In [6]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
import seaborn as sns
```

In [3]:
```
#Defining the Business Problem
#1.What are the best selling products and categories?
#2.What is the average order value?
#3.What is the customer retention rate?
#4.What is the customer acquisition cost?
#5.Which marketing channels drive the most traffic and sales?
```

In [46]:
```
df1=pd.read_csv('Customer.csv')
df1.head()
df2=pd.read_csv('Transactions.csv')
df2.head()
df3=pd.read_csv('prod_cat_info.csv')
df3.head()
```

Out[46]:

	prod_cat_code	prod_cat	prod_sub_cat_code	prod_subcat
0	1	Clothing	4	Mens
1	1	Clothing	1	Women
2	1	Clothing	3	Kids
3	2	Footwear	1	Mens
4	2	Footwear	3	Women

In [47]:
```
df1.head()
```

Out[47]:

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

In [48]:
```
df2.head()
```

Out[48]:

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	Tax	Tc
0	80712190438	270351	28-02-2014	1	1	-5	-772	3860	405.300	

	transaction_id	cust_id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	Tax	Tc
1	29258453508	270384	27-02-2014	5	3	-5	-1497	7485	785.925	
2	51750724947	273420	24-02-2014	6	5	-2	-791	1582	166.110	
3	93274880719	271509	24-02-2014	11	6	-3	-1363	4089	429.345	
4	51750724947	273420	23-02-2014	6	5	-2	-791	1582	166.110	

In [49]:
```
df2=df2.rename(columns={'cust_id':'customer_Id'})
```

In [50]:
```
#Merging Entire dataset
```

In [51]:
```
df4=pd.merge(df2,df1,how='left',on='customer_Id')
```

In [52]:
```
df4.head()
```

Out[52]:

	transaction_id	customer_Id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	Ta
0	80712190438	270351	28-02-2014	1	1	-5	-772	3860	405.30
1	29258453508	270384	27-02-2014	5	3	-5	-1497	7485	785.92
2	51750724947	273420	24-02-2014	6	5	-2	-791	1582	166.11
3	93274880719	271509	24-02-2014	11	6	-3	-1363	4089	429.34
4	51750724947	273420	23-02-2014	6	5	-2	-791	1582	166.11

In [53]:
```
E_commerce_data=pd.merge(df4,df3,how='left',on='prod_cat_code')
E_commerce_data.head()
```

Out[53]:

	transaction_id	customer_Id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	Ta
0	80712190438	270351	28-02-2014	1	1	-5	-772	3860	405.30
1	80712190438	270351	28-02-2014	1	1	-5	-772	3860	405.30
2	80712190438	270351	28-02-2014	1	1	-5	-772	3860	405.30
3	29258453508	270384	27-02-2014	5	3	-5	-1497	7485	785.92
4	29258453508	270384	27-02-2014	5	3	-5	-1497	7485	785.92

In [54]:
```
#checking info about dataset
```

In [55]:
```
print(E_commerce_data.shape)
print(E_commerce_data.info())
print(E_commerce_data.describe())
```

```
(99293, 18)
<class 'pandas.core.frame.DataFrame'>
Int64Index: 99293 entries, 0 to 99292
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   transaction_id      99293 non-null  int64  
 1   customer_Id         99293 non-null  int64  
 2   tran_date           99293 non-null  object 
 3   prod_subcat_code    99293 non-null  int64  
 4   prod_cat_code       99293 non-null  int64  
 5   Qty                 99293 non-null  int64  
 6   Rate                99293 non-null  int64  
 7   Total Amount        99293 non-null  int64  
 8   Tax                 99293 non-null  float64
 9   Total_amt_with_tax  99293 non-null  float64
 10  Store_type          99293 non-null  object 
 11  DOB                 99293 non-null  object 
 12  Gender              99253 non-null  object 
 13  city_code           99257 non-null  float64
 14  city_name           99257 non-null  object 
 15  prod_cat            99293 non-null  object 
 16  prod_sub_cat_code   99293 non-null  int64  
 17  prod_subcat         99293 non-null  object 
dtypes: float64(3), int64(8), object(7)
memory usage: 14.4+ MB
None
       transaction_id    customer_Id  prod_subcat_code  prod_cat_code  \
count    9.929300e+04   99293.000000      99293.000000   99293.000000   
mean     5.007320e+10  271030.010635          6.796894       4.003243   
```

```
std      2.899361e+10    2429.333624          3.609439       1.563991
min      3.268991e+06  266783.000000          1.000000       1.000000
25%      2.492150e+10  268956.000000          4.000000       3.000000
50%      5.011083e+10  270982.000000          7.000000       5.000000
75%      7.528121e+10  273120.000000         10.000000       5.000000
max      9.998755e+10  275265.000000         12.000000       6.000000

                Qty          Rate  Total Amount           Tax  \
count  99293.000000  99293.000000  99293.000000  99293.000000
mean       2.438017    637.919884   2370.226058    248.873736
std        2.260726    621.576326   1781.089736    187.014422
min       -5.000000  -1499.000000     70.000000      7.350000
25%        1.000000    313.000000    936.000000     98.280000
50%        3.000000    713.000000   1904.000000    199.920000
75%        4.000000   1109.000000   3495.000000    366.975000
max        5.000000   1500.000000   7500.000000    787.500000

       Total_amt_with_tax     city_code  prod_sub_cat_code
count        99293.000000  99257.000000       99293.000000
mean          2114.616420      5.467221           6.806985
std           2502.306768      2.859343           3.615952
min          -8270.925000      1.000000           1.000000
25%            762.450000      3.000000           4.000000
50%           1761.370000      5.000000           7.000000
75%           3585.725000      8.000000          10.000000
max           8287.500000     10.000000          12.000000
```

In [56]:

```
# Check for missing values
print(E_commerce_data.isnull().sum())
```

```
transaction_id         0
customer_Id            0
tran_date              0
prod_subcat_code       0
prod_cat_code          0
Qty                    0
Rate                   0
Total Amount           0
Tax                    0
Total_amt_with_tax     0
Store_type             0
DOB                    0
Gender                40
city_code             36
city_name             36
prod_cat               0
prod_sub_cat_code      0
prod_subcat            0
dtype: int64
```

In [29]:

```
E_commerce_data[E_commerce_data['Gender'].isnull()]
```

Out[29]:

	transaction_id	customer_Id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	
7582	85496594077	267199	27-11-2013	1	1	-4	-366	1464	15
7583	85496594077	267199	27-11-2013	1	1	-4	-366	1464	15

	transaction_id	customer_Id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	
7584	85496594077	267199	27-11-2013	1	1	-4	-366	1464	15
7841	85496594077	267199	24-11-2013	1	1	4	366	1464	15
7842	85496594077	267199	24-11-2013	1	1	4	366	1464	15
7843	85496594077	267199	24-11-2013	1	1	4	366	1464	15
16680	5723163001	267199	15-08-2013	6	5	1	244	244	2
16681	5723163001	267199	15-08-2013	6	5	1	244	244	2
16682	5723163001	267199	15-08-2013	6	5	1	244	244	2
16683	5723163001	267199	15-08-2013	6	5	1	244	244	2
16684	5723163001	267199	15-08-2013	6	5	1	244	244	2
16685	5723163001	267199	15-08-2013	6	5	1	244	244	2
29637	55955314599	271626	20-03-2013	11	6	5	1039	5195	54
29638	55955314599	271626	20-03-2013	11	6	5	1039	5195	54
29639	55955314599	271626	20-03-2013	11	6	5	1039	5195	54
29640	55955314599	271626	20-03-2013	11	6	5	1039	5195	54
37483	51951874983	271626	20-12-2012	10	5	4	845	3380	35

	transaction_id	customer_Id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	
37484	51951874983	271626	20-12-2012	10	5	4	845	3380	35
37485	51951874983	271626	20-12-2012	10	5	4	845	3380	35
37486	51951874983	271626	20-12-2012	10	5	4	845	3380	35
37487	51951874983	271626	20-12-2012	10	5	4	845	3380	35
37488	51951874983	271626	20-12-2012	10	5	4	845	3380	35
53114	78371516927	271626	29-06-2012	11	5	3	1022	3066	32
53115	78371516927	271626	29-06-2012	11	5	3	1022	3066	32
53116	78371516927	271626	29-06-2012	11	5	3	1022	3066	32
53117	78371516927	271626	29-06-2012	11	5	3	1022	3066	32
53118	78371516927	271626	29-06-2012	11	5	3	1022	3066	32
53119	78371516927	271626	29-06-2012	11	5	3	1022	3066	32
54991	24340761293	271626	05-06-2012	4	2	3	769	2307	24
54992	24340761293	271626	05-06-2012	4	2	3	769	2307	24
54993	24340761293	271626	05-06-2012	4	2	3	769	2307	24
68148	56749259881	267199	11-01-2012	7	5	4	1049	4196	44

	transaction_id	customer_Id	tran_date	prod_subcat_code	prod_cat_code	Qty	Rate	Total Amount	
68149	56749259881	267199	11-01-2012	7	5	4	1049	4196	44
68150	56749259881	267199	11-01-2012	7	5	4	1049	4196	44
68151	56749259881	267199	11-01-2012	7	5	4	1049	4196	44
68152	56749259881	267199	11-01-2012	7	5	4	1049	4196	44
68153	56749259881	267199	11-01-2012	7	5	4	1049	4196	44
98078	29245958438	267199	07-02-2011	3	1	1	278	278	2
98079	29245958438	267199	07-02-2011	3	1	1	278	278	2
98080	29245958438	267199	07-02-2011	3	1	1	278	278	2

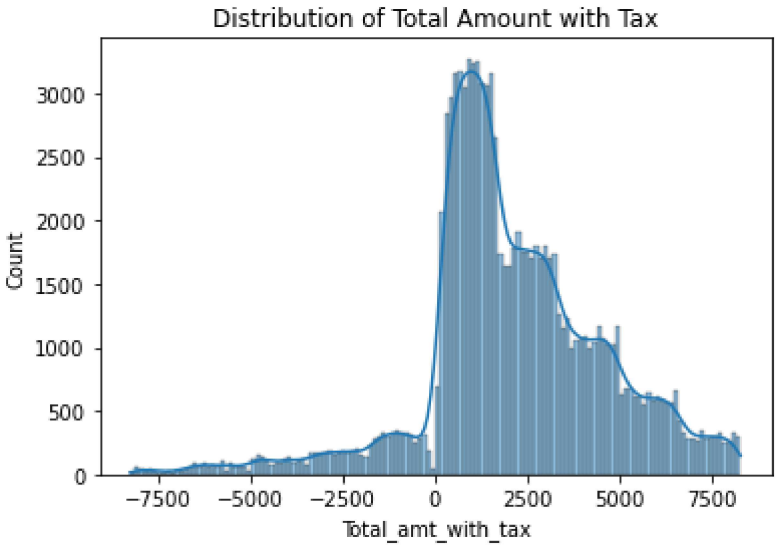
In [30]:
```
#Replacing null values with NA
```

In [32]:
```
E_commerce_data=E_commerce_data.fillna('Na',inplace=True)
```

In [58]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

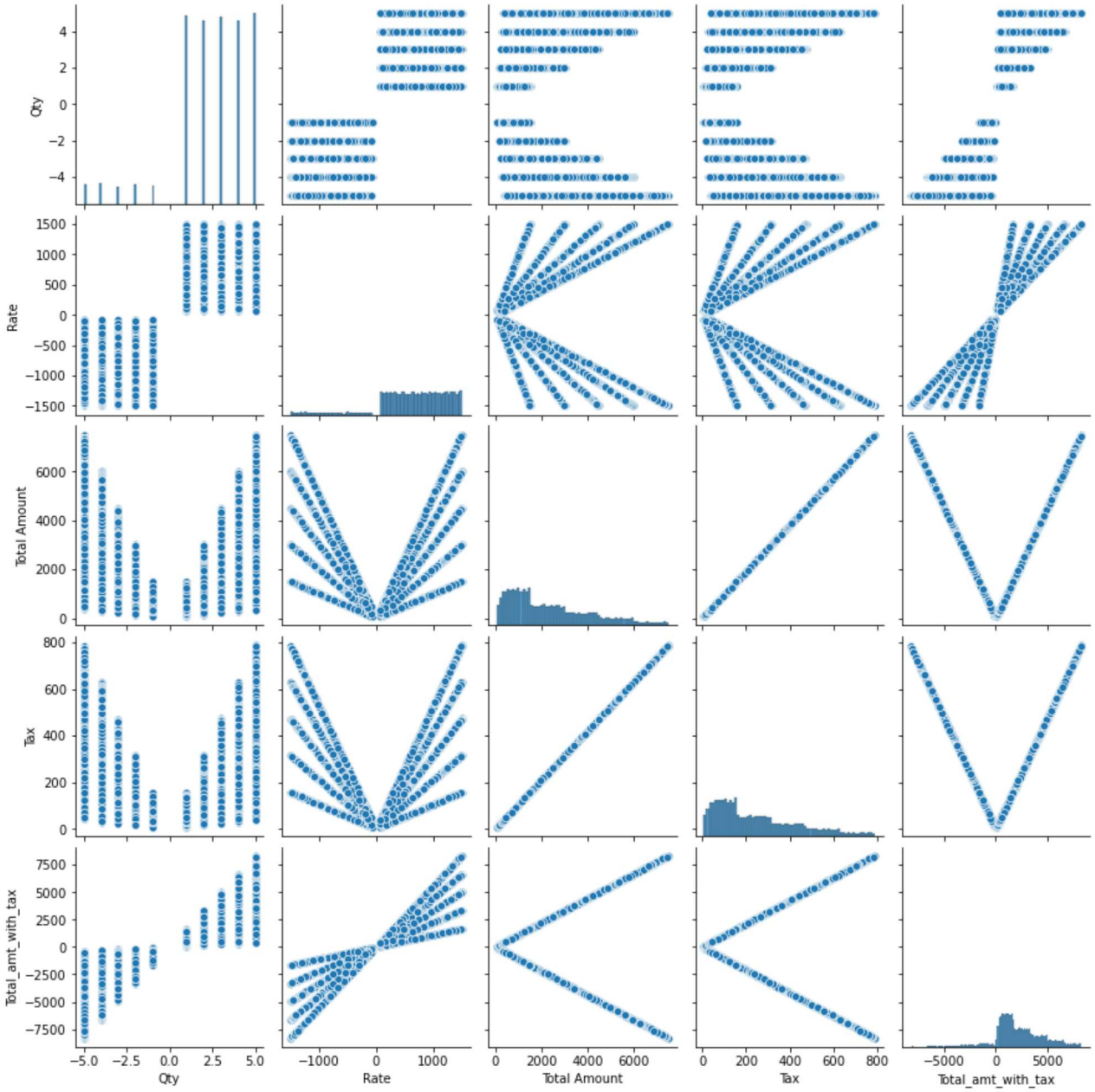
In [36]:
```
# Explore the distribution of each numerical variable
```

In [63]:
```
sns.histplot(E_commerce_data['Total_amt_with_tax'], kde=True)
plt.title('Distribution of Total Amount with Tax')
plt.show()
```



In [64]:

```
# Explore the relationship between numerical variables
sns.pairplot(E_commerce_data[['Qty', 'Rate', 'Total Amount', 'Tax', 'Total_amt_with_tax
plt.show()
```

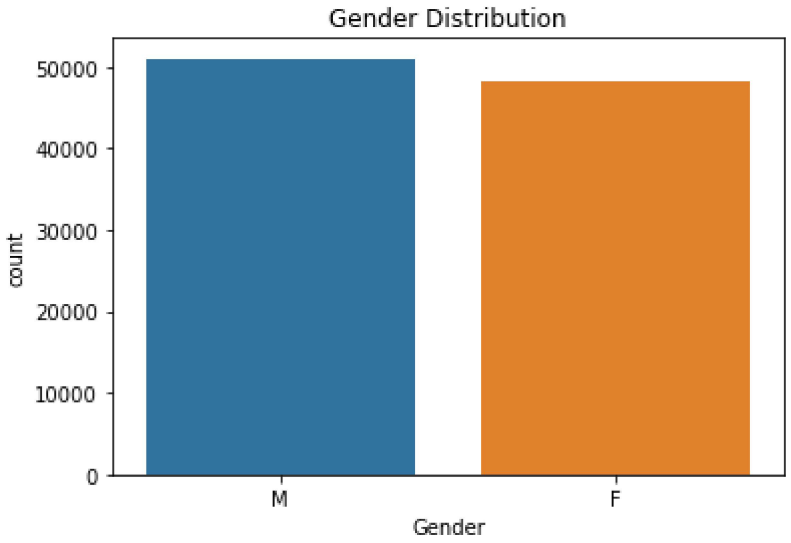


In [65]:

```
# Explore the relationship between categorical variables
#4. Which marketing channels drive the most traffic and sales?
sns.countplot(x='Store_type', data=E_commerce_data)
plt.title('Store Type Distribution')
plt.show()
```

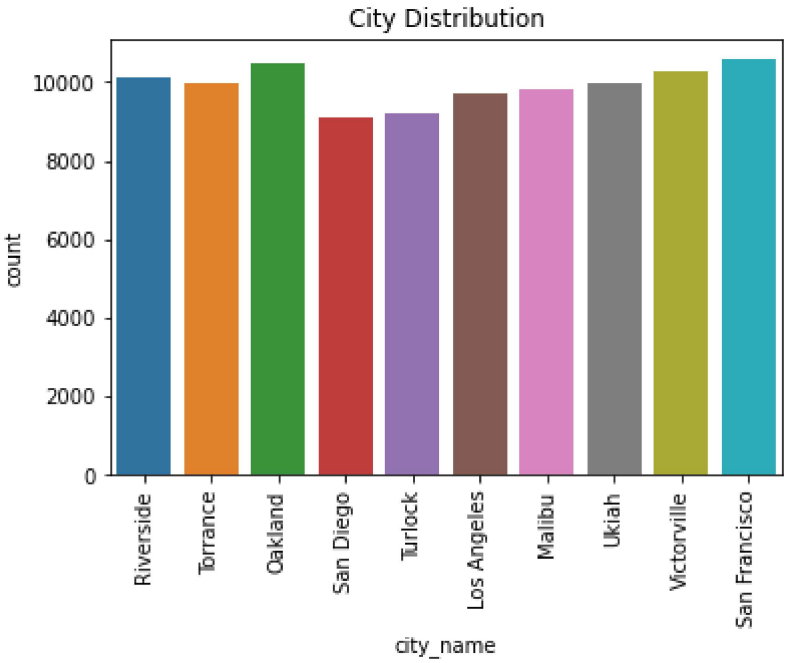
In [66]:

```
sns.countplot(x='Gender', data=E_commerce_data)
plt.title('Gender Distribution')
plt.show()
```



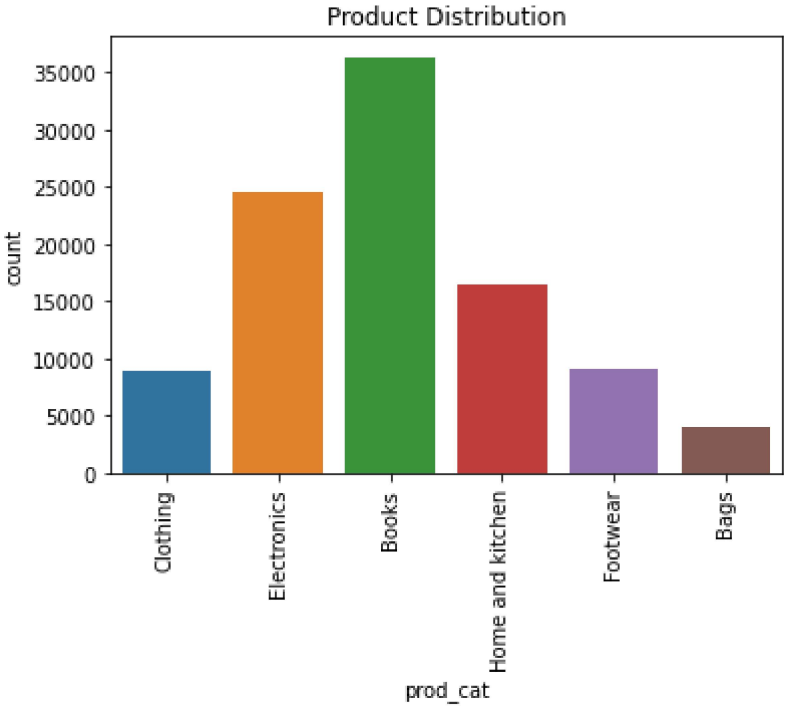
In [67]:

```
sns.countplot(x='city_name', data=E_commerce_data)
plt.title('City Distribution')
plt.xticks(rotation=90)
plt.show()
```



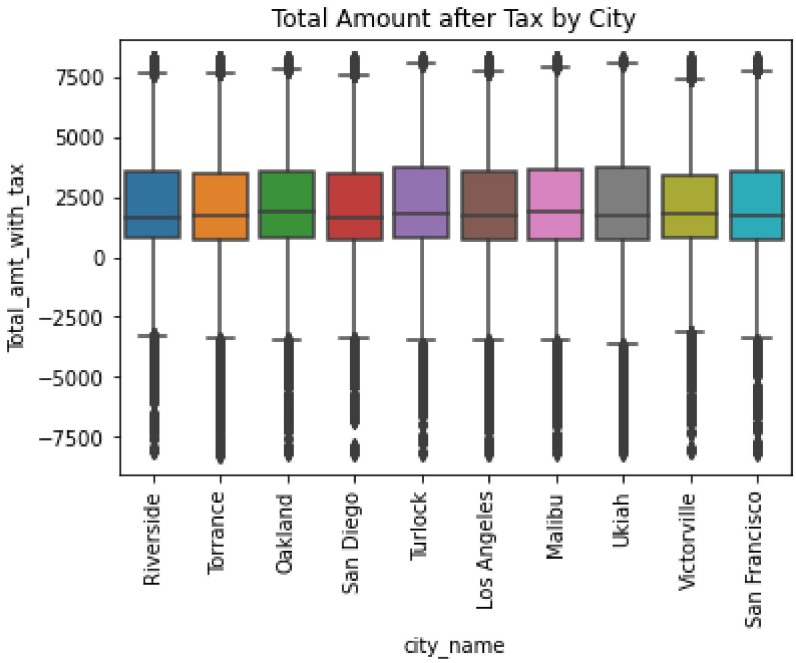
In [68]:

```
sns.countplot(x='prod_cat', data=E_commerce_data)
plt.title('Product Distribution')
plt.xticks(rotation=90)
plt.show()
```



In [71]:

```
sns.boxplot(x='city_name', y='Total_amt_with_tax', data=E_commerce_data)
plt.title('Total Amount after Tax by City')
plt.xticks(rotation=90)
plt.show()
```


Total Amount after Tax by City
Total_amt_with_tax
7500
5000
2500
0
−2500
−5000
−7500
Riverside
Torrance
Oakland
San Diego
Turlock
Los Angeles
Malibu
Ukiah
Victorville
San Francisco
city_name

In [ ]: