aa22dd3e48c8b143116a8403dd52	offer completed	fafdcd668e3743c1bb461111dcafc2a4, w 2	0	1
12679	629fc02d56414d91bca360decdfa9288	offer completed	9b98b8c7a33c4b65b9aebfe6a799e6d9, w 5	0	1
12692	676506bad68e4161b9bbaffeb039626b	offer completed	ae264e3637204a6fb9bb56bc8210ddfd, w 10	0	1
12697	8f7dd3b2afe14c078eb4f6e6fe4ba97d	offer completed	4d5c57ea9a6940dd891ad53e9dbe8da0, w 10	0	1
...	...	...	...	...	...
306475	0c027f5f34dd4b9eba0a25785c611273	offer completed	2298d6c36e964ae4a3e7e9706d1fb8c2, w 3	714	30
306497	a6f84f4e976f44508c358cc9aba6d2b3	offer completed	2298d6c36e964ae4a3e7e9706d1fb8c2, w 3	714	30
306506	b895c57e8cd047a8872ce02aa54759d6	offer completed	fafdcd668e3743c1bb461111dcafc2a4, w 2	714	30
306509	8431c16f8e1d440880db371a68f82dd0	offer completed	fafdcd668e3743c1bb461111dcafc2a4, w 2	714	30
306527	24f56b5e1849462093931b164eb803b5	offer completed	fafdcd668e3743c1bb461111dcafc2a4, w 2	714	30

33579 rows × 5 columns

In [23]:

```
Offers['offer_id'].unique()
```

Out[23]:

```
array(['ae264e3637204a6fb9bb56bc8210ddfd',
       '4d5c57ea9a6940dd891ad53e9dbe8da0',
       '3f207df678b143eea3cee63160fa8bed',
       '9b98b8c7a33c4b65b9aebfe6a799e6d9',
       '0b1e1539f2cc45b7b9fa7c272da2e1d7',
       '2298d6c36e964ae4a3e7e9706d1fb8c2',
       'fafdcd668e3743c1bb461111dcafc2a4',
       '5a8bc65990b245e5a138643cd4eb9837',
       'f19421c1d4aa40978ebb69ca19b0e20d',
       '2906b810c7d4411798c6938adc9daaa5'], dtype=object)
```

In [24]:
```
Offerscompleted['offer_id'].unique()
```

Out[24]:
```
array([' 2906b810c7d4411798c6938adc9daaa5, w 2',
       ' fafdcd668e3743c1bb461111dcafc2a4, w 2',
       ' 9b98b8c7a33c4b65b9aebfe6a799e6d9, w 5',
       ' ae264e3637204a6fb9bb56bc8210ddfd, w 10',
       ' 4d5c57ea9a6940dd891ad53e9dbe8da0, w 10',
       ' 2298d6c36e964ae4a3e7e9706d1fb8c2, w 3',
       ' f19421c1d4aa40978ebb69ca19b0e20d, w 5',
       ' 0b1e1539f2cc45b7b9fa7c272da2e1d7, w 5'], dtype=object)
```

In [59]:
```
Offerscompleted['offer_id']=Offerscompleted['offer_id'].str.split(',').str[0]
```

In [60]:
```
Offerscompleted['offer_id']=Offerscompleted['offer_id'].str.strip()
```

In [61]:
```
Offerscompleted
```

Out[61]:

	customer_id	event	offer_id	time	day
12658	9fa9ae8f57894cc9a3b8a9bbe0fc1b2f	offer completed	2906b810c7d4411798c6938adc9daaa5	0	1
12672	fe97aa22dd3e48c8b143116a8403dd52	offer completed	fafdcd668e3743c1bb461111dcafc2a4	0	1
12679	629fc02d56414d91bca360decdfa9288	offer completed	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1
12692	676506bad68e4161b9bbaffeb039626b	offer completed	ae264e3637204a6fb9bb56bc8210ddfd	0	1
12697	8f7dd3b2afe14c078eb4f6e6fe4ba97d	offer completed	4d5c57ea9a6940dd891ad53e9dbe8da0	0	1
...	...	...	...	...	...
306475	0c027f5f34dd4b9eba0a25785c611273	offer completed	2298d6c36e964ae4a3e7e9706d1fb8c2	714	30
306497	a6f84f4e976f44508c358cc9aba6d2b3	offer completed	2298d6c36e964ae4a3e7e9706d1fb8c2	714	30
306506	b895c57e8cd047a8872ce02aa54759d6	offer completed	fafdcd668e3743c1bb461111dcafc2a4	714	30
306509	8431c16f8e1d440880db371a68f82dd0	offer completed	fafdcd668e3743c1bb461111dcafc2a4	714	30
306527	24f56b5e1849462093931b164eb803b5	offer completed	fafdcd668e3743c1bb461111dcafc2a4	714	30

33579 rows × 5 columns

Customer Insights :

Merging Data files

In [54]:
```
Customer_Details=pd.merge(Events,Customers,on='customer_id',how='left')
```

In [55]:
```
Customer_Details.head(5)
```

Out[55]:

	customer_id	event	offer_id	time	day	Membership	gender	age	incom
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1	2017-05-09	F	75	100000.00000
1	a03223e636434f42ac4c3df47e8bac43	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1	2017-08-04	unknown	118	85000.00000
2	e2127556f4f64592b11af22de27a7932	offer received	2906b810c7d4411798c6938adc9daaa5	0	1	2018-04-26	M	68	70000.00000
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	fafdcd668e3743c1bb461111dcafc2a4	0	1	2017-09-25	unknown	118	64333.33333
4	68617ca6246f4fbc85e91a2a49552598	offer received	4d5c57ea9a6940dd891ad53e9dbe8da0	0	1	2017-10-02	unknown	118	58666.66666

1.What is the total Customers Count?

In [69]:
```
len(Customer_Details)
```

Out[69]:
```
306534
```

2. What is the Average Age of the Customers ?

In [115…
```
CustomerAge=round(Customer_Details['age'].mean())
print(f"\033[1m----------\n{CustomerAge}")
```

```
----------
61
```

3. What is the Average Income of the Customers?

In [116…

```
CustomersAverageIncome=round(Customer_Details['income'].mean())
print(f"\033[1m----------------\n{CustomersAverageIncome}")
```

```
----------------
64489
```

- The **64489** means the customers buying the membership are the upper middle class family having the good income source.
- and the **61** shows generally the Senior age customers are likely to buy the membership .
- the cafe can do more marketing amongst this age category people.

4.Offers Completed Customer count?

In [70]:

```
Offerscompleted=Customer_Details[Customer_Details['event']=='offer completed']
len(Offerscompleted)
```

Out[70]: 33579

5. customers count by the Genders?

In [119…

```
GenderWiseCustomers= Customer_Details.groupby('gender')['customer_id'].count().reset_index()
GenderWiseCustomers
```

Out[119…

	gender	customer_id
0	F	113101
1	M	155690
2	O	3971
3	unknown	33772

6. Offers Completed customers Count by the Gender?

In [122…

```
Offerscompleted=Customer_Details[Customer_Details['event']=='offer completed']
OfferscompletedCustomers=Offerscompleted.groupby('gender')['customer_id'].count().reset_index()
OfferscompletedCustomers
```

Out[122…

	gender	customer_id
0	F	15477
1	M	16466
2	O	501
3	unknown	1135

7. Total Unique Customers available?

In [120…

```
Totalcustomers=Customer_Details['customer_id'].nunique()
print(f"\033[1mTotal Unique Customers----------------\n{Totalcustomers}")
```

```
Total Unique Customers----------------
17000
```

8. Unique Offers Available?

In [30]:

```
TotalOffers=Offers['offer_id'].unique()
TotalOffers.tolist()
```

Out[30]:

```
['ae264e3637204a6fb9bb56bc8210ddfd',
 '4d5c57ea9a6940dd891ad53e9dbe8da0',
 '3f207df678b143eea3cee63160fa8bed',
 '9b98b8c7a33c4b65b9aebfe6a799e6d9',
 '0b1e1539f2cc45b7b9fa7c272da2e1d7',
 '2298d6c36e964ae4a3e7e9706d1fb8c2',
 'fafdcd668e3743c1bb461111dcafc2a4',
 '5a8bc65990b245e5a138643cd4eb9837',
 'f19421c1d4aa40978ebb69ca19b0e20d',
 '2906b810c7d4411798c6938adc9daaa5']
```

9.Total Offers Completed?

In [31]:
```
Offerscompleted=Customer_Details[Customer_Details['event']=='offer completed']
len(Offerscompleted)
```

Out[31]: 33579

10. Total Events happened?

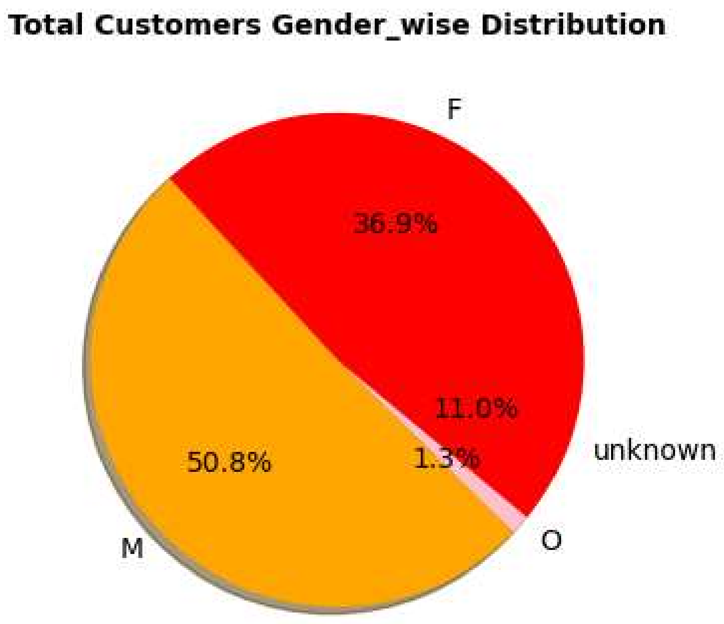
In [42]:
```
len(Events)
```

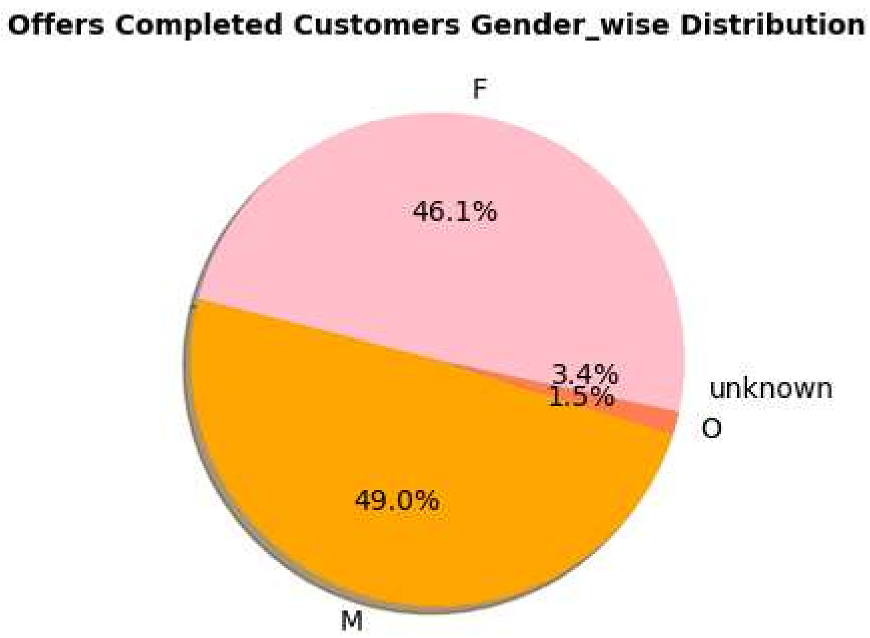
Out[42]: 306534

1.Gender_Wise Overall Customers VS Offer Completed Customers Distribution :

In [126…
```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
col=['red','orange','pink']
Gender=['Female','Male','Other']
plt.pie(GenderWiseCustomers['customer_id'],autopct='%1.1f%%',colors=col,labels=GenderWiseCustomers['gender'],shadow=True)
plt.title('Total Customers Gender_wise Distribution',fontweight='bold',fontsize=10)

plt.subplot(1,2,2)
col=['pink','orange','coral']
Gender=['Female','Male','Other']
plt.pie(OfferscompletedCustomers['customer_id'],autopct='%1.1f%%',colors=col,labels=OfferscompletedCustomers['gender'],shadow=
plt.title('Offers Completed Customers Gender_wise Distribution',fontweight='bold',fontsize=10)
plt.show()
```





- The business has received the male customers more.
- both in terms of event and offers completed that means where the other events refers to offer receive,offer viewed,and transactions happened the male customers count is more than the female count.
- and even in terms of completing the offers also,**male** has contributed the **49%** where as **female** customes have contributed to the **46.1%.**
- Other category is contributing generally low.
- **Unknown category** showing the data of customes where **1.3%** customers are unknown because gender information is not provided of this customers and it can lead to generate wrong customer insights.
- The business employees are responsible to take the proper information of the customers for the future marketing and promotional planning in the different customer segments.

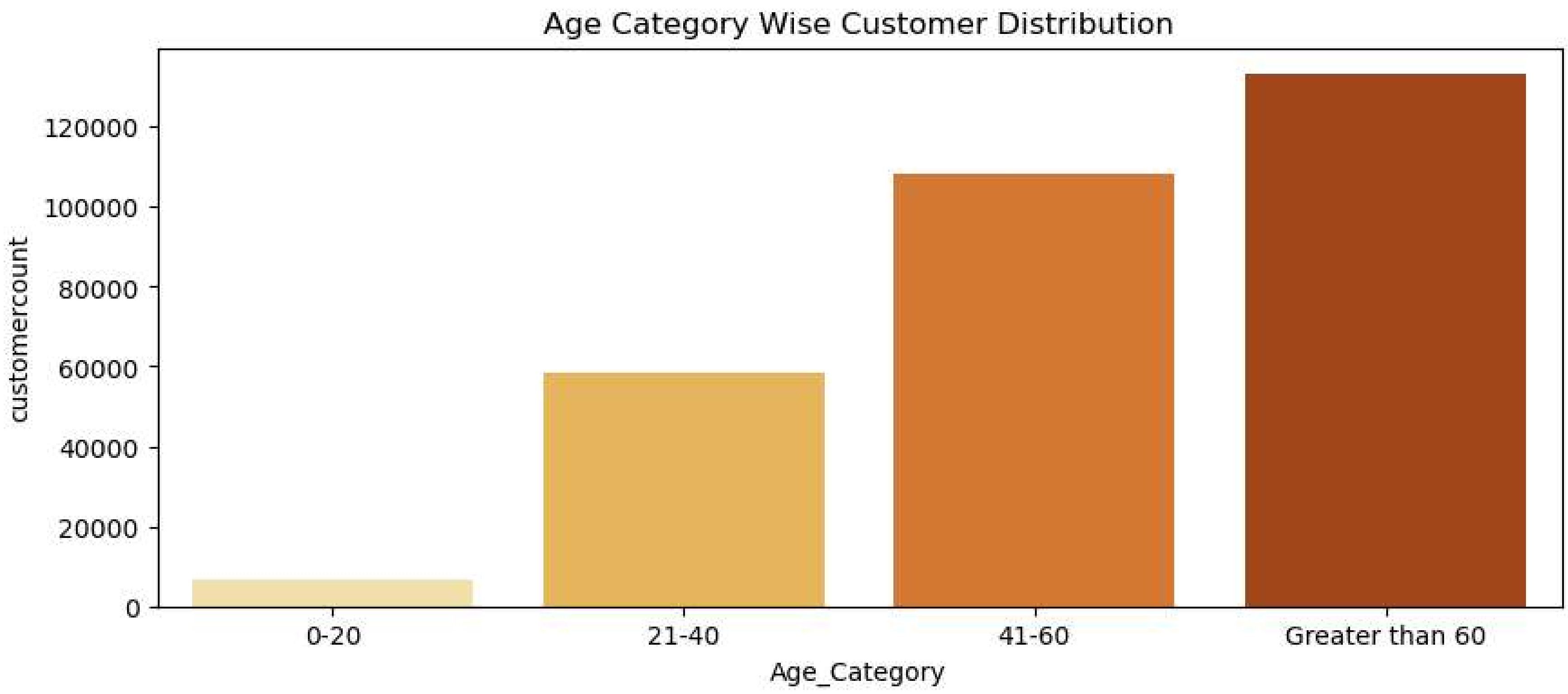
In [32]:
```
Age_CategoryCustomers=Customer_Details.groupby('Age_Category')['customer_id'].count().reset_index().rename(columns={'customer_
Data=pd.DataFrame(Age_CategoryCustomers)
Data
```

Out[32]:

	Age_Category	customercount
0	0-20	6544
1	21-40	58569
2	41-60	108318
3	Greater than 60	133103

2.Age Wise Customers Distribution:

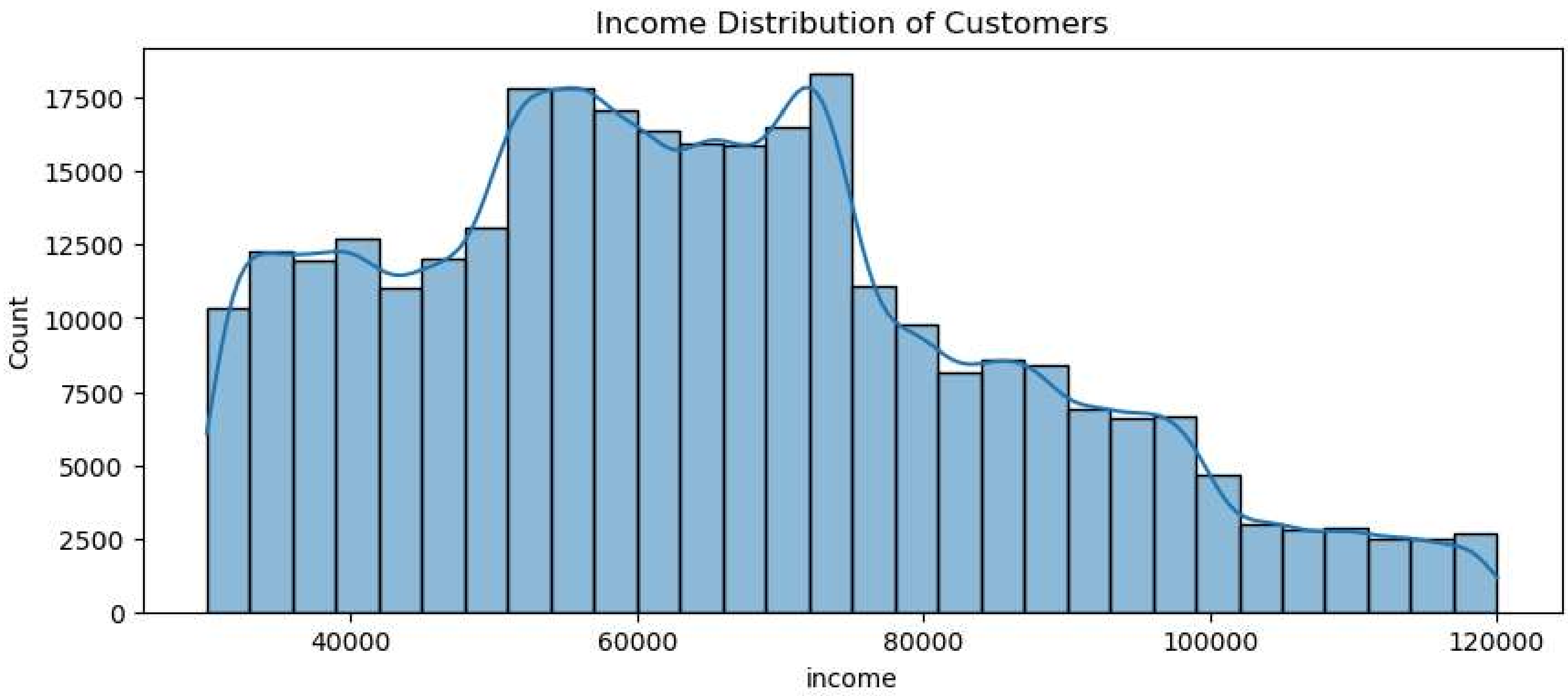
In [128…
```
plt.figure(figsize=(10,4))
sns.barplot(data=Data,x='Age_Category',y='customercount',palette='YlOrBr')
plt.title('Age Category Wise Customer Distribution')
plt.show()
```



- Customers **below 20 years (kids)** form the smallest group, with fewer than **15,000 customers**.
- The largest customer group belongs to the 60+ age category (old age customers).
- A significant variation is observed in customer counts across the range of **20,000 to 120,000**, showing notable differences between age groups.
- The **41–60** age group contributes a substantial customer base of more than **100,000 customers**.
- The **21–40** age group (adults) accounts for only around **60,000** customers, which is half the size of the old age category.
- This wide variation in age-based customer distribution highlights the need for stronger marketing strategies to improve awareness and engagement with products and offers across all age categories.

In [129…
```
Income=Customer_Details['income'].reset_index()
```

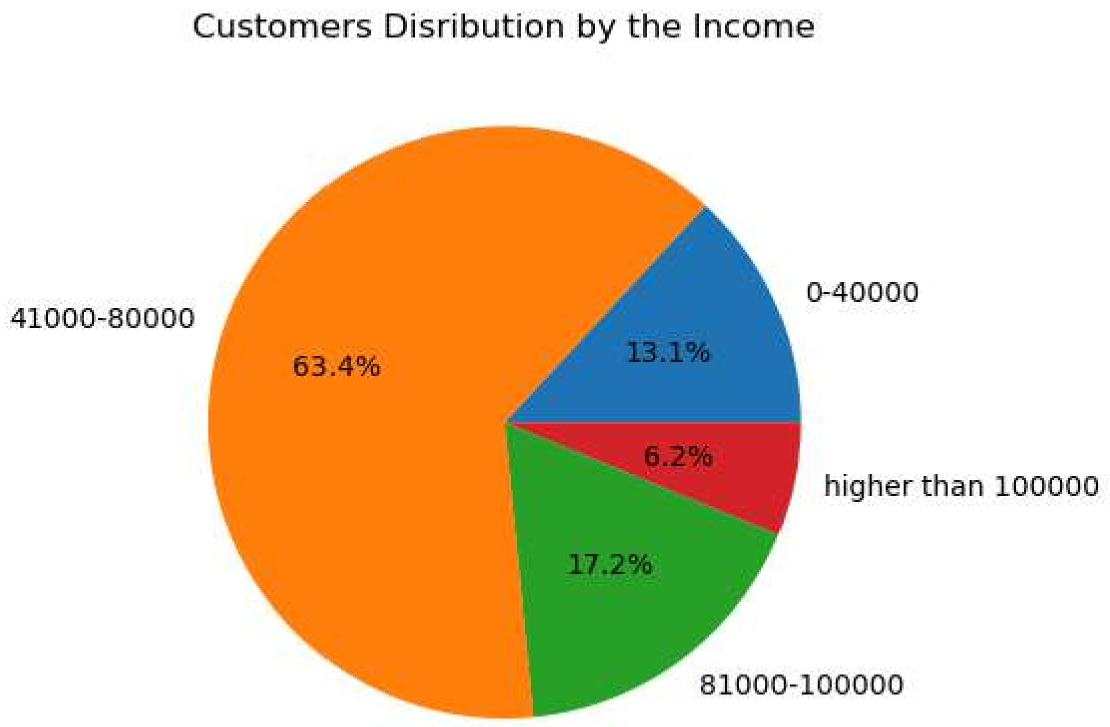
In [71]:
```
plt.figure(figsize=(10,4))
sns.histplot(data=Customer_Details,x='income',bins=30,kde=True)
plt.title('Income Distribution of Customers')
plt.show()
```



- The data shows that the **40 to 50%** of the customers falls under the income range of **(USD 40000 to 80000)** as the average income is also **64489**
- and can say that the possibility of other **25%** of the customers distribution falls under the income range of **(80000 to 100000 USD)** or less than the **40000**.
- as the huge number of **12500** customers are falling under the income range of less than **40000**.
- There is also some of the customers we received having the income more than **100000** shows the customer count less than the **50000** ,that is not even the **(25% /one 3rd part)** of highest income range customers count **17500**.

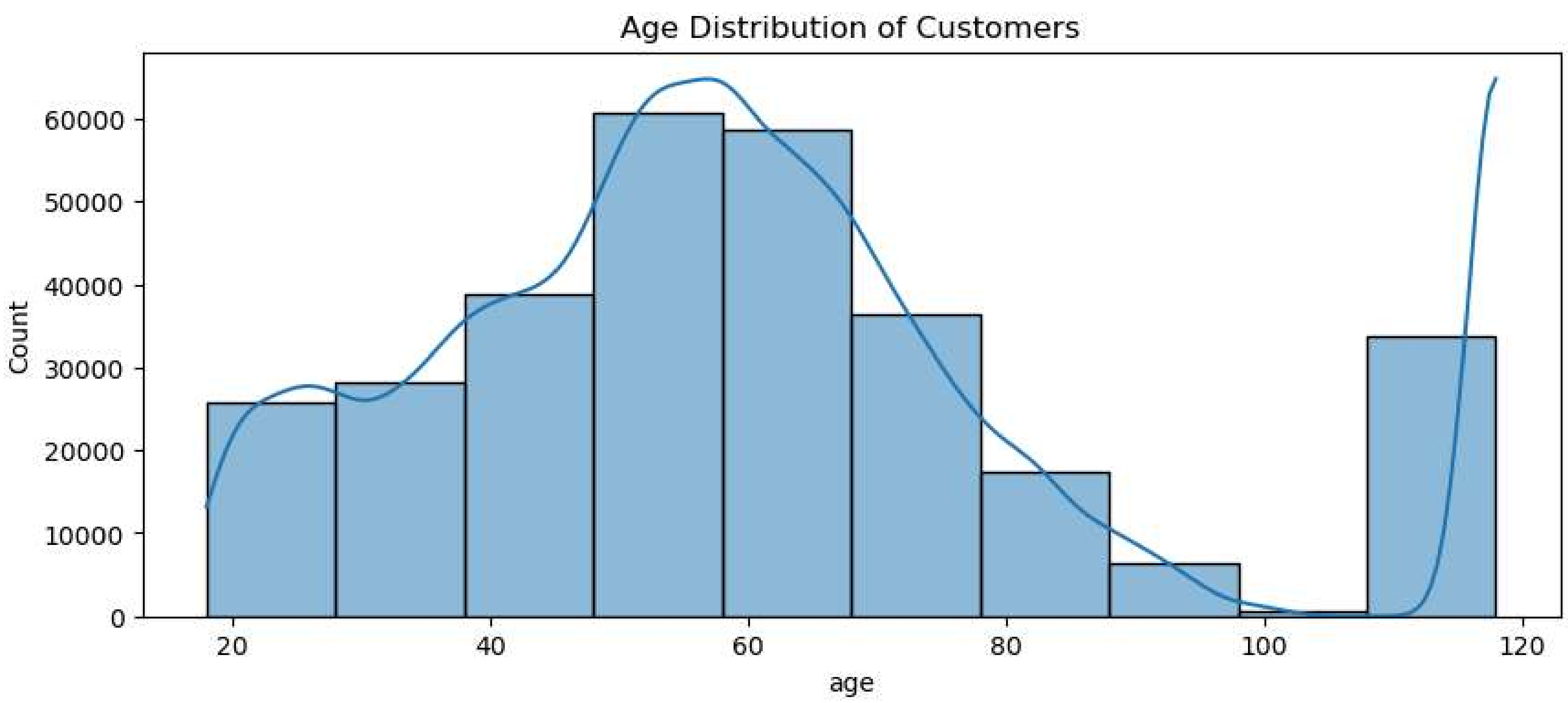
In [149…

```
IncomeDistribute=Customers.groupby('Income_Category')['customer_id'].count().reset_index()
plt.pie(IncomeDistribute['customer_id'],labels=IncomeDistribute['Income_Category'],autopct='%1.1f%%')
plt.title('Customers Disribution by the Income')
plt.show()
```



In [43]:

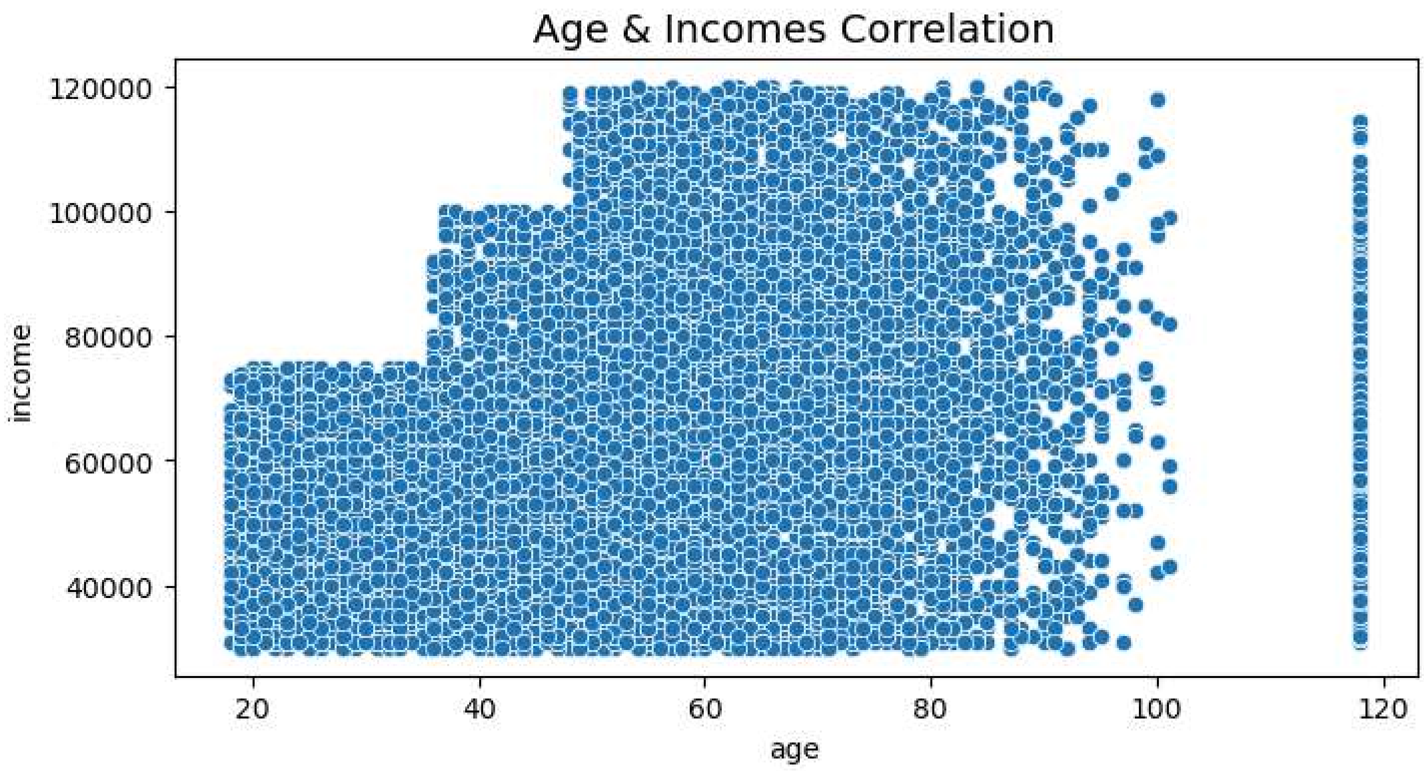
```
plt.figure(figsize=(10,4))
sns.histplot(data=Customer_Details,x='age',bins=10,kde=True)
plt.title('Age Distribution of Customers')
plt.show()
```



- As we have calculated the age, **average age is 61** and the kde line simply shows that the mean,means the average age is between range of **60 to 65** .
- This customers are the real assets of the business.
- and the business should make good engagement with this customer section by providing the best customers services as this are regular customers and the backbone of the business.

In [132…

```
AgeandIncome=Customer_Details[['age','income']]
plt.figure(figsize=(8,4))
sns.scatterplot(data=AgeandIncome,x='age',y='income')
plt.title('Age & Incomes Correlation',fontsize=14)
plt.show()
```



In [33]:
```
AgeandIncome=Customer_Details[['age','income']].corr()
AgeandIncome
```

Out[33]:

	age	income
age	1.000000	0.199931
income	0.199931	1.000000

In [134…
```
print('Correlation coefficient\n------------')
print('AgeIncomeCorr:',round(AgeandIncome.values[0,1],2))
```

```
Correlation coefficient
------------
AgeIncomeCorr: 0.2
```

- The **0.2 is showing a positive correlation but the correlation is is very week**.
- That means as age increases ,income tends to increase slightly,but the relationship is not strong.
- That means there is no as such connection is there between the earning of the customers with the age,so if the business focusing on high paying customers the business should check the income of the person individually age priority will not be the good option in that case.

3. Is there a difference between offers received VS completed distribution by Age_Category ?

In [84]:
```
Offersreceived=Customer_Details[Customer_Details['event']=='offer received']
OffersreceivedData=Offersreceived.groupby('Age_Category')['offer_id'].count().reset_index().sort_values(by='offer_id',ascendin
OffersreceivedData
```

Out[84]:

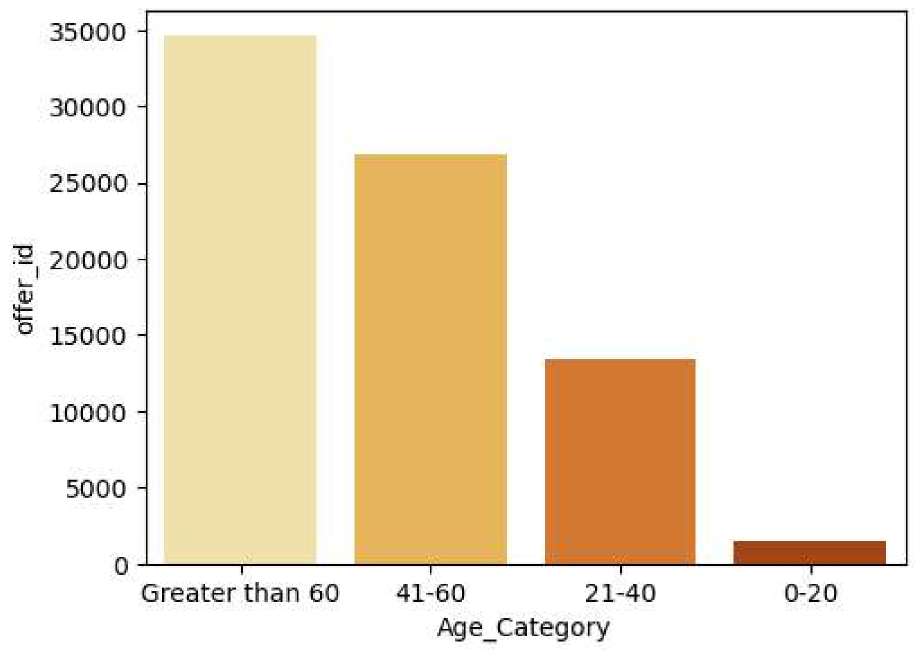
	Age_Category	offer_id
3	Greater than 60	34584
2	41-60	26803
1	21-40	13382
0	0-20	1508

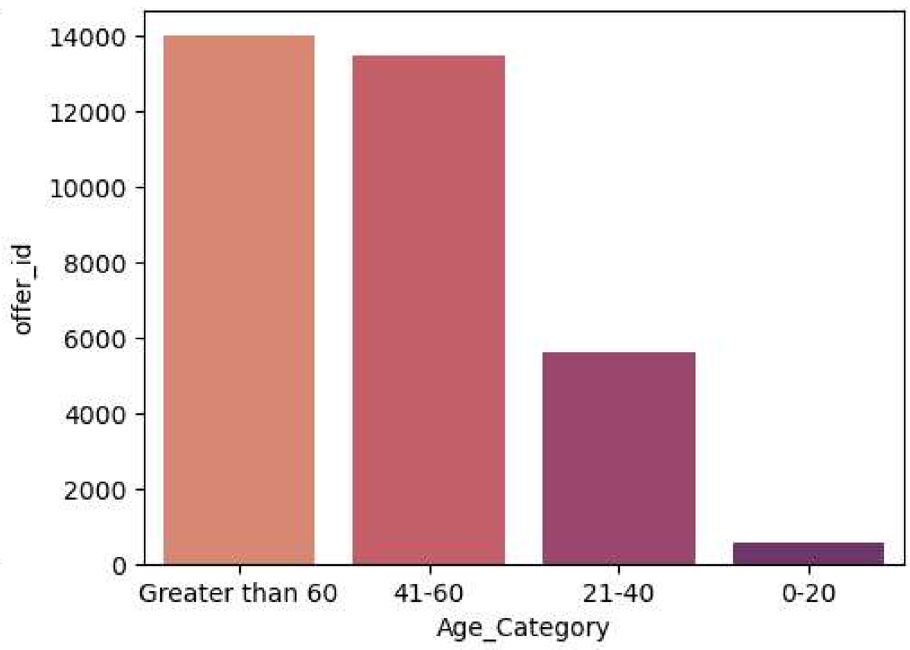
In [83]:
```
Offerscompleted=Customer_Details[Customer_Details['event']=='offer completed']
OfferscompletedData=Offerscompleted.groupby('Age_Category')['offer_id'].count().reset_index().sort_values(by='offer_id',ascend
OfferscompletedData
```

Out[83]:

	Age_Category	offer_id
3	Greater than 60	13977
2	41-60	13463
1	21-40	5603
0	0-20	536

In [95]:
```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
sns.barplot(data=OffersreceivedData,x='Age_Category',y='offer_id',palette='YlOrBr')
plt.subplot(1,2,2)
sns.barplot(data=OfferscompletedData,x='Age_Category',y='offer_id',palette='flare')
plt.show()
```





- **There is a bias in the offers distribution** .
- Most offers were received by the old age category,which is why that segment completed the majority of offers.
- This could be due to uneven tergeting or limited use of marketing channnels.
- Equal marketing across all age group is needed.

4.Is there a difference between offers received VS completed customers income?

In [144…

```
round(Offersreceived['income'].mean())
```

Out[144… 65407

In [143…

```
round(Offerscompleted['income'].mean())
```

Out[143… 69333

- There is not a huge difference but the slight difference is absolutely there in the incomes of this 2 sections.
- The Income can also be the reason to complete the orders as the person should willing to pay the difficulty amount to complete offers.

In [65]:

```
TransactionsData=Customer_Details[Customer_Details['event']=='transaction'].rename(columns={'offer_id':'amount','time':'hours'
TransactionsData
```

Out[65]:

	customer_id	event	amount	hours	day	Membership	gender	age	income	Age_Categ
12654	02c083884c7d45b39cc68e1314fec56c	transaction	0.8300000000000001	0	1	2016-07-11	F	20	30000.0	
12657	9fa9ae8f57894cc9a3b8a9bbe0fc1b2f	transaction	34.56	0	1	2016-01-17	M	42	96000.0	4
12659	54890f68699049c2a04d415abc25e717	transaction	13.23	0	1	2017-12-28	M	36	56000.0	2
12670	b2f1cd155b864803ad8334cdf13c4bd2	transaction	19.51	0	1	2017-10-16	F	55	94000.0	4
12671	fe97aa22dd3e48c8b143116a8403dd52	transaction	18.97	0	1	2017-12-17	F	39	67000.0	2
...	...	...	...	...	...	...	...	...	...	
306529	b3a1272bc9904337b331bf348c3e8c17	transaction	1.5899999999999999	714	30	2018-01-01	M	66	47000.0	Greater
306530	68213b08d99a4ae1b0dcb72aebd9aa35	transaction	9.53	714	30	2018-04-08	M	52	62000.0	4
306531	a00058cf10334a308c68e7631c529907	transaction	3.61	714	30	2013-09-22	F	63	52000.0	Greater
306532	76ddbd6576844afe811f1a3c0fbb5bec	transaction	3.5300000000000002	714	30	2016-07-09	M	57	40000.0	4
306533	c02b10e8752c4d8e9b73f918558531f7	transaction	4.05	714	30	2015-12-11	unknown	118	83500.0	Greater

138953 rows × 11 columns

In [66]:

```
TransactionsData.dtypes
```

```
Out[66]: customer_id             object
         event                   object
         amount                  object
         hours                    int64
         day                      int64
         Membership      datetime64[ns]
         gender                  object
         age                      int64
         income                 float64
         Age_Category            object
         Year                     int32
         dtype: object
```

```
In [69]: TransactionsData['amount']=pd.to_numeric(TransactionsData['amount'])
```

5.How many Transactions Made?

```
In [70]: len(TransactionsData)
```

```
Out[70]: 138953
```

- Total **138953** transactions has made by customers in this 30 days period,that doesnt refers to any offers.
- that means customers are there who doesn't wait for the any offers and discounts often likely to show craze or love for the coffee.
- we can provide the little discounts and better services to these customers to become a regular customers.

6.How Was the Overall Transactions Performance?

Transactions=round(TransactionsData.agg({'age':'mean','income':'mean','hours':'mean','amount':'mean','day':'mean'})) Data=Transactions.rename({'age':'Averge_Age','income':'Average_Income','hours':'Average_hours', 'amount':'Average_Amountpaid','day':'Avergae_DaysTook'}).reset_index().rename(columns={'index':'category',0:'Averages'}) Data

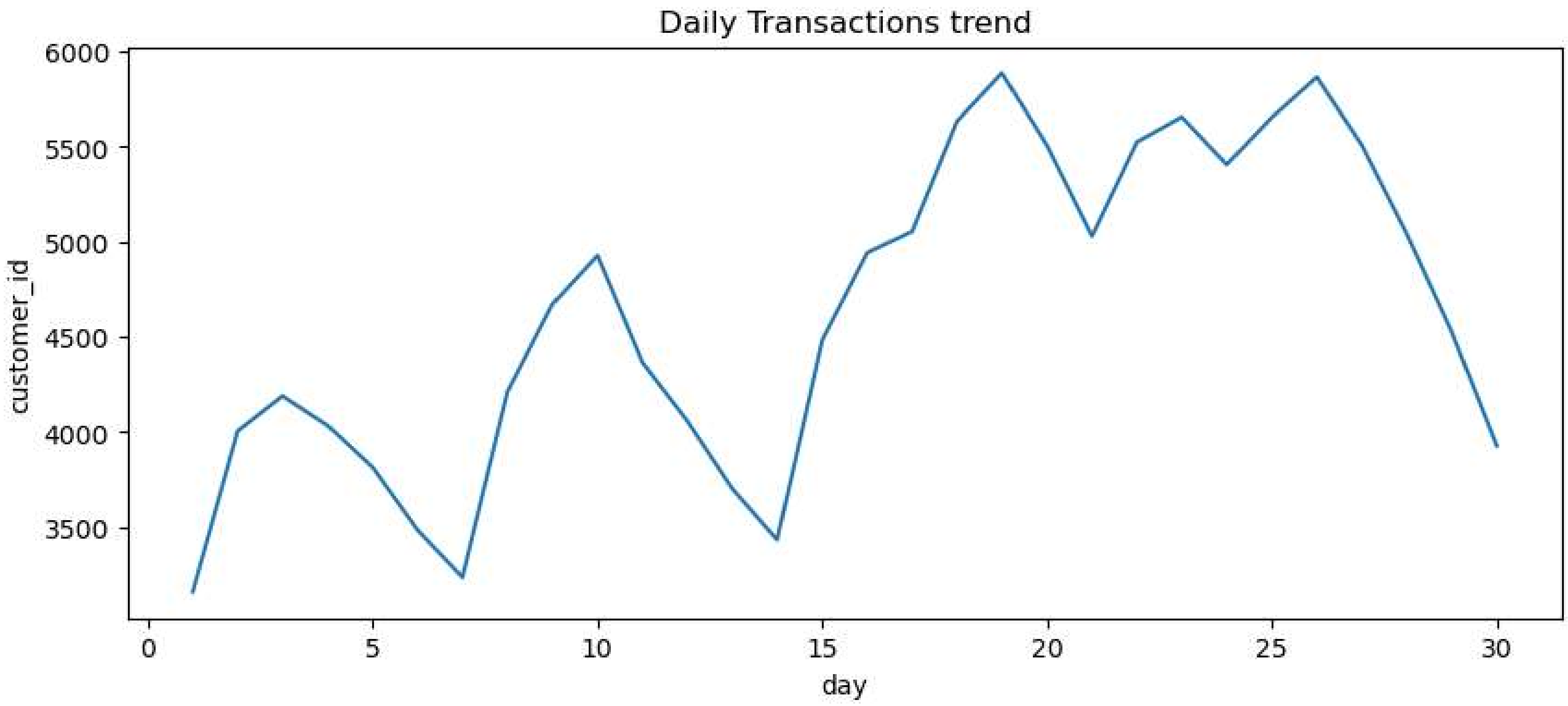
- Average age **60** and the average income **62271**$

- Average Amount paid by the customer is **13** $,shows that the High earning well earned customers are those who are likely to buy without waiting for the offers.

7.Is there a difference between Offers VS regular transactions daily Trend?

Is there a weekly trend?

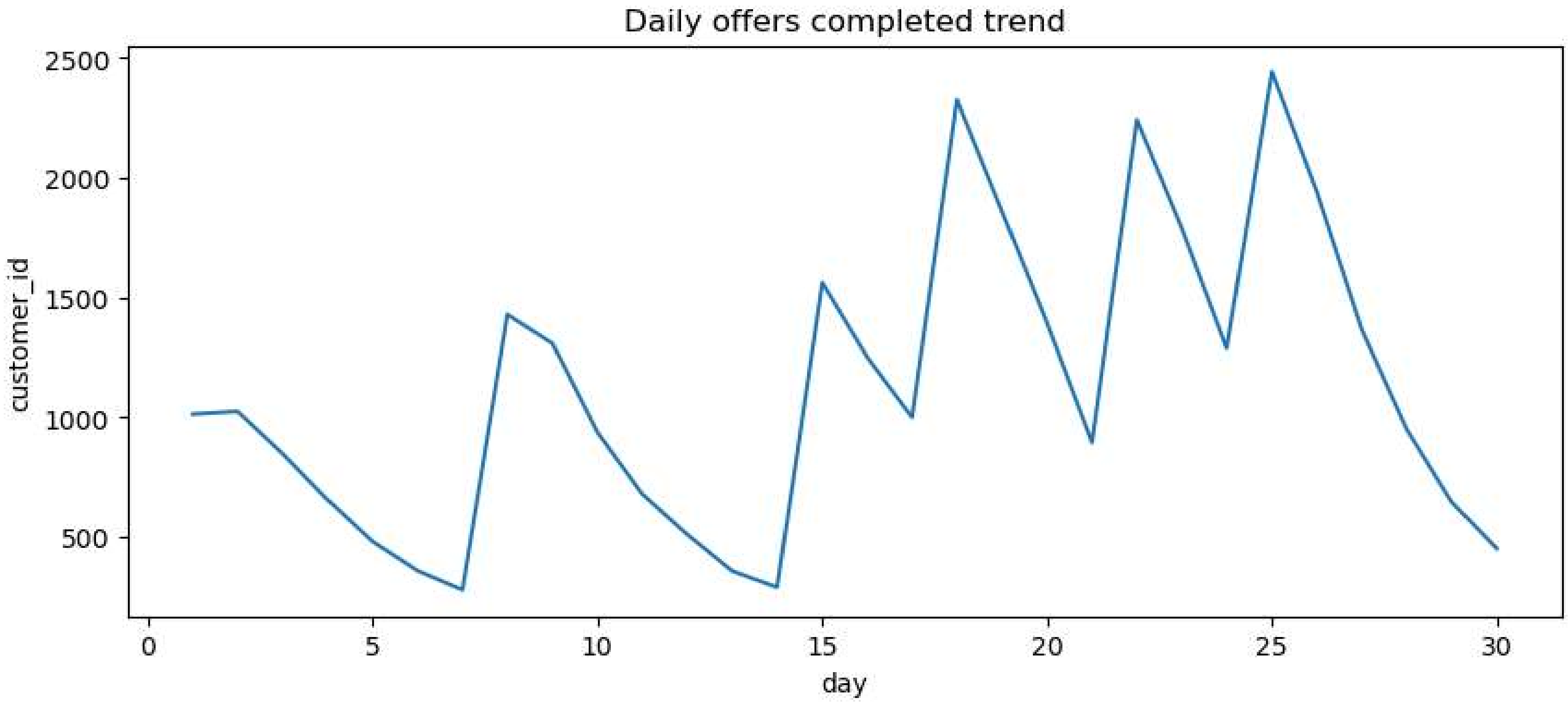
```
In [73]: DailyTransactions=TransactionsData.groupby('day')['customer_id'].count().reset_index()
         plt.figure(figsize=(10,4))
         sns.lineplot(data=DailyTansactions,x='day',y='customer_id',palette='flare')
         plt.title('Daily Transactions trend')
         plt.show()
```



- Daily transaction data is starts from the min range of **3000** around to till the **6000** it lasts.
- That means maximum we can expect to have **6000** tansaction
- and average transactions limit will fall between the range of **4000 to 4500**.

- If on any particular day if the transactions will comes under this range ,even below **3000** the business can be easily track that sales has declined.

In [74]:
```
DailyOffersCompleted=Offerscompleted.groupby('day')['customer_id'].count().reset_index()
plt.figure(figsize=(10,4))
sns.lineplot(data=DailyOffersCompleted,x='day',y='customer_id',palette='flare')
plt.title('Daily offers completed trend')
plt.show()
```



- In this 30 day trial period the offers completed by the customers on the 1st day starts from the count of **1000** which was a great start.
- but the maximum offers completed is below the range of **2500** where as regular transactions maximum limit is **6000**, that means there is a huge difference between the regular transactions and Offers completed transactions.
- There is a upward trend is there from the 15 day the half of month.
- But incontinuty is there where regular patterns is showing in each 5 to 7 days sales increase then decrease and vise-versa.
- If we look at the both patterns in the first 10 day period the sale has been made in a moderate limit.
- But suddenly from the **10 th day to 15 th** day there is a downward trend showing,this days are **performing very low** as compared to other days,
- Similarly for the next 10 days from **15 th to 20** sales increases ,then again drop at the end day of the month.
- To understsnd what exactly the monthly pattern is? The comparison of this pattern with the overall years monthly pattern is neeed be perform better understanding.
- However the weekly trend is easily understandable.
- the businees should provide best discounts and offers in this low performing days to create the continuty and and positive tend in the business.
- **The day 20 and and 25 are the peak days of the business.**
- in simple words there is not a stability in this business and to control this incontinuty preplanning is important regarding on which weekdays sholud provide discout and offers ,how to engage with customers provide best services to the customers.

In [159…
```
Membership_Details=Customers.groupby('Year')['customer_id'].count().reset_index()
Membership_Details
```

Out[159…

	Year	customer_id
0	2013	286
1	2014	691
2	2015	1830
3	2016	3526
4	2017	6469
5	2018	4198

In [160…
```
MembershipCompleted=Offerscompleted.groupby('Year')['offer_id'].count().reset_index()
MembershipCompleted
```

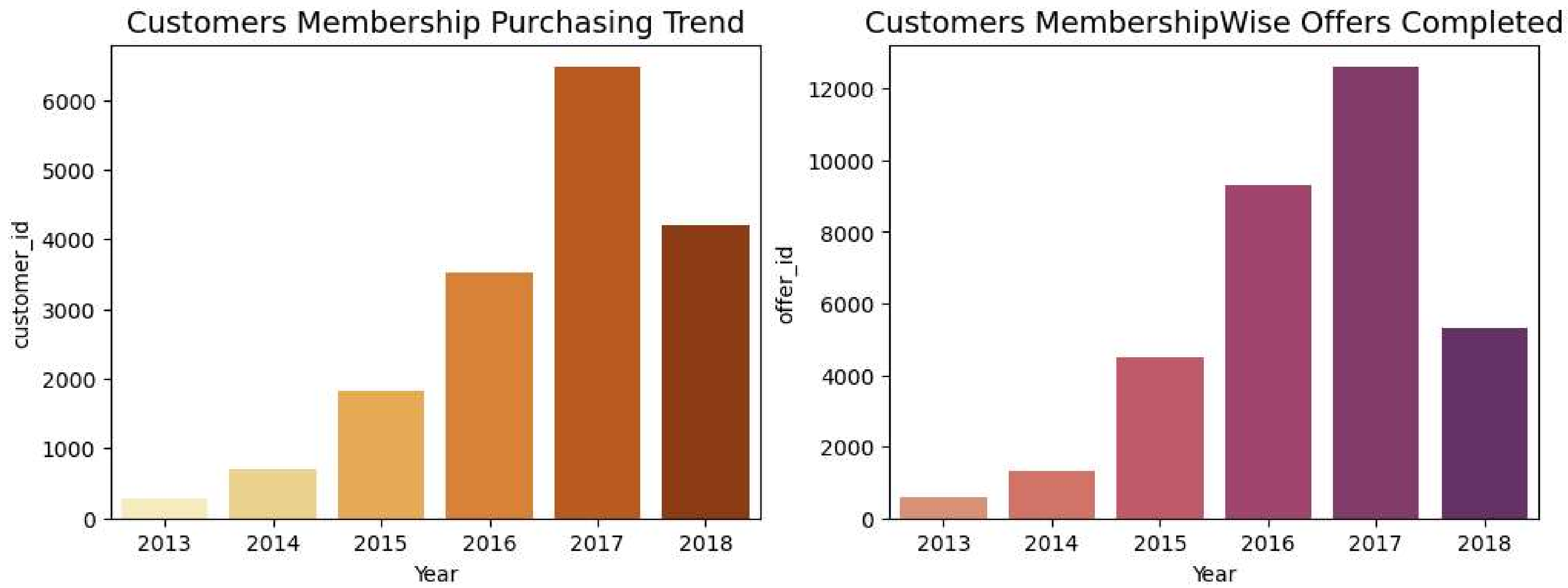
Out[160…

	Year	offer_id
0	2013	568
1	2014	1334
2	2015	4489
3	2016	9298
4	2017	12595
5	2018	5295

8.Do all the membership-year wise members completing the offers or not?

In [163…

```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
sns.barplot(data=Membership_Details,x=Membership_Details['Year'],y=Membership_Details['customer_id'],palette='YlOrBr')
plt.title('Customers Membership Purchasing Trend',fontsize=14)
plt.subplot(1,2,2)
sns.barplot(data=MembershipCompleted,x=MembershipCompleted['Year'],y=MembershipCompleted['offer_id'],palette='flare')
plt.title('Customers MembershipWise Offers Completed',fontsize=14)
plt.show()
```



- If we look at consumers membership details the numbers has dropped in the recent year 2018.
- from the year 2013-2017 their was a positive trend but in the Year 2018 the line has dropped to nearest of 4000 where as in the year 2017 the cafe got membership from the more than 6000 people.
- the sudden drop of membership for above 2000 consumers is a huge value and cannot be get ignored.
- Need of business marketing campaigns.
- Here plotting membership year wise customers offer completed count to see whether the all years members do they completes offers and takes the interest in the offers ,so the answer is yes as per membership count offers completed count plot is also similar showing as membership increases offers completed count also increases.

Events Insights:

In [116…

```
Events
```

Out[116…

	customer_id	event	offer_id	time	day
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1
1	a03223e636434f42ac4c3df47e8bac43	offer received	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1
2	e2127556f4f64592b11af22de27a7932	offer received	2906b810c7d4411798c6938adc9daaa5	0	1
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	fafdcd668e3743c1bb461111dcafc2a4	0	1
4	68617ca6246f4fbc85e91a2a49552598	offer received	4d5c57ea9a6940dd891ad53e9dbe8da0	0	1
...	...	...	...	...	...
306529	b3a1272bc9904337b331bf348c3e8c17	transaction	1.5899999999999999	714	30
306530	68213b08d99a4ae1b0dcb72aebd9aa35	transaction	9.53	714	30
306531	a00058cf10334a308c68e7631c529907	transaction	3.61	714	30
306532	76ddbd6576844afe811f1a3c0fbb5bec	transaction	3.5300000000000002	714	30
306533	c02b10e8752c4d8e9b73f918558531f7	transaction	4.05	714	30

306534 rows × 5 columns

In [120…

```
EventsCount= Events['event'].value_counts().reset_index()
```

In [121…

```
EventsCount
```

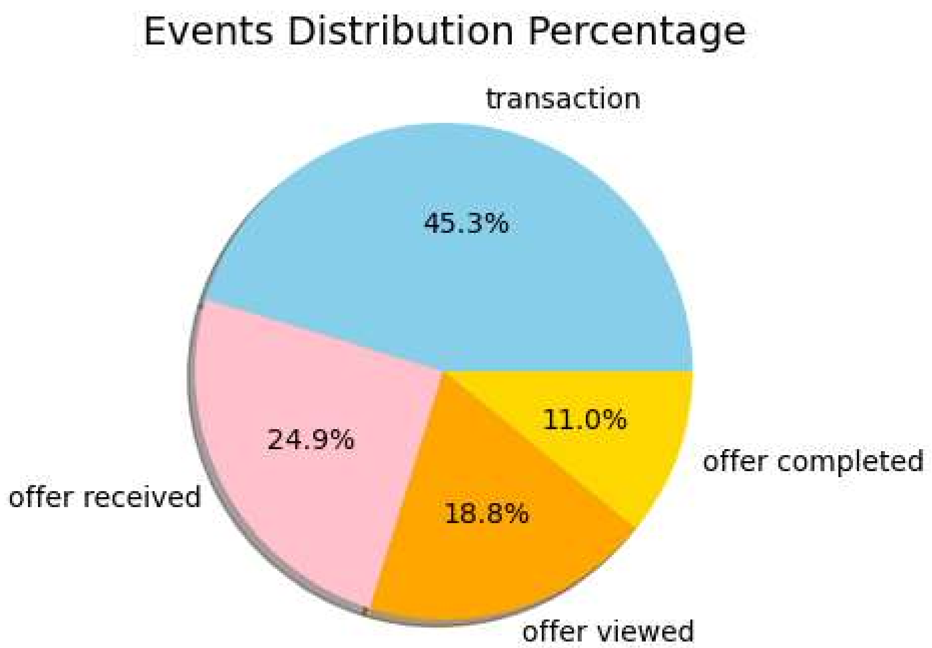
Out[121…

	event	count
0	transaction	138953
1	offer received	76277
2	offer viewed	57725
3	offer completed	33579

9.How was the events distribution in the business?

In [135…

```
col=['skyblue','pink','orange','gold']
plt.figure(figsize=(4,8))
plt.pie(EventsCount['count'],autopct='%1.1f%%',colors=col,labels=EventsCount['event'],shadow=True)
plt.title('Events Distribution Percentage',fontsize =14)
plt.show()
```



- The transaction count percentage **45.3%** is higher than the other events.
- Offers received percentage is **24.9%** ,on the other side offers completed distribution is only **11%** .
- Offers viewed percentage **18.8%** shows the offers information has reached to customers but completed rate is very low the half of offer received percentage.

In [152…

```
OffersData=pd.merge(Offerscompleted,Offers,on='offer_id')
OffersData
```

Out[152…

	customer_id	event	offer_id	time	day	offer_type	difficulty	reward	dura
0	9fa9ae8f57894cc9a3b8a9bbe0fc1b2f	offer completed	2906b810c7d4411798c6938adc9daaa5	0	1	discount	10	2	
1	fe97aa22dd3e48c8b143116a8403dd52	offer completed	fafdcd668e3743c1bb461111dcafc2a4	0	1	discount	10	2	
2	629fc02d56414d91bca360decdfa9288	offer completed	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1	bogo	5	5	
3	676506bad68e4161b9bbaffeb039626b	offer completed	ae264e3637204a6fb9bb56bc8210ddfd	0	1	bogo	10	10	
4	8f7dd3b2afe14c078eb4f6e6fe4ba97d	offer completed	4d5c57ea9a6940dd891ad53e9dbe8da0	0	1	bogo	10	10	
...	...	...	...	...	...	...	...	...	
33574	0c027f5f34dd4b9eba0a25785c611273	offer completed	2298d6c36e964ae4a3e7e9706d1fb8c2	714	30	discount	7	3	
33575	a6f84f4e976f44508c358cc9aba6d2b3	offer completed	2298d6c36e964ae4a3e7e9706d1fb8c2	714	30	discount	7	3	
33576	b895c57e8cd047a8872ce02aa54759d6	offer completed	fafdcd668e3743c1bb461111dcafc2a4	714	30	discount	10	2	
33577	8431c16f8e1d440880db371a68f82dd0	offer completed	fafdcd668e3743c1bb461111dcafc2a4	714	30	discount	10	2	
33578	24f56b5e1849462093931b164eb803b5	offer completed	fafdcd668e3743c1bb461111dcafc2a4	714	30	discount	10	2	

33579 rows × 10 columns

10. Which channels are more effective?

In [152…

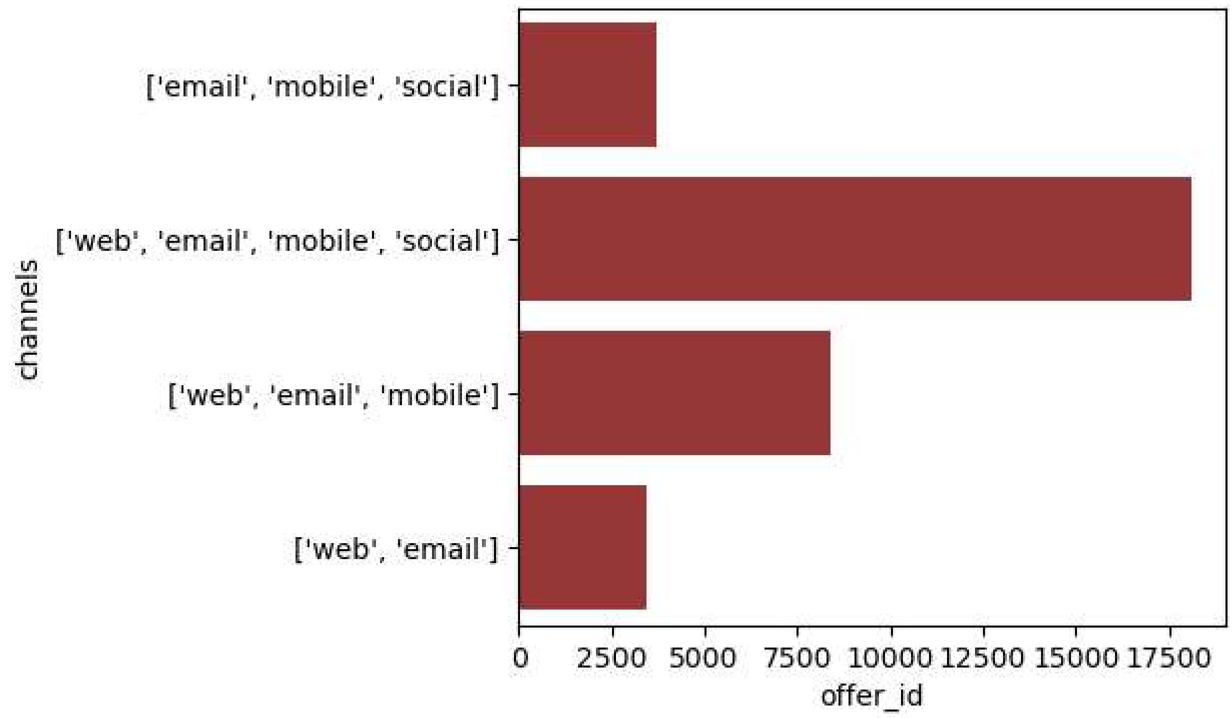
```
ChannelWise_OffersComplete=OffersData.groupby('channels')['offer_id'].count().reset_index()
ChannelWise_OffersComplete
```

Out[152…

	channels	offer_id
0	['email', 'mobile', 'social']	3688
1	['web', 'email', 'mobile', 'social']	18100
2	['web', 'email', 'mobile']	8371
3	['web', 'email']	3420

In [162…

```
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
sns.barplot(data=ChannelWise_OffersComplete,x='offer_id',y='channels',color='brown')
plt.show()
```



- **('web','email','mobile','social')** this channels set are most effective helped to complete **more than 17500 total 18100 offers**.
- that means the busines should use all the channels available fo marketing to get maximum offers complete.
- **('web','email','mobile')** is the second effective channel set.

11.Which offer_type has mostly choosed by Customers?

In [163…

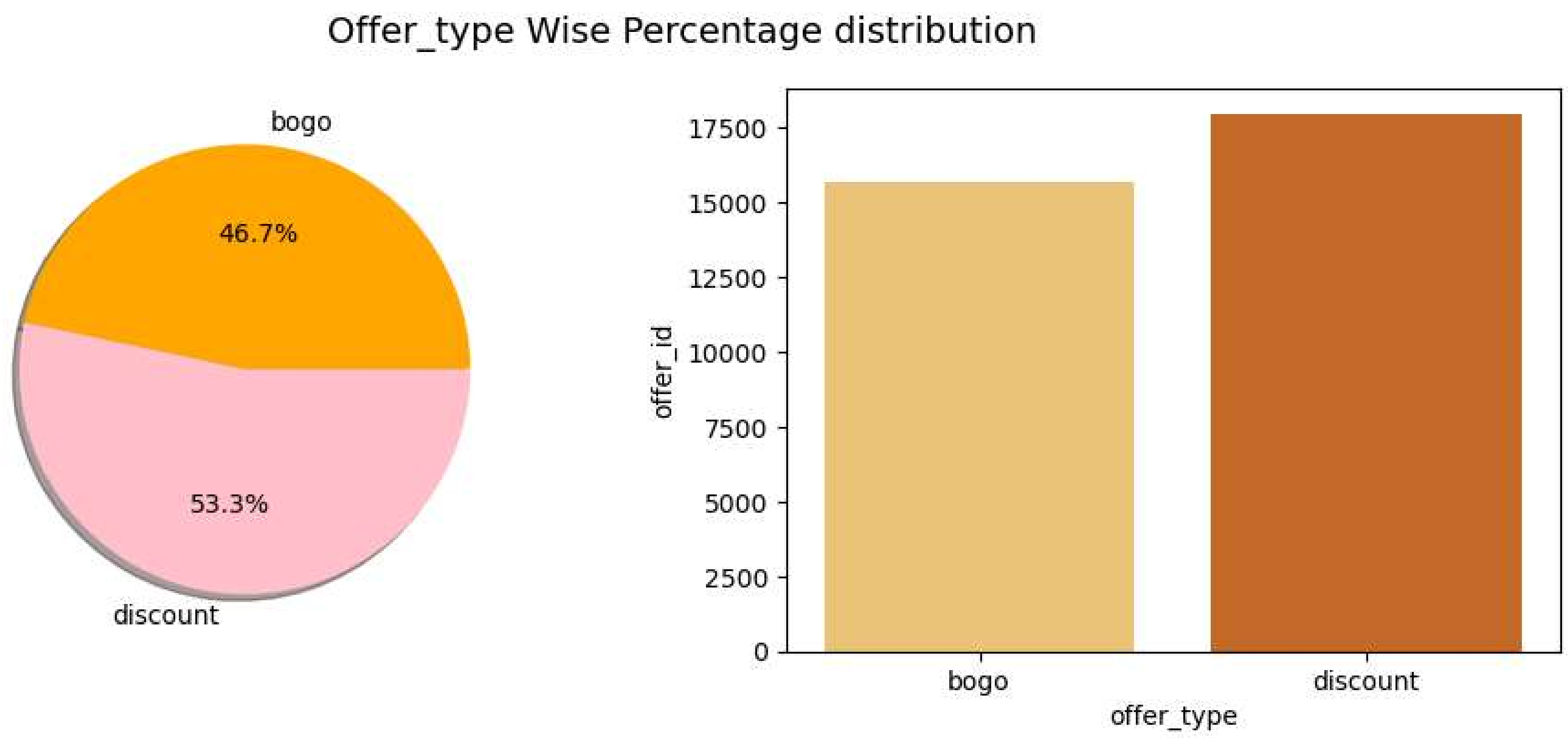
```
Offertype_OffersComplete=OffersData.groupby('offer_type')['offer_id'].count().reset_index()
Offertype_OffersComplete
```

Out[163…

	offer_type	offer_id
0	bogo	15669
1	discount	17910

In [200…

```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
col=['orange','pink']
plt.pie(Offertype_OffersComplete['offer_id'],autopct='%1.1f%%',shadow=True,labels=Offertype_OffersComplete['offer_type'],color
plt.subplot(1,2,2)
sns.barplot(data=Offertype_OffersComplete,y='offer_id',x='offer_type',palette='YlOrBr')
plt.suptitle('Offer_type Wise Percentage distribution',fontsize=14)
plt.show()
```



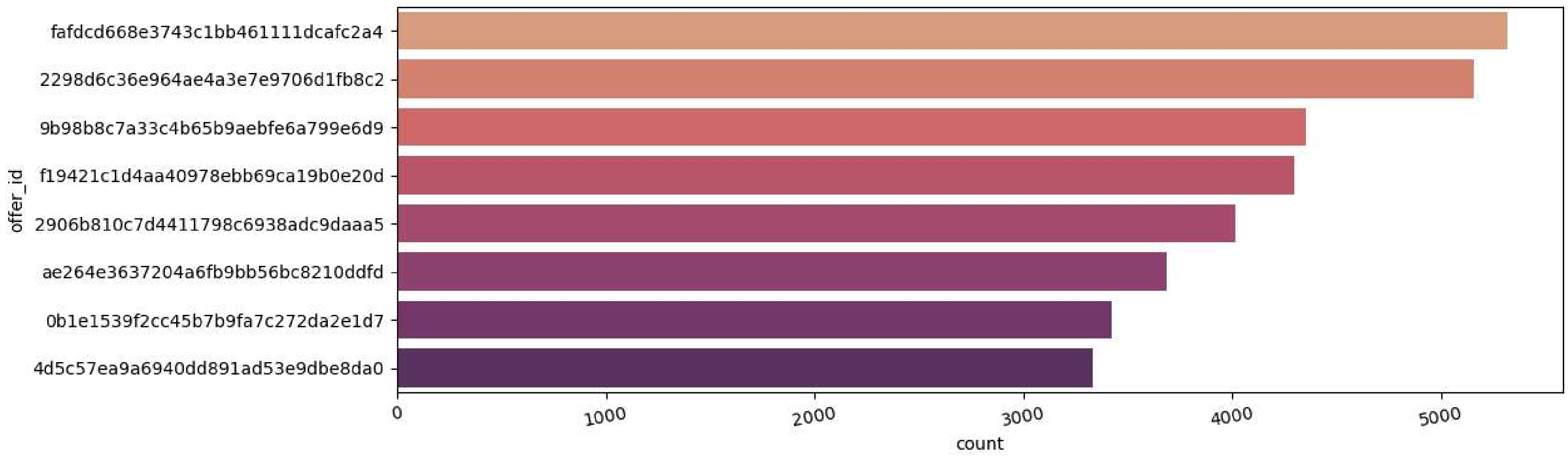
- Discount Offer type is most frequently choosed by the customers.
- The Discout offers completed count **more than 17500** ,Bogo offers are **more than 15000** total count around **2000 difference** is there.
- However both offer_type has received and completed by the customers having only **7% difference**.

12.Which offer is mostly completed by the customer?

In [187…

```
Offer_idCompleted=OffersData['offer_id'].value_counts().reset_index()
plt.figure(figsize=(12,4))
```

```
sns.barplot(data=Offer_idCompleted,y='offer_id',x='count',palette='flare')
plt.xticks(rotation=10)
plt.show()
```



- Offer_id's **('fafdcd668e3743c1bb461111dcafc2a4','2298d6c36e964ae4a3e7e9706d1fb8c2')** are the top 2 completed offers by the customers,**More than 5000 offers completed by both offers** .
- However **other 3** offers after the top 2 also achived good complete count ranges between **4000 to more than 4000**.
- **The lowest 3 perfoming offers** complete count is quite lower than top 2 completed count falls **under less than 4000 between 3000 to 3500**

13.Is there a difference between the reward provided,duration,difficulty of each offer_id?

In [204…

```
OffersDifference=OffersData.groupby('offer_id').agg({'reward':'mean','difficulty':'mean','duration':'mean'}).reset_index()
OffersDifference
```

Out[204…

	offer_id	reward	difficulty	duration
0	0b1e1539f2cc45b7b9fa7c272da2e1d7	5.0	20.0	10.0
1	2298d6c36e964ae4a3e7e9706d1fb8c2	3.0	7.0	7.0
2	2906b810c7d4411798c6938adc9daaa5	2.0	10.0	7.0
3	4d5c57ea9a6940dd891ad53e9dbe8da0	10.0	10.0	5.0
4	9b98b8c7a33c4b65b9aebfe6a799e6d9	5.0	5.0	7.0
5	ae264e3637204a6fb9bb56bc8210ddfd	10.0	10.0	7.0
6	f19421c1d4aa40978ebb69ca19b0e20d	5.0	5.0	5.0
7	fafdcd668e3743c1bb461111dcafc2a4	2.0	10.0	10.0

In [76]:

```
Offers
```

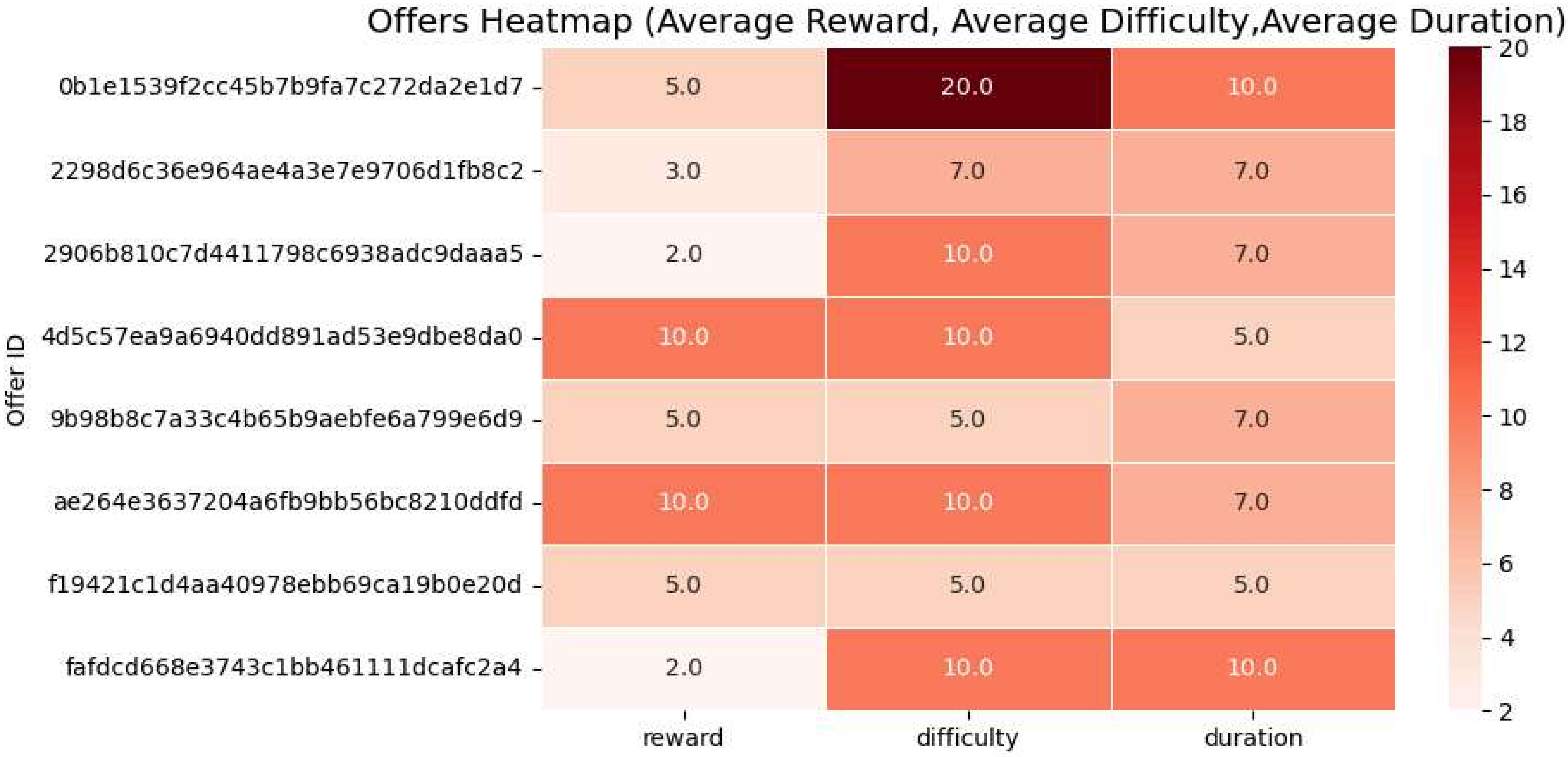
Out[76]:

	offer_id	offer_type	difficulty	reward	duration	channels
0	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10	10	7	['email', 'mobile', 'social']
1	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10	10	5	['web', 'email', 'mobile', 'social']
2	3f207df678b143eea3cee63160fa8bed	informational	0	0	4	['web', 'email', 'mobile']
3	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5	5	7	['web', 'email', 'mobile']
4	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	20	5	10	['web', 'email']
5	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	7	3	7	['web', 'email', 'mobile', 'social']
6	fafdcd668e3743c1bb461111dcafc2a4	discount	10	2	10	['web', 'email', 'mobile', 'social']
7	5a8bc65990b245e5a138643cd4eb9837	informational	0	0	3	['email', 'mobile', 'social']
8	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5	5	5	['web', 'email', 'mobile', 'social']
9	2906b810c7d4411798c6938adc9daaa5	discount	10	2	7	['web', 'email', 'mobile']

In [218…

```
plt.figure(figsize=(8,5))
sns.heatmap(OffersDifference.set_index('offer_id'),
            annot=True, fmt=".1f", cmap="Reds", linewidths=0.5)

plt.title("Offers Heatmap (Average Reward, Average Difficulty,Average Duration)", fontsize=14)
plt.ylabel("Offer ID")
plt.show()
```



- **('fafdcd668e3743c1bb461111dcafc2a4')** was the most frequent offer,at the top 1, Average (duration & difficulty)was 10.
- Second most frquent offers **('2298d6c36e964ae4a3e7e9706d1fb8c2')** difficulty and duration was 7 .
- This both offers are belong to the discount offer types,the details speaks louder that discount offers are often preffered and loved by the customers.
- If look neatly at the numbers the difficulty amount and duration both matters to the customers.
- Offer_id's **('ae264e3637204a6fb9bb56bc8210ddfd',4d5c57ea9a6940dd891ad53e9dbe8da0')** this were the offers providing the highest reward price **10 $**., then also didnt got customer response because difficulty was 10 for both and duration was 7 and 5 ,Difficulty amount was high and duration to complete was low was not a great combo of difficulty and duration.and **comes in the list of least 3 performing offers**.
- offer_id **('0b1e1539f2cc45b7b9fa7c272da2e1d7')** with highest difficulty **20** on second least position.

14. Customers with highest complete counts?

In [111…

```
Completecount =Offerscompleted.groupby('customer_id')['offer_id'].count().reset_index().rename(columns={'offer_id':'count'})
```

In [113…

```
Completecount['count'].unique()
```

Out[113…

```
array([3, 1, 2, 5, 4, 6])
```

In [133…

```
HighestCompletedCustomers=Completecount[Completecount['count']==6]
len(HighestCompletedCustomers)
```

Out[133…

```
200
```

In [150…

```
TopCustomers=HighestCompletedCustomers.merge(Customers,on='customer_id',how='inner')
TopCustomers
```

Out[150…

	customer_id	count	Membership	gender	age	income	Age_Category	Year	Income_Category
0	0200f61c69da4c2ea078842cdaf234e6	6	2016-12-10	M	64	76000.0	Greater than 60	2016	41000-80000
1	0335d274249f4eb6b3c51527f02a3216	6	2016-11-24	F	21	74000.0	21-40	2016	41000-80000
2	043bcfeacb874bbc837300701ce25870	6	2017-11-29	F	70	72000.0	Greater than 60	2017	41000-80000
3	0494aa6671414fab9837fa3cd45e72bc	6	2016-02-10	F	71	36000.0	Greater than 60	2016	0-40000
4	04dc7c54fc8147069875a5def52d711e	6	2016-11-03	F	74	57000.0	Greater than 60	2016	41000-80000
...	...	...	...	...	...	...	...	...	...
195	f8aedd0cbea0419c806842b4265b82e5	6	2016-08-11	F	72	72000.0	Greater than 60	2016	41000-80000
196	fd90af4b9b784b268efa9d349a762491	6	2018-06-01	M	30	64000.0	21-40	2018	41000-80000
197	ff932c6f8bb641bd816955337d153676	6	2015-09-29	M	65	76000.0	Greater than 60	2015	41000-80000
198	fff29fb549084123bd046dbc5ceb4faa	6	2017-08-31	F	59	93000.0	41-60	2017	81000-100000
199	ffff82501cea40309d5fdd7edcca4a07	6	2016-11-25	F	45	62000.0	41-60	2016	41000-80000

200 rows × 9 columns

In [138…

```
TopCustomers['income'].mean()
```

Out[138… np.float64(71760.0)

In [153…
```
TopCustomers.groupby('gender')['customer_id'].count().reset_index()
```

Out[153…

	gender	customer_id
0	F	103
1	M	93
2	O	4

- The in overall data ,generally males have contributed more but in Top complete count data females are more than male.
- this 103 females have completed total 6 offers.
- details of this customers are important to business to get get more engage with them by sending more offer details and discout details through different channels.

Conclusion :

- Male customers are more active than female customers in events and offer completions.
- A small percentage **(1.3%)** of customers have unknown gender, which may distort customer insights.
- The old age **(60+)** group is the largest customer base, while under 20 group is the smallest.
- A major portion of customers **(40–50%)** fall in the income range of **$40,000–80,000**.
- The correlation between age and income is weak **(0.2)**, meaning income cannot be predicted reliably by age.
- Most offers were targeted at old age customers, leading to higher completion in that segment.
- Regular transactions **(138,000)** are much higher than offers completed **(11%)**, showing many customers buy without offers.
- Daily transactions range between **3,000–6,000**, with fluctuations and unstable sales patterns.
- Customer membership peaked until 2017 but dropped sharply in 2018, showing reduced loyalty.
- Discount offers were the most popular, with two specific offers completing **5,000+** times each, while offers with high difficulty + short duration performed poorly.

Recommendations:

- Collect complete and accurate gender data to avoid bias in analysis.
- Improve targeting by ensuring equal marketing across all age groups, not just older customers.
- Focus marketing campaigns on the 21–40 age group, which is underrepresented compared to older customers.
- Provide loyalty programs and consistent discounts to stabilize sales and increase membership.
- Design offers with a balanced difficulty and duration (not too hard, not too short) for better completion.
- Track low-performing days and apply dynamic offers or promotions to maintain continuity in sales.