QUESTION - 8

- **Name: - Aryan Omkar Ashar**
- **Reg.Number: - 19BAI10094**
- **Course Code: - NAS2001**
- **Course Title: - Advanced Data Analysis**
- **Slot: - A21+A22+A23**
- **Faculty: - Dr. Kamlesh**

QUESTION - 8: - Retail Customer Analysis

A Retail store is required to analyze the day-to-day transactions and keep a track of its customers spread across various locations along with their purchases/returns across various categories.

Import necessary libraries

In [1]:

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

Import the data set

In [2]:

```
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/Transactions.csv
/kaggle/input/Customer.csv
/kaggle/input/prod_cat_info.csv
```

In [3]:

```
customer = pd.read_csv("/kaggle/input/Customer.csv")
prod_info = pd.read_csv("/kaggle/input/prod_cat_info.csv")
transaction = pd.read_csv("/kaggle/input/Transactions.csv")
```

Exploratory Data Analysis

In [4]:

```
customer.shape
```

Out[4]:

```
(5647, 4)
```

In [5]:

```
prod_info.shape
```

Out[5]:

```
(23, 4)
```

In [6]:

```
transaction.shape
```

Out[6]:

```
(23053, 10)
```

In [7]:

```
customer.head(2)
```

Out[7]:

	customer_Id	DOB	Gender	city_code
0	268408	02-01-1970	M	4.0
1	269696	07-01-1970	F	8.0

In [8]:

```
prod_info.head(2)
```

Out[8]:

	prod_cat_code	prod_cat	prod_sub_cat_code	prod_subcat
0	1	Clothing	4	Mens
1	1	Clothing	1	Women

In [9]:

```
# renaming "prod_sub_cat_code" column in 'prod_info' table to make it similar to 'transac
tion' table
# to merge the both the tables easily
prod_info.rename(columns={"prod_sub_cat_code":"prod_subcat_code"},inplace=True)
```

In [10]:

```
transaction.head()
```

Out[10]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type
0	80712190438	270351	28-02-2014	1	1	-5	-772	405.300	-4265.300	e-Shop
1	29258453508	270384	27-02-2014	5	3	-5	-1497	785.925	-8270.925	e-Shop
2	51750724947	273420	24-02-2014	6	5	-2	-791	166.110	-1748.110	TeleShop
3	93274880719	271509	24-02-2014	11	6	-3	-1363	429.345	-4518.345	e-Shop
4	51750724947	273420	23-02-2014	6	5	-2	-791	166.110	-1748.110	TeleShop

1. Merge the datasets Customers, Product Hierarchy and Transactions as Customer_Final

Merge 'transaction' and 'prod_info' tables

In [11]:

```
# merge transaction and prod_info table and create a new table "prod_concat"
prod_concat = pd.merge(left=transaction, right=prod_info,on=["prod_cat_code","prod_subcat_code"],how="left")
```

In [12]:

```
prod_concat
```

Out[12]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	proc
0	80712190438	270351	28-02-2014	1	1	-5	-772	405.300	-4265.300	e-Shop	Clo
1	29258453508	270384	27-02-2014	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electro
2	51750724947	273420	24-02-2014	6	5	-2	-791	166.110	-1748.110	TeleShop	B
3	93274880719	271509	24-02-2014	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home kit
4	51750724947	273420	23-02-2014	6	5	-2	-791	166.110	-1748.110	TeleShop	B
...	...	...	...	...	...	...	...	...	...	...	
23048	94340757522	274550	25-01-2011	12	5	1	1264	132.720	1396.720	e-Shop	B
23049	89780862956	270022	25-01-2011	4	1	1	677	71.085	748.085	e-Shop	Clo
23050	85115299378	271020	25-01-2011	2	6	4	1052	441.840	4649.840	MBR	Home kit
23051	72870271171	270911	25-01-2011	11	5	3	1142	359.730	3785.730	TeleShop	B
23052	77960931771	271961	25-01-2011	11	5	1	447	46.935	493.935	TeleShop	B

23053 rows × 12 columns

In [13]:

```
prod_concat.isnull().sum()
```

Out[13]:

```
transaction_id      0
cust_id             0
tran_date           0
prod_subcat_code    0
prod_cat_code       0
Qty                 0
Rate                0
Tax                 0
total_amt           0
Store_type          0
prod_cat            0
prod_subcat         0
dtype: int64
```

Merge 'customer' and 'prod_concat' tables

In [14]:

```
customer.head()
```

Out[14]:

	customer_Id	DOB	Gender	city_code
0	268408	02-01-1970	M	4.0
1	269696	07-01-1970	F	8.0
2	268159	08-01-1970	F	8.0
3	270181	10-01-1970	F	2.0
4	268073	11-01-1970	M	1.0

In [15]:

```
#merge "prod_concat" and "customer" table and create the final table "customer_final"
customer_final = pd.merge(left=prod_concat, right=customer,right_on="customer_Id", left_o
n="cust_id", how="left")
```

In [16]:

```
customer_final.head()
```

Out[16]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	28-02-2014	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	27-02-2014	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics
2	51750724947	273420	24-02-2014	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
3	93274880719	271509	24-02-2014	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home and kitchen
4	51750724947	273420	23-02-2014	6	5	-2	-791	166.110	-1748.110	TeleShop	Books

In [17]:

```
customer_final.shape
```

Out[17]:

```
(23053, 16)
```

In [18]:

```
transaction.shape
```

Out[18]:

```
(23053, 10)
```

In [19]:

```
print('''Rows of both the 'customer_final' and 'transaction' table are same. That means a
ll the transactions done at the
        Retail Store are present in the final table ''')
```

```
Rows of both the 'customer_final' and 'transaction' table are same. That means all the tr
ansactions done at the
        Retail Store are present in the final table
```

In [20]:

```
customer_final.dtypes
```

Out[20]:

```
transaction_id        int64
cust_id               int64
tran_date            object
prod_subcat_code      int64
prod_cat_code         int64
Qty                   int64
Rate                  int64
Tax                 float64
total_amt           float64
Store_type           object
prod_cat             object
prod_subcat          object
customer_Id           int64
DOB                  object
Gender               object
city_code           float64
dtype: object
```

In [21]:

```
customer_final.isnull().sum()
```

Out[21]:

```
transaction_id      0
cust_id             0
tran_date           0
prod_subcat_code    0
prod_cat_code       0
Qty                 0
Rate                0
Tax                 0
total_amt           0
Store_type          0
prod_cat            0
prod_subcat         0
customer_Id         0
DOB                 0
Gender              9
city_code           8
dtype: int64
```

In [22]:

```
# converting "DOB" and "tran_date" from object dtype to dates
customer_final["DOB"] = pd.to_datetime(customer_final["DOB"], format="%d-%m-%Y")
```

In [23]:

```
customer_final['DOB'].head(10)
```

Out[23]:

```
0   1981-09-26
1   1973-05-11
2   1992-07-27
3   1981-06-08
4   1992-07-27
5   1982-10-09
6   1981-05-29
7   1971-04-21
8   1971-11-04
9   1979-11-27
Name: DOB, dtype: datetime64[ns]
```

In [24]:

```
customer_final["tran_date"] = pd.to_datetime(customer_final["tran_date"])
```

In [25]:

```
customer_final["tran_date"].head(10)
```

Out[25]:

```
0   2014-02-28
1   2014-02-27
2   2014-02-24
3   2014-02-24
4   2014-02-23
5   2014-02-23
6   2014-02-22
7   2014-02-22
8   2014-02-22
9   2014-02-21
Name: tran_date, dtype: datetime64[ns]
```

Checking for duplicate values

In [26]:

```
customer_final.duplicated().sum()
```

Out[26]:

```
13
```

In [27]:

```
# dropping duplicate rows
customer_final.drop_duplicates(inplace=True)
```

In [28]:

```
customer_final.duplicated().sum()
```

Out[28]:

```
0
```

2. Prepare a summary report for the merged data set.

(a) Get the column names and their corresponding data types

In [29]:

```
#column names of "customer_final" dataframe
customer_final.columns
```

Out[29]:

```
Index(['transaction_id', 'cust_id', 'tran_date', 'prod_subcat_code',
       'prod_cat_code', 'Qty', 'Rate', 'Tax', 'total_amt', 'Store_type',
       'prod_cat', 'prod_subcat', 'customer_Id', 'DOB', 'Gender', 'city_code'],
      dtype='object')
```

In [30]:

```
# data types of all columns of "customer_final" dataframe
customer_final.dtypes
```

Out[30]:

```
transaction_id               int64
cust_id                      int64
tran_date           datetime64[ns]
prod_subcat_code             int64
```

```
prod_cat_code                int64
Qty                          int64
Rate                         int64
Tax                        float64
total_amt                  float64
Store_type                  object
prod_cat                    object
prod_subcat                 object
customer_Id                  int64
DOB                 datetime64[ns]
Gender                      object
city_code                  float64
dtype: object
```

(b) Top/Bottom 10 observations

In [31]:

```
# top 10 observations
customer_final.head(10)
```

Out[31]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
3	93274880719	271509	2014-02-24	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home and kitchen
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
5	97439039119	272357	2014-02-23	8	3	-2	-824	173.040	-1821.040	TeleShop	Electronics
6	45649838090	273667	2014-02-22	11	6	-1	-1450	152.250	-1602.250	e-Shop	Home and kitchen
7	22643667930	271489	2014-02-22	12	6	-1	-1225	128.625	-1353.625	TeleShop	Home and kitchen
8	79792372943	275108	2014-02-22	3	1	-3	-908	286.020	-3010.020	MBR	Clothing
9	50076728598	269014	2014-02-21	8	3	-4	-581	244.020	-2568.020	e-Shop	Electronics

In [32]:

```
#bottom 10 observations
customer_final.tail(10)
```

Out[32]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	proc
23043	49882891062	271982	2011-01-25	10	5	4	1330	558.600	5878.600	e-Shop	B
23044	14787475597	273982	2011-01-25	4	3	5	969	508.725	5353.725	e-Shop	Electro
23045	50691119572	273031	2011-01-25	6	5	1	1148	120.540	1268.540	TeleShop	B
23046	40893803228	272049	2011-01-25	11	6	3	1077	339.255	3570.255	e-Shop	Home kit

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod
23047	30856003613	266866	2011-01-25	4	2	2	444	93.240	981.240	TeleShop	Foot
23048	94340757522	274550	2011-01-25	12	5	1	1264	132.720	1396.720	e-Shop	B
23049	89780862956	270022	2011-01-25	4	1	1	677	71.085	748.085	e-Shop	Clo
23050	85115299378	271020	2011-01-25	2	6	4	1052	441.840	4649.840	MBR	Home kit
23051	72870271171	270911	2011-01-25	11	5	3	1142	359.730	3785.730	TeleShop	B
23052	77960931771	271961	2011-01-25	11	5	1	447	46.935	493.935	TeleShop	B

(c) "Five-number summary" for continuous variables (min, Q1, median, Q3 and max)

In [33]:

```
customer_final.describe()
```

Out[33]:

	transaction_id	cust_id	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_a
count	2.304000e+04	23040.000000	23040.000000	23040.000000	23040.000000	23040.000000	23040.000000	23040.000
mean	5.006955e+10	271021.880252	6.148785	3.763498	2.435764	637.094965	248.677488	2109.865
std	2.898062e+10	2431.573668	3.726197	1.677091	2.264326	621.727374	187.188311	2505.610
min	3.268991e+06	266783.000000	1.000000	1.000000	-5.000000	-1499.000000	7.350000	-8270.925
25%	2.493315e+10	268935.000000	3.000000	2.000000	1.000000	312.000000	98.280000	762.450
50%	5.009188e+10	270980.500000	5.000000	4.000000	3.000000	710.000000	199.080000	1756.950
75%	7.532632e+10	273114.250000	10.000000	5.000000	4.000000	1109.000000	365.767500	3570.255
max	9.998755e+10	275265.000000	12.000000	6.000000	5.000000	1500.000000	787.500000	8287.500

(d) Frequency tables for all the categorical variables

In [34]:

```
customer_final.loc[:,customer_final.dtypes=="object"].describe()
```

Out[34]:

	Store_type	prod_cat	prod_subcat	Gender
count	23040	23040	23040	23031
unique	4	6	18	2
top	e-Shop	Books	Women	M
freq	9304	6066	3046	11804

(3) Generate histograms for all continuous variables and frequency bars for categorical variables

Histogram of all continuous variables

In [35]:

In [35]:

```
conti_customer = customer_final.loc[:,['prod_subcat_code','prod_cat_code', 'Qty', 'Rate',
'Tax', 'total_amt']]
```

In [36]:

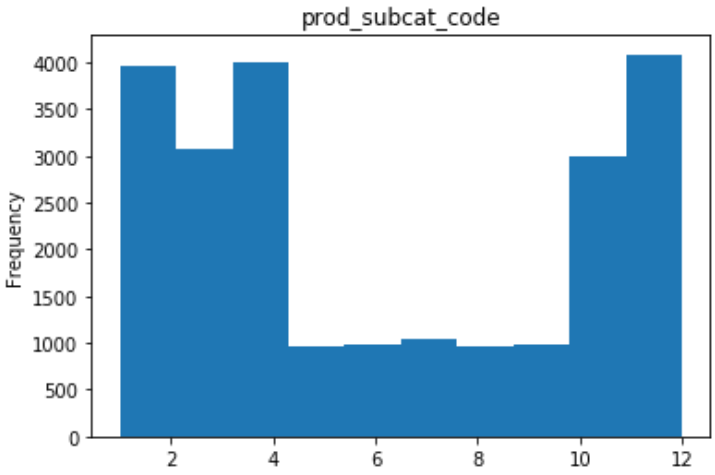
```
conti_customer.columns
```

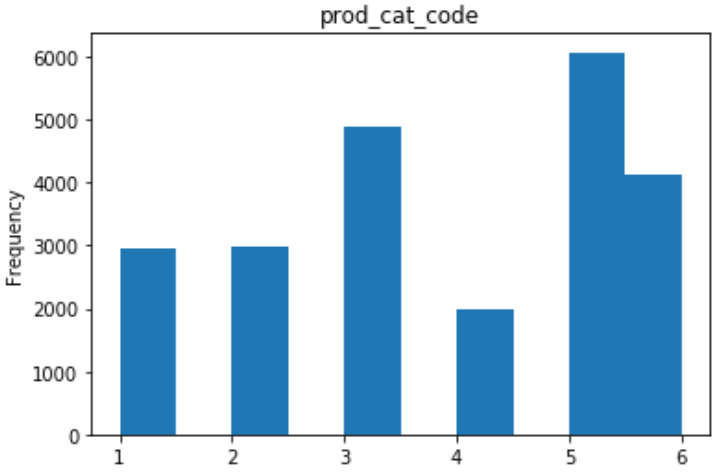
Out[36]:

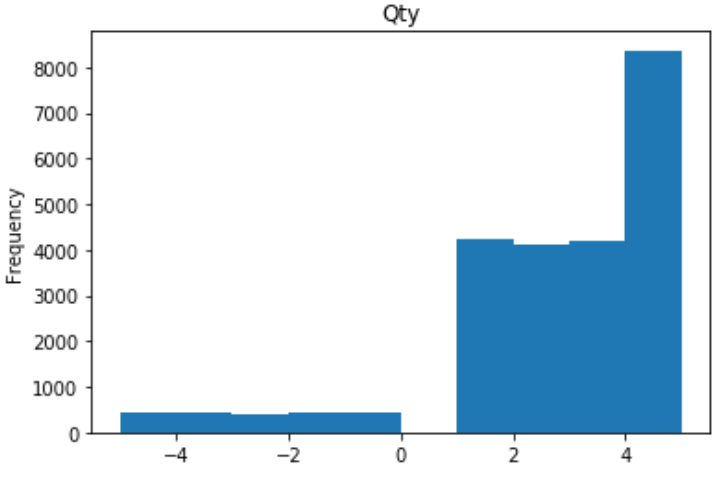
```
Index(['prod_subcat_code', 'prod_cat_code', 'Qty', 'Rate', 'Tax', 'total_amt'], dtype='ob
ject')
```

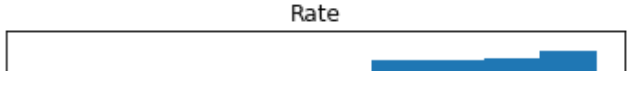
In [37]:

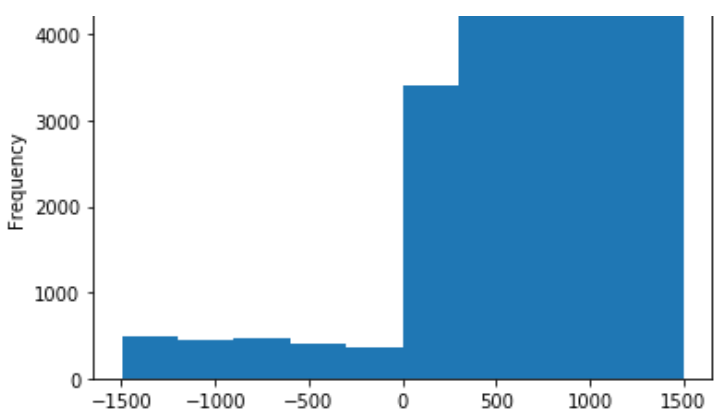
```
for var in conti_customer.columns:
    conti_customer[var].plot(kind='hist')
    plt.title(var)
    plt.show()
```

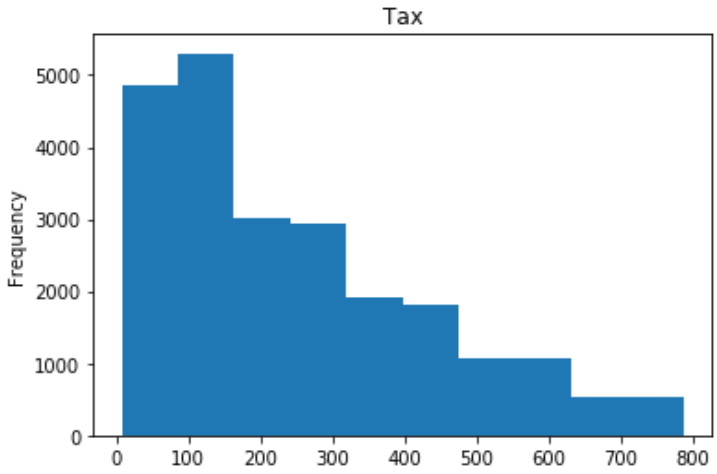


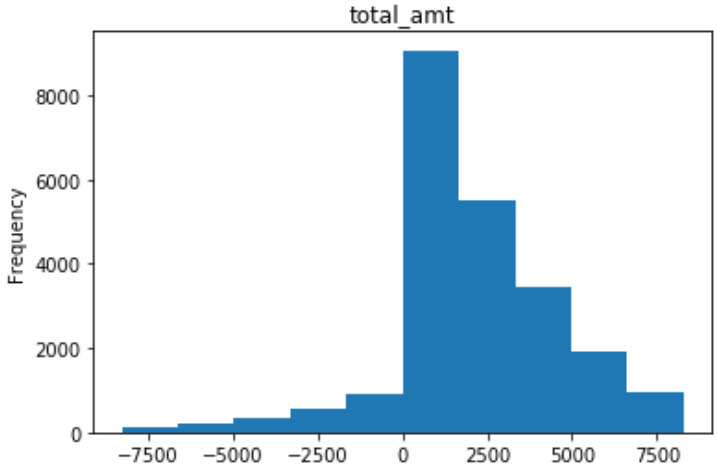












Bar chart of categorical variables

In [38]:

```
category_customer = customer_final.loc[:,customer_final.dtypes=='object']
```

In [39]:

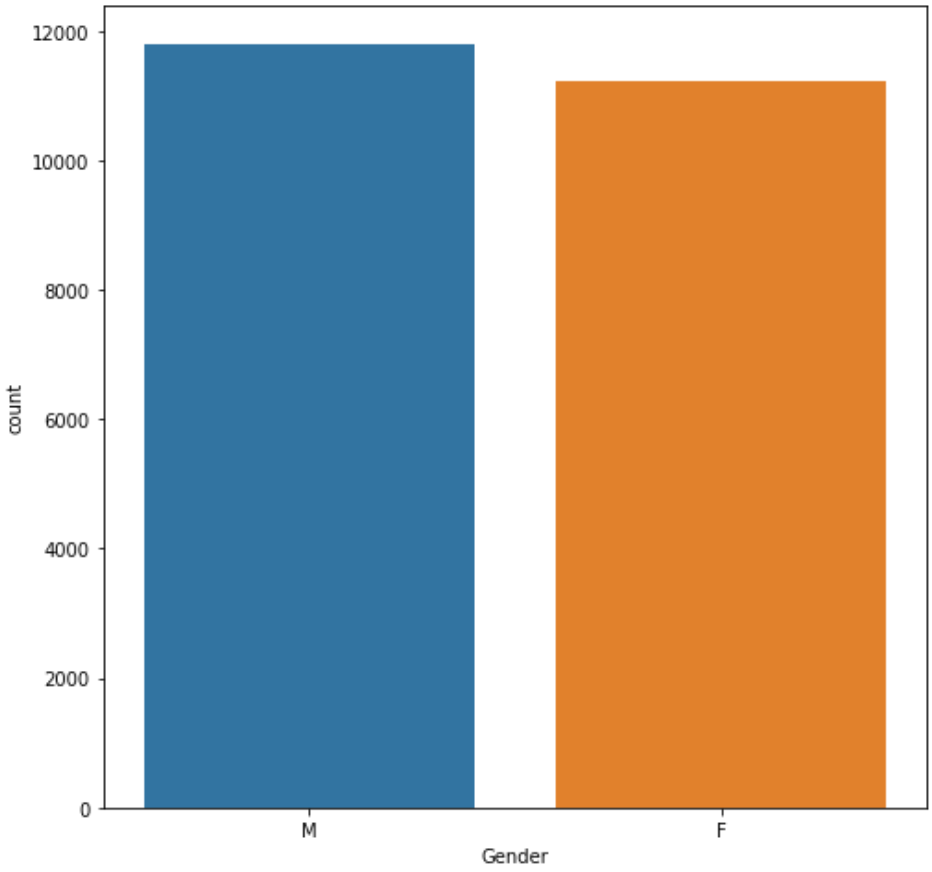
```
category_customer.head()
```

Out[39]:

	Store_type	prod_cat	prod_subcat	Gender
0	e-Shop	Clothing	Women	M
1	e-Shop	Electronics	Computers	F
2	TeleShop	Books	DIY	M
3	e-Shop	Home and kitchen	Bath	M
4	TeleShop	Books	DIY	M

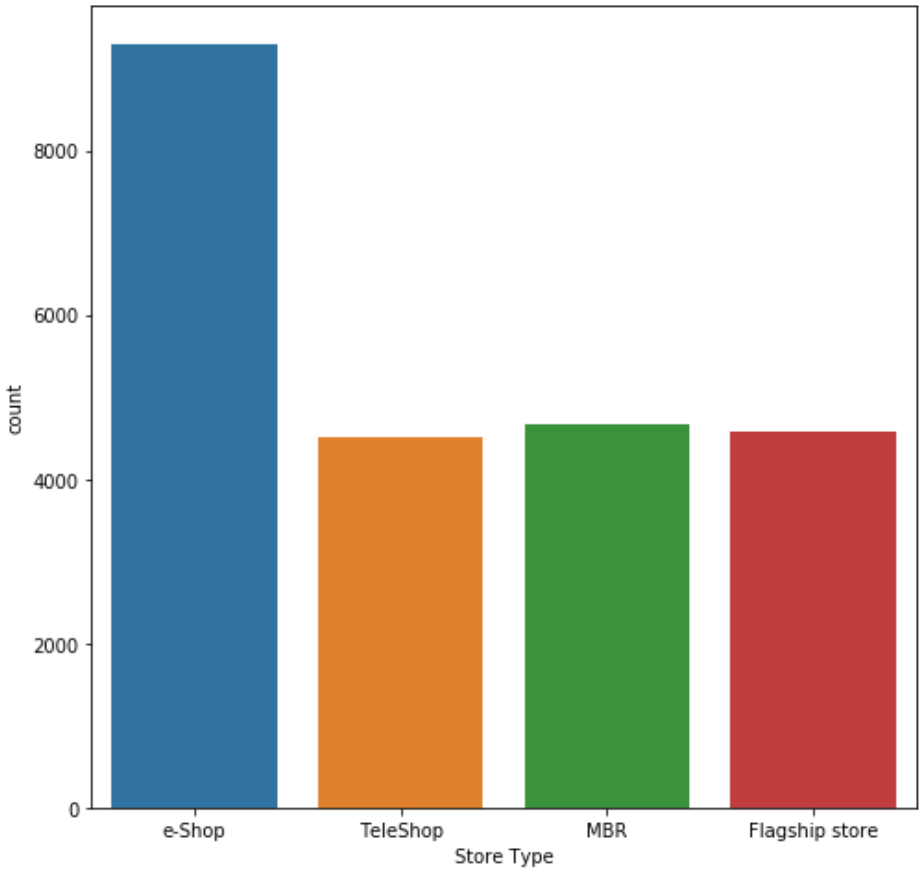
In [40]:

```
plt.figure(figsize=(8,8))
sns.countplot(category_customer['Gender'])
plt.show()
```



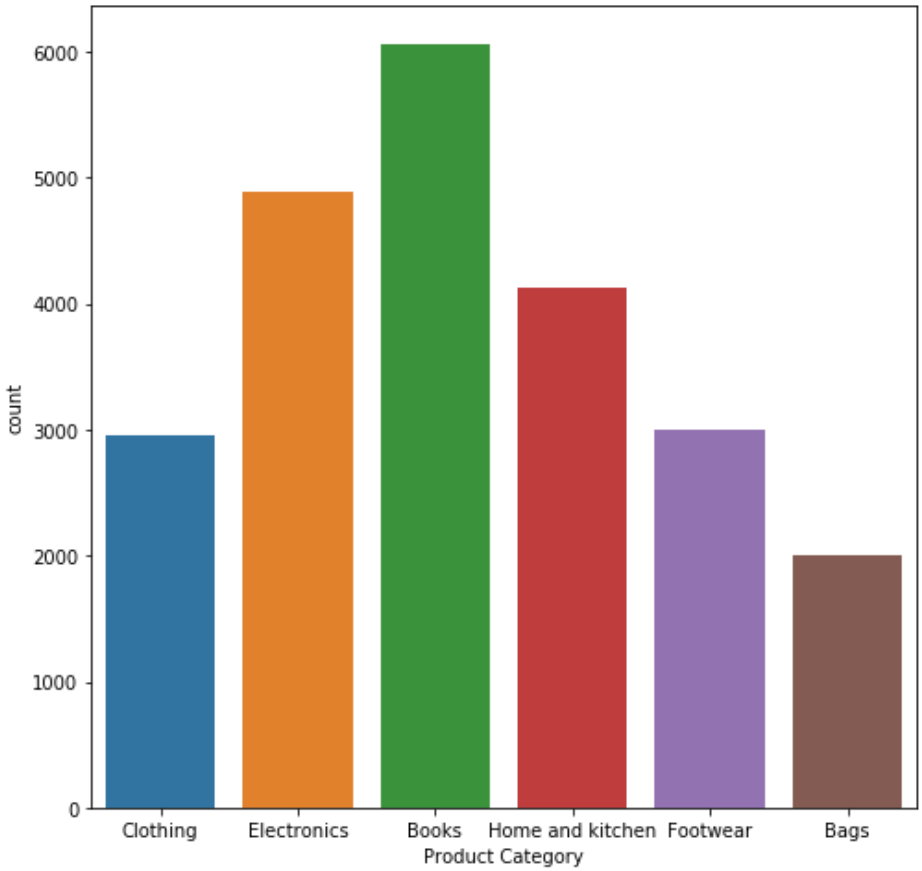
In [41]:

```
plt.figure(figsize=(8,8))
sns.countplot(category_customer['Store_type'])
plt.xlabel('Store Type')
plt.show()
```



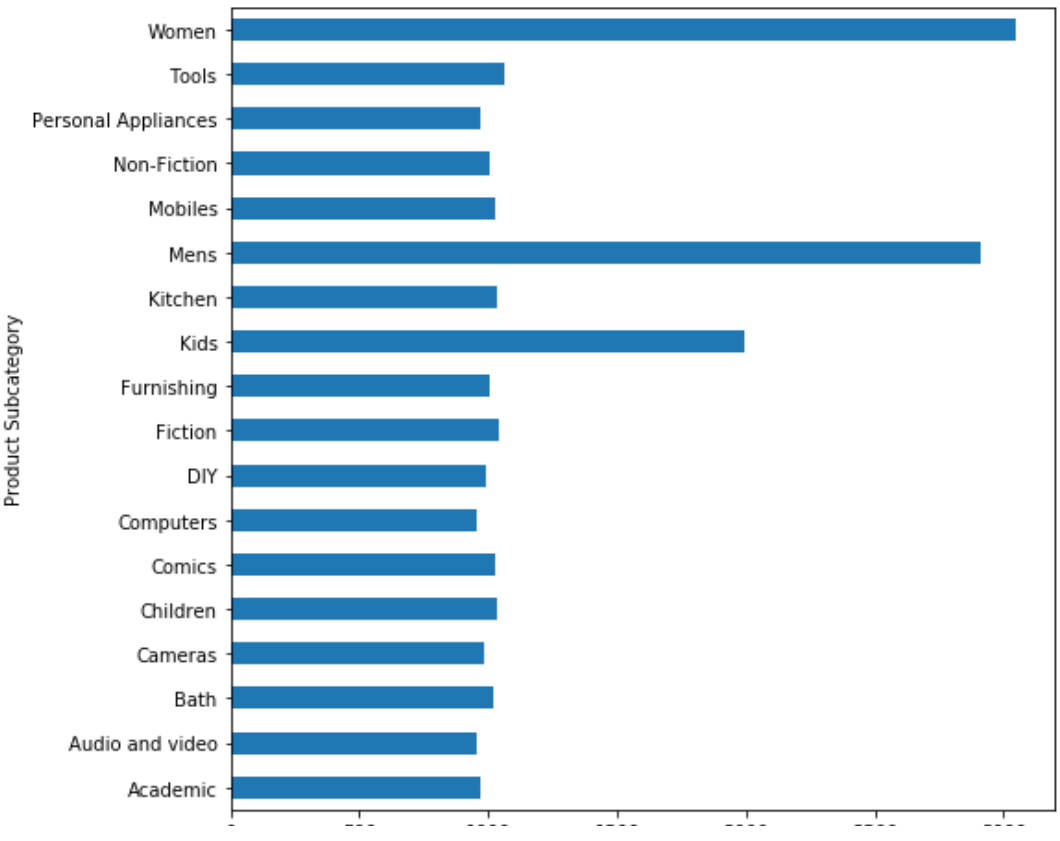
In [42]:

```
plt.figure(figsize=(8,8))
sns.countplot(category_customer['prod_cat'])
plt.xlabel('Product Category')
plt.show()
```



In [43]:

```
plt.figure(figsize=(8,8))
category_customer.groupby('prod_subcat')['prod_subcat'].count().plot(kind='barh')
plt.xlabel('Count')
plt.ylabel('Product Subcategory')
plt.show()
```



(4) Calculate the following information using the merged dataset:

(a) Time period of the available transaction data

In [44]:

```
customer_final.sort_values(by="tran_date")
```

Out[44]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	proc
22899	36332303449	268624	2011-01-02	10	6	-4	-295	123.900	-1303.900	Flagship store	Home kit
22893	25374972356	268904	2011-01-02	2	6	5	821	431.025	4536.025	MBR	Home kit
22894	15662366857	272756	2011-01-02	5	3	3	527	166.005	1747.005	e-Shop	Electro
22895	28972634039	275227	2011-01-02	9	3	-1	-334	35.070	-369.070	MBR	Electro
22896	60041644943	267309	2011-01-02	3	2	1	392	41.160	433.160	Flagship store	Foot
...	...	...	...	...	...	...	...	...	...	...	
161	65228973233	270009	2014-12-02	11	5	2	301	63.210	665.210	e-Shop	B
162	83661978186	274678	2014-12-02	10	3	2	325	68.250	718.250	e-Shop	Electro
147	36792372906	275246	2014-12-02	2	6	1	1185	124.425	1309.425	e-Shop	Home kit
154	74023090711	271180	2014-12-02	8	3	3	271	85.365	898.365	Flagship store	Electro
146	17146707816	274897	2014-12-02	12	5	3	622	195.930	2061.930	MBR	B

23040 rows × 16 columns

In [45]:

```
min_date = customer_final["tran_date"].min()
```

In [46]:

```
max_date = customer_final["tran_date"].max()
```

In [47]:

```
print("Time period of the available transaction data is from "+ pd.Timestamp.strftime(min_date,format="%d-%m-%Y") + " to " + pd.Timestamp.strftime(max_date,format="%d-%m-%Y"))
```

```
Time period of the available transaction data is from 02-01-2011 to 02-12-2014
```

(b) Count of transactions where the total amount of transaction was negative

In [48]:

```
customer_final.head()
```

Out[48]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
3	93274880719	271509	2014-02-24	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home and kitchen
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Books

In [49]:

```
#count of transaction_ids where total_amt was negative
negative_transaction = customer_final.loc[customer_final["total_amt"] < 0,"transaction_id
"].count()
```

In [50]:

```
print("Count of transactions where the total amount of transaction was negative is",negat
ive_transaction)
```

```
Count of transactions where the total amount of transaction was negative is 2164
```

(5) Analyze which product categories are more popular among females vs male customers

In [51]:

```
#groupby the data set on the basis of "Gender" and "prod_cat"
product_gender = customer_final.groupby(["Gender","prod_cat"])[["Qty"]].sum().reset_inde
x()
```

In [52]:

```
product_gender
```

Out[52]:

	Gender	prod_cat	Qty
0	F	Bags	2364
1	F	Books	7080
2	F	Clothing	3425
3	F	Electronics	5832
4	F	Footwear	3721
5	F	Home and kitchen	4898
6	M	Bags	2346
7	M	Books	7587
8	M	Clothing	3748
9	M	Electronics	6486
10	M	Footwear	3561
11	M	Home and kitchen	5051

In [53]:

```
#converting to pivot table for better view
product_gender.pivot(index="Gender",columns="prod_cat",values="Qty")
```

Out[53]:

prod_cat	Bags	Books	Clothing	Electronics	Footwear	Home and kitchen
Gender						
F	2364	7080	3425	5832	3721	4898
M	2346	7587	3748	6486	3561	5051

Products that are popular among males are:

- **Books**
- **Clothing**
- **Electronics**
- **Home and kitchen**

Products that are popular among females are:

- **Bags**
- **Footwear**

(6) Which City code has the maximum customers and what was the percentage of customers from that city?

In [54]:

```
customer_final.head(2)
```

Out[54]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics

In [55]:

```
customer_group = customer_final.groupby('city_code')['customer_Id'].count().sort_values(
ascending =False)
```

In [56]:

```
customer_group
```

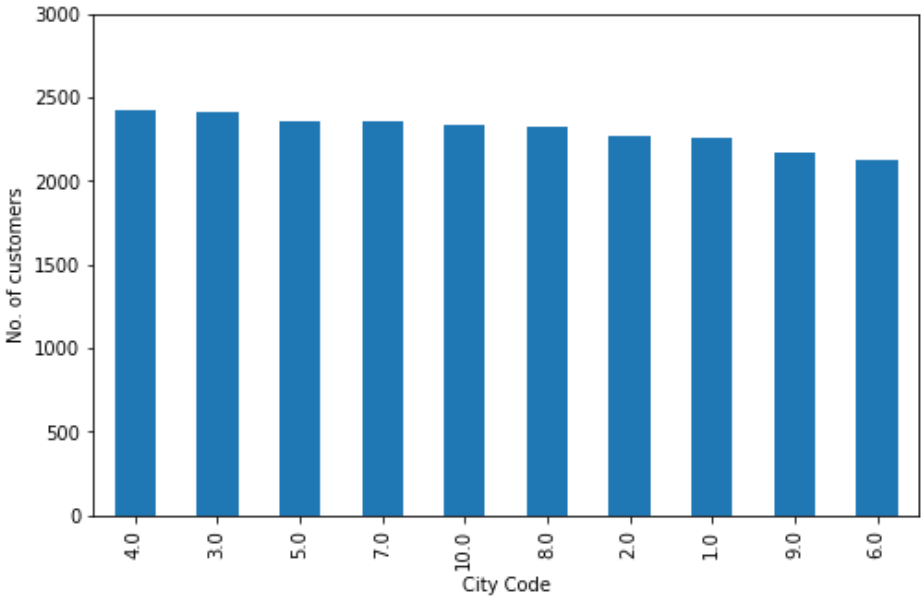
Out[56]:

```
city_code
4.0     2422
3.0     2410
5.0     2357
7.0     2356
10.0    2333
8.0     2328
2.0     2268
1.0     2255
9.0     2176
6.0     2127
```

Name: customer_Id, dtype: int64

In [57]:

```
plt.figure(figsize=(8,5))
customer_group.plot(kind="bar")
plt.xlabel("City Code")
plt.ylabel("No. of customers")
plt.yticks(np.arange(0, 3500, step=500))
plt.show()
```



In [58]:

```
percentage = round((customer_group[4.0] / customer_group.sum()) * 100,2)
```

In [59]:

```
percentage
```

Out[59]:

10.52

In [60]:

```
print("City code 4.0 has the maximum customers and the percentage of customers from that city is ",percentage)
```

City code 4.0 has the maximum customers and the percentage of customers from that city is 10.52

(7) Which store type sells the maximum products by value and by quantity?

In [61]:

```
customer_final.head(2)
```

Out[61]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics

In [62]:

```
customer_final.groupby("Store_type")["Qty","Rate"].sum().sort_values(by="Qty",ascending=
False)
```

Out[62]:

Store_type	Qty	Rate
e-Shop	22790	5945770
MBR	11195	2953665
Flagship store	11142	2942874
TeleShop	10993	2836359

In [63]:

```
print('e-Shop store sell the maximum products by value and by quantity')
```

```
e-Shop store sell the maximum products by value and by quantity
```

(8) What was the total amount earned from the "Electronics" and "Clothing" categories from Flagship Stores?

In [64]:

```
store_group = round(customer_final.pivot_table(index = "prod_cat",columns="Store_type",
values="total_amt", aggfunc='sum'),2)
```

In [65]:

```
store_group
```

Out[65]:

Store_type	Flagship store	MBR	TeleShop	e-Shop
prod_cat				
Bags	870548.83	848678.68	789181.06	1617933.27
Books	2493677.81	2496039.20	2545714.47	5297161.15
Clothing	1194423.23	1287686.34	1241834.36	2527193.57
Electronics	2215136.04	2107969.82	1978457.19	4429142.77
Footwear	1234806.56	1112163.72	1235719.29	2643215.25
Home and kitchen	1713004.15	1822403.57	1581227.37	3327977.12

In [66]:

```
store_group.loc[["Clothing","Electronics"],"Flagship store"]
```

Out[66]:

```
prod_cat
Clothing       1194423.23
Electronics    2215136.04
Name: Flagship store, dtype: float64
```

In [67]:

```
# if we have to find total amount of both 'Clothing' and 'Electronics' from ' Flagship St
ore'
store_group.loc[["Clothing","Electronics"],"Flagship store"].sum()
```

Out[67]:

Out[67]:

3409559.27

(9) What was the total amount earned from "Male" customers under the "Electronics" category?

In [68]:

```
gender_group = round(customer_final.pivot_table(index = "prod_cat",columns="Gender", values="total_amt", aggfunc='sum'),2)
```

In [69]:

```
gender_group
```

Out[69]:

Gender	F	M
prod_cat		
Bags	2079618.84	2046722.99
Books	6174590.82	6645972.77
Clothing	3026750.80	3224079.50
Electronics	5019354.21	5711351.62
Footwear	3203155.21	3020200.37
Home and kitchen	4133702.23	4305169.51

In [70]:

```
male_earning = gender_group.loc["Electronics","M"]
```

In [71]:

```
print("The total amount earned from Male customers under the Electronics category is",male_earning)
```

The total amount earned from Male customers under the Electronics category is 5711351.62

(10) How many customers have more than 10 unique transactions, after removing all transactions which have any negative amounts?

In [72]:

```
#creating a new dataframe that does not contain transactions with negative values
pos_trans = customer_final.loc[customer_final["total_amt"]>0,:]
```

In [73]:

```
pos_trans
```

Out[73]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	proc
10	29258453508	270384	2014-02-20	5	3	5	1497	785.925	8270.925	e-Shop	Electr
11	25455265351	267750	2014-02-20	12	6	3	1360	428.400	4508.400	e-Shop	Home kit
12	1571002198	275023	2014-02-20	6	5	4	587	246.540	2594.540	e-Shop	B

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	proc
14	36554696014	269345	2014-02-20	3	5	3	1253	394.695	4153.695	e-Shop	B
15	56814940239	268799	2014-02-20	7	5	5	368	193.200	2033.200	e-Shop	B
...	...	...	...	...	...	...	...	...	...	...	
23048	94340757522	274550	2011-01-25	12	5	1	1264	132.720	1396.720	e-Shop	B
23049	89780862956	270022	2011-01-25	4	1	1	677	71.085	748.085	e-Shop	Clo
23050	85115299378	271020	2011-01-25	2	6	4	1052	441.840	4649.840	MBR	Home kit
23051	72870271171	270911	2011-01-25	11	5	3	1142	359.730	3785.730	TeleShop	B
23052	77960931771	271961	2011-01-25	11	5	1	447	46.935	493.935	TeleShop	B

20876 rows × 16 columns

In [74]:

```
# creating a dataframe that contains unique transactions
unique_trans = pos_trans.groupby(['customer_Id','prod_cat','prod_subcat'])['transaction_id'].count().reset_index()
```

In [75]:

```
unique_trans
```

Out[75]:

	customer_Id	prod_cat	prod_subcat	transaction_id
0	266783	Books	Non-Fiction	1
1	266783	Clothing	Mens	2
2	266783	Footwear	Mens	1
3	266784	Books	Fiction	1
4	266784	Books	Non-Fiction	1
...	...	...	...	...
19273	275264	Books	Non-Fiction	1
19274	275264	Home and kitchen	Tools	1
19275	275265	Bags	Mens	1
19276	275265	Books	Academic	1
19277	275265	Home and kitchen	Furnishing	1

19278 rows × 4 columns

In [76]:

```
# now finding the customers which have unique transactions greater than 10
unique_trans_count = unique_trans.groupby('customer_Id')['transaction_id'].count().reset_index()
```

In [77]:

```
unique_trans_count.head()
```

Out[77]:

	customer_Id	transaction_id

	customer_Id	transaction_id
0	266783	3
1	266784	3
2	266785	5
3	266788	4
4	266794	8

In [78]:

```
unique_trans_count[unique_trans_count['transaction_id'] > 10]
```

Out[78]:

customer_Id	transaction_id

In [79]:

```
print('There are no unique transactions greater than 10')
```

```
There are no unique transactions greater than 10
```

(11) For all customers aged between 25-35, find out:

(a) What was the total amount spent for 'Electronics' and 'Books' product categories?

Adding new column 'age'

In [80]:

```
now = pd.Timestamp('now')
customer_final['DOB'] = pd.to_datetime(customer_final['DOB'], format='%m%d%y')     # 1
customer_final['DOB'] = customer_final['DOB'].where(customer_final['DOB'] < now, customer
_final['DOB'] -  np.timedelta64(100, 'Y'))    # 2
customer_final['AGE'] = (now - customer_final['DOB']).astype('<m8[Y]')
```

In [81]:

```
customer_final.head()
```

Out[81]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
3	93274880719	271509	2014-02-24	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home and kitchen
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Books

as we have to deal with customers aged between 25-35, so creating new column 'Age_cat'

In [82]:

```
customer_final['Age_cat'] = pd.cut(customer_final['AGE'],bins=[24,35,46,57],labels=['25-35','36-46','47-57'],include_lowest=True)
```

In [83]:

```
customer_final.head()
```

Out[83]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
3	93274880719	271509	2014-02-24	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home and kitchen
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Books

In [84]:

```
# grouping the dataframe 'customer_final' on the basis of 'Age_cat' and 'prod_cat'
customer_25_35 = customer_final.groupby(['Age_cat','prod_cat'])['total_amt'].sum()
```

In [85]:

```
customer_25_35
```

Out[85]:

```
Age_cat  prod_cat
25-35    Bags                1602196.960
         Books               4905583.410
         Clothing            2693713.750
         Electronics         4407137.800
         Footwear            2455791.780
         Home and kitchen    3346553.275
36-46    Bags                2020725.655
         Books               6377315.230
         Clothing            2898137.645
         Electronics         4981580.890
         Footwear            2951376.545
         Home and kitchen    4137682.445
47-57    Bags                 503419.215
         Books               1549693.990
         Clothing             659286.095
         Electronics         1341987.140
         Footwear             818736.490
         Home and kitchen     960376.495
Name: total_amt, dtype: float64
```

In [86]:

```
customer_25_35.loc['25-35',['Books','Electronics']]
```

Out[86]:

```
Age_cat  prod_cat
25-35    Books          4905583.41
         Electronics    4407137.80
Name: total_amt, dtype: float64
```

In [87]:

```
print("Total amount spent on 'Electronics' and 'Books' product categories is",
```

```
      customer_25_35.loc['25-35',['Books','Electronics']].sum().round(2))
```

```
Total amount spent on 'Electronics' and 'Books' product categories is 9312721.21
```

(b) What was the total amount spent by these customers between 1st Jan 2014 to 1st Mar 2014?

In [88]:

```
customer_final.head()
```

Out[88]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
0	80712190438	270351	2014-02-28	1	1	-5	-772	405.300	-4265.300	e-Shop	Clothing
1	29258453508	270384	2014-02-27	5	3	-5	-1497	785.925	-8270.925	e-Shop	Electronics
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
3	93274880719	271509	2014-02-24	11	6	-3	-1363	429.345	-4518.345	e-Shop	Home and kitchen
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Books

In [89]:

```
# filtering out data that belongs to the 'age_cat' = 25-35
customer_total_amount_25_35 = customer_final[customer_final['Age_cat']=='25-35']
```

In [90]:

```
customer_total_amount_25_35.head()
```

Out[90]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_cat
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Books
11	25455265351	267750	2014-02-20	12	6	3	1360	428.400	4508.400	e-Shop	Home and kitchen
13	43134751727	268487	2014-02-20	3	2	-1	-611	64.155	-675.155	e-Shop	Footwear
17	25963520987	274829	2014-02-20	4	4	3	502	158.130	1664.130	Flagship store	Bags

In [91]:

```
# getting all the data with transaction date between 1st Jan 2014 to 1st Mar 2014?
total_amount = customer_total_amount_25_35[(customer_total_amount_25_35['tran_date'] >='
2014-01-01') & (customer_total_amount_25_35['tran_date'] <='2014-03-01')]
```

In [92]:

```
total_amount
```

Out[92]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Tax	total_amt	Store_type	prod_c
2	51750724947	273420	2014-02-24	6	5	-2	-791	166.110	-1748.110	TeleShop	Bool
4	51750724947	273420	2014-02-23	6	5	-2	-791	166.110	-1748.110	TeleShop	Bool
11	25455265351	267750	2014-02-20	12	6	3	1360	428.400	4508.400	e-Shop	Hom ar kitche
13	43134751727	268487	2014-02-20	3	2	-1	-611	64.155	-675.155	e-Shop	Footwe
17	25963520987	274829	2014-02-20	4	4	3	502	158.130	1664.130	Flagship store	Bag
...	...	...	...	...	...	...	...	...	...	...	
1051	32889219128	269536	2014-01-01	10	5	5	1423	747.075	7862.075	e-Shop	Bool
1054	42711619809	271701	2014-01-01	1	2	5	336	176.400	1856.400	MBR	Footwe
1059	67088172893	271877	2014-01-01	1	1	1	902	94.710	996.710	e-Shop	Clothir
1061	63635040022	268886	2014-01-01	3	2	5	652	342.300	3602.300	e-Shop	Footwe
1063	69368153122	273627	2014-01-01	7	5	3	311	97.965	1030.965	TeleShop	Bool

271 rows × 18 columns

In [93]:

```
print('The total amount spent by customers aged 25-35 between 1st Jan 2014 to 1st Mar 201
4 is',
      total_amount['total_amt'].sum())
```

```
The total amount spent by customers aged 25-35 between 1st Jan 2014 to 1st Mar 2014 is 60
2196.27
```