In [1]:
```
#setting up the working directory
import os
cwd=os.getcwd()
print(cwd)
```

```
C:\Users\Dell
```

In [2]:
```
os.chdir("C:\\Users\\Dell\\3D Objects\\one drive\\OneDrive\\Documents\\datasets")
```

In [29]:
```
#importing the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statistics import mean
import seaborn as sns
import plotly.express as px
```

In [4]:
```
data=pd.read_csv("train.csv")
```

# data exploration

In [5]:
```
data.head()
```

Out[5]:

| | User_ID | Product_ID | Gender | Age | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Category_1 | Product_Cate |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1000001 | P00069042 | F | 0-17 | 10 | A | 2 | 0 | 3 | |
| **1** | 1000001 | P00248942 | F | 0-17 | 10 | A | 2 | 0 | 1 | |
| **2** | 1000001 | P00087842 | F | 0-17 | 10 | A | 2 | 0 | 12 | |
| **3** | 1000001 | P00085442 | F | 0-17 | 10 | A | 2 | 0 | 12 | |
| **4** | 1000002 | P00285442 | M | 55+ | 16 | C | 4+ | 0 | 8 | |

In [6]:
```
#checking size of the data
data.shape
```

Out[6]:
```
(550068, 12)
```

In [7]:
```
#checking if there are any null values
data.isnull().sum()
```

Out[7]:
```
User_ID                           0
Product_ID                        0
Gender                            0
Age                               0
Occupation                        0
City_Category                     0
Stay_In_Current_City_Years        0
Marital_Status                    0
Product_Category_1                0
Product_Category_2           173638
Product_Category_3           383247
Purchase                          0
dtype: int64
```

In [8]:
```
data=data.dropna()
```

In [9]:
```
data.dtypes
```

Out[9]:
```
User ID                          int64
```

```
Product_ID                    object
Gender                        object
Age                           object
Occupation                     int64
City_Category                 object
Stay_In_Current_City_Years    object
Marital_Status                 int64
Product_Category_1             int64
Product_Category_2           float64
Product_Category_3           float64
Purchase                       int64
dtype: object
```

In [21]:
```
data.describe()
```

Out[21]:

| | User_ID | Occupation | Marital_Status | Product_Category_1 | Product_Category_2 | Product_Category_3 | Purchase | New_Ag |
|---|---|---|---|---|---|---|---|---|
| count | 1.668210e+05 | 166821.000000 | 166821.000000 | 166821.000000 | 166821.000000 | 166821.000000 | 166821.000000 | 166821.00000 |
| mean | 1.003037e+06 | 8.178886 | 0.402839 | 2.742766 | 6.896871 | 12.668243 | 11658.114980 | 33.93345 |
| std | 1.732907e+03 | 6.487522 | 0.490470 | 2.573969 | 4.500288 | 4.125338 | 5082.287959 | 10.81890 |
| min | 1.000001e+06 | 0.000000 | 0.000000 | 1.000000 | 2.000000 | 3.000000 | 185.000000 | 8.50000 |
| 25% | 1.001523e+06 | 2.000000 | 0.000000 | 1.000000 | 2.000000 | 9.000000 | 7869.000000 | 30.50000 |
| 50% | 1.003101e+06 | 7.000000 | 0.000000 | 1.000000 | 6.000000 | 14.000000 | 11756.000000 | 30.50000 |
| 75% | 1.004480e+06 | 14.000000 | 1.000000 | 4.000000 | 10.000000 | 16.000000 | 15626.000000 | 40.50000 |
| max | 1.006040e+06 | 20.000000 | 1.000000 | 15.000000 | 16.000000 | 18.000000 | 23959.000000 | 55.00000 |

# checking the correlation between purchase and other columns

In [10]:
```
#checking the correlation between purchase and other columns
df_numerized=data
```

In [11]:
```
b=[]
for i in data["Age"]:
    if i=='55+':
        p=55
    else:
        a=i.split('-')
        p=mean(int(j) for j in a)
    b.append(p)
```

In [12]:
```
df_numerized["New_Age"]=np.array(b)
df_numerized.head()
```

Out[12]:

| | User_ID | Product_ID | Gender | Age | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Category_1 | Product_Cat |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1000001 | P00248942 | F | 0-17 | 10 | A | 2 | 0 | 1 | |
| 6 | 1000004 | P00184942 | M | 46-50 | 7 | B | 2 | 1 | 1 | |
| 13 | 1000005 | P00145042 | M | 26-35 | 20 | A | 1 | 1 | 1 | |
| 14 | 1000006 | P00231342 | F | 51-55 | 9 | A | 1 | 0 | 5 | |
| 16 | 1000006 | P0096642 | F | 51-55 | 9 | A | 1 | 0 | 2 | |

In [13]:
```
df_numerized=df_numerized.drop(["Age"],axis=1)
df_numerized.head()
```

Out[13]:

| | User_ID | Product_ID | Gender | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Category_1 | Product_Category |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1000001 | P00248942 | F | 10 | A | 2 | 0 | 1 | 6 |
| 6 | 1000004 | P00184942 | M | 7 | B | 2 | 1 | 1 | 8 |
| 13 | 1000005 | P00145042 | M | 20 | A | 1 | 1 | 1 | 2 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **14** | 1000006 | P00231342 | F | 9 | A | 1 | 0 | 5 | 8 |
| **16** | 1000006 | P0096642 | F | 9 | A | 1 | 0 | 2 | 3 |

In [14]:
```
df_numerized["User_ID"]=df_numerized["User_ID"].astype('object')
```

In [15]:
```
for col_name in df_numerized.columns:
    if(df_numerized[col_name].dtype == 'object'):
        df_numerized[col_name]= df_numerized[col_name].astype('category')
        df_numerized[col_name] = df_numerized[col_name].cat.codes

df_numerized.head()
```

Out[15]:

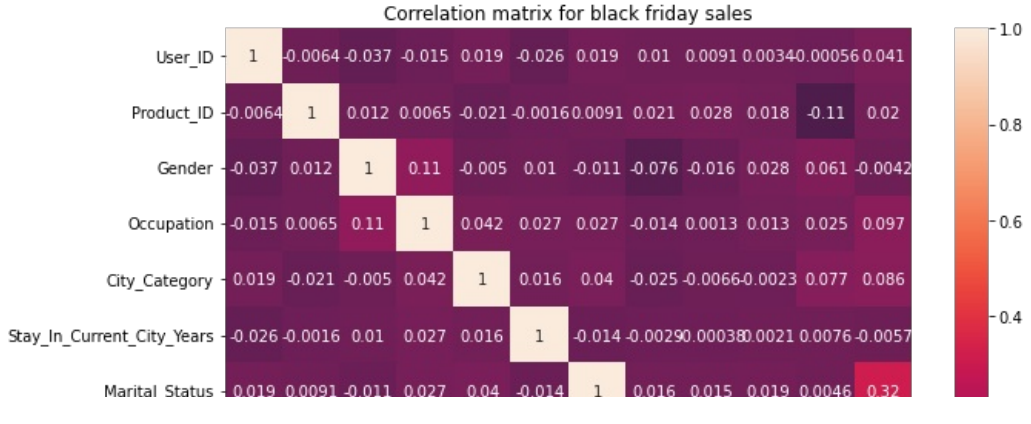
| | User_ID | Product_ID | Gender | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Category_1 | Product_Category_ |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 394 | 0 | 10 | 0 | 2 | 0 | 1 | 6 |
| **6** | 3 | 287 | 1 | 7 | 1 | 2 | 1 | 1 | 8 |
| **13** | 4 | 214 | 1 | 20 | 0 | 1 | 1 | 1 | 2 |
| **14** | 5 | 366 | 0 | 9 | 0 | 1 | 0 | 5 | 8 |
| **16** | 5 | 520 | 0 | 9 | 0 | 1 | 0 | 2 | 3 |

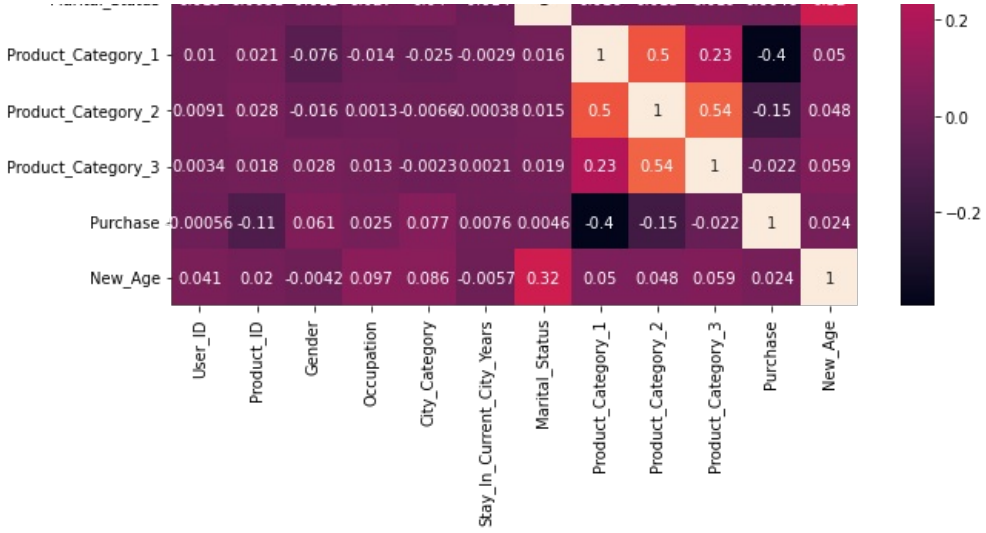
In [16]:
```
df_numerized.corr()
```

Out[16]:

| | User_ID | Product_ID | Gender | Occupation | City_Category | Stay_In_Current_City_Years | Marital_Status | Product_Ca |
|---|---|---|---|---|---|---|---|---|
| **User_ID** | 1.000000 | -0.006358 | -0.036817 | -0.014516 | 0.018853 | -0.026391 | 0.018670 | |
| **Product_ID** | -0.006358 | 1.000000 | 0.011904 | 0.006502 | -0.021432 | -0.001629 | 0.009090 | |
| **Gender** | -0.036817 | 0.011904 | 1.000000 | 0.111920 | -0.004953 | 0.010200 | -0.010872 | - |
| **Occupation** | -0.014516 | 0.006502 | 0.111920 | 1.000000 | 0.041711 | 0.026696 | 0.027368 | - |
| **City_Category** | 0.018853 | -0.021432 | -0.004953 | 0.041711 | 1.000000 | 0.016395 | 0.039678 | - |
| **Stay_In_Current_City_Years** | -0.026391 | -0.001629 | 0.010200 | 0.026696 | 0.016395 | 1.000000 | -0.014053 | - |
| **Marital_Status** | 0.018670 | 0.009090 | -0.010872 | 0.027368 | 0.039678 | -0.014053 | 1.000000 | |
| **Product_Category_1** | 0.010354 | 0.021317 | -0.076321 | -0.013682 | -0.024514 | -0.002906 | 0.015682 | |
| **Product_Category_2** | 0.009070 | 0.028008 | -0.016093 | 0.001336 | -0.006612 | -0.000382 | 0.014813 | |
| **Product_Category_3** | 0.003398 | 0.017570 | 0.028069 | 0.013263 | -0.002347 | 0.002093 | 0.019473 | |
| **Purchase** | -0.000564 | -0.108375 | 0.060852 | 0.025048 | 0.077344 | 0.007598 | 0.004603 | - |
| **New_Age** | 0.041168 | 0.020064 | -0.004205 | 0.097323 | 0.086229 | -0.005653 | 0.316897 | |

In [19]:
```
correlation_matrix = df_numerized.corr(method='pearson')

#sns.heatmap(correlation_matrix, annot = True)
fig, ax = plt.subplots(figsize=(10,8))

# Create a heatmap using the correlation matrix
sns.heatmap(correlation_matrix, annot=True, ax=ax)

# Set the title of the plot
plt.title("Correlation matrix for black friday sales")

# Display the plot
plt.show()
```






# data visualization

In [26]:
```
#no of users participated in black friday sales
data["User_ID"].nunique()
```

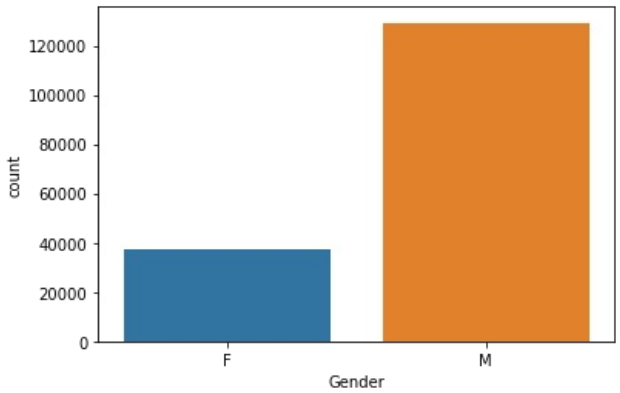
Out[26]: 5870

In [27]:
```
#no of products sold in black friday sales
data["Product_ID"].nunique()
```

Out[27]: 528

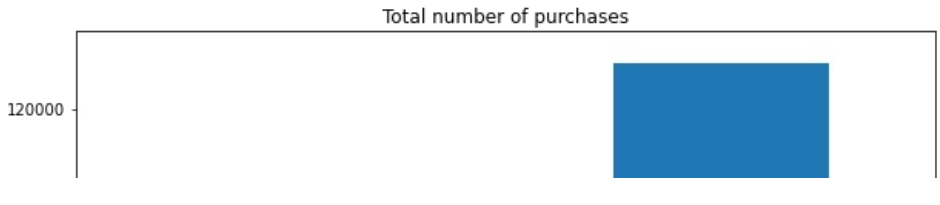
# Analysing by gender column

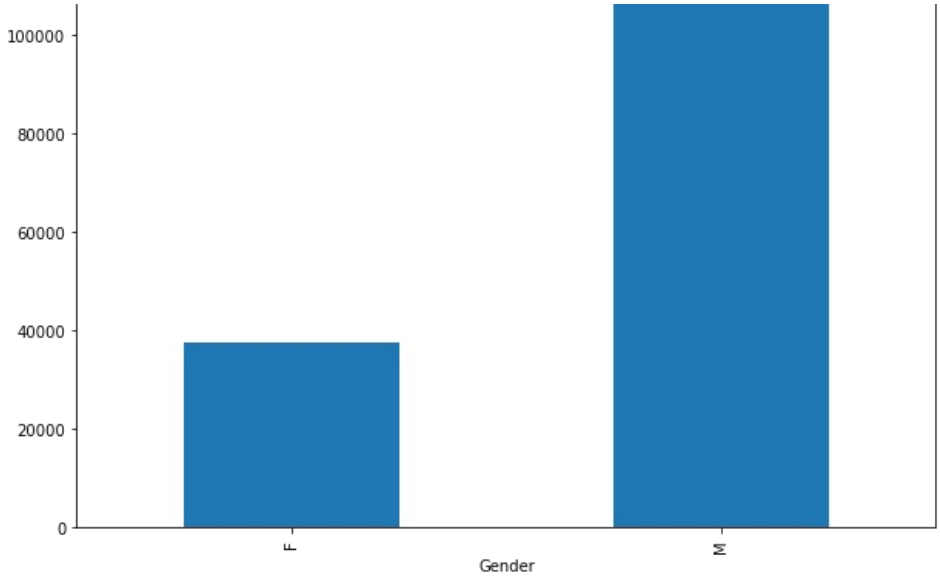
In [28]:
```
#no of men and women participated in black friday sales
sns.countplot(x="Gender",data=data)
plt.show()
```




In [35]:
```
#Total no of purchases by gender
data.groupby('Gender').size().plot(kind='bar',figsize=(10,8),title='Total number of purchases')
```

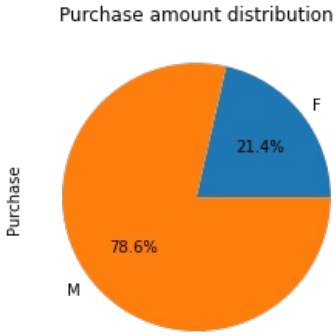
Out[35]: <AxesSubplot:title={'center':'Total number of purchases'}, xlabel='Gender'>





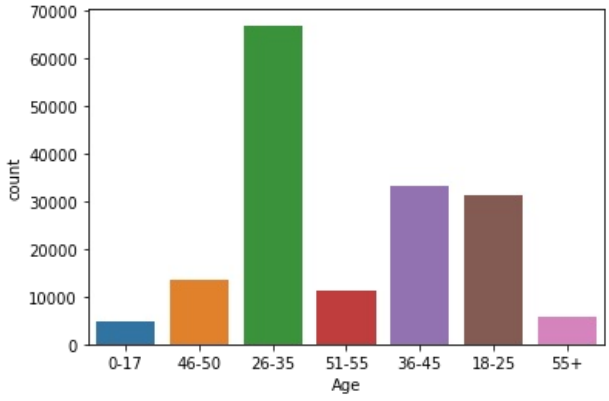

In [36]:
```
#Purchase distribution by gender
data.groupby('Gender').sum()['Purchase'].plot(kind='pie',autopct='%0.1f%%',title='Purchase amount distribution',t
```

Out[36]: <AxesSubplot:title={'center':'Purchase amount distribution'}, ylabel='Purchase'>



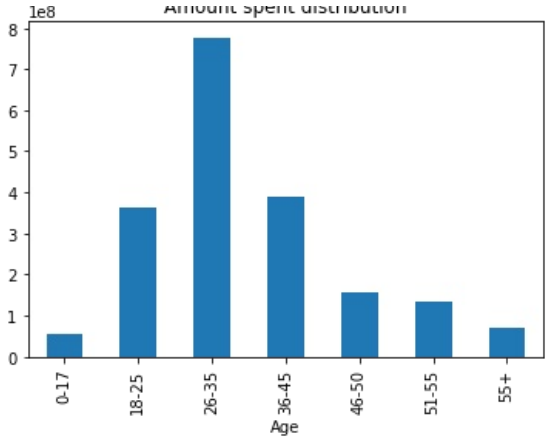

# Analysing by age column

In [38]:
```
#count of different ages of people participated in black friday sales
sns.countplot(x="Age",data=data)
plt.show()
```




In [71]:
```
#Amount spent by each age group in the sale
data.groupby('Age').sum()['Purchase'].plot(kind = 'bar',
                                           title = 'Amount spent distribution')
```

Out[71]: <AxesSubplot:title={'center':'Amount spent distribution'}, xlabel='Age'>

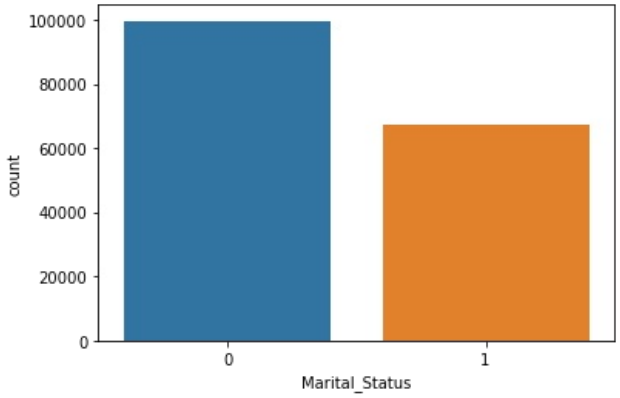



In [72]:
```
#people of age between 26-35 are purchasing more
```

# Analysing by marital status column

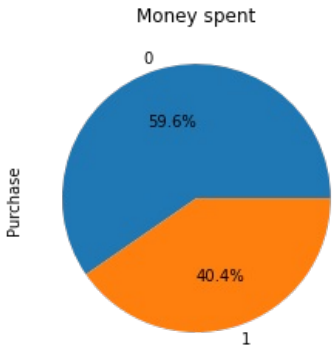
In [43]:
```
#count of married and unmarried people participated in black friday sales
sns.countplot(x="Marital_Status",data=data)
```

Out[43]: <AxesSubplot:xlabel='Marital_Status', ylabel='count'>




In [45]:
```
data.groupby('Marital_Status').sum()['Purchase'].plot(kind = 'pie',
                                                      autopct = '%0.1f%%',
                                                      figsize = (4,4),
                                                      title = 'Money spent')
```

Out[45]: <AxesSubplot:title={'center':'Money spent'}, ylabel='Purchase'>

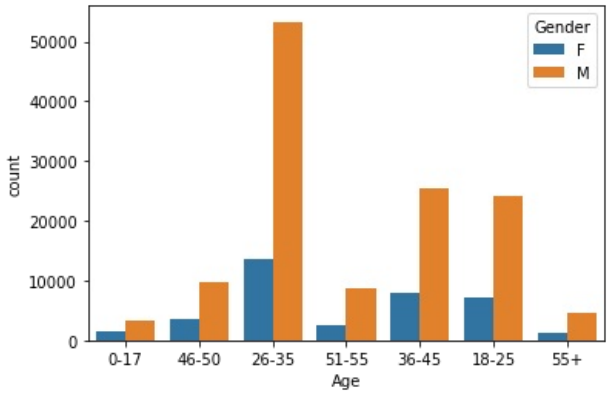



In [44]:
```
#More than 50% people are unmarried and 60% of the revenue comes from unmarried people
```

In [48]:
```
sns.countplot(x = 'Age',hue = 'Gender',data = data)
```

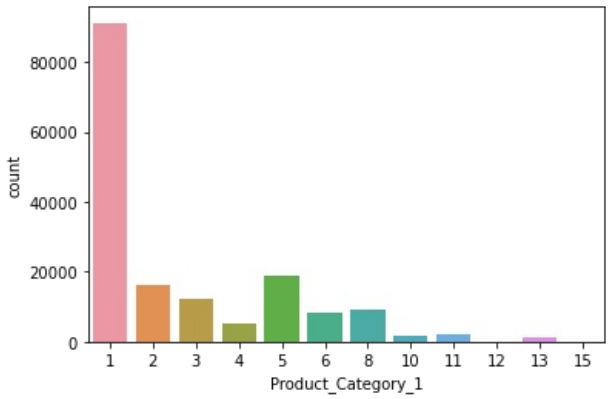
Out[48]: <AxesSubplot:xlabel='Age', ylabel='count'>

Out[48]: <AxesSubplot:xlabel='Age', ylabel='count'>



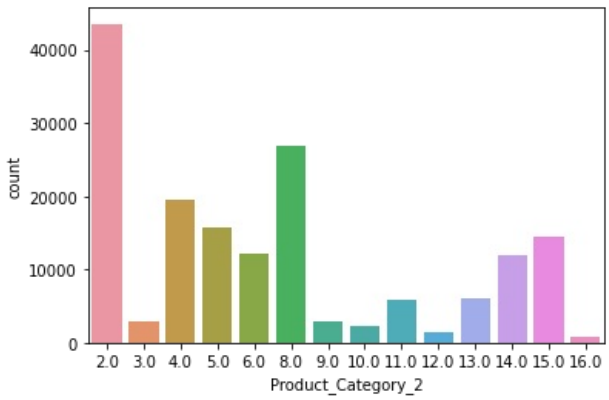

In [49]:
```
sns.countplot(x = 'Product_Category_1', data = data)
```

Out[49]: <AxesSubplot:xlabel='Product_Category_1', ylabel='count'>



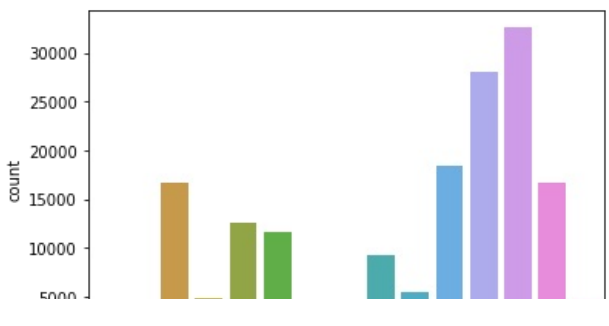

In [51]:
```
sns.countplot(x = 'Product_Category_2', data = data)
```

Out[51]: <AxesSubplot:xlabel='Product_Category_2', ylabel='count'>



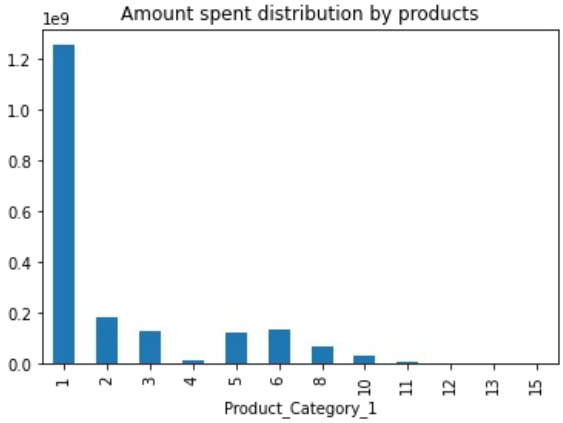

In [52]:
```
sns.countplot(x = 'Product_Category_3', data = data)
```

Out[52]: <AxesSubplot:xlabel='Product_Category_3', ylabel='count'>




In [56]:
```
data.groupby('Product_Category_1').sum()['Purchase'].plot(kind = 'bar',
                                                          title = 'Amount spent distribution by products')
```

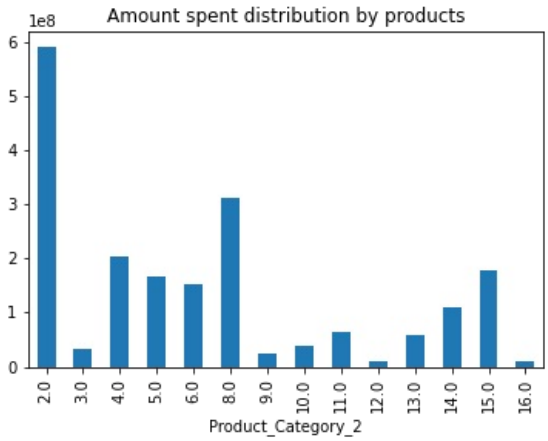
Out[56]: <AxesSubplot:title={'center':'Amount spent distribution by products'}, xlabel='Product_Category_1'>




In [ ]:
```
```

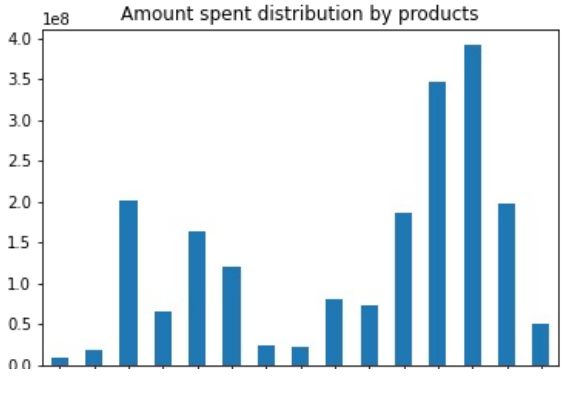
In [57]:
```
data.groupby('Product_Category_2').sum()['Purchase'].plot(kind = 'bar',
                                                          title = 'Amount spent distribution by products')
```

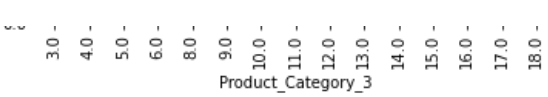
Out[57]: <AxesSubplot:title={'center':'Amount spent distribution by products'}, xlabel='Product_Category_2'>




In [58]:
```
data.groupby('Product_Category_3').sum()['Purchase'].plot(kind = 'bar',
                                                          title = 'Amount spent distribution by products')
```

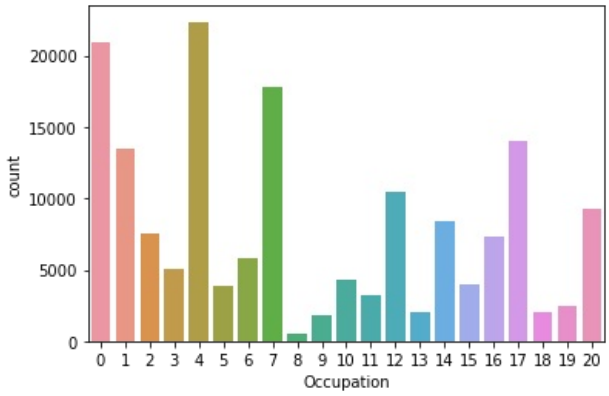
Out[58]: <AxesSubplot:title={'center':'Amount spent distribution by products'}, xlabel='Product_Category_3'>






In [53]:
```
sns.countplot(x = 'Occupation', data = data)
```

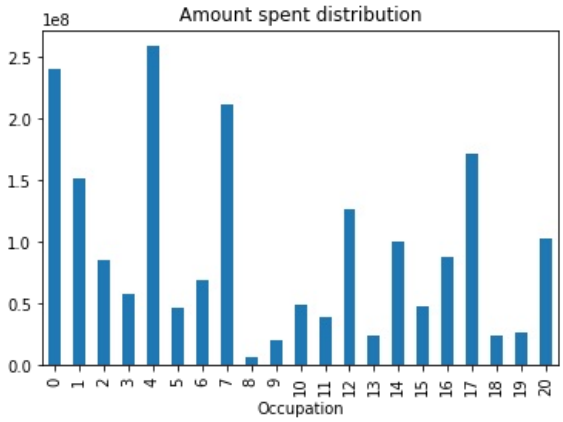
Out[53]: <AxesSubplot:xlabel='Occupation', ylabel='count'>




In [ ]:
```
#occupation 0,4,7,17 participated more in the sale
```

In [54]:
```
#amount spent distribution by occupation
data.groupby('Occupation').sum()['Purchase'].plot(kind = 'bar',
                                        title = 'Amount spent distribution by occupation')
```

Out[54]: <AxesSubplot:title={'center':'Amount spent distribution'}, xlabel='Occupation'>

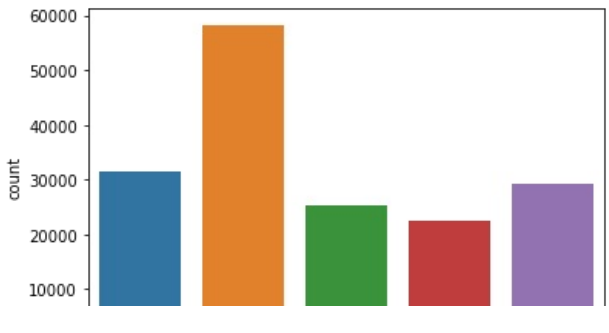


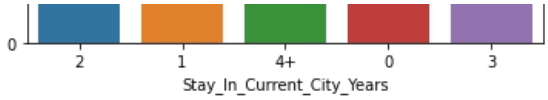

In [ ]:
```
#occupation 0,4,7,17 purchased more in the sale
```

# Analysing by Stay_In_Current_City_Years

In [59]:
```
sns.countplot(x="Stay_In_Current_City_Years",data=data)
```

Out[59]: <AxesSubplot:xlabel='Stay_In_Current_City_Years', ylabel='count'>

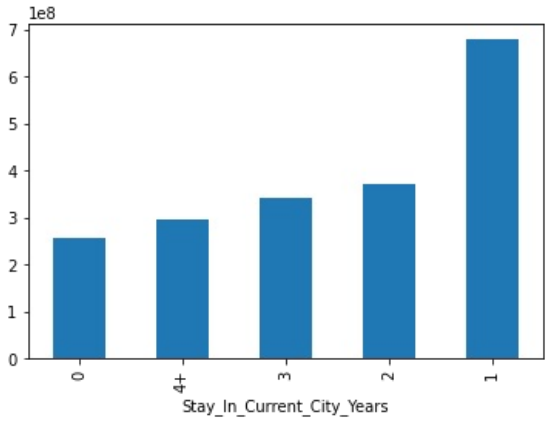





In [68]:
```
#people who lived 1 and 3 years participated more in sales
```

In [61]:
```
data.groupby('Stay_In_Current_City_Years').sum()['Purchase'].sort_values().plot(kind = 'bar')
```

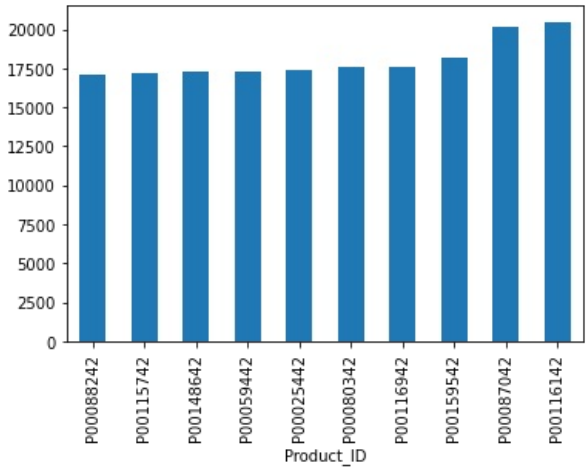
Out[61]: <AxesSubplot:xlabel='Stay_In_Current_City_Years'>




In [62]:
```
#people who lived for 1 year are purchasing more items
```

In [64]:
```
data.groupby('Product_ID').mean()['Purchase'].nlargest(10).sort_values().plot(kind = 'bar')
```

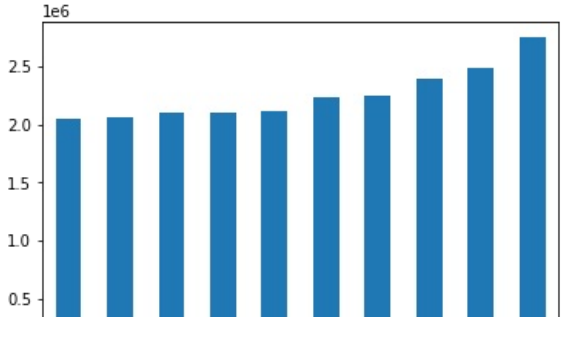
Out[64]: <AxesSubplot:xlabel='Product_ID'>




In [65]:
```
#product ids p00116142,p00087042 are being purchased more
```

In [66]:
```
data.groupby('User_ID').sum()['Purchase'].nlargest(10).sort_values().plot(kind = 'bar')
```

Out[66]: <AxesSubplot:xlabel='User_ID'>




In [67]:
```
#userids 1004277,1004448 are being purchased more
```

# conclusion regarding black friday sales

In [69]:
```
#Males of age between 18-45 are purchasing more number of products.

#Though the more number of purchases are made by males, the average money spent on each procuct by both males and

#The number of purchases and the total money spent is more for the age groups between 18-45 with 26-35 being the

#More than 50% people are unmarried and 60% of the revenue comes from unmarried people.

#People with occupations [4,0,7] are purchasing more items and [17,12,15] are purchasing expensive items.

#People who stayed for 1 year are purchasing more items. And everyone are purchasing more or less same price item

#Products of category [5,1,8] are being purchased more. Products of category [10,7,6] are most expensive ones.

#Product with ids [P00265242, P00110742, P00025442] are being purchased more and those with ids [P00086242,P00085
```

Out[69]:
```
'Males of age between 18-45 are purchasing more number of products.\n\nThough the more number of purchases are ma
de by males, the average money spent on each procuct by both males and females is same.\n\nThe number of purchase
s and the total money spent is more for the age groups between 18-45 with 26-35 being the most. Also the number o
f unique products available are more in the same age groups. The average money spent per product is more or less
same for all the age groups.\n\nMore than 50% people are unmarried and 60% of the revenue comes from unmarried pe
ople.\n\nPeople from City category B.\n\nPeople with occupations [4,0,7] are purchasing more items and [17,12,15]
are purchasing expensive items.\n\nPeople who stayed for 1 year are purchasing more items. And everyone are purch
asing more or less same price items.\n\nProducts of category [5,1,8] are being purchased more. Products of catego
ry [10,7,6] are most expensive ones.\n\nProduct with ids [P00265242, P00110742, P00025442] are being purchased mo
re and those with ids [P00086242,P00085342,P00200642] are the expensive ones compared to others.'
```

In [ ]:
```

```