In [3]:

```python
import pandas as pd #impoerting pandas library to help in computation
import seaborn as sns #for visualation of data in graphical format
```

In [4]:

```python
df=pd.read_csv("BlackFriday.csv") #df means dataframe
```

Dataset Walkthrough

In [6]:

```python
df
```

Out[6]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Produ
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	
...	...	...	...	...	...	...	...	...	...	
537572	1004737	P00193542	M	36-45	16	C	1	0	1	
537573	1004737	P00111142	M	36-45	16	C	1	0	1	
537574	1004737	P00345942	M	36-45	16	C	1	0	8	
537575	1004737	P00285842	M	36-45	16	C	1	0	5	
537576	1004737	P00118242	M	36-45	16	C	1	0	5	

537577 rows × 12 columns

In [6]:

```
df.info()#it gives the count of total rows,columns and data type of the values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 537577 entries, 0 to 537576
Data columns (total 12 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   User_ID                     537577 non-null  int64  
 1   Product_ID                  537577 non-null  object 
 2   Gender                      537577 non-null  object 
 3   Age                         537577 non-null  object 
 4   Occupation                  537577 non-null  int64  
 5   City_Category               537577 non-null  object 
 6   Stay_In_Current_City_Years  537577 non-null  object 
 7   Marital_Status              537577 non-null  int64  
 8   Product_Category_1          537577 non-null  int64  
 9   Product_Category_2          370591 non-null  float64
 10  Product_Category_3          164278 non-null  float64
 11  Purchase                    537577 non-null  int64  
dtypes: float64(2), int64(5), object(5)
memory usage: 49.2+ MB
```

In [7]:

```
df.isnull().sum()#isnull gives the is there is any null values or not and with sum() it gies the total no of null value
```

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            166986
Product_Category_3            373299
Purchase                           0
dtype: int64
```

In [8]:

```
del df["Product_Category_2"]# deleting two column which containing null values
del df["Product_Category_3"]
```

In [9]:

```
df
```

Out[9]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purch
0	1000001	P00069042	F	0-17	10	A	2	0	3	8
1	1000001	P00248942	F	0-17	10	A	2	0	1	15
2	1000001	P00087842	F	0-17	10	A	2	0	12	1
3	1000001	P00085442	F	0-17	10	A	2	0	12	1
4	1000002	P00285442	M	55+	16	C	4+	0	8	7
...	...	...	...	...	...	...	...	...	...	
537572	1004737	P00193542	M	36-45	16	C	1	0	1	11
537573	1004737	P00111142	M	36-45	16	C	1	0	1	19
537574	1004737	P00345942	M	36-45	16	C	1	0	8	8
537575	1004737	P00285842	M	36-45	16	C	1	0	5	7
537576	1004737	P00118242	M	36-45	16	C	1	0	5	6

537577 rows × 10 columns

Analyzing Columns

In [10]:

```
df.head() # by default head gives details of the 5 rows
```

Out[10]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

In [11]:

```
df["User_ID"].nunique()#nunique give all the unique values present in a particular column
```

Out[11]:

```
5891
```

In [12]:

```
df["User_ID"].nunique()
```

Out[12]:

```
5891
```

In [13]:

```
df["Gender"].unique()
```

Out[13]:

```
array(['F', 'M'], dtype=object)
```

In [14]:

```
df["Age"].unique()
```

Out[14]:

```
array(['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25'],
      dtype=object)
```

In [15]:

```
df["Occupation"].nunique()
```

Out[15]:

```
21
```

In [16]:

```
df["City_Category"].unique()
```

Out[16]:

```
array(['A', 'C', 'B'], dtype=object)
```

In [17]:

```
df["Stay_In_Current_City_Years"].unique()
```

Out[17]:

```
array(['2', '4+', '3', '1', '0'], dtype=object)
```

In [18]:

```python
df["Marital_Status"].unique()
```

Out[18]:

```
array([0, 1], dtype=int64)
```

In [19]:

```python
df["Product_Category_1"].nunique()
```

Out[19]:

```
18
```

In [20]:

```python
df["Purchase"].sum()
```

Out[20]:

```
5017668378
```

In [21]:

```python
for i in df.columns:#  Automating the process for each column to find the no of unique values
    print(i,df[i].nunique())
```

```
User_ID 5891
Product_ID 3623
Gender 2
Age 7
Occupation 21
City_Category 3
Stay_In_Current_City_Years 5
Marital_Status 2
Product_Category_1 18
Purchase 17959
```

Analyzing Gender

In [22]:

```python
s=0
for i in df['Gender']:#to find the total no of male
    if(i=='M'):
        s+=1
print(s)
```

```
405380
```

In [23]:

```python
s=0
for i in df['Gender']:#to find the total no of female
    if(i=='F'):
        s+=1
print(s)
```

```
132197
```

In [24]:

```
data=pd.DataFrame({'Ratio':[len(df[df['Gender']=='M']),len(df[df['Gender']=='F'])]},index=['Male','Female'])#creating a
```

In [25]:

```
data
```

Out[25]:

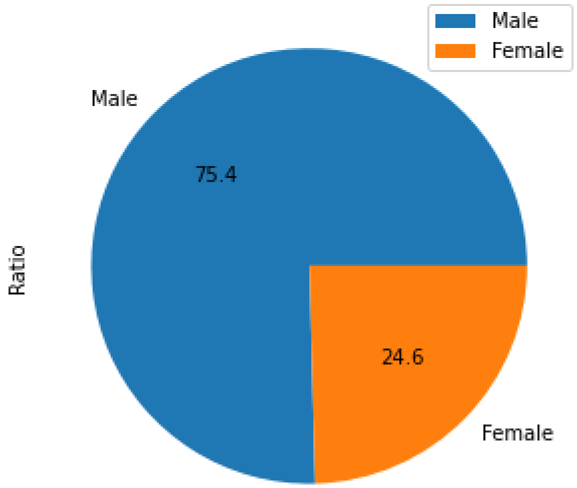
	Ratio
Male	405380
Female	132197

In [26]:

```
data.plot.pie(y='Ratio',figsize=(6,5),autopct="%.1f")#ploting a pie chart of male and female ratio
```

Out[26]:

```
<AxesSubplot:ylabel='Ratio'>
```

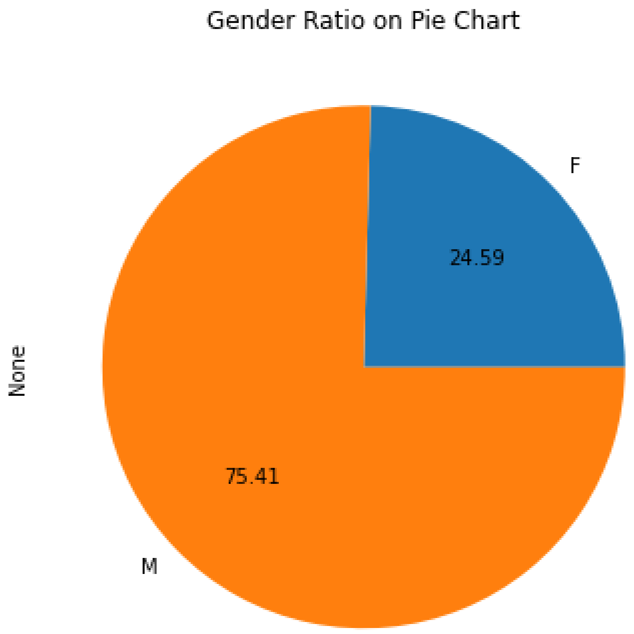


In [27]:

```
df.groupby('Gender').size().plot.pie(x='Ratio',figsize=(6,6),autopct="%.2f",title='Gender Ratio on Pie Chart')#this bet
```

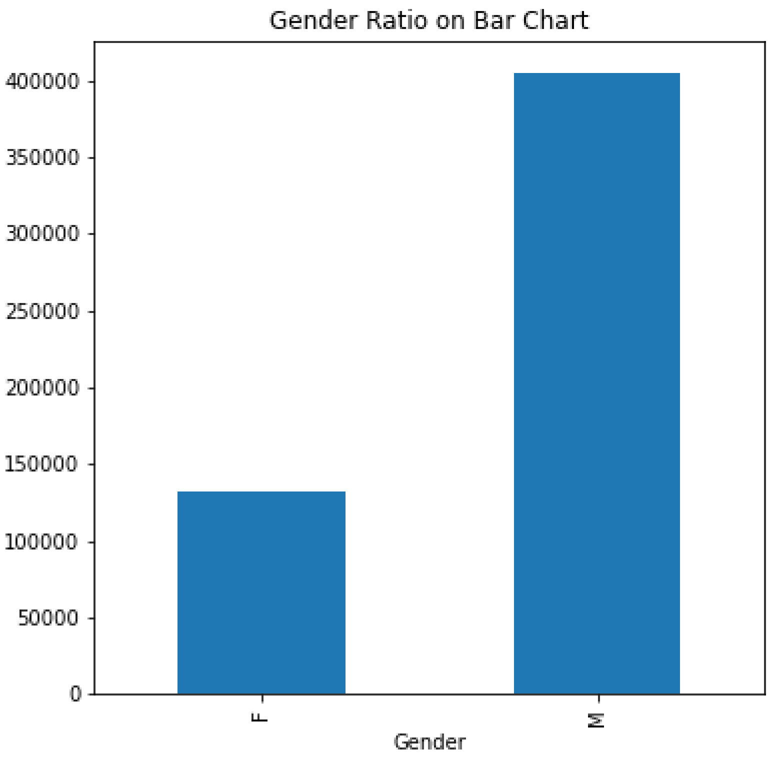
Out[27]:

```
<AxesSubplot:title={'center':'Gender Ratio on Pie Chart'}, ylabel='None'>
```



```
In [28]: df.groupby('Gender').size().plot.bar(figsize=(6,6),title='Gender Ratio on Bar Chart')
```

```
Out[28]: <AxesSubplot:title={'center':'Gender Ratio on Bar Chart'}, xlabel='Gender'>
```



In [29]:

```
df.groupby('Gender').sum()['Purchase']#to show the total no of items purchased by M or F.
```

Out[29]:

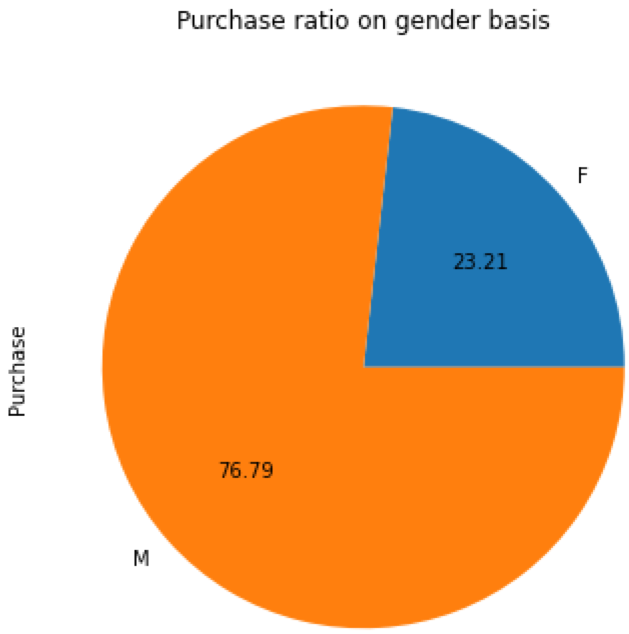
```
Gender
F    1164624021
M    3853044357
Name: Purchase, dtype: int64
```

In [30]:

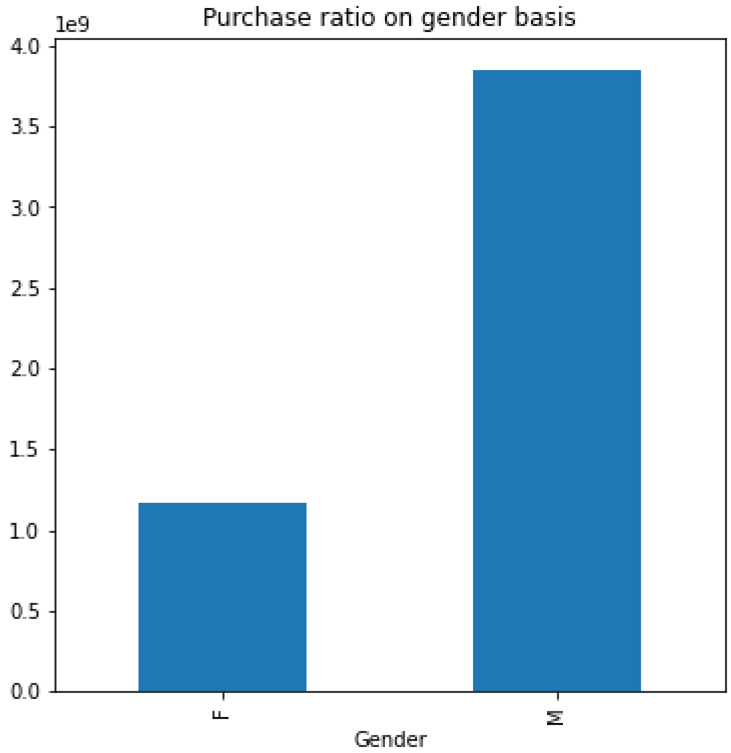
```
df.groupby('Gender').sum()['Purchase'].plot.pie(x='Ratio',figsize=(6,6),autopct="%.2f",title='Purchase ratio on gender
```

Out[30]:

```
<AxesSubplot:title={'center':'Purchase ratio on gender basis'}, ylabel='Purchase'>
```

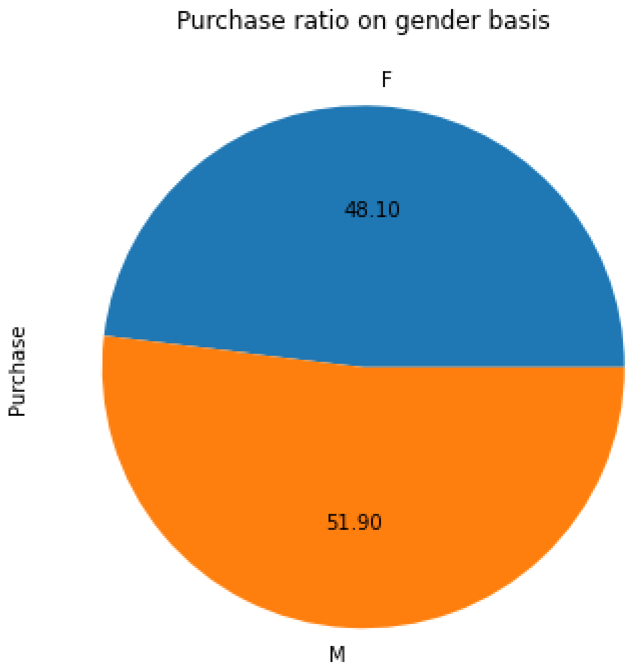


```
In [31]: df.groupby('Gender').sum()['Purchase'].plot.bar(figsize=(6,6),title='Purchase ratio on gender basis')
Out[31]: <AxesSubplot:title={'center':'Purchase ratio on gender basis'}, xlabel='Gender'>
```



In [32]:
```
df.groupby('Gender').mean()['Purchase'].plot.pie(x='Ratio',figsize=(6,6),autopct="%.2f",title='Purchase ratio on gender
```

Out[32]:
```
<AxesSubplot:title={'center':'Purchase ratio on gender basis'}, ylabel='Purchase'>
```



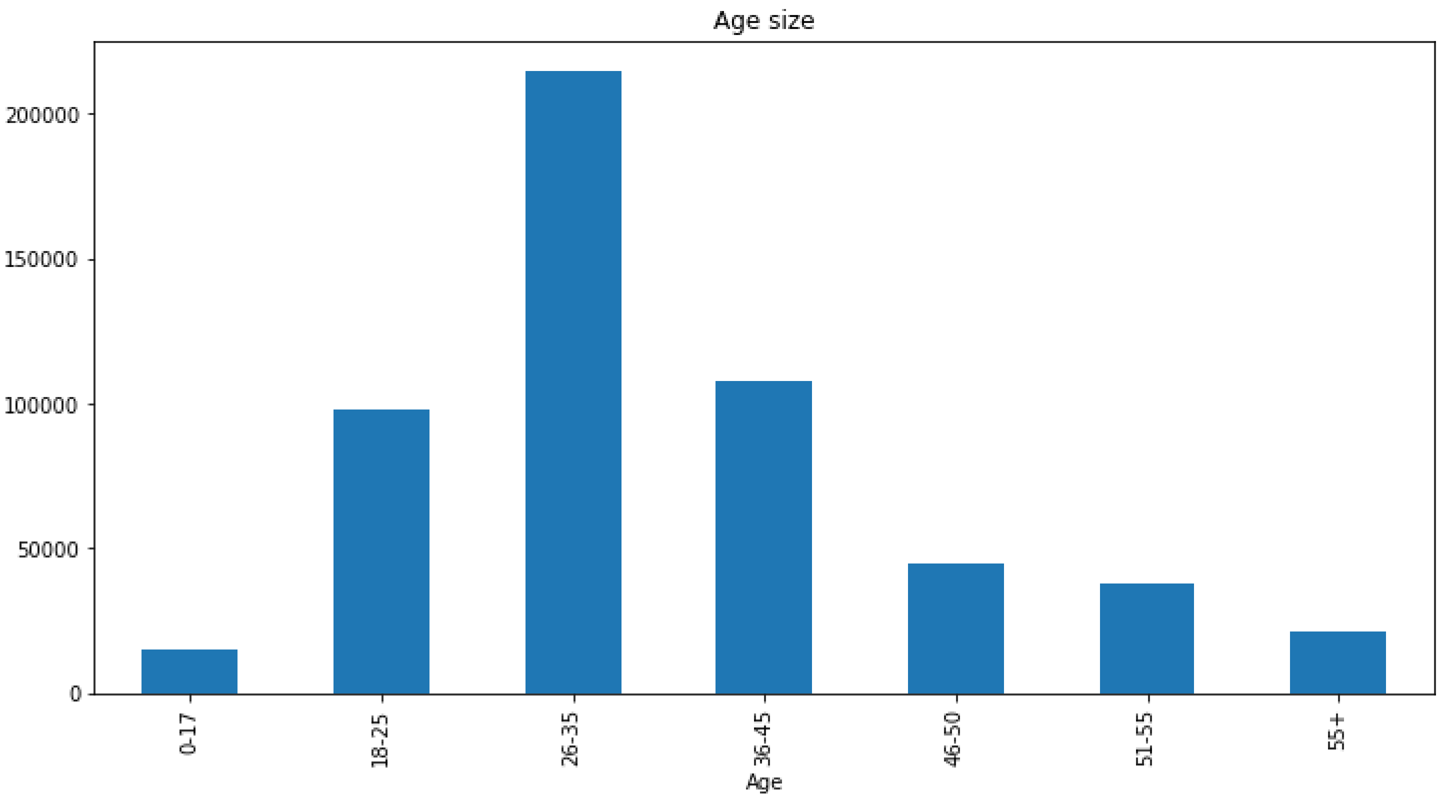
Analysing Age and Marital Status

```
In [33]: df.groupby("Age").size()#to count the no of people on a spcified age range
```

```
Out[33]: Age
0-17      14707
18-25     97634
26-35    214690
36-45    107499
46-50     44526
51-55     37618
55+       20903
dtype: int64
```

```
In [7]: df.groupby("Age").size().plot.bar(figsize=(12,6),title='Age size')#ploting a bar graph on the basis of size of each age
```

```
Out[7]: <AxesSubplot:title={'center':'Age size'}, xlabel='Age'>
```



In [35]:

```
l=[]
for i in df["Age"].unique():
    l.append([i,df[df['Age']==i]['Product_ID'].nunique()])# how many unique proucts bought by each age range
```

In [36]:

```
data=pd.DataFrame(l,columns=['Age','Products'])
```

In [37]:

```
data
```

Out[37]:

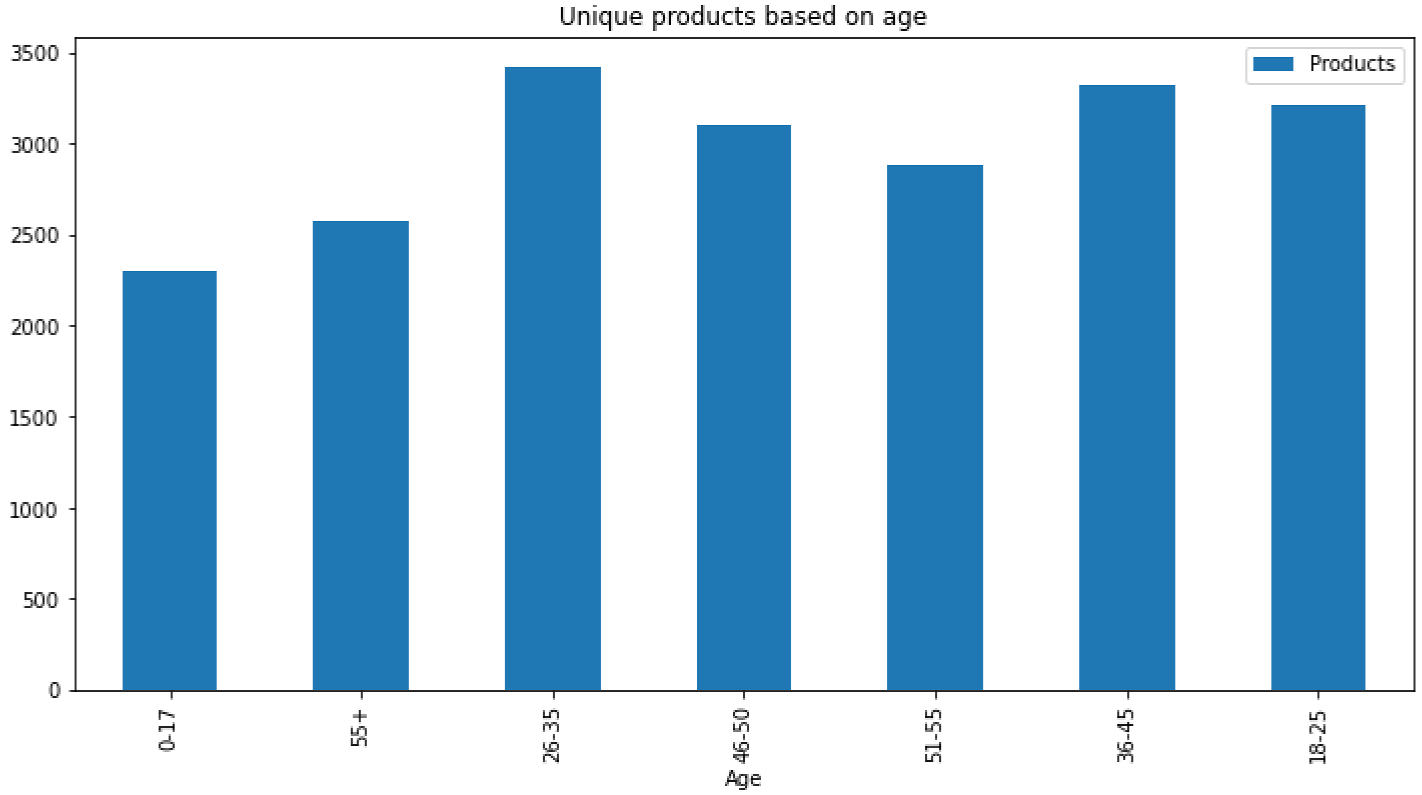
	Age	Products
0	0-17	2300
1	55+	2573
2	26-35	3419
3	46-50	3099
4	51-55	2877
5	36-45	3318
6	18-25	3213

In [38]:

```
data.plot.bar(x='Age',figsize=(12,6),title="Unique products based on age")#ploting a bar graph of unique proucts bought
```

Out[38]:

```
<AxesSubplot:title={'center':'Unique products based on age'}, xlabel='Age'>
```

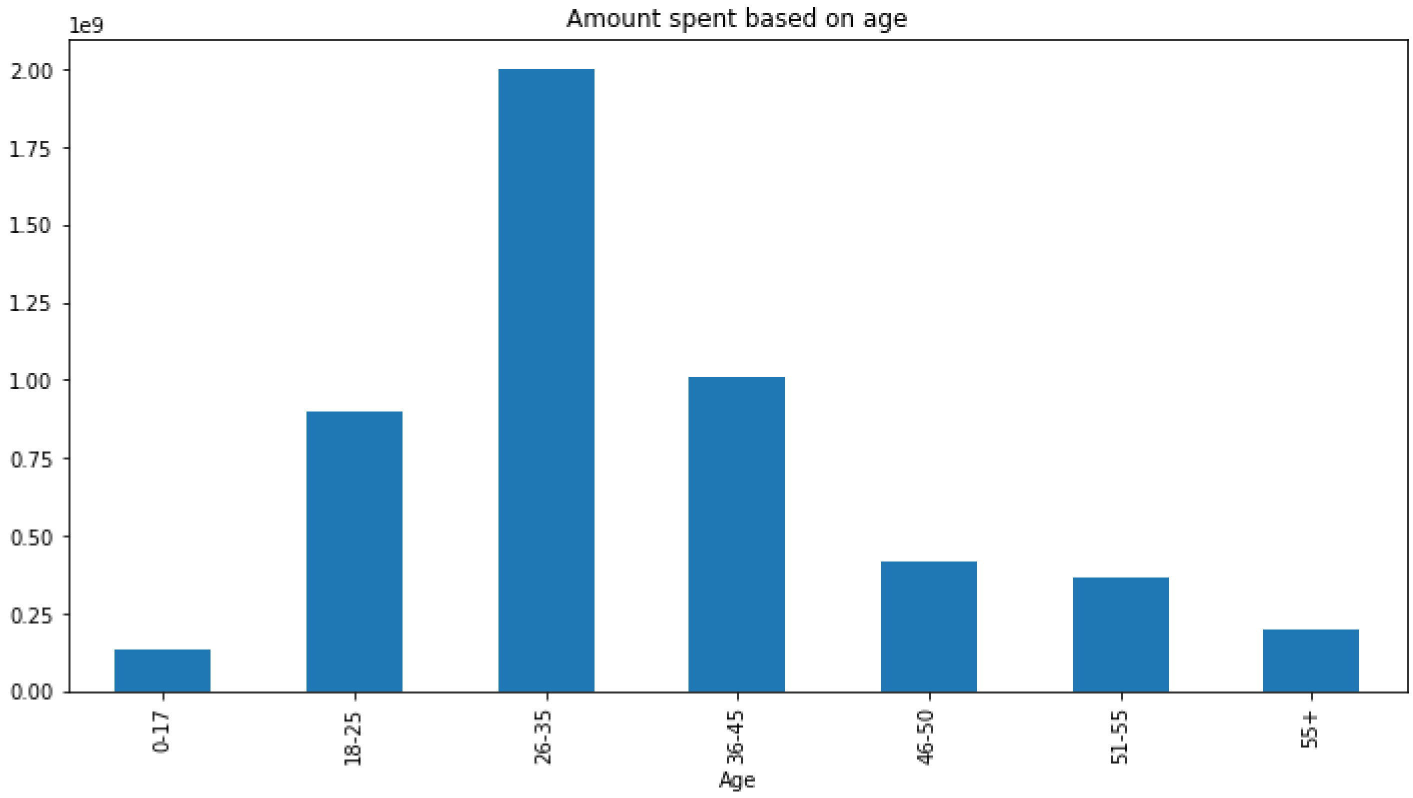


In [39]:

```
df.groupby('Age').sum()['Purchase'].plot.bar(figsize=(12,6),title="Amount spent based on age")
```

Out[39]:

```
<AxesSubplot:title={'center':'Amount spent based on age'}, xlabel='Age'>
```

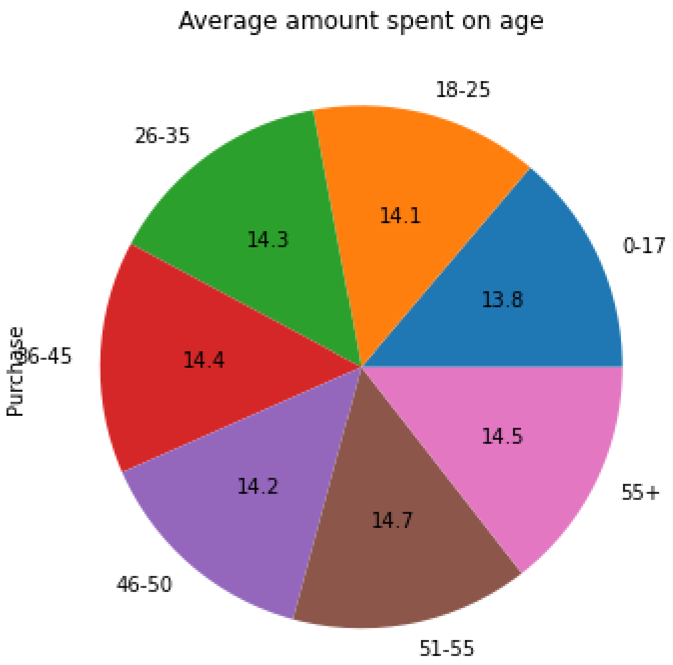


In [40]:

```
df.groupby('Age').mean()['Purchase'].plot.pie(figsize=(12,6),title="Average amount spent on age",autopct="%.1f")# avera
```

Out[40]:

```
<AxesSubplot:title={'center':'Average amount spent on age'}, ylabel='Purchase'>
```

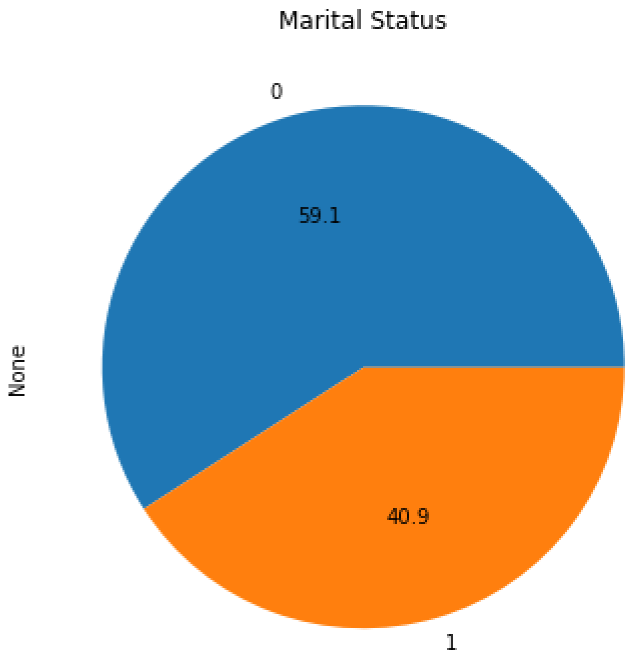


In [41]:

```
df.groupby('Marital_Status').size().plot.pie(figsize=(12,6),title="Marital Status",autopct="%.1f")# to plot total perce
```

Out[41]:

```
<AxesSubplot:title={'center':'Marital Status'}, ylabel='None'>
```



Multi column analysis

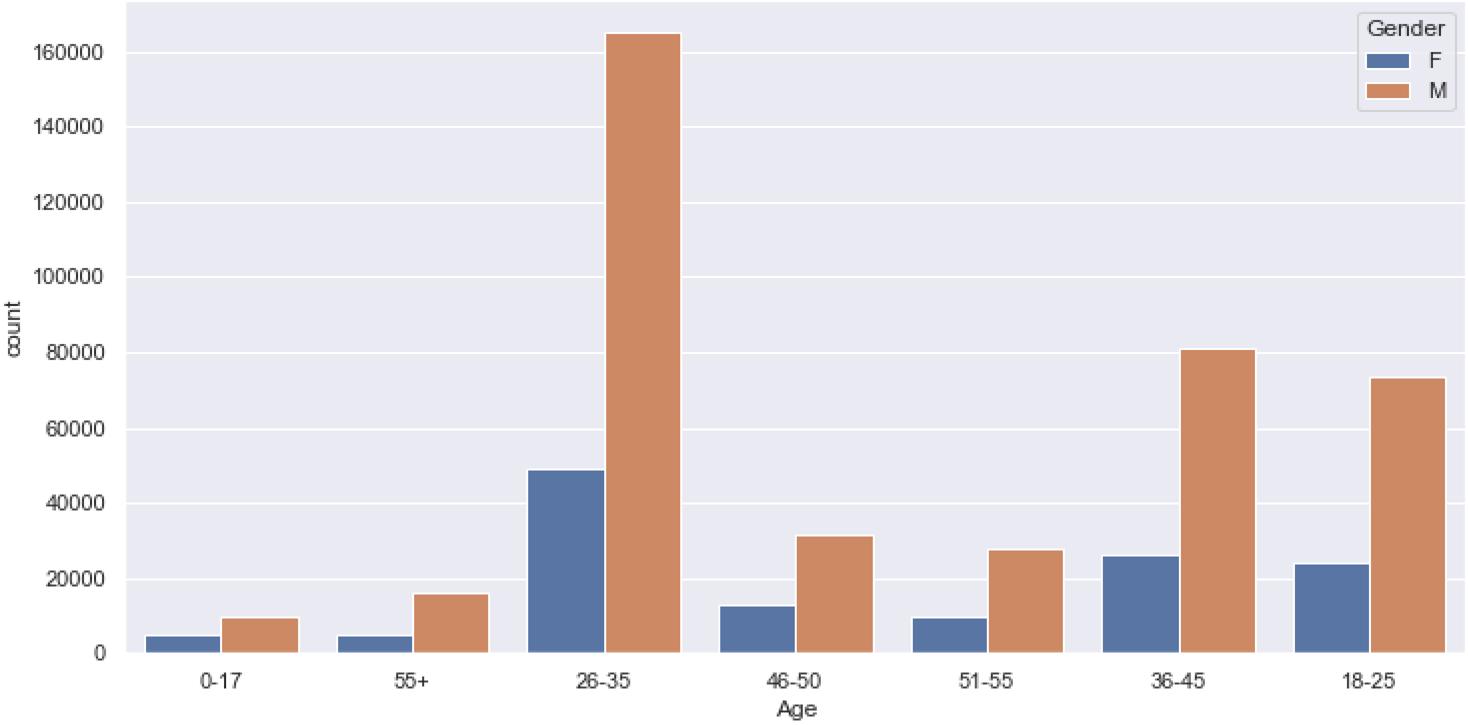
```
In [42]: df.head()
```

Out[42]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

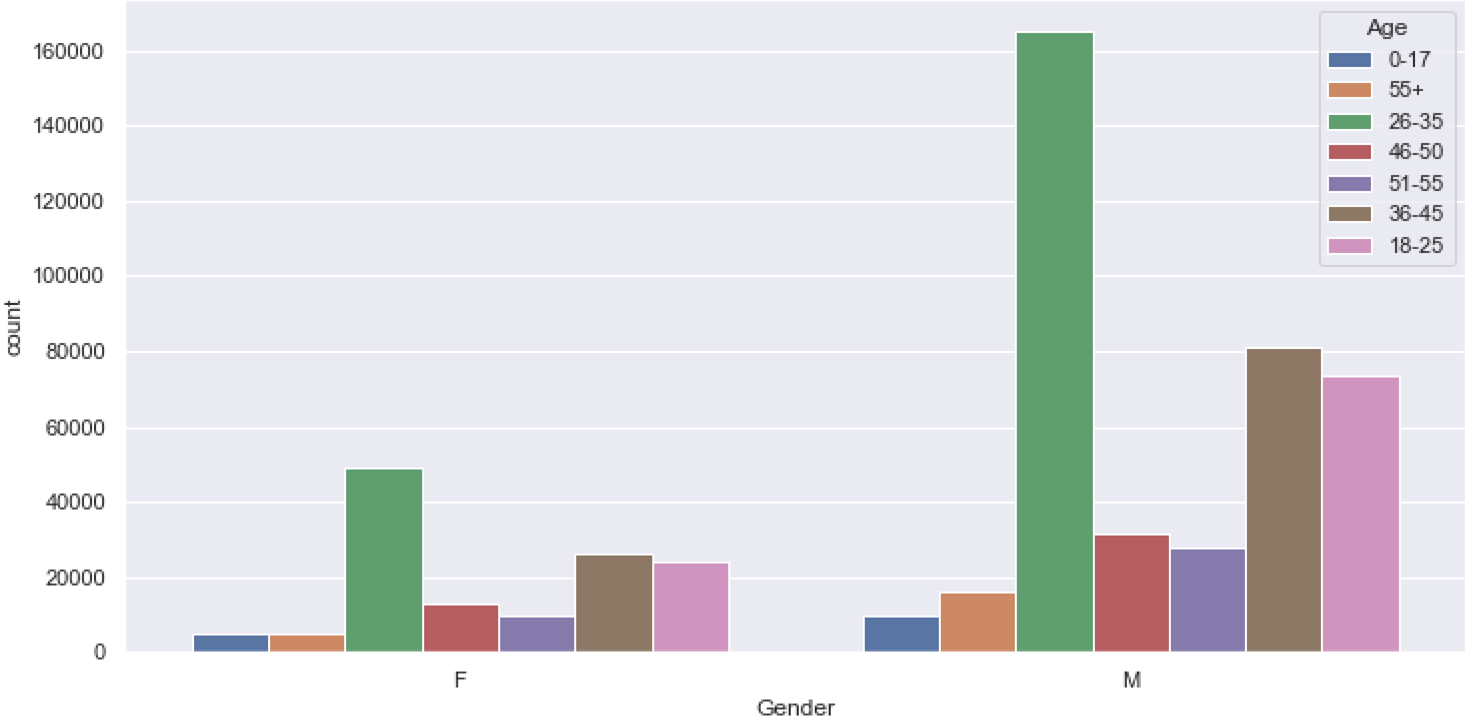
In [43]:
```
sns.set(rc = {'figure.figsize' : (12,6)})
sns.countplot(x = "Age", hue = 'Gender', data = df)#ploting the no of male and female on each age range
```

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



In [44]:
```
sns.set(rc = {'figure.figsize' : (12,6)})
sns.countplot(x = "Gender", hue = 'Age', data = df)#ploting the no of male and female on each age range
```

Out[44]:
```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```

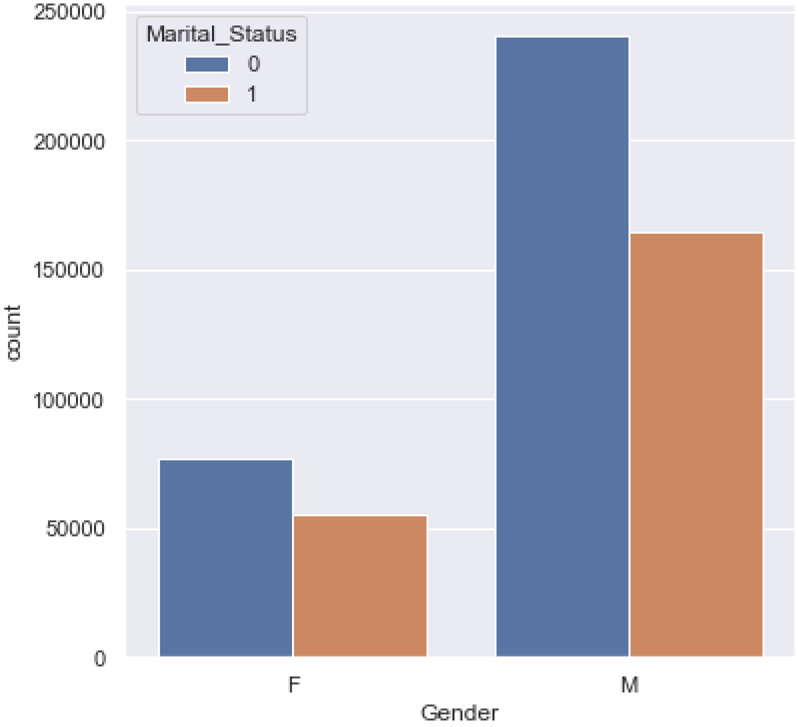


In [45]:

```
sns.set(rc = {'figure.figsize' : (6,6)})
sns.countplot(x = "Gender", hue = 'Marital_Status', data = df)# to plot marital status based on gender
```

Out[45]:

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```

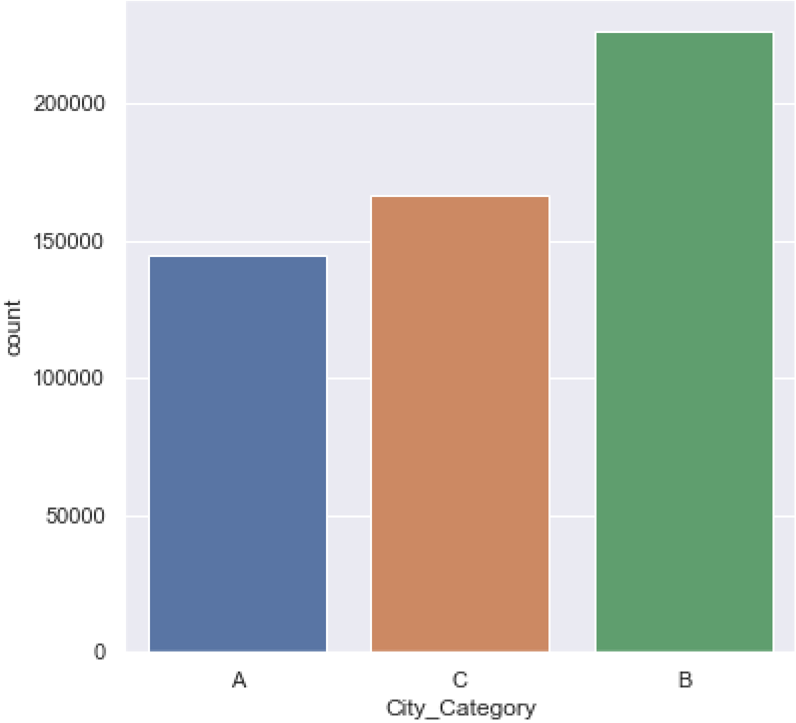


In [46]:

```
sns.countplot(x=df['City_Category'])#to plot the no of people living in each city
```

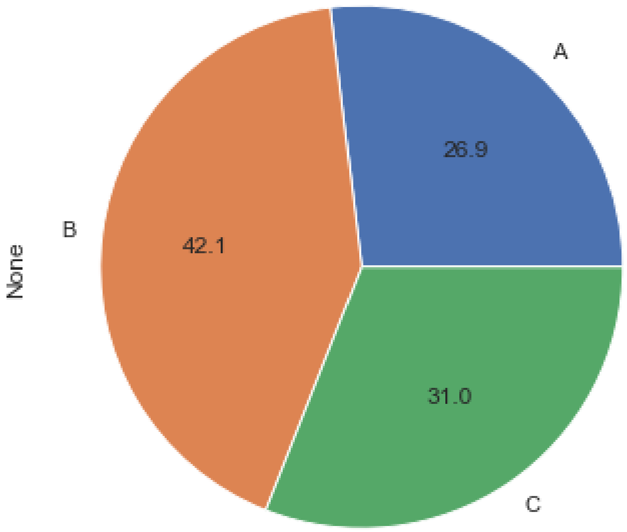
Out[46]:

```
<AxesSubplot:xlabel='City_Category', ylabel='count'>
```



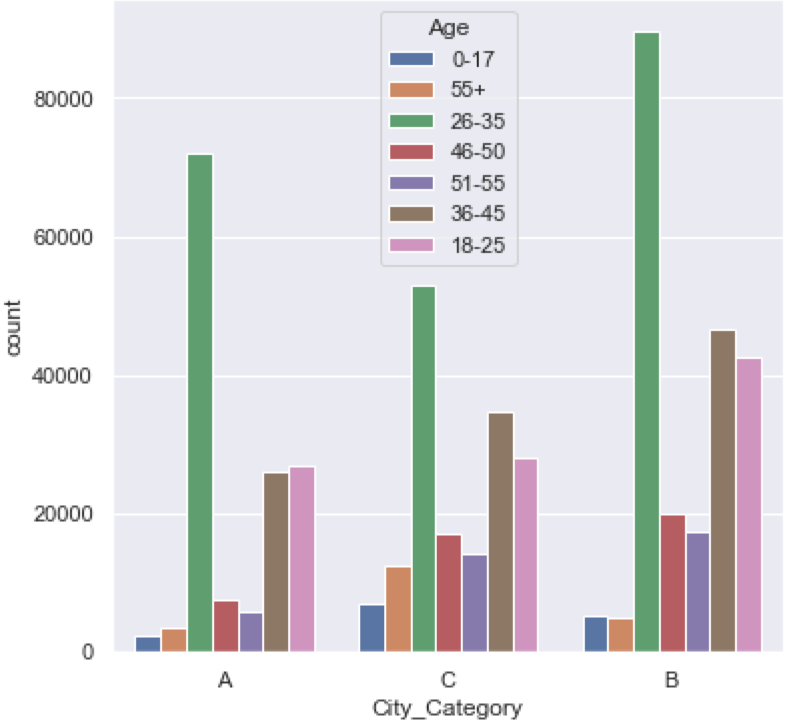
In [47]:
```
df.groupby('City_Category').size().plot.pie(autopct="%.1f")
```

Out[47]:
```
<AxesSubplot:ylabel='None'>
```



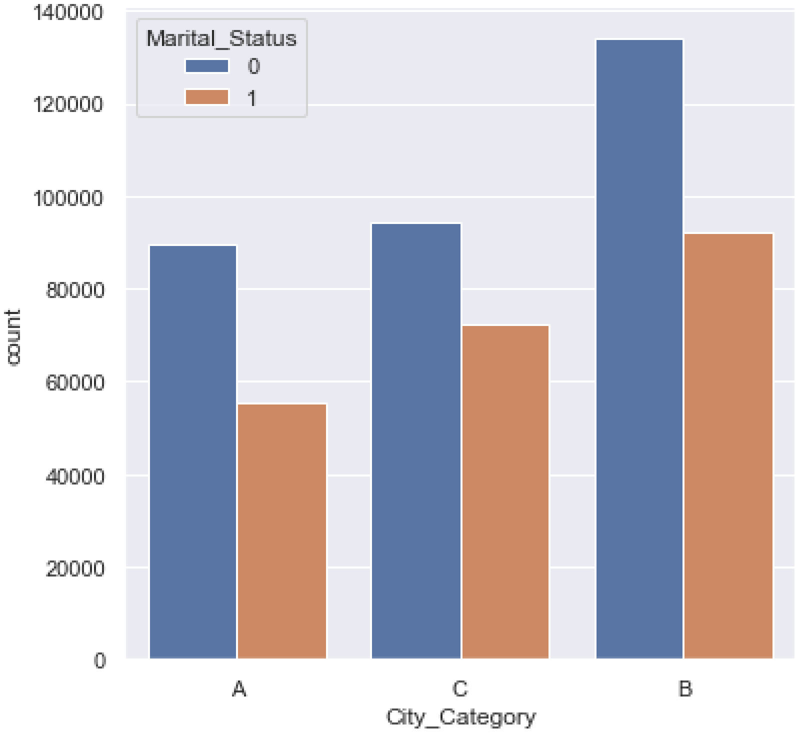
```
In [48]: sns.countplot(x='City_Category',hue='Age',data=df)#plotig diff age group people living on each city
```

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



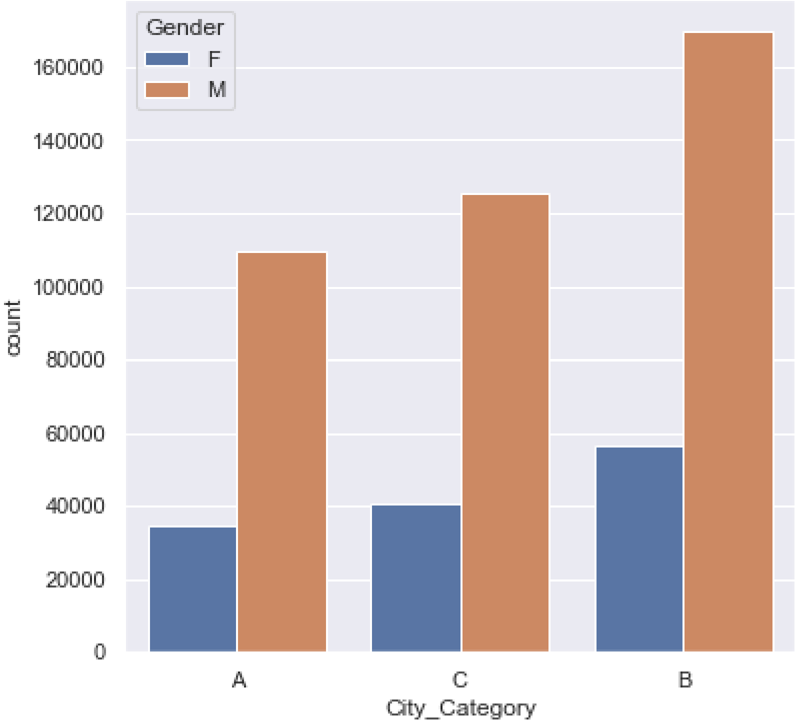
```
In [49]: sns.countplot(x='City_Category',hue='Marital_Status',data=df)#marital status on each city
```

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



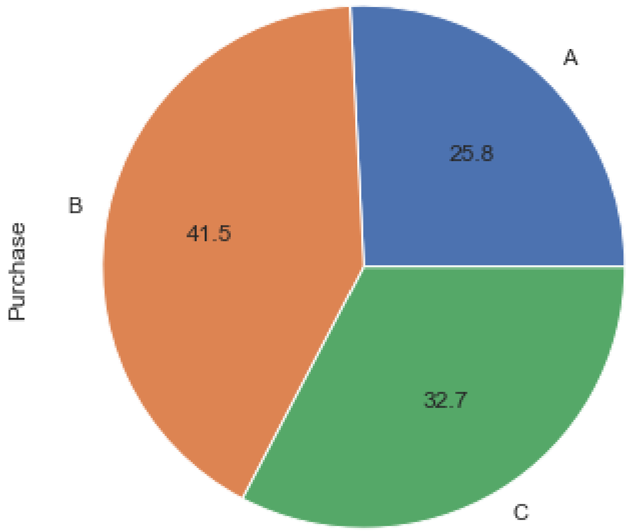
In [50]:
```
sns.countplot(x='City_Category',hue='Gender',data=df)#no of males and female in each city
```

Out[50]:
```
<AxesSubplot:xlabel='City_Category', ylabel='count'>
```



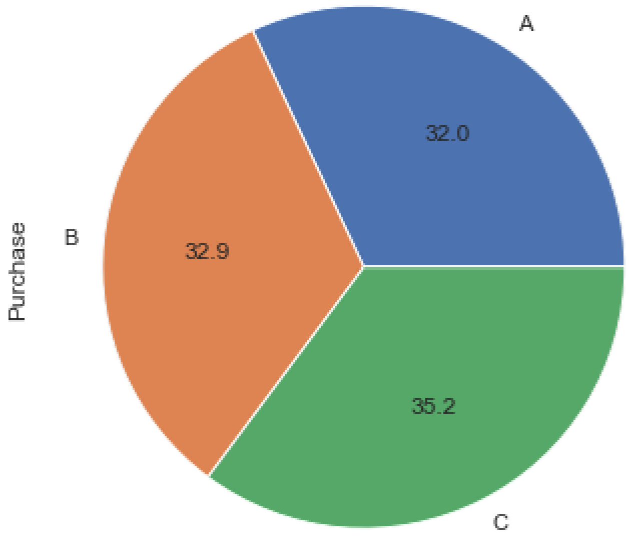
In [51]:
```
df.groupby('City_Category').sum()['Purchase'].plot.pie(autopct="%.1f")#which city purchase more
```

Out[51]:
```
<AxesSubplot:ylabel='Purchase'>
```



In [52]:
```
df.groupby('City_Category').mean()['Purchase'].plot.pie(autopct="%.1f")#average amount spent by each city
```

Out[52]:
```
<AxesSubplot:ylabel='Purchase'>
```



Occupation and product Analysis

In [53]:

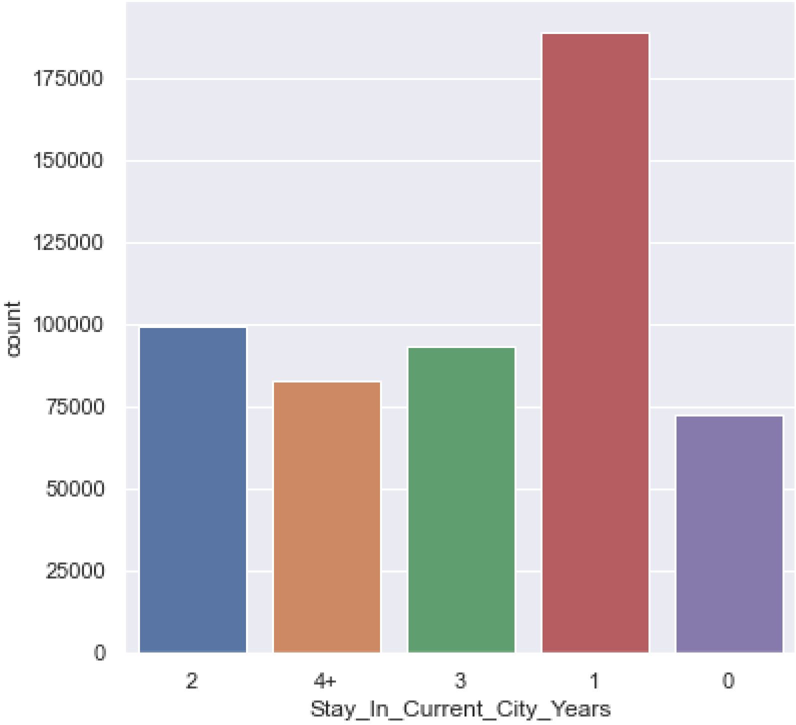
```
df.head()
```

Out[53]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

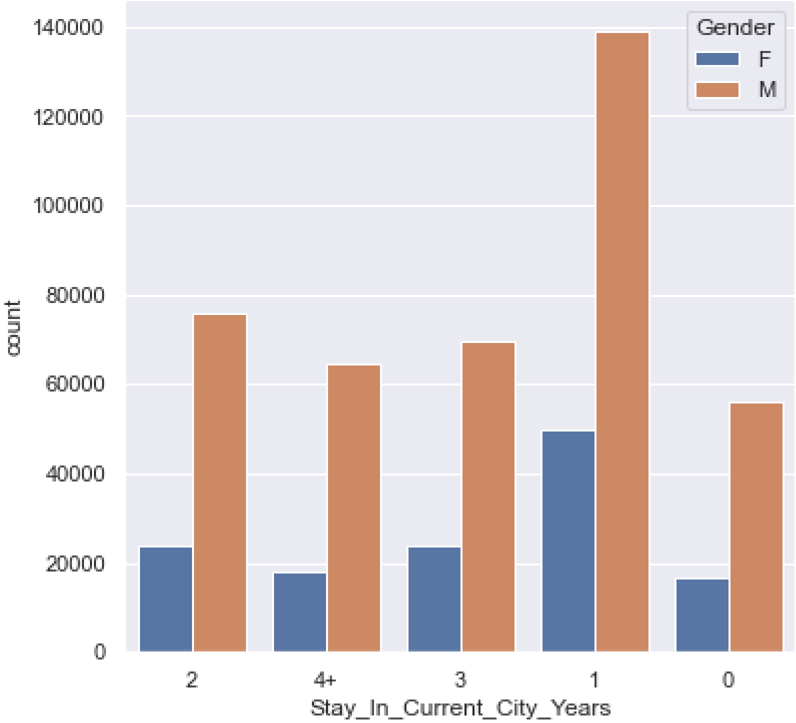
In [54]:
```
sns.countplot(x=df['Stay_In_Current_City_Years'])#tendancy to stay in current city in years
```

Out[54]:
```
<AxesSubplot:xlabel='Stay_In_Current_City_Years', ylabel='count'>
```



In [55]:
```
sns.countplot(x='Stay_In_Current_City_Years',hue='Gender',data=df)#tendancy to stay in current city in years on the bas
```

Out[55]:
```
<AxesSubplot:xlabel='Stay_In_Current_City_Years', ylabel='count'>
```

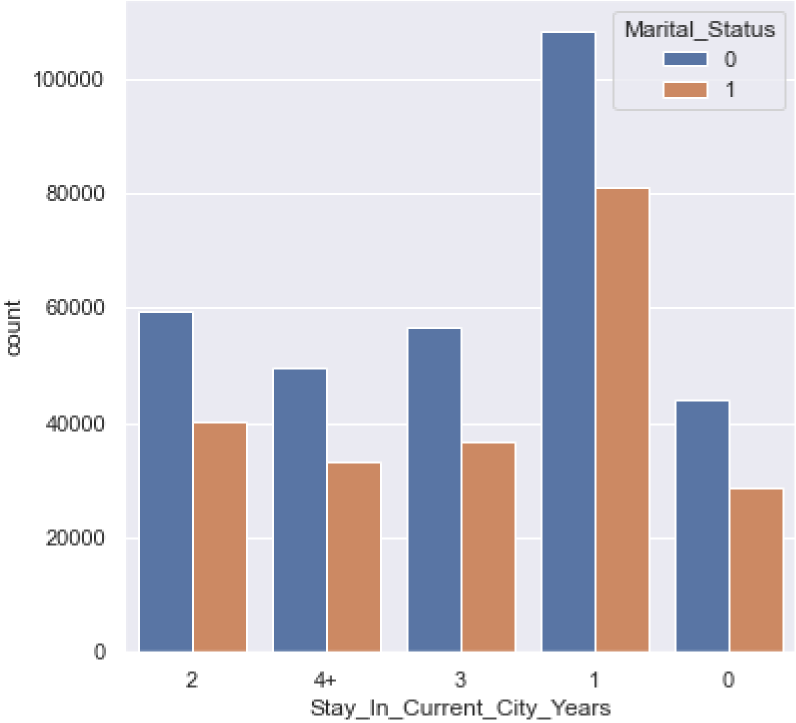


In [56]:

```
sns.countplot(x='Stay_In_Current_City_Years',hue='Marital_Status',data=df)#tendancy to stay in current city in years on
```

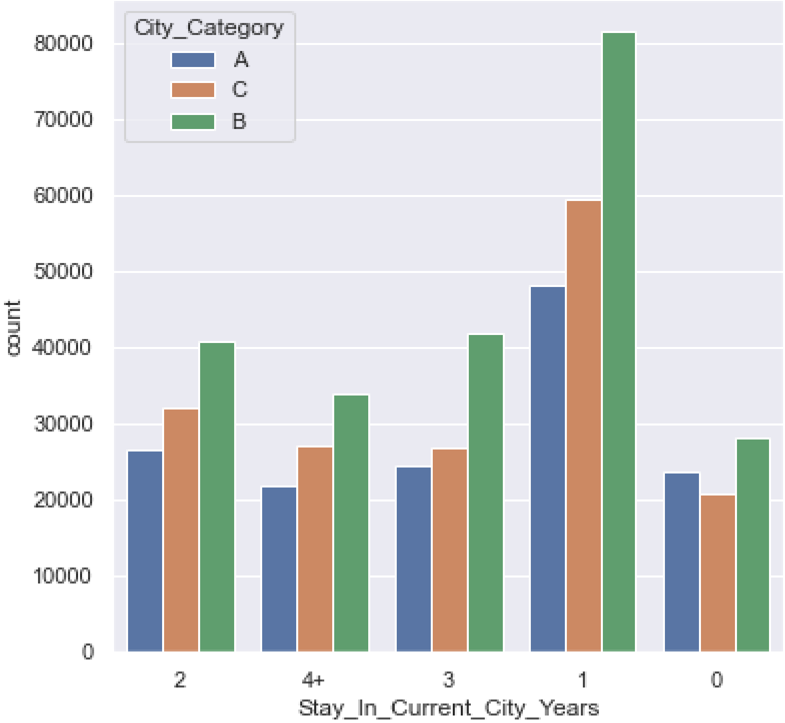
Out[56]:

```
<AxesSubplot:xlabel='Stay_In_Current_City_Years', ylabel='count'>
```



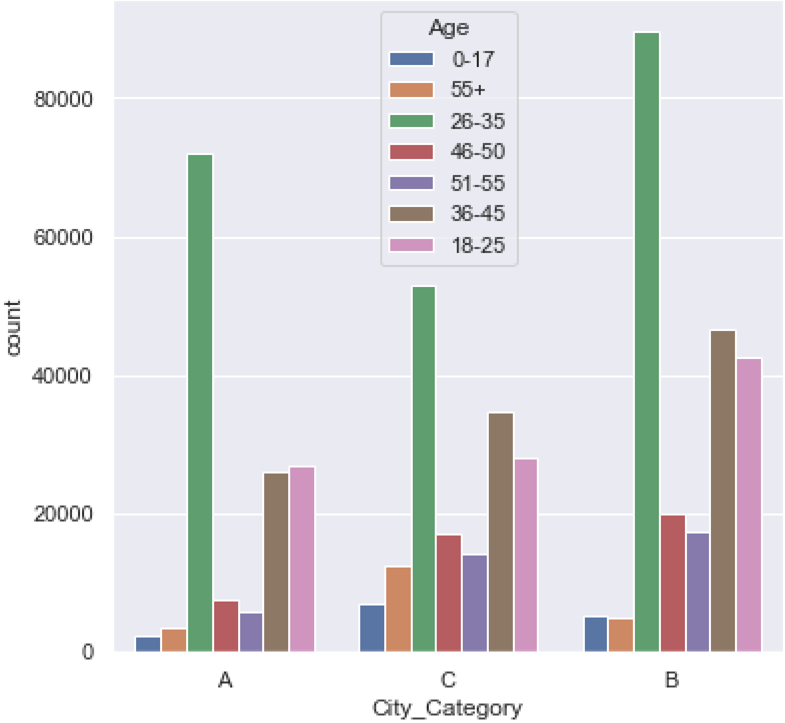
In [57]:
```
sns.countplot(x='Stay_In_Current_City_Years',hue='City_Category',data=df)#tendancy to stay in current city in years on
```

Out[57]:
```
<AxesSubplot:xlabel='Stay_In_Current_City_Years', ylabel='count'>
```



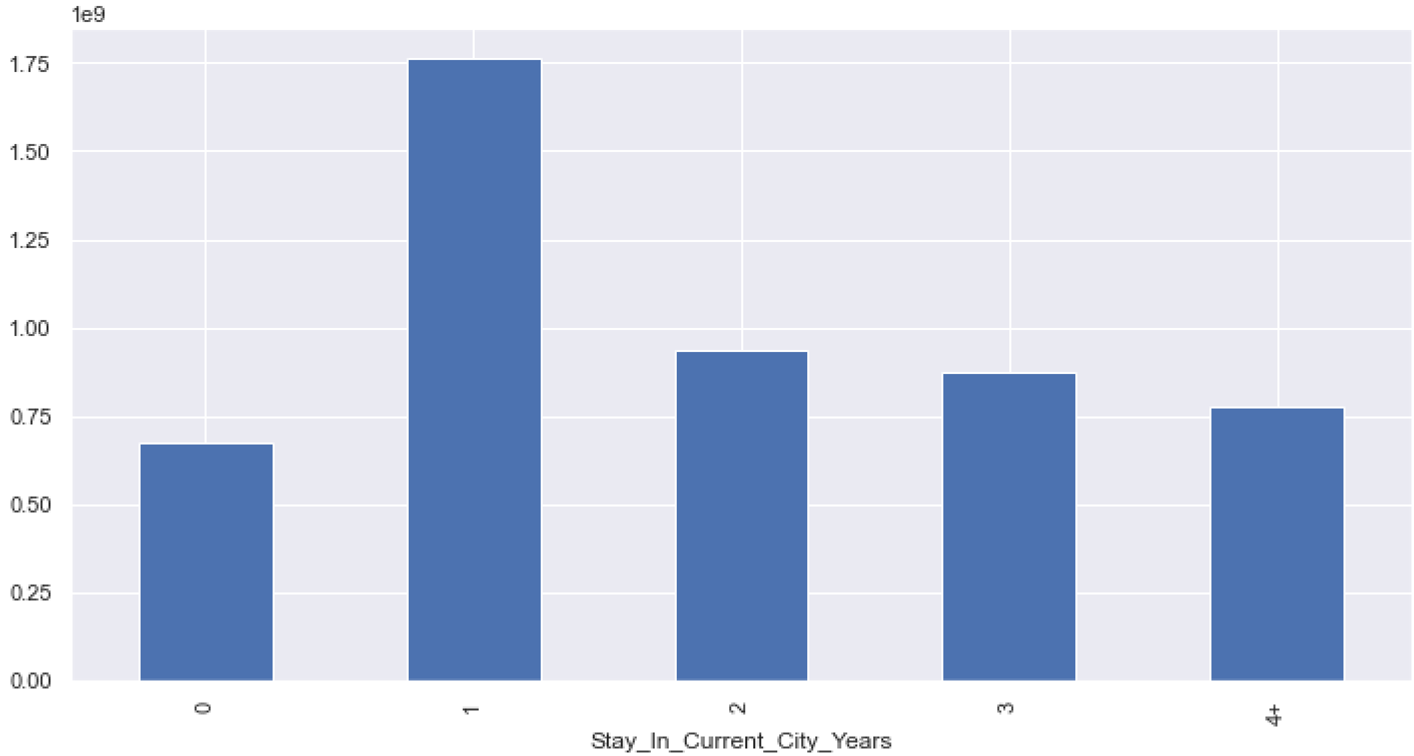
```
In [58]: sns.countplot(x='City_Category',hue='Age',data=df)#which age group stay in each city
```

```
Out[58]: <AxesSubplot:xlabel='City_Category', ylabel='count'>
```



In [59]:
```
df.groupby('Stay_In_Current_City_Years').sum()['Purchase'].plot.bar(figsize=(12,6))#which stay in current city has the
```

Out[59]:
```
<AxesSubplot:xlabel='Stay_In_Current_City_Years'>
```

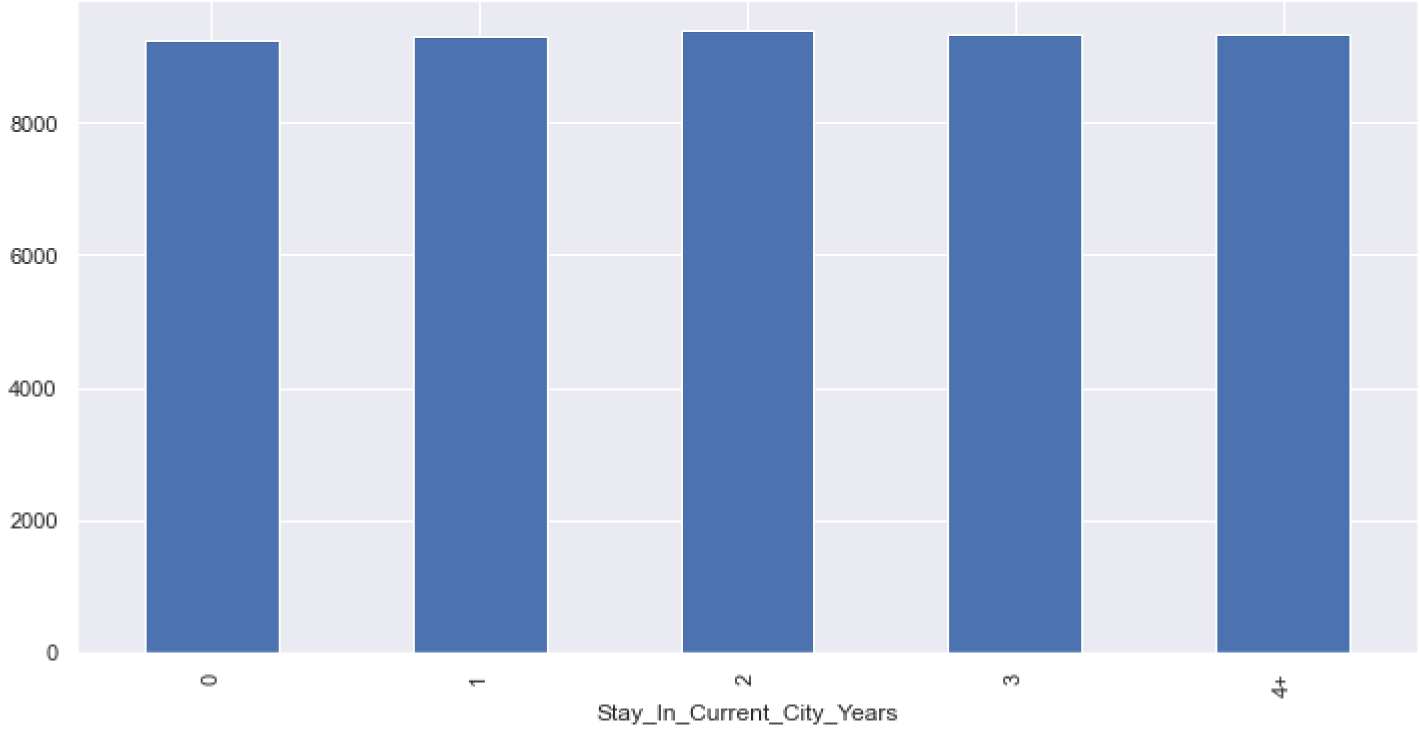


In [60]:

```
df.groupby('Stay_In_Current_City_Years').mean()['Purchase'].plot.bar(figsize=(12,6))# averag purchasing power of peropl
```

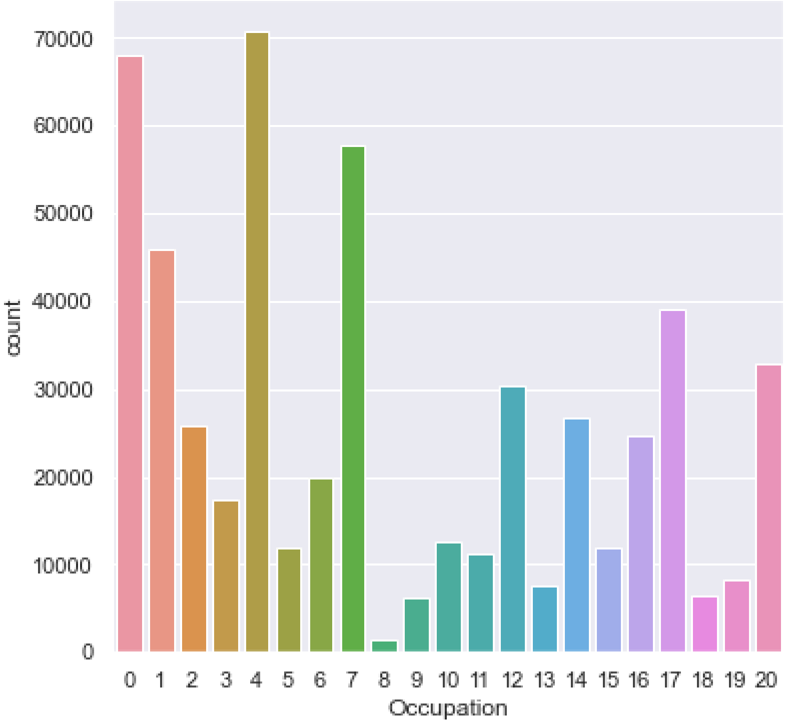
Out[60]:

```
<AxesSubplot:xlabel='Stay_In_Current_City_Years'>
```



In [61]:
```
sns.countplot(x=df['Occupation'])#to plot the no of occupation
```

Out[61]:
```
<AxesSubplot:xlabel='Occupation', ylabel='count'>
```

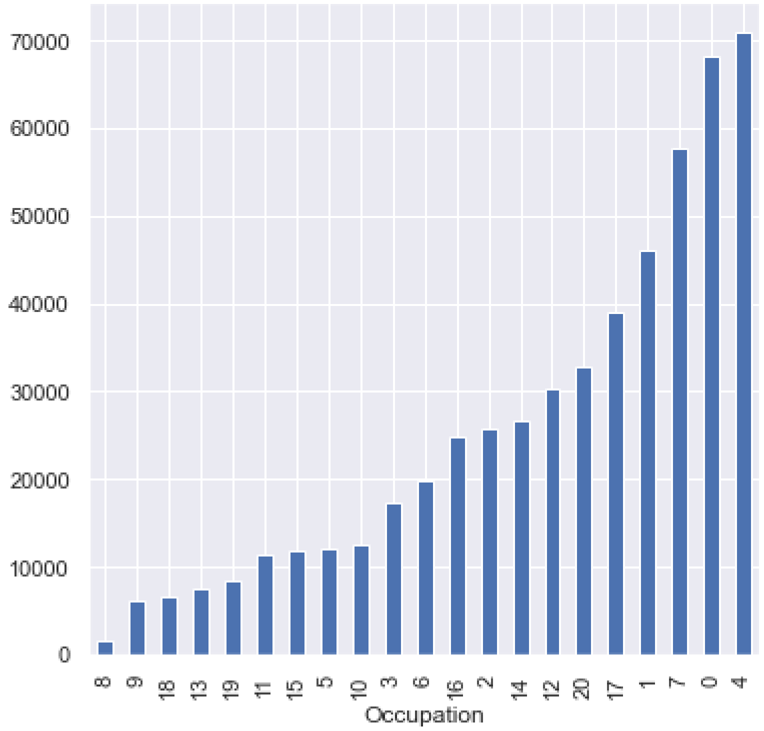


In [62]:

```
df.groupby('Occupation').size().sort_values().plot.bar(figsize=(6,6))#occupation on size
```

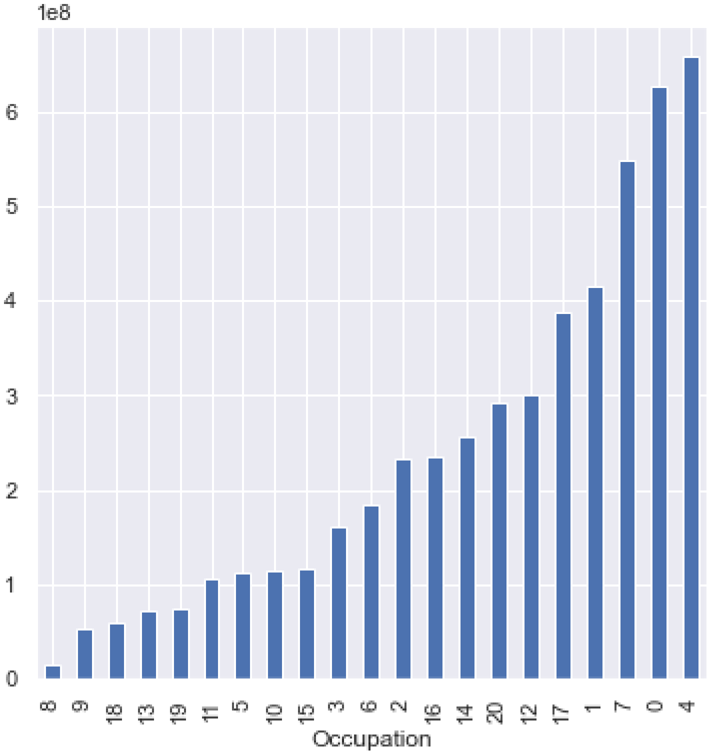
Out[62]:

```
<AxesSubplot:xlabel='Occupation'>
```



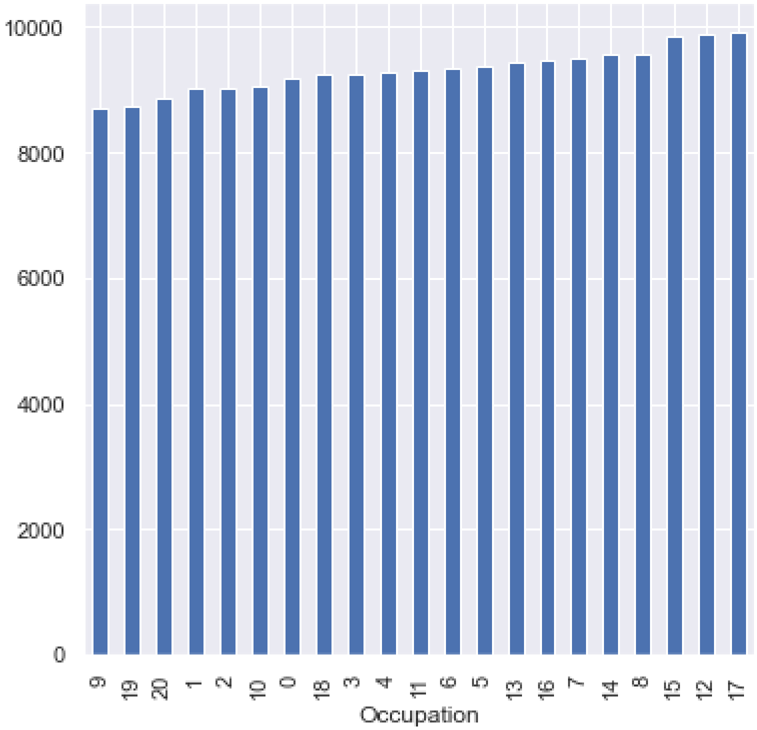
In [63]:
```
df.groupby('Occupation').sum()['Purchase'].sort_values().plot.bar(figsize=(6,6))#which occupation people purchasing mor
```

Out[63]:
```
<AxesSubplot:xlabel='Occupation'>
```



In [64]:
```
df.groupby('Occupation').mean()['Purchase'].sort_values().plot.bar(figsize=(6,6))#which occupation people spending more
```

Out[64]:
```
<AxesSubplot:xlabel='Occupation'>
```



```
In [65]: df.groupby('Occupation').mean()['Purchase'].sort_values()
```

Out[65]:

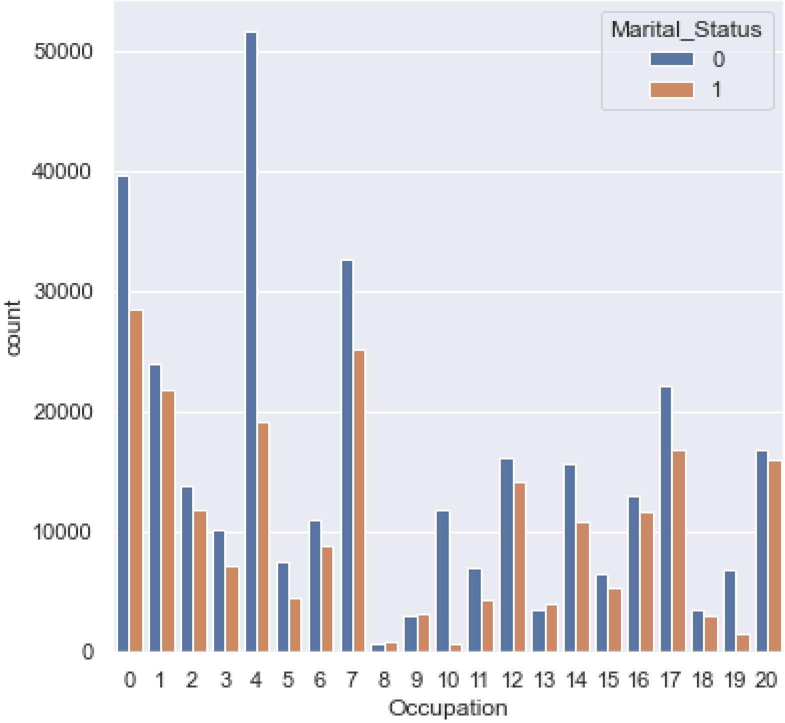
```
Occupation
9     8714.335934
19    8754.249162
20    8881.099514
1     9017.703095
2     9025.938982
10    9052.836410
0     9186.946726
18    9233.671418
3     9238.077277
4     9279.026742
11    9299.467190
6     9336.378620
5     9388.848978
13    9424.449391
16    9457.133118
7     9502.175276
14    9568.536426
8     9576.508530
15    9866.239925
12    9883.052460
17    9906.378997
Name: Purchase, dtype: float64
```

In [66]:

```
sns.countplot(x='Occupation',hue='Marital_Status',data=df)#occupation based on marital status
```

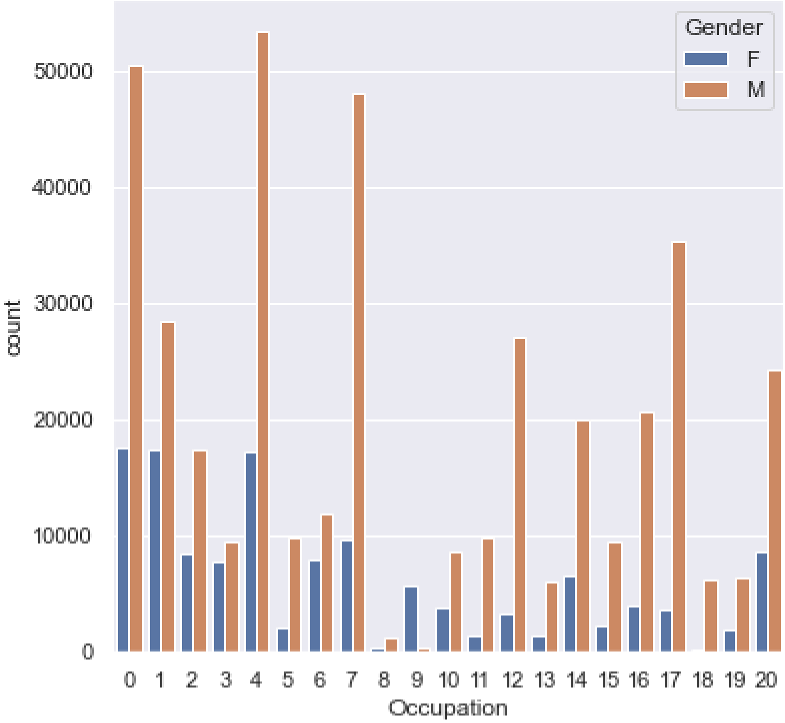
Out[66]:

```
<AxesSubplot:xlabel='Occupation', ylabel='count'>
```



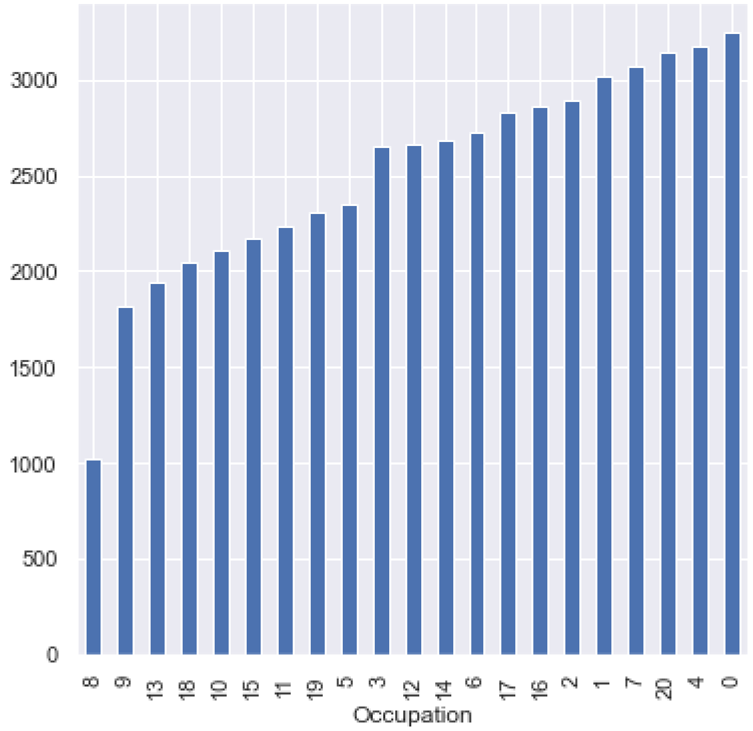
In [67]:
```
sns.countplot(x='Occupation',hue='Gender',data=df)#ocuupation on the basis of gender
```

Out[67]:
```
<AxesSubplot:xlabel='Occupation', ylabel='count'>
```



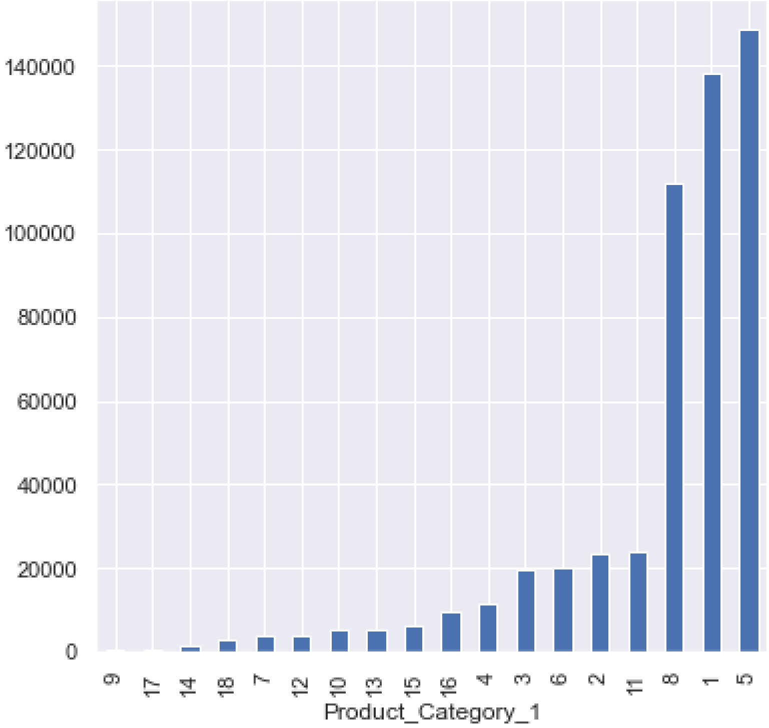
In [68]:
```
df.groupby('Occupation').nunique()['Product_ID'].sort_values().plot.bar(figsize=(6,6))#which occupation purchasing uniq
```

Out[68]:
```
<AxesSubplot:xlabel='Occupation'>
```



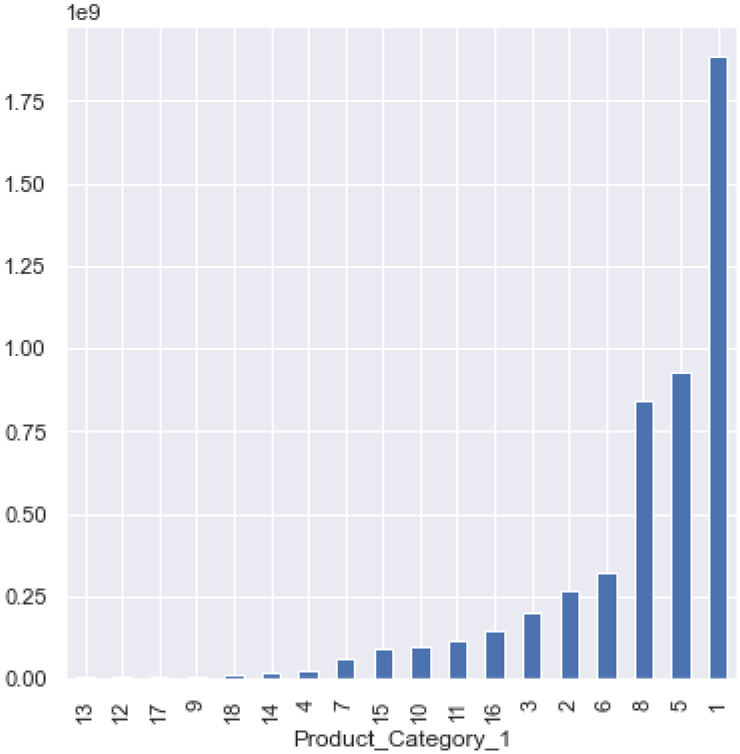
In [69]:
```
df.groupby('Product_Category_1').size().sort_values().plot(kind='bar')#no of product category
```

Out[69]:
```
<AxesSubplot:xlabel='Product_Category_1'>
```



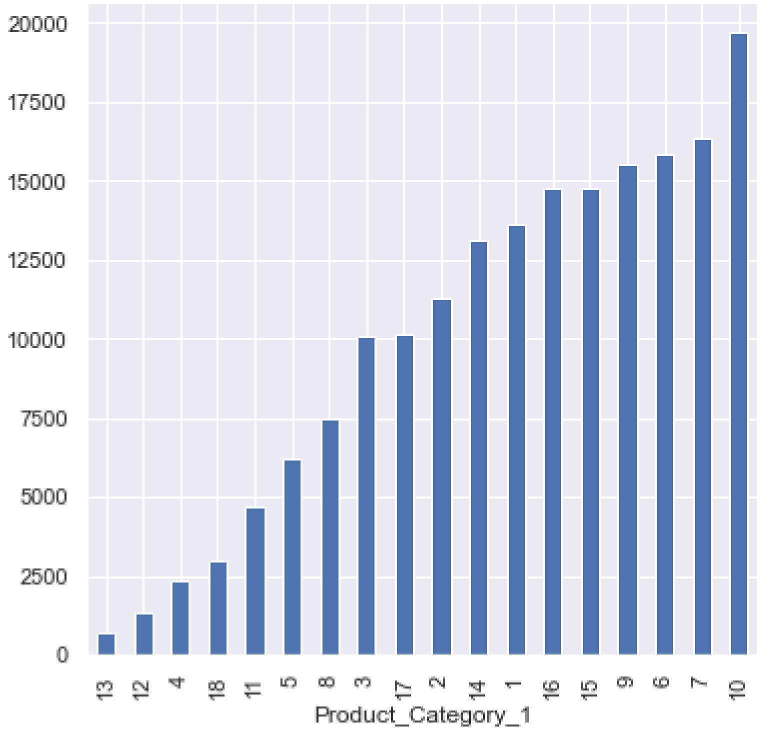
```
In [70]: df.groupby('Product_Category_1').sum()['Purchase'].sort_values().plot(kind='bar')#which category purchased more
```

```
Out[70]: <AxesSubplot:xlabel='Product_Category_1'>
```



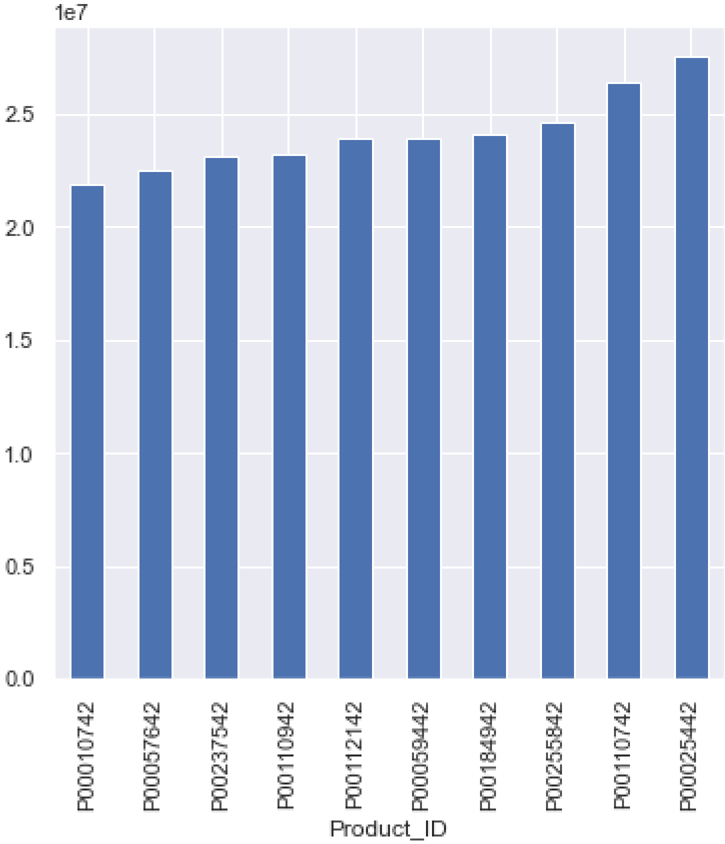
```
In [71]: df.groupby('Product_Category_1').mean()['Purchase'].sort_values().plot(kind='bar')#on which category more money spent
```

```
Out[71]: <AxesSubplot:xlabel='Product_Category_1'>
```



```
In [72]: df.groupby('Product_ID').sum()['Purchase'].nlargest(10).sort_values().plot(kind='bar')#top 10 product id interms of pur
```

```
Out[72]: <AxesSubplot:xlabel='Product_ID'>
```

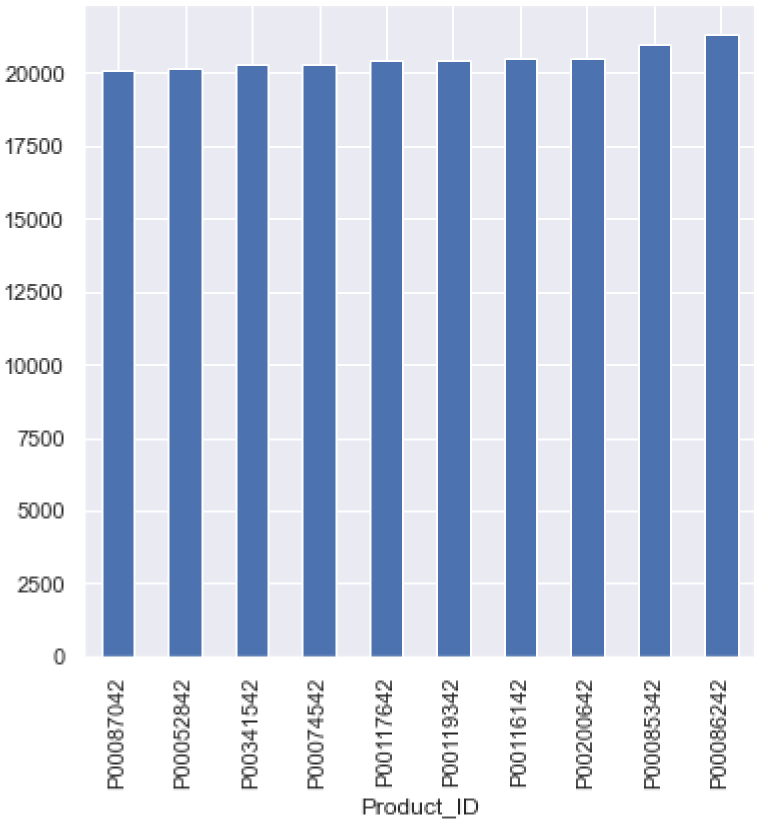


In [73]:

```
df.groupby('Product_ID').mean()['Purchase'].nlargest(10).sort_values().plot(kind='bar')#top 10 product idon which more
```

Out[73]:

```
<AxesSubplot:xlabel='Product_ID'>
```



Combining Gender and Marital Status

In [74]:
```
df.head()
```

Out[74]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

In [75]:

```
l=[]
for i in range(len(df)):
    l.append(df['Gender'][i]+'_'+str(df['Marital_Status'][i]))#combining gender and marital status column
```

In [77]:

```
df['MaritalGender']=l
```

In [78]:

```
df.head()
```

Out[78]:

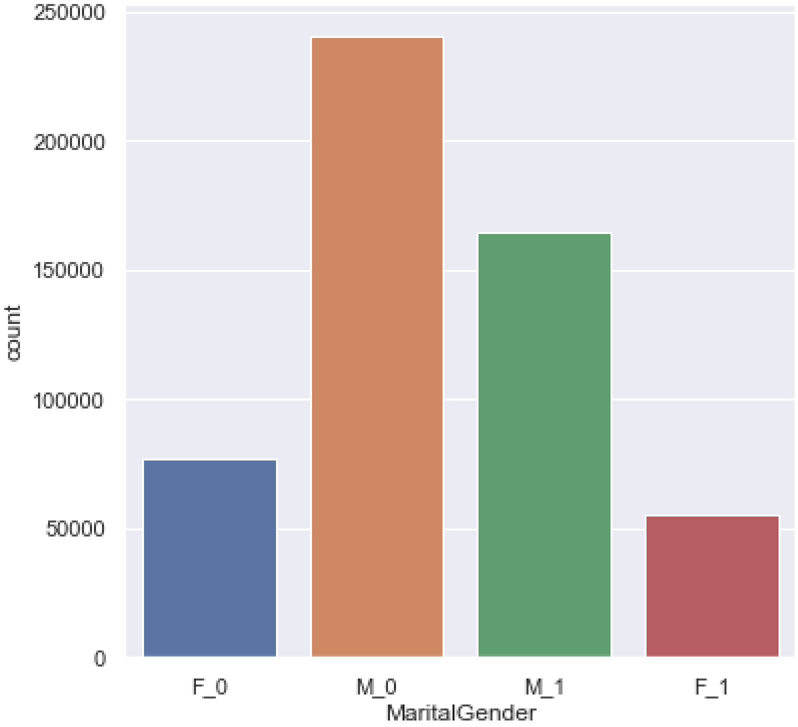
	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

In [82]:

```
sns.countplot(x=df['MaritalGender'])#no of female and male on the basis of maried and unbaried
```

Out[82]:

```
<AxesSubplot:xlabel='MaritalGender', ylabel='count'>
```

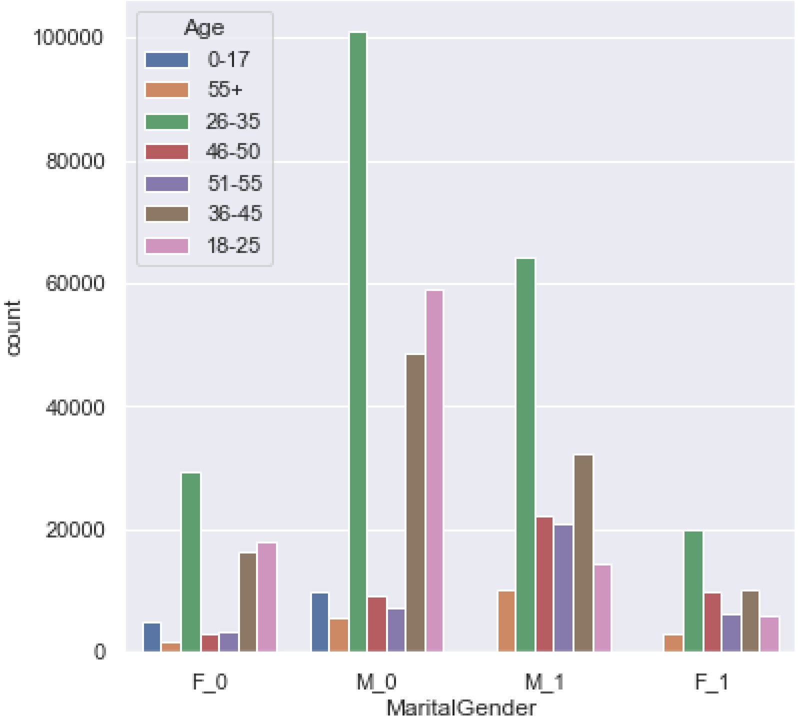


In [83]:

```python
sns.countplot(x='MaritalGender',hue='Age',data=df)#marital gender on the basis of gender
```

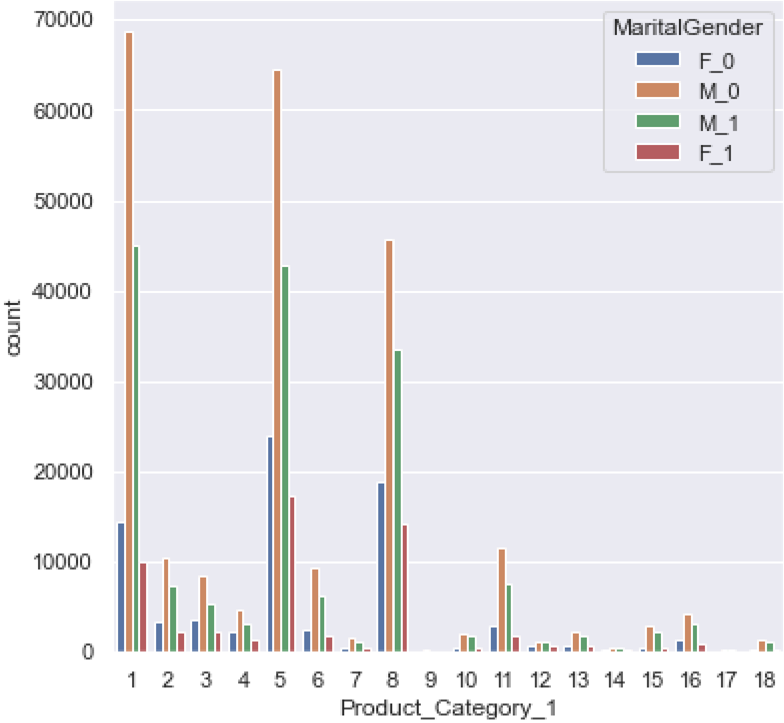
Out[83]:

```
<AxesSubplot:xlabel='MaritalGender', ylabel='count'>
```



In [85]:
```
sns.countplot(x='Product_Category_1',hue='MaritalGender',data=df)#product category on the basis of marital gender
```

Out[85]:
```
<AxesSubplot:xlabel='Product_Category_1', ylabel='count'>
```

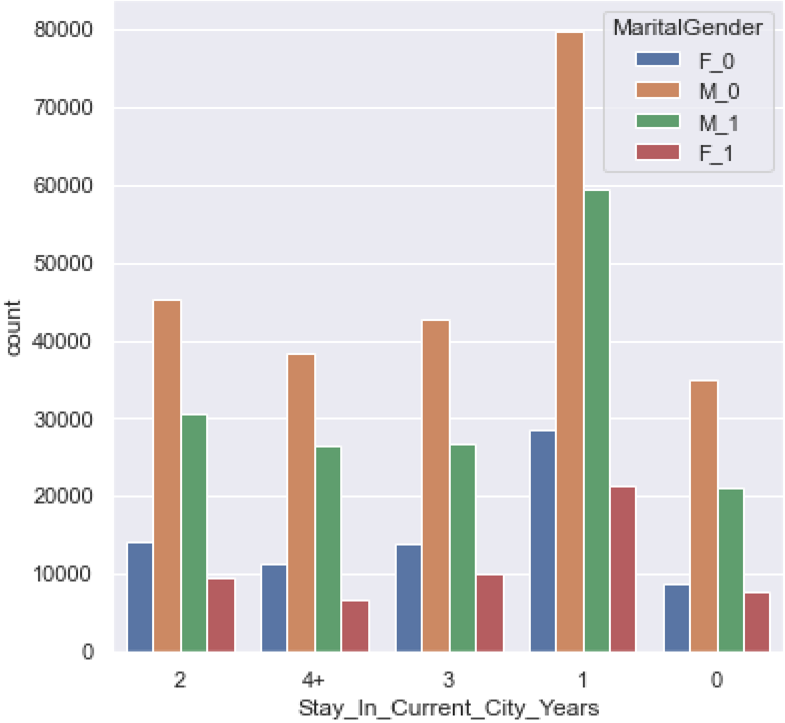


In [87]:

```
sns.countplot(x='Stay_In_Current_City_Years',hue='MaritalGender',data=df)#stay in current city on the basisof marital g
```

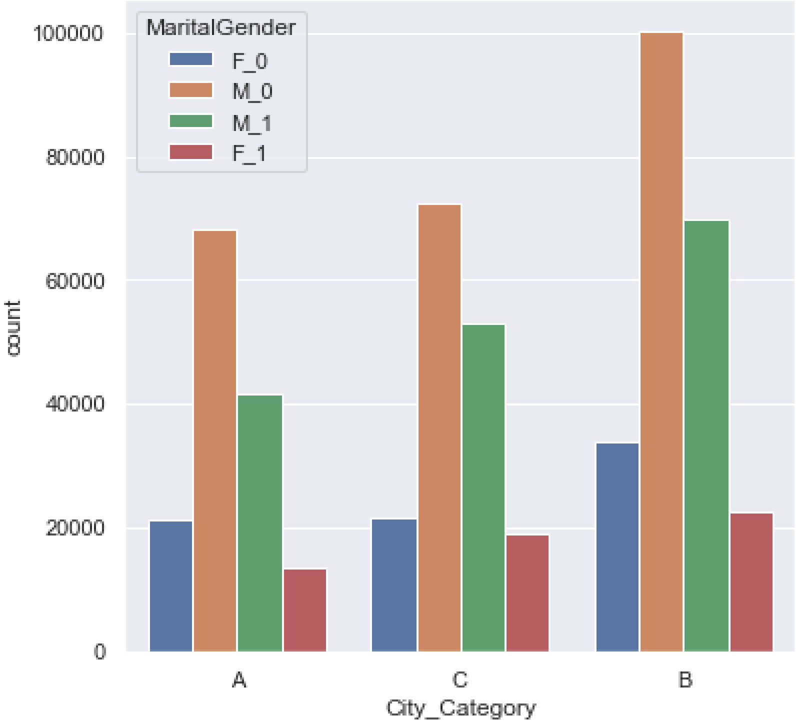
Out[87]:

```
<AxesSubplot:xlabel='Stay_In_Current_City_Years', ylabel='count'>
```



In [88]:
```
sns.countplot(x='City_Category',hue='MaritalGender',data=df)# city category on the basisof marital gender
```

Out[88]:
```
<AxesSubplot:xlabel='City_Category', ylabel='count'>
```



In [1]:

```
pip install Pandoc
```

```
Requirement already satisfied: Pandoc in c:\anaconda\lib\site-packages (2.2)Note: you may need to restart the kernel to
use updated packages.

Requirement already satisfied: ply in c:\anaconda\lib\site-packages (from Pandoc) (3.11)
Requirement already satisfied: plumbum in c:\anaconda\lib\site-packages (from Pandoc) (1.7.2)
Requirement already satisfied: pywin32 in c:\anaconda\lib\site-packages (from plumbum->Pandoc) (302)
```