Black Friday Sale-Exploratory Data Analysis

-Harsh Chaudhary

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sb
```

[2]:
```
df=pd.read_csv('BlackFriday.csv')
df.head()
```

[2]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	NaN	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	14.0	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	NaN	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	NaN	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	NaN	7969

[3]:
```
df.shape
```

[3]:
```
(537577, 12)
```

[4]:
```
df.info()
```

```
<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
```

[5]:
```
df.isnull().sum()
```

[5]:
```
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
```

[6]:
```
df.dropna()
```

[6]:
```
        User_ID Product_ID Gender    Age  Occupation City_Category  \
1       1000001  P00248942      F   0-17          10             A
6       1000004  P00184942      M  46-50           7             B
13      1000005  P00145042      M  26-35          20             A
14      1000006  P00231342      F  51-55           9             A
16      1000006   P0096642      F  51-55           9             A
…         …        …    …    …          …             …
537549  1004734  P00345842      M  51-55           1             B
537551  1004735  P00313442      M  46-50           3             C
537562  1004736  P00146742      M  18-25          20             A
537571  1004737  P00221442      M  36-45          16             C
537573  1004737  P00111142      M  36-45          16             C
```

```
        Stay_In_Current_City_Years  Marital_Status  Product_Category_1  \
1                                2               0                   1
6                                2               1                   1
13                               1               1                   1
14                               1               0                   5
16                               1               0                   2
…                              …               …                 …
537549                           1               1                   2
537551                           3               0                   5
537562                           1               1                   1
537571                           1               0                   1
537573                           1               0                   1

        Product_Category_2  Product_Category_3  Purchase
1                      6.0                14.0     15200
6                      8.0                17.0     19215
13                     2.0                 5.0     15665
14                     8.0                14.0      5378
16                     3.0                 4.0     13055
…                    …                 …       …
537549                 8.0                14.0     13082
537551                 6.0                 8.0      6863
537562                13.0                14.0     11508
537571                 2.0                 5.0     11852
537573                15.0                16.0     19196

[164278 rows x 12 columns]
```

[7]:
```
del df['Product_Category_2']
del df['Product_Category_3']
```

[8]:
```
print('-'*60)
print('Updated Shape of Dataset :',df.shape)
print('-'*60)
print('Updated Dataset :')
df.isnull().sum()
```

```
------------------------------------------------------------
Updated Shape of Dataset : (537577, 10)
------------------------------------------------------------
Updated Dataset :
```

[8]:
```
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
Purchase                      0
dtype: int64
```

1 Analyzing Columns

1.0.1 Column : User_ID

[9]:
```
df['User_ID'].nunique()
```

[9]:
```
5891
```

[10]:
```
df['User_ID'].unique()
```

[10]:
```
array([1000001, 1000002, 1000003, …, 1004113, 1005391, 1001529],
      dtype=int64)
```

1.0.2 Column : Product_ID

[11]:
```
df['Product_ID'].nunique()
```

[11]:
```
3623
```

[12]:
```
df['Product_ID'].unique()
```

[12]:
```
array(['P00069042', 'P00248942', 'P00087842', …, 'P00038842',
       'P00295642', 'P00091742'], dtype=object)
```

1.0.3 Column : Gender

[13]:
```
df['Gender'].unique()
```

[13]:
```
array(['F', 'M'], dtype=object)
```

1.0.4 Column : Age

[14]:
```
df['Age'].unique()
```

[14]:
```
array(['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25'],
      dtype=object)
```

1.0.5 Column : Occupation

[15]:
```
df['Occupation'].unique()
```

[15]:
```
array([10, 16, 15,  7, 20,  9,  1, 12, 17,  0,  3,  4, 11,  8, 19,  2, 18,
        5, 14, 13,  6], dtype=int64)
```

1.0.6 Column : City_Category

[16]:
```
df['City_Category'].unique()
```

[16]:
```
array(['A', 'C', 'B'], dtype=object)
```

1.0.7 Column : Stay_In_Current_City_Years

[17]:
```
df['Stay_In_Current_City_Years'].unique()
```

[17]:
```
array(['2', '4+', '3', '1', '0'], dtype=object)
```

1.0.8 Column : Marital_Status

[18]:
```
df['Marital_Status'].unique()
```

[18]:
```
array([0, 1], dtype=int64)
```

1.0.9 Column : Product_Category_1

[19]:
```
df['Product_Category_1'].unique()
```

[19]:
```
array([ 3,  1, 12,  8,  5,  4,  2,  6, 14, 11, 13, 15,  7, 16, 18, 10, 17,
        9], dtype=int64)
```

1.0.10 Column : Purchase

[20]:
```
# Total Amount spent
df['Purchase'].sum()
```

[20]:
```
5017668378
```

[21]:
```
# avg spent on 1 product
df['Purchase'].sum()/len(df['Purchase'])
```

[21]:
```
9333.859852635065
```

[22]:
```
# Automating
for i in df.columns:
```

```
    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
```

2 Analyzing Gender Column

2.0.1 Approach 1

[23]:
```
countM=0
for i in df['Gender']:
    if i=='M':
        countM+=1
countF=0
for i in df['Gender']:
    if i=='F':
        countF+=1
print('No. of Males : ',countM)
print('No. of Females : ',countF)
```

```
No. of Males :  405380
No. of Females :  132197
```

[24]:
```
data=pd.DataFrame({ 'Ratio' : [countM,countF]}, index=['Male','Female'])
```

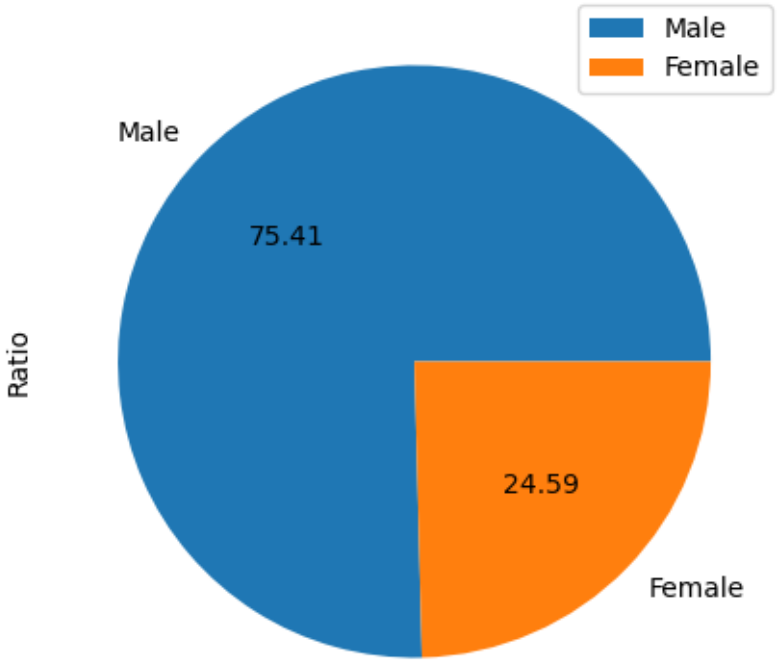
[25]:
```
data
```

[25]:

	Ratio
Male	405380
Female	132197

[26]:
```
data.plot(kind='pie', y='Ratio', autopct='%.2f', figsize=(5,5))
```

[26]:
```
<Axes: ylabel='Ratio'>
```



2.0.2 Approach 2

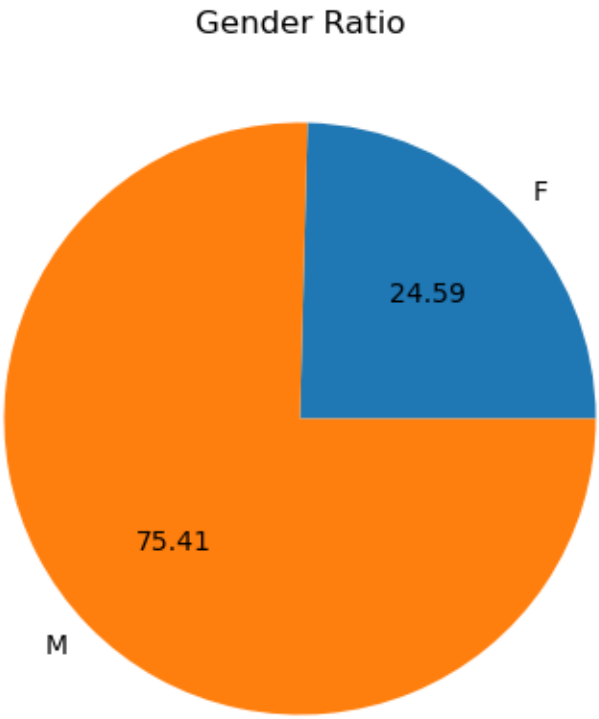
[27]:
```
df.groupby('Gender').size()
```

[27]:
```
Gender
F    132197
M    405380
dtype: int64
```

2.1 Pie Chart-Gender

[28]:
```
df.groupby('Gender').size().plot(kind = 'pie',
                                 autopct='%.2f',
                                 title='Gender Ratio',
                                 figsize=(5,5))
```

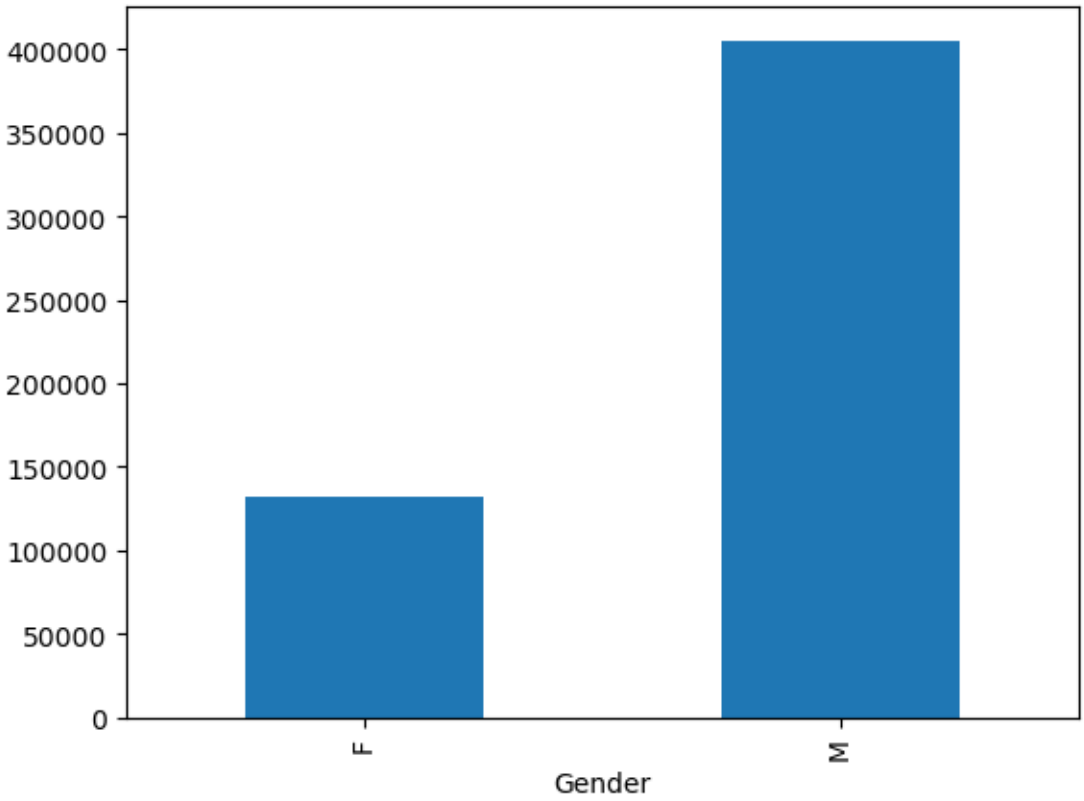
[28]:
```
<Axes: title={'center': 'Gender Ratio'}>
```



2.2 Bar Graph-Gender

[29]:
```
df.groupby('Gender').size().plot(kind='bar')
```

[29]:
```
<Axes: xlabel='Gender'>
```



2.2.1 % of men & women individually spending on purchase

[30]:
```
#df.groupby('Gender').sum()['Purchase'].plot(kind = 'pie', autopct = "%0.1f")
#Not working
```

3 Age Column

[31]:
```
df.head(5)
```

[31]:

	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				
4	1000002	P00285442	M	55+	16	C				

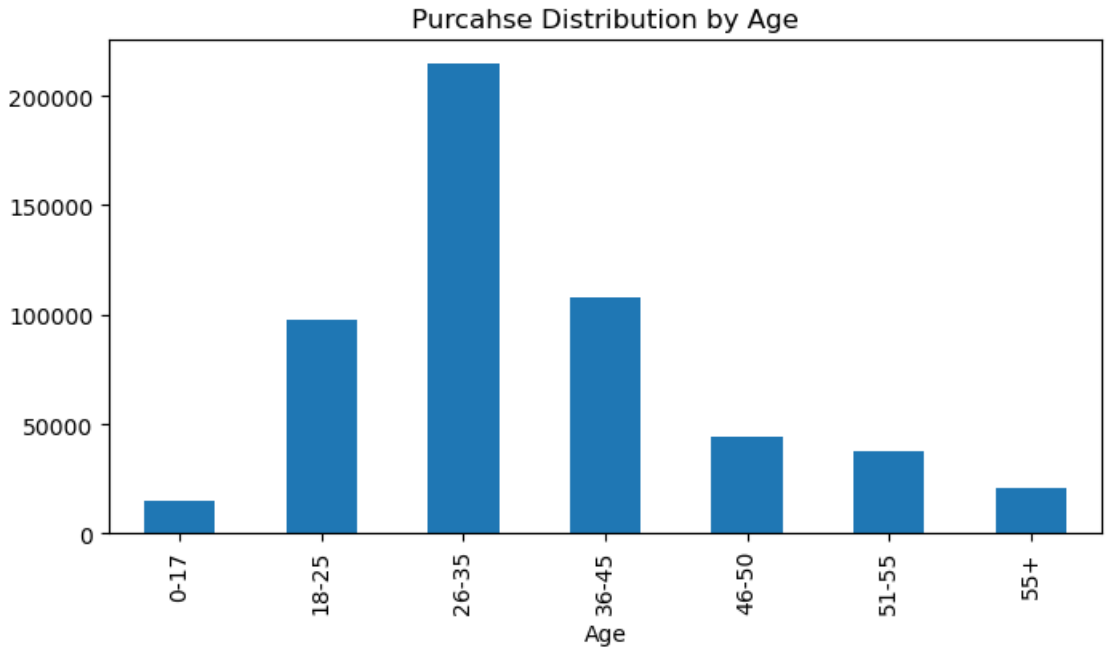
```
3                          2                  0                  12        1057
4                         4+                  0                   8        7969
```

[32]:
```
df.groupby('Age').size()
```

[32]:
```
Age
0-17       14707
18-25      97634
26-35     214690
36-45     107499
46-50      44526
51-55      37618
55+        20903
dtype: int64
```

[33]:
```
df.groupby('Age').size().plot(kind='bar', figsize=(8,4), title='Purcahse␣
↪Distribution by Age')
```

[33]:
```
<Axes: title={'center': 'Purcahse Distribution by Age'}, xlabel='Age'>
```



[34]:
```
len(df[df['Age']=='0-17'])
```

[34]:
```
14707
```

[35]:
```
lst=[]

for i in df['Age'].unique():
    lst.append([i,df[df['Age']==i]['Product_ID'].nunique()])
lst
```

[35]:
```
[['0-17', 2300],
 ['55+', 2573],
 ['26-35', 3419],
 ['46-50', 3099],
 ['51-55', 2877],
 ['36-45', 3318],
 ['18-25', 3213]]
```

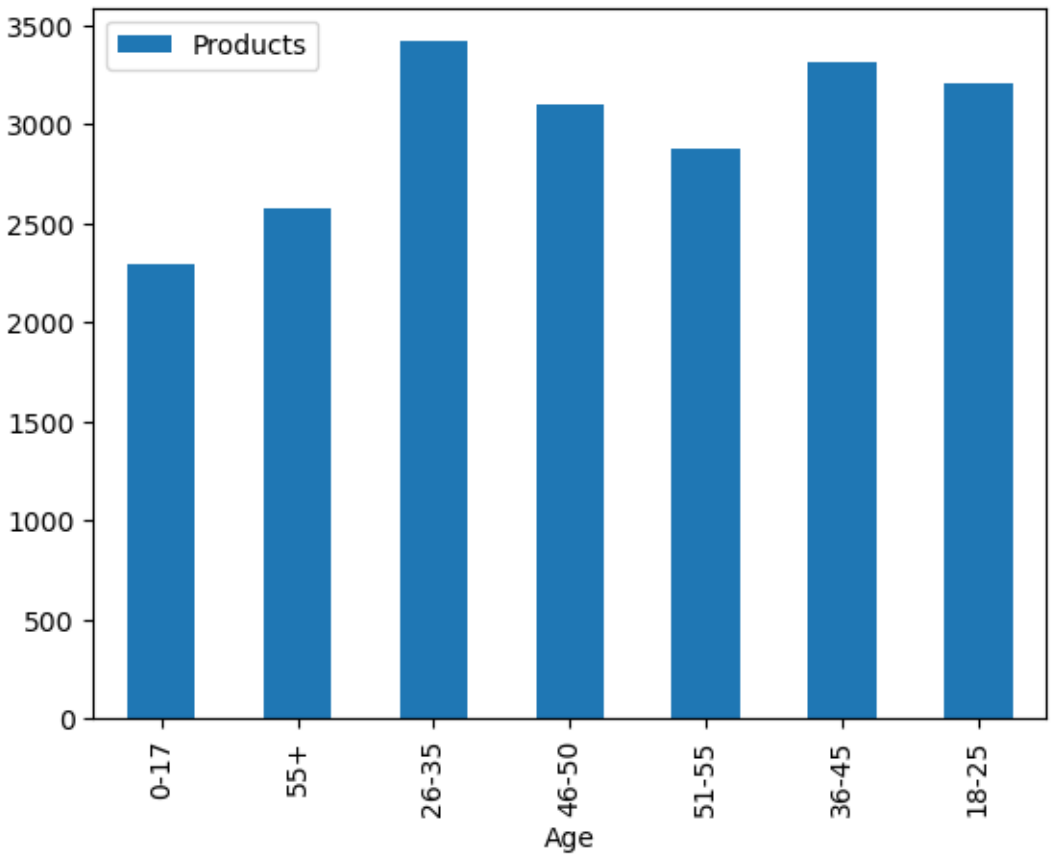
[36]:
```
data=pd.DataFrame(lst, columns=['Age','Products'])
data
```

[36]:

	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

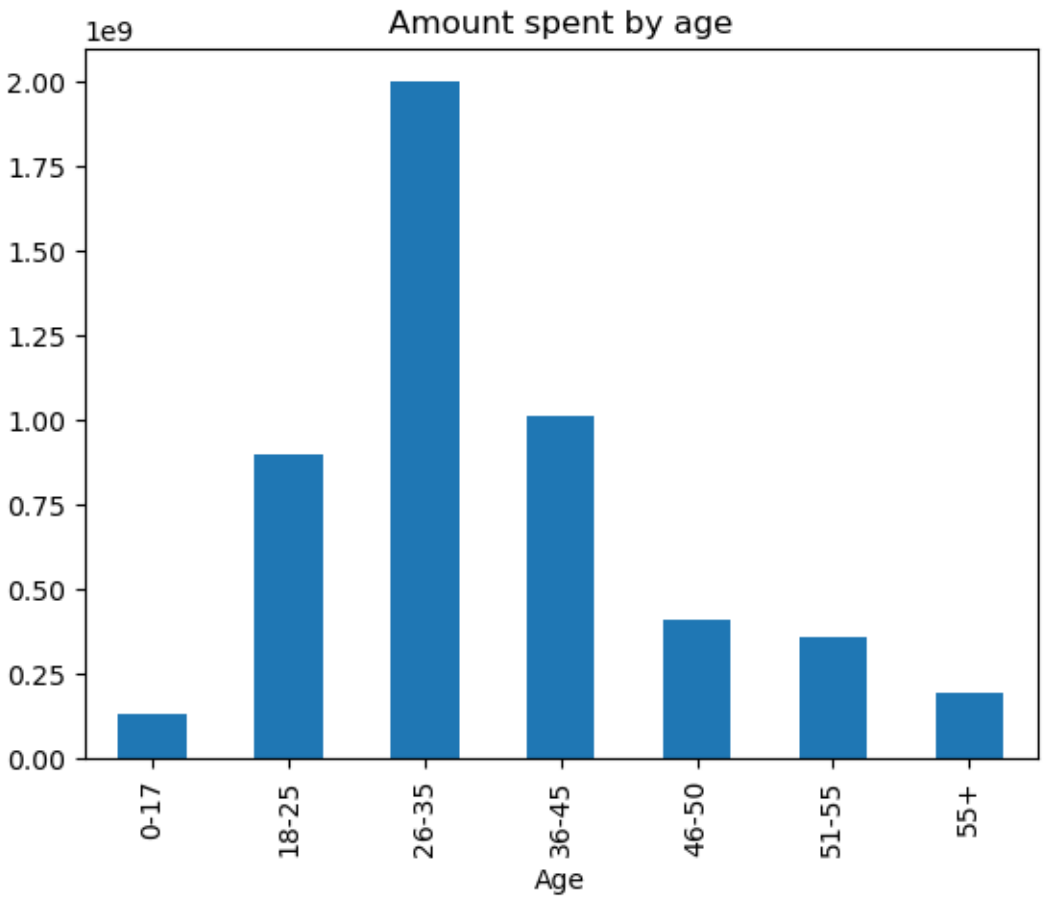
[37]:
```
data.plot(kind='bar', x='Age', y='Products')
```

[37]:
```
<Axes: xlabel='Age'>
```



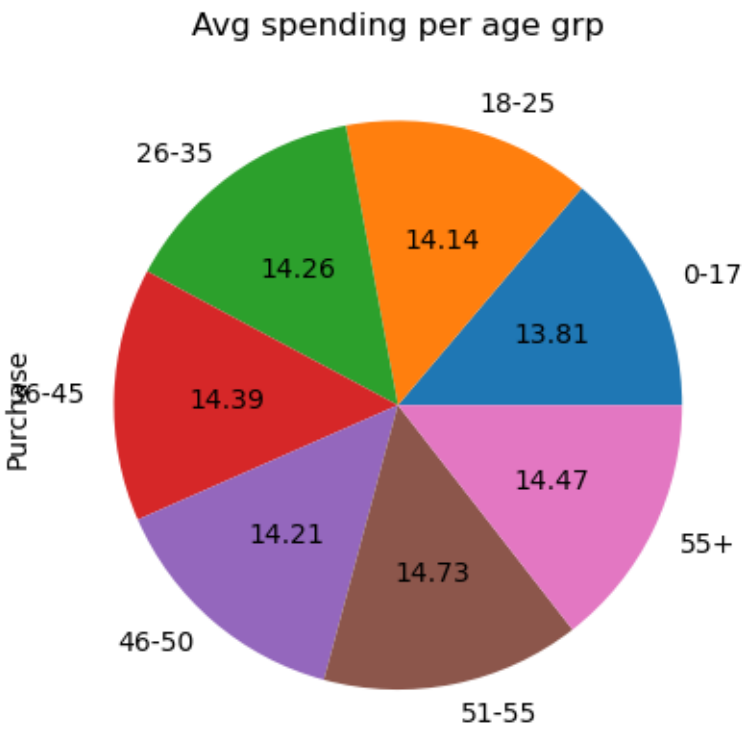
[38]:
```
df.groupby('Age').sum()['Purchase'].plot(kind='bar', title='Amount spent by␣
↪age')
```

[38]: <Axes: title={'center': 'Amount spent by age'}, xlabel='Age'>



[39]:
```
# Average spendings per age group
df.groupby('Age')['Purchase'].mean().plot(kind='pie', title='Avg spending per␣
 ↪age grp', autopct="%.2f")
```

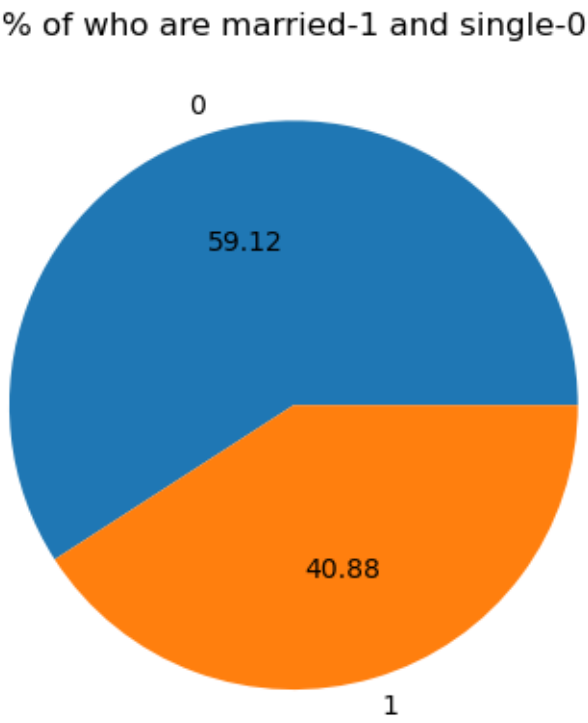
[39]: <Axes: title={'center': 'Avg spending per age grp'}, ylabel='Purchase'>



4 Marital_status column

[40]:
```
df.groupby('Marital_Status').size().plot(kind='pie', title='% of who are␣
↪married-1 and single-0', autopct="%.2f")
```

[40]: <Axes: title={'center': '% of who are married-1 and single-0'}>

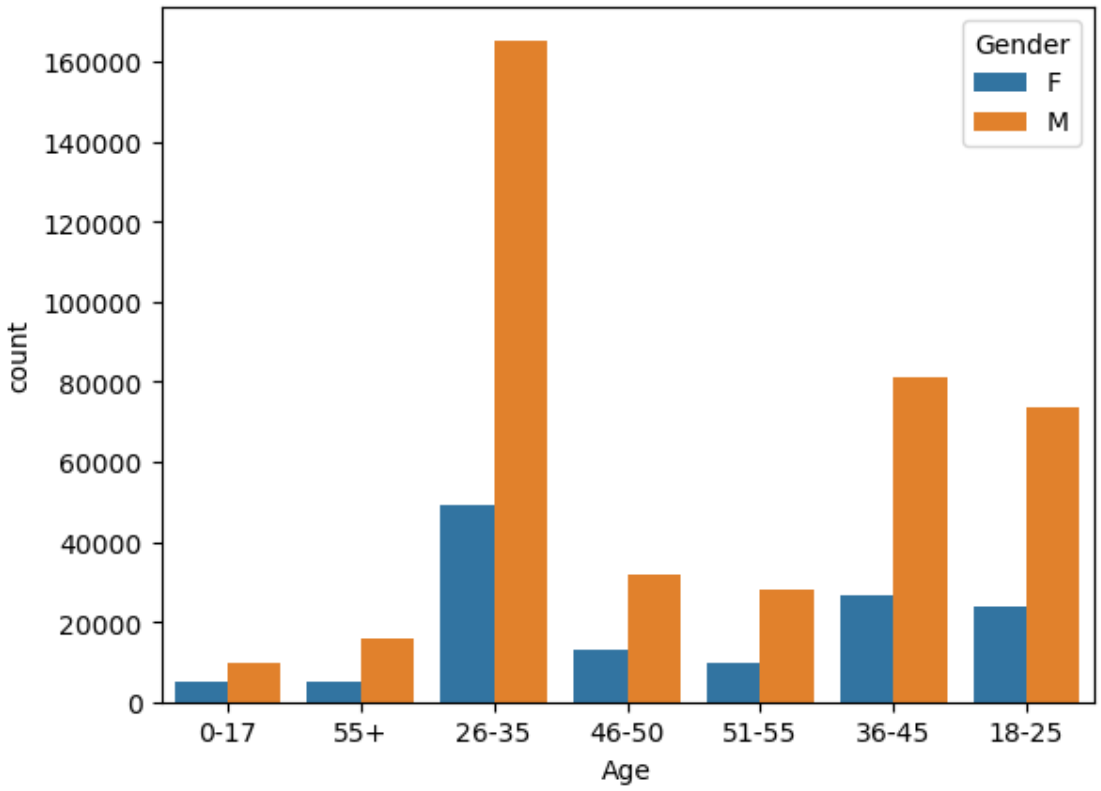


5 Multicolumn Analysis

5.0.1 Based on Age , count of men & women

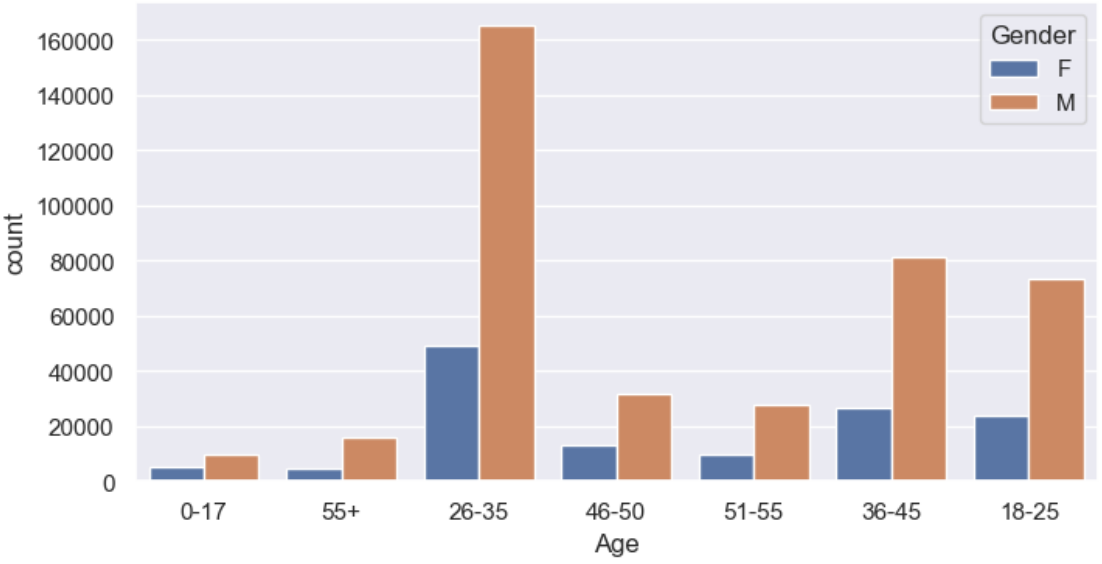
[41]:
```
sb.countplot(x='Age', hue='Gender', data=df)
```

[41]:
```
<Axes: xlabel='Age', ylabel='count'>
```



[42]:
```
sb.set(rc={'figure.figsize': (8,4)})
sb.countplot(x='Age', hue='Gender', data=df)
```

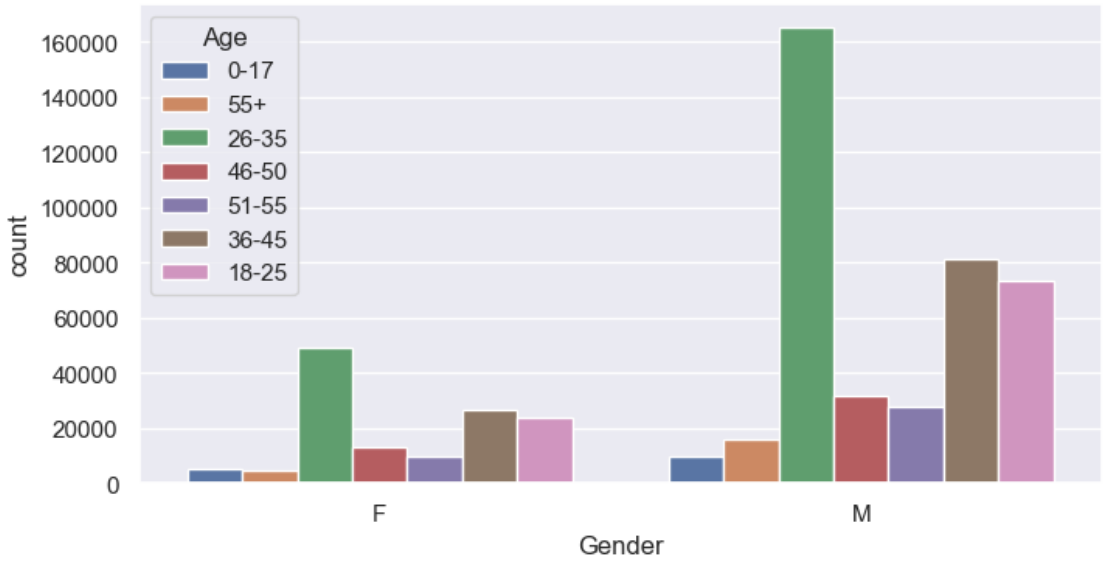
[42]: <Axes: xlabel='Age', ylabel='count'>



5.0.2 Based on Gender , count of age groups

[43]:
```
sb.countplot(x='Gender', hue='Age', data=df)
```

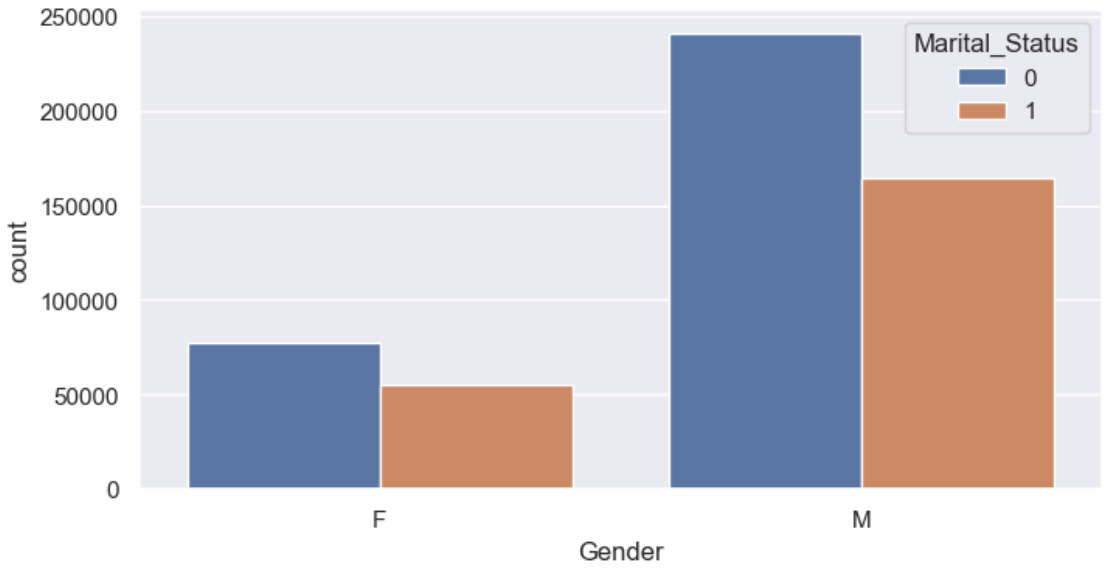
[43]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



5.0.3 Based on Gender , maital staus of M & F

[44]:
```
sb.countplot(x='Gender', hue='Marital_Status', data=df)
```

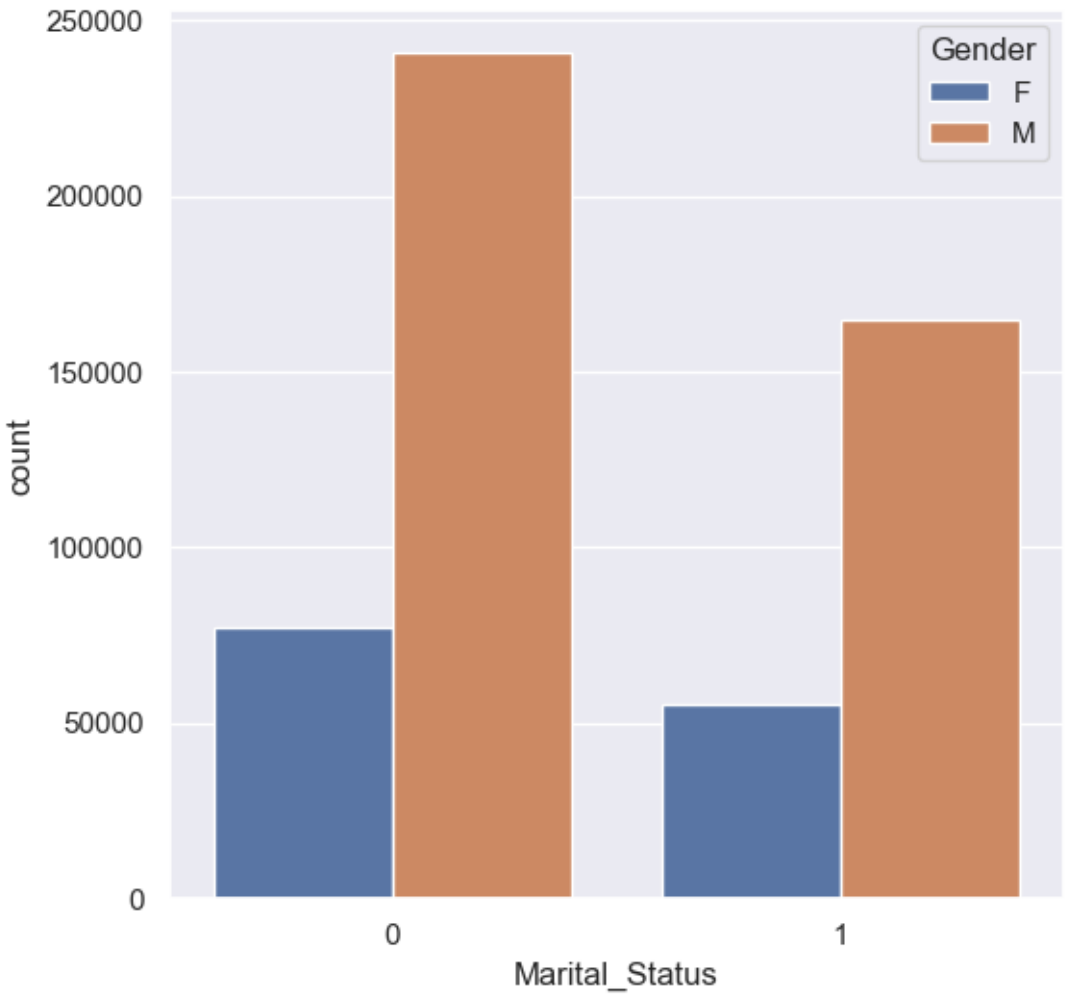
[44]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



5.0.4 Based on Marital status, count of Gender

[45]:
```
sb.set(rc={'figure.figsize':(6,6)})
sb.countplot(x='Marital_Status', hue='Gender', data=df)
```

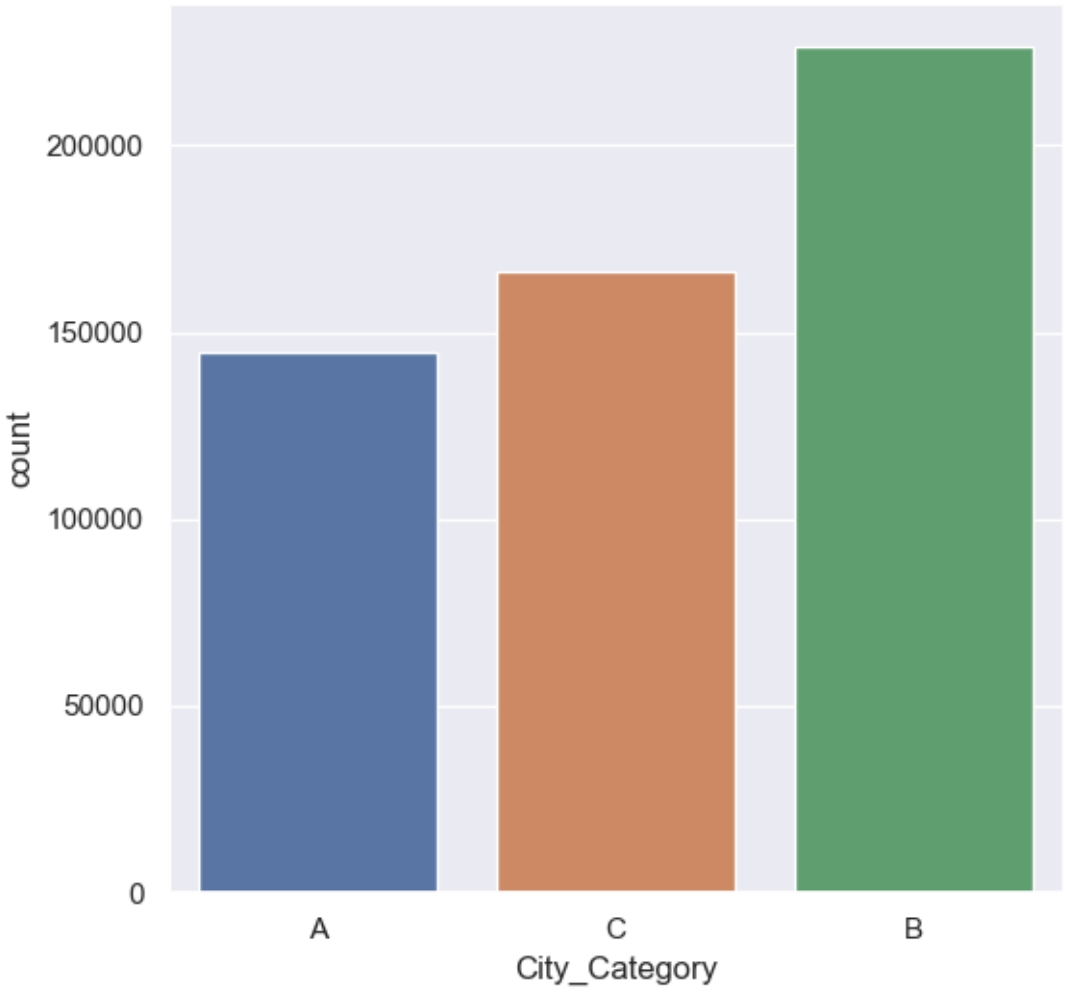
[45]:
```
<Axes: xlabel='Marital_Status', ylabel='count'>
```



6 City_Category column

[46]:
```
sb.countplot(x='City_Category', data=df)
```

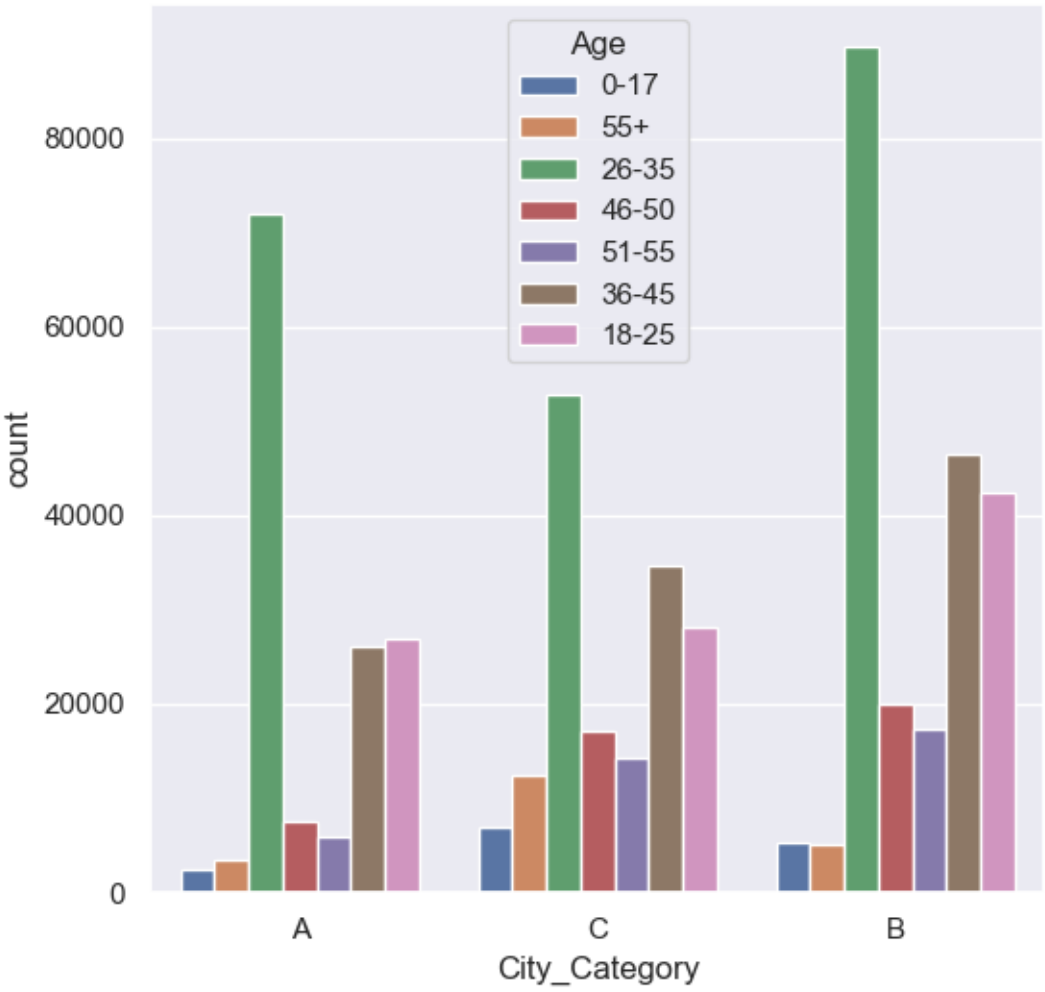
[46]:
```
<Axes: xlabel='City_Category', ylabel='count'>
```



6.0.1 Age group belonging to what all city's

[47]:
```
sb.countplot(x='City_Category', hue='Age', data=df)
```

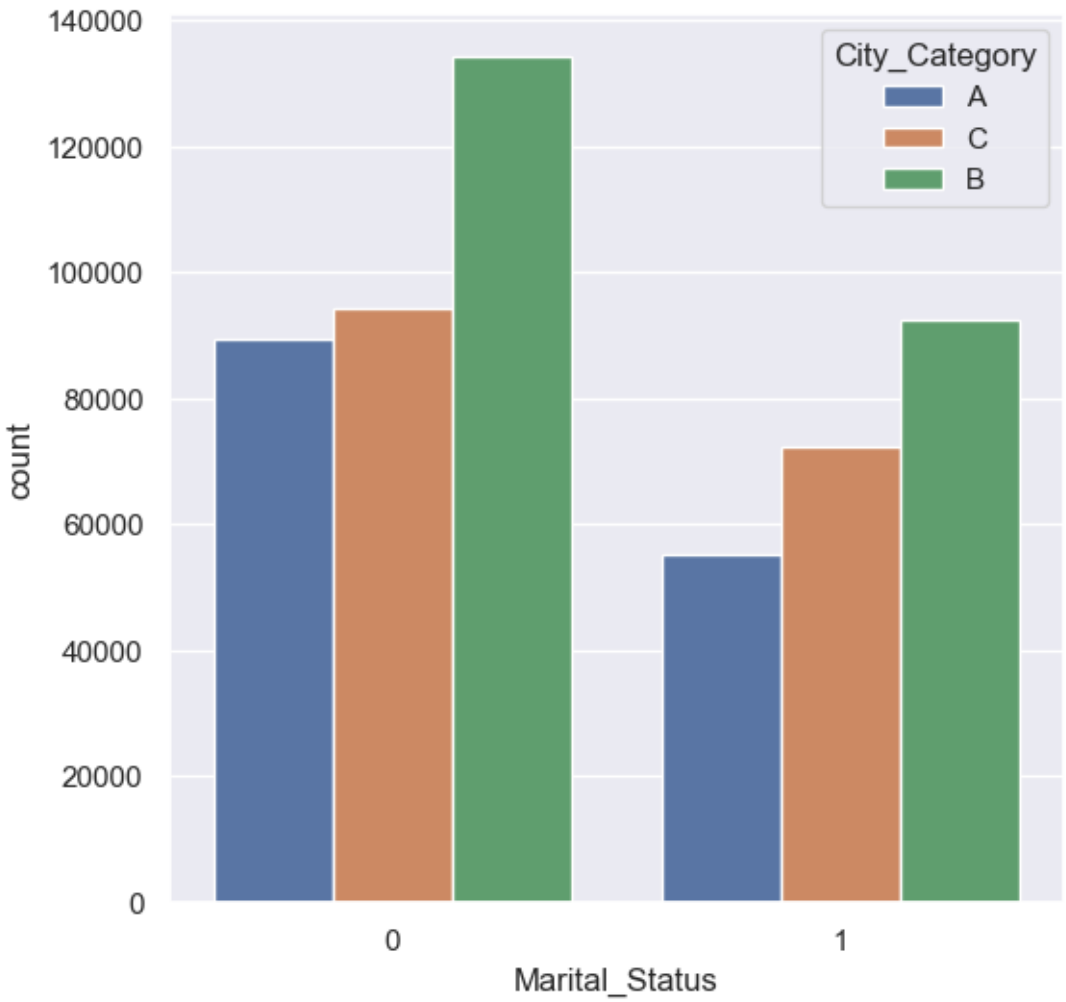
[47]:
```
<Axes: xlabel='City_Category', ylabel='count'>
```



6.0.2 Marital status based on city's

[48]:
```
sb.countplot(x='Marital_Status', hue='City_Category', data=df)
```

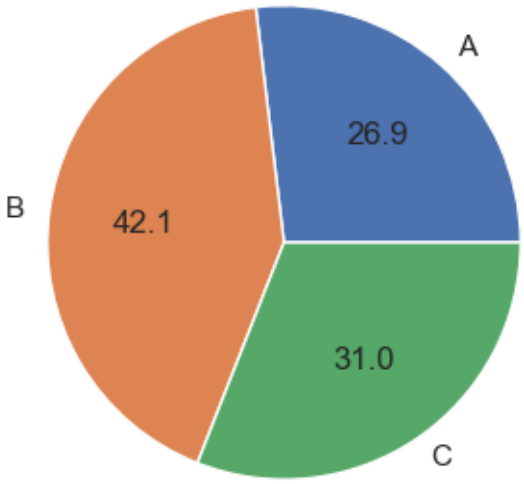
[48]:
```
<Axes: xlabel='Marital_Status', ylabel='count'>
```



6.0.3 Pie-chart : City_Category distribution

[49]:
```
df.groupby('City_Category').size().plot(kind='pie', autopct='%.1f',␣
↪figsize=(4,4))
```

[49]: <Axes: >



6.0.4 Count of Gender's in each city

[50]:
```
sb.countplot(x = 'City_Category', hue = 'Gender', data = df)
```

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

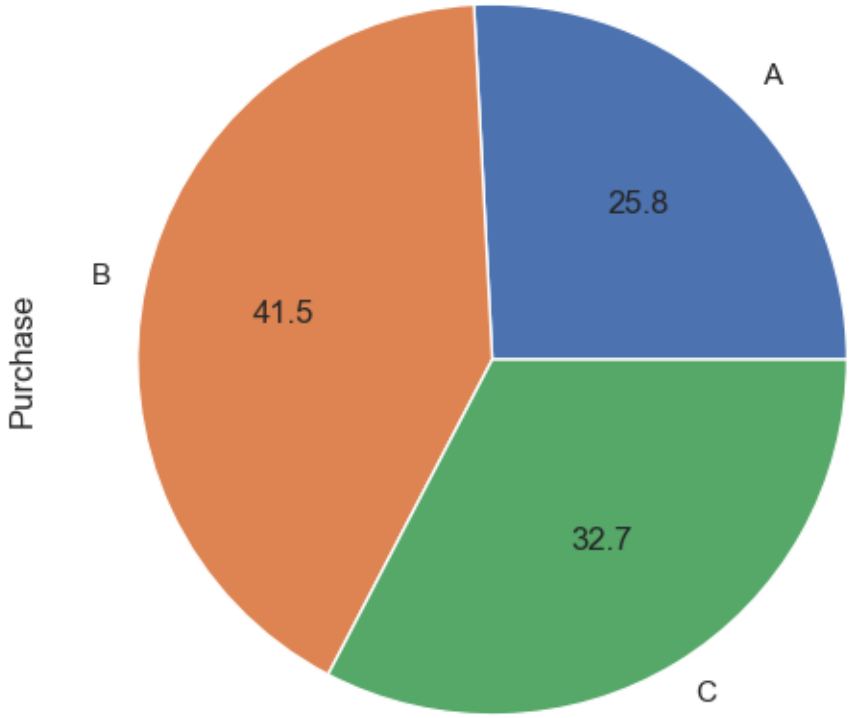
6.1 Amount spent in city

[51]:
```
df.groupby('City_Category').size()
```

[51]:
```
City_Category
A    144638
B    226493
C    166446
dtype: int64
```

[52]:
```
df.groupby('City_Category').sum()['Purchase'].plot(kind='pie', autopct='%.1f')
```

[52]:
```
<Axes: ylabel='Purchase'>
```



6.2 Avg spending by cutomers in each city

[57]:
```
#df.groupby('City_Category').mean()['Purchase'].plot(kind='pie', autopct='%.1f')
```

[56]:
```
df.head(10)
```

[56]:

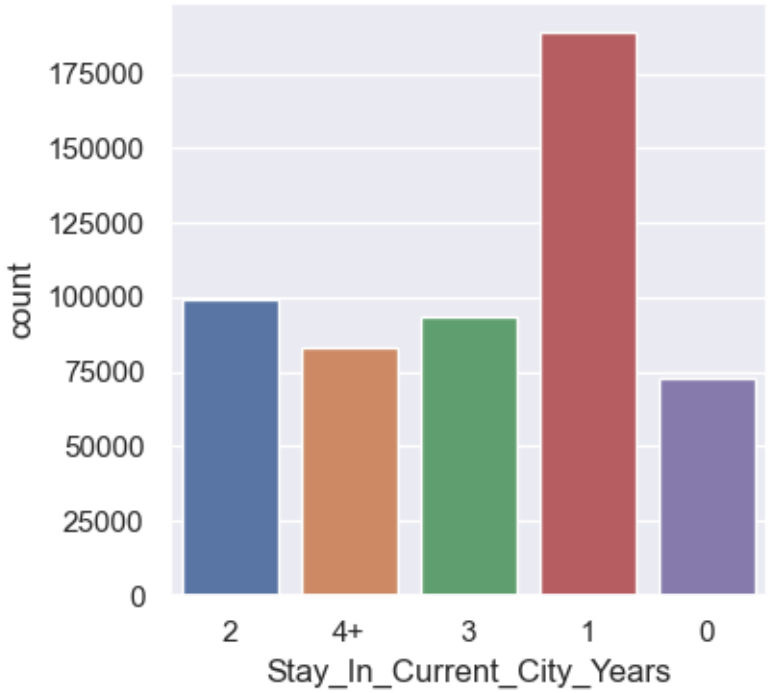
	User_ID	Product_ID	Gender	Age	Occupation	City_Category
0	1000001	P00069042	F	0-17	10	A
1	1000001	P00248942	F	0-17	10	A
2	1000001	P00087842	F	0-17	10	A
3	1000001	P00085442	F	0-17	10	A
4	1000002	P00285442	M	55+	16	C
5	1000003	P00193542	M	26-35	15	A
6	1000004	P00184942	M	46-50	7	B
7	1000004	P00346142	M	46-50	7	B
8	1000004	P0097242	M	46-50	7	B
9	1000005	P00274942	M	26-35	20	A

	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	2	0	3	8370
1	2	0	1	15200
2	2	0	12	1422
3	2	0	12	1057
4	4+	0	8	7969
5	3	0	1	15227
6	2	1	1	19215
7	2	1	1	15854
8	2	1	1	15686
9	1	1	8	7871

7 Stay_In_Current_City_Years Coulmn

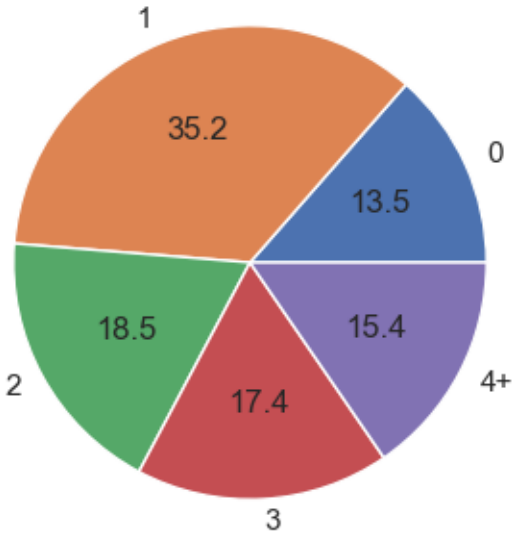
[60]:
```
sb.set(rc={'figure.figsize':(4,4)})
sb.countplot(x='Stay_In_Current_City_Years', data=df)
```

[60]: <Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>



[68]:
```
df.groupby('Stay_In_Current_City_Years').size().plot(kind='pie', autopct="%.
↪1f",)
```

[68]: <Axes: >



7.0.1 How many Years are Male and female staying in city

[69]:
```
sb.countplot(x='Stay_In_Current_City_Years', hue='Gender', data=df)
```

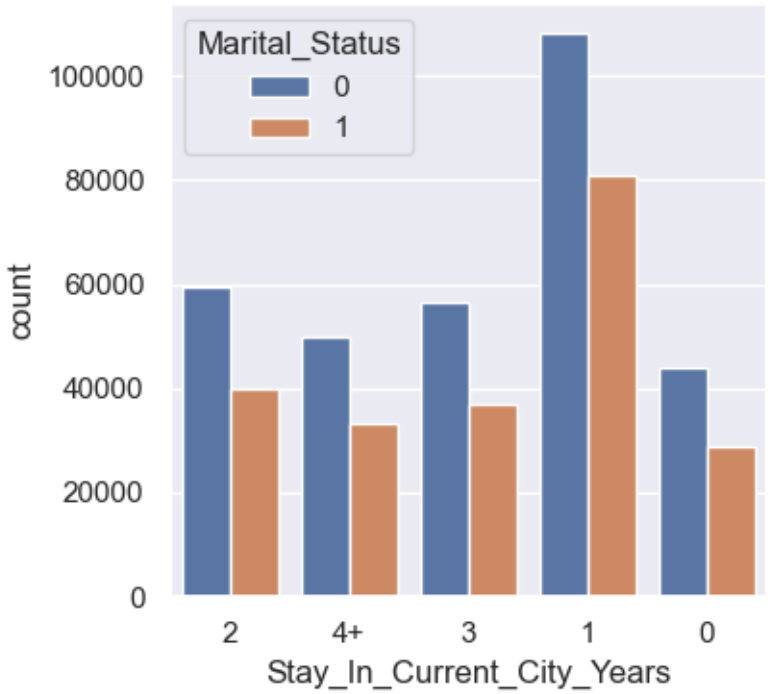
[69]:
```
<Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>
```

Males are staying more than females for 1 year

7.0.2 How many of them are married or bachelor

[71]:
```
sb.countplot(x='Stay_In_Current_City_Years', hue='Marital_Status', data=df)
```

[71]:
```
<Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>
```

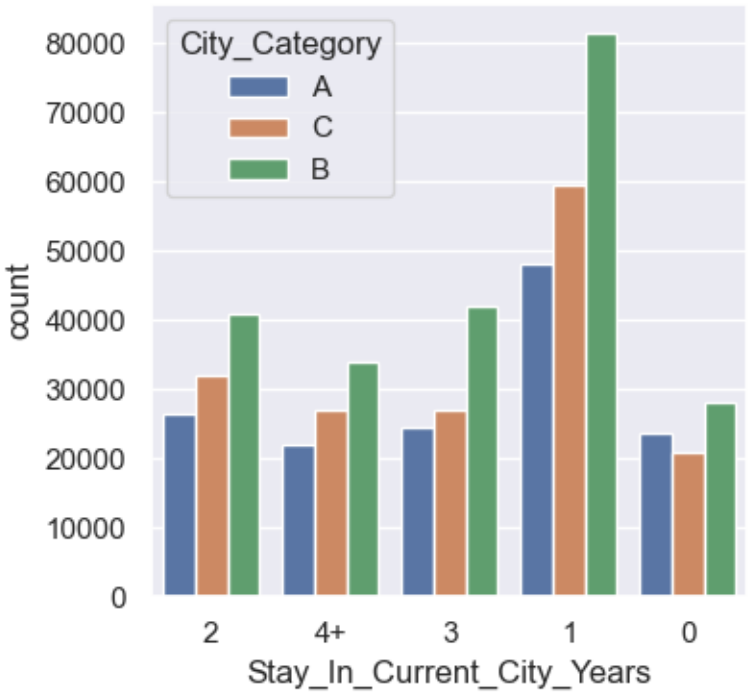


MOst of the men who are staying for 1 year are bachelor's–>target audience

7.0.3 In which city , Men are staying for 1 year

[72]:
```
sb.countplot(x='Stay_In_Current_City_Years', hue='City_Category', data=df)
```

[72]:
```
<Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>
```

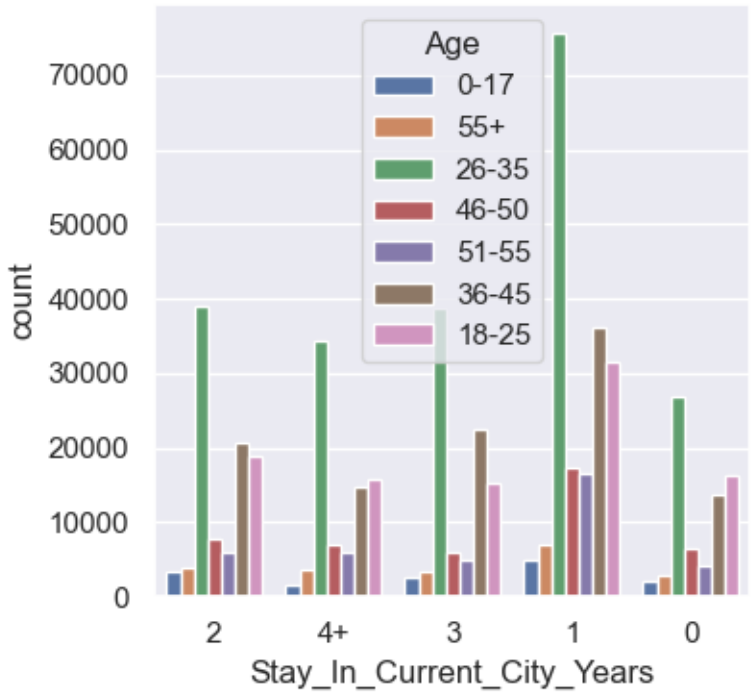


In City B , most of our target audience lies

7.0.4 In this , What is the age grp spending most

[73]:
```
sb.countplot(x='Stay_In_Current_City_Years', hue='Age', data=df)
```

[73]:
```
<Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>
```



7.1 Conclusion :

Target audience : staying-1yr,men,bachelor,city B, 26-35yrs old

7.2 Occupation Column

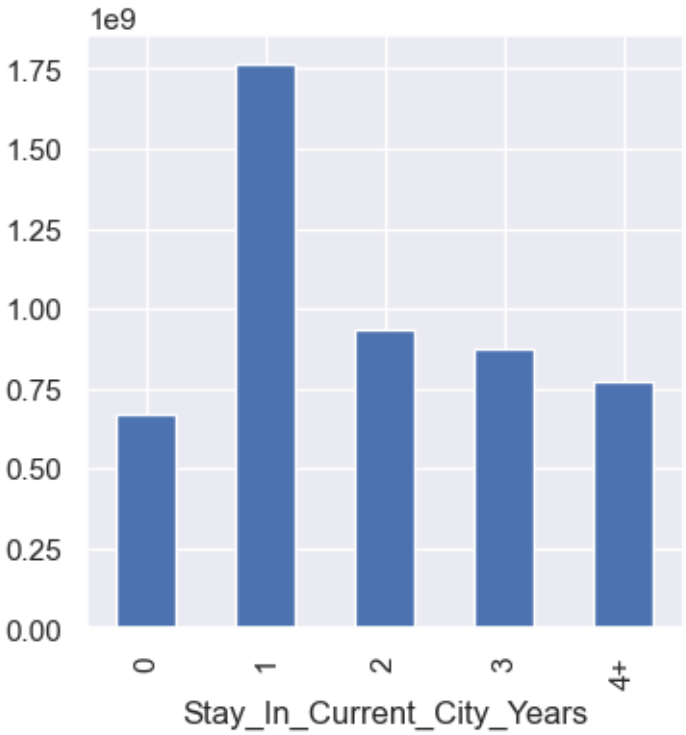
7.2.1 Total spending 's

[74]:
```
df.groupby('Stay_In_Current_City_Years').size()
```

[74]:
```
Stay_In_Current_City_Years
0      72725
1     189192
2      99459
3      93312
4+     82889
dtype: int64
```

[76]:
```
df.groupby('Stay_In_Current_City_Years').sum()['Purchase'].plot(kind='bar')
```

[76]:
```
<Axes: xlabel='Stay_In_Current_City_Years'>
```

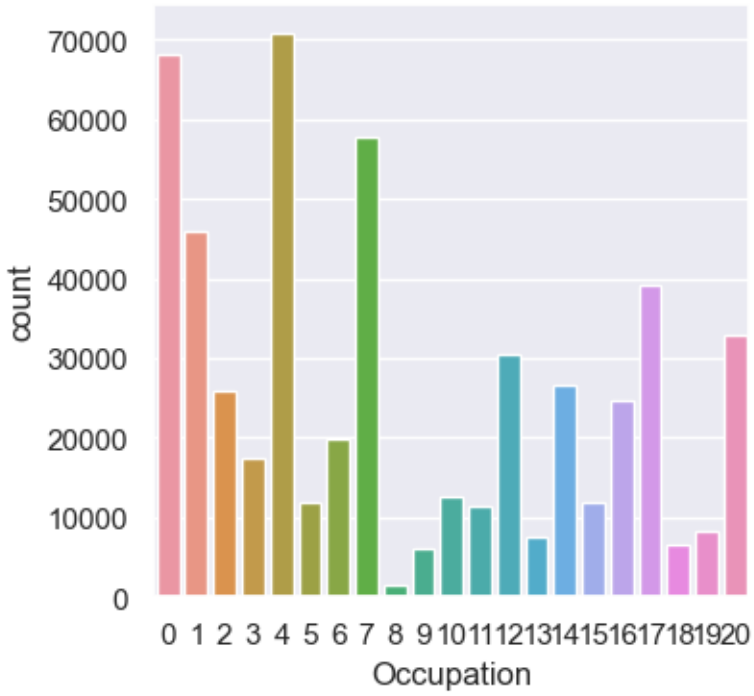


[ ]:
```
## MEAN()
```

8 Occupation Column

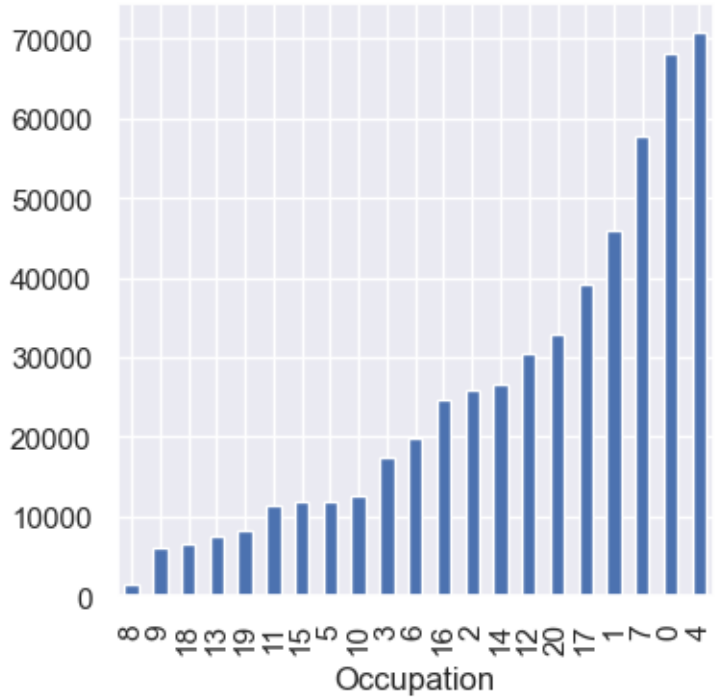
[77]:
```
sb.countplot(x=df['Occupation'])
```

[77]:
```
<Axes: xlabel='Occupation', ylabel='count'>
```



[78]:
```
df.groupby('Occupation').size().sort_values().plot(kind='bar')
```

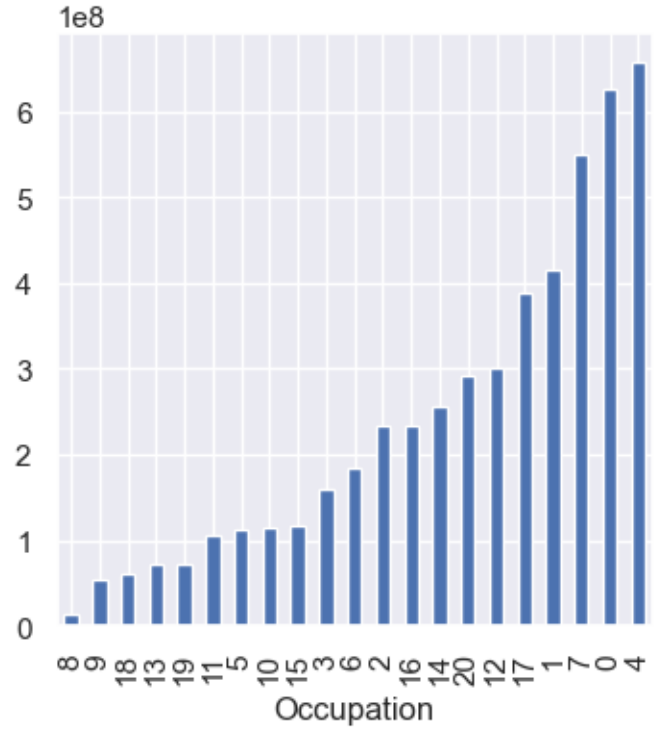
[78]: <Axes: xlabel='Occupation'>



occupation 4 spends more

[80]:
```
df.groupby('Occupation').sum()['Purchase'].sort_values().plot(kind='bar')
```

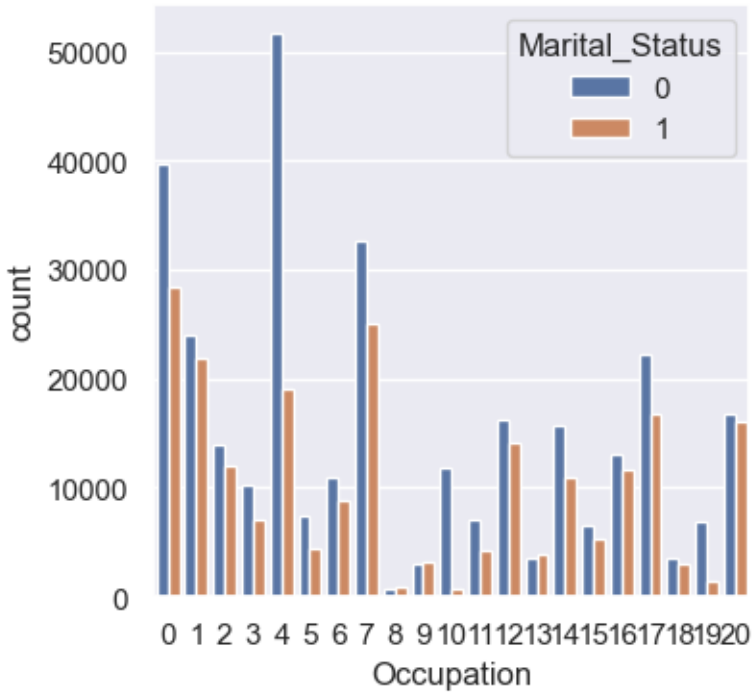
[80]:
```
<Axes: xlabel='Occupation'>
```



8.1 Marital status of occupation

[81]:
```
sb.countplot(x='Occupation', hue='Marital_Status', data=df)
```

[81]:
```
<Axes: xlabel='Occupation', ylabel='count'>
```

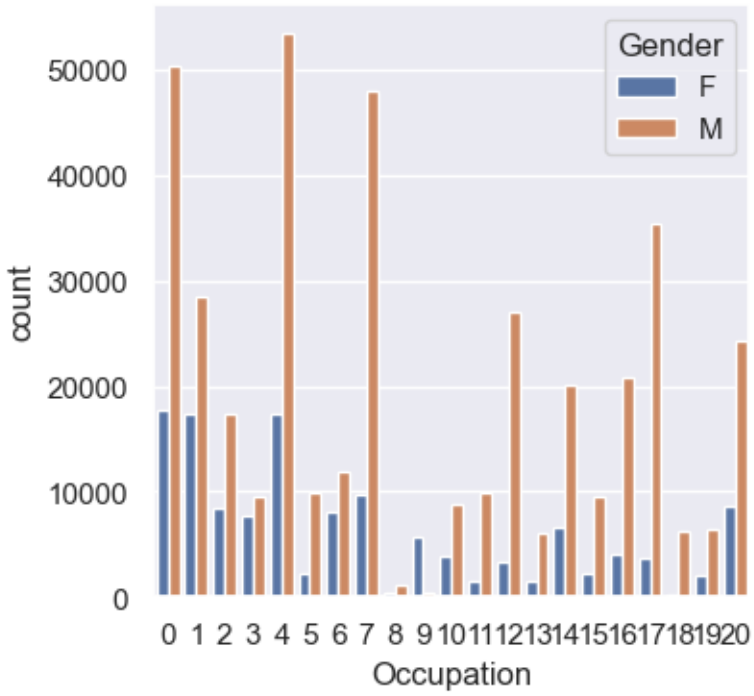


More bachelor's in all occupations

8.1.1 Gender's divison in Occupation

[82]:
```
sb.countplot(x='Occupation', hue='Gender', data=df)
```

[82]:
```
<Axes: xlabel='Occupation', ylabel='count'>
```

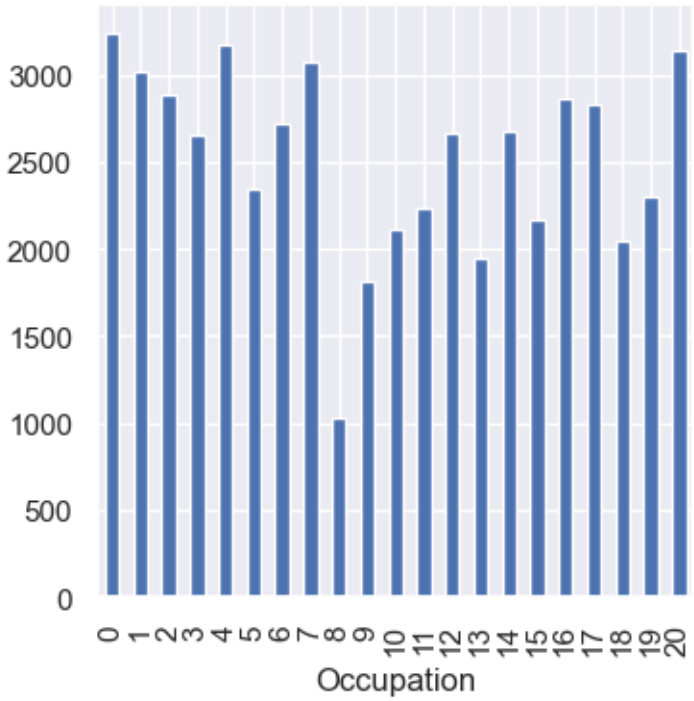


more dominance of men in all occupation except in 9

8.2 Different productsID in all occupations

[84]:
```
df.groupby('Occupation').nunique()['Product_ID'].plot(kind='bar')
```

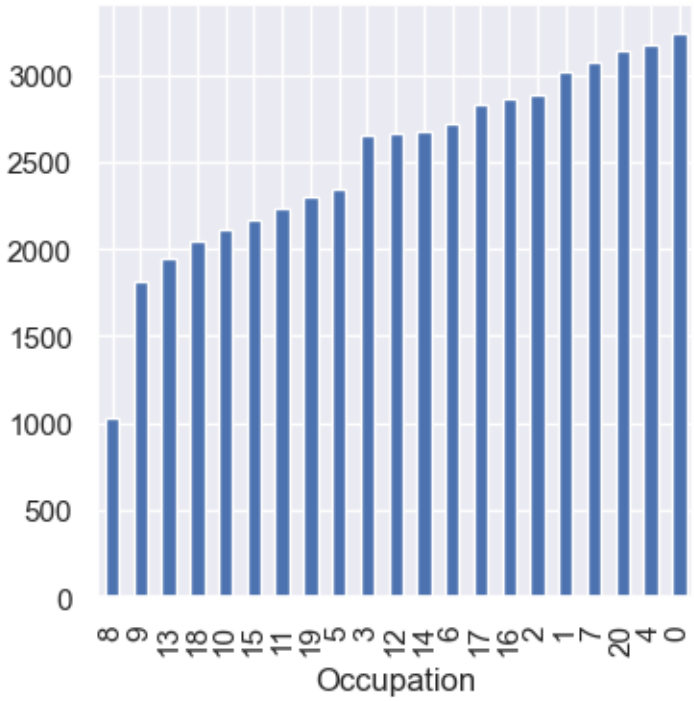
[84]:
```
<Axes: xlabel='Occupation'>
```



8.2.1 sorting

[85]:
```
df.groupby('Occupation').nunique()['Product_ID'].sort_values().plot(kind='bar')
```

[85]:
```
<Axes: xlabel='Occupation'>
```



0 productID is having highest quantity being sold use mean() to find which productID is expensive and which is cheap

9 Product_Category_1 Column

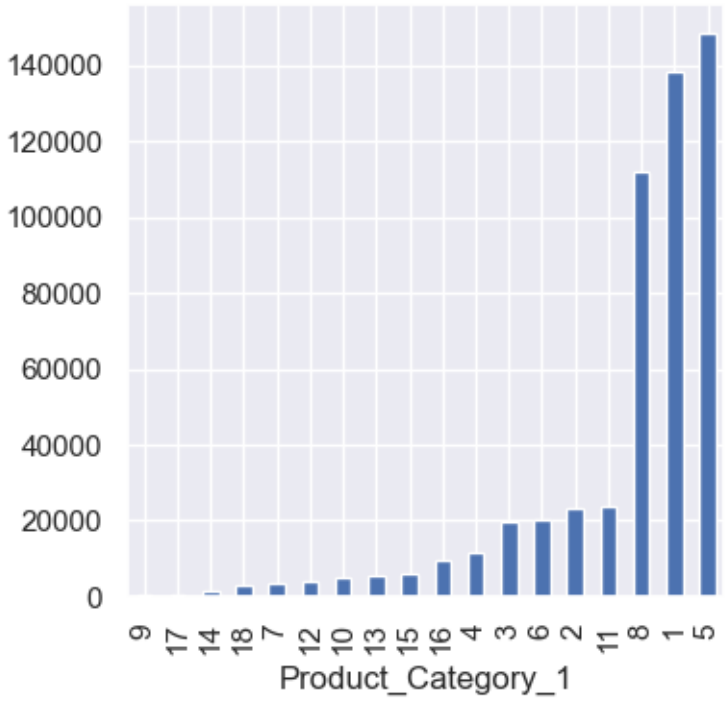
[87]:
```
df.groupby('Product_Category_1').size().plot(kind = 'bar')
```

[87]:
```
<Axes: xlabel='Product_Category_1'>
```

sorting:

[88]:
```
df.groupby('Product_Category_1').size().sort_values().plot(kind='bar')
```

[88]:
```
<Axes: xlabel='Product_Category_1'>
```

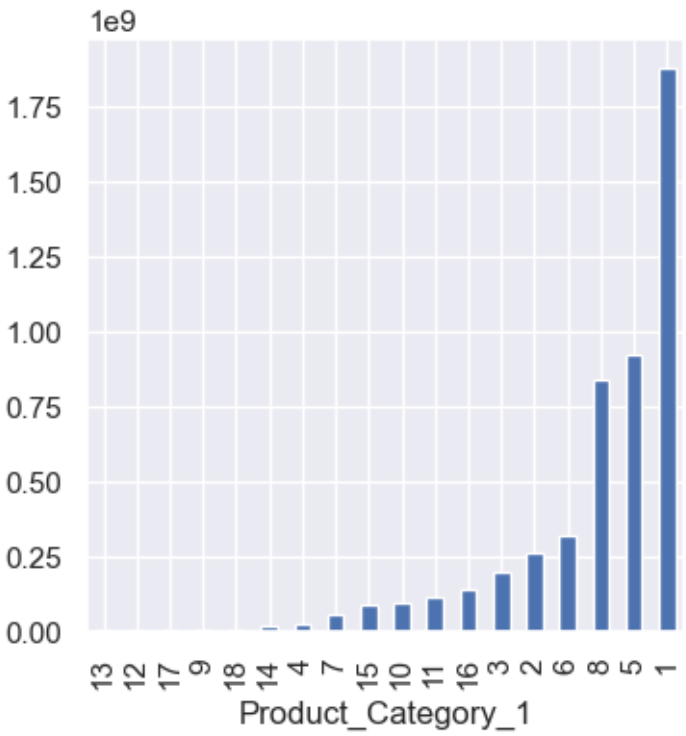


Prodct 5 having largest stocks being sold

9.0.1 Amount generation from each element of product_category_1

[90]:
```
df.groupby('Product_Category_1').sum()['Purchase'].sort_values().
↪plot(kind='bar')
```

[90]: <Axes: xlabel='Product_Category_1'>

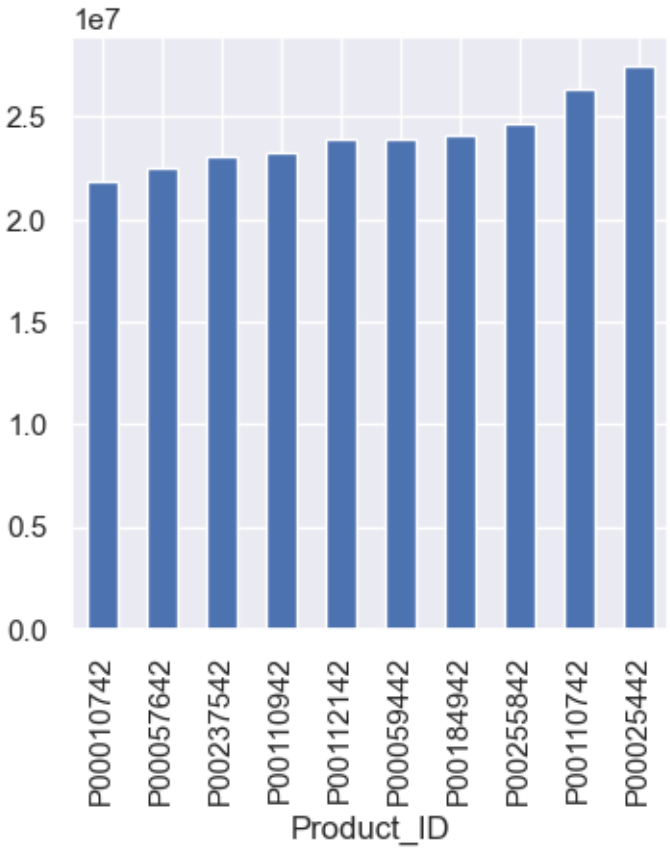


product 1 generating the highest revenue

9.0.2 Top 10 largest amount generating products , whose individual cost is high

[91]:
```
df.groupby('Product_ID').sum()['Purchase'].nlargest(10).sort_values().plot(kind␣
↪= 'bar')
```

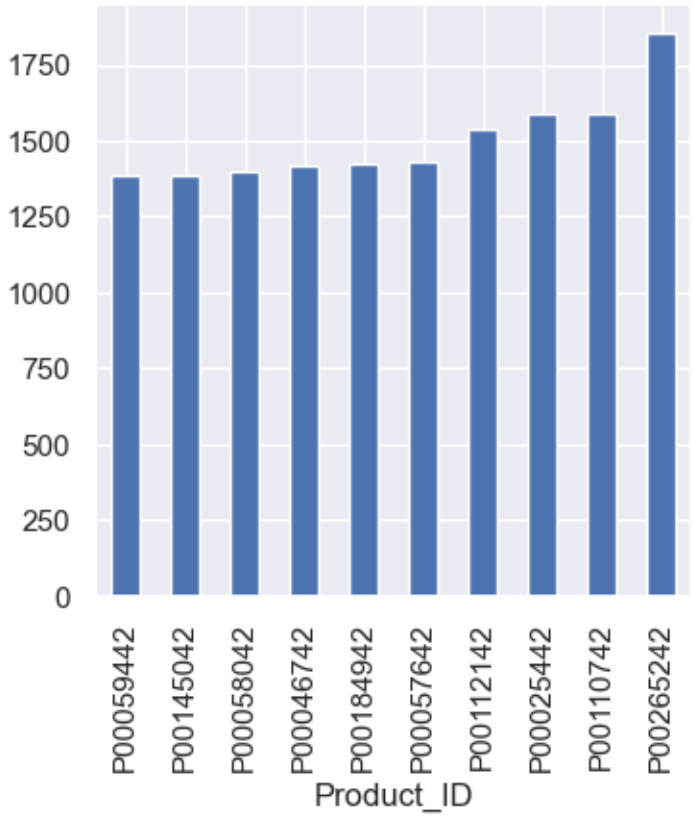
[91]:
```
<Axes: xlabel='Product_ID'>
```



9.1 Product selling the most but individual cost - cheap

[92]:
```
df.groupby('Product_ID').size().nlargest(10).sort_values().plot(kind = 'bar')
```

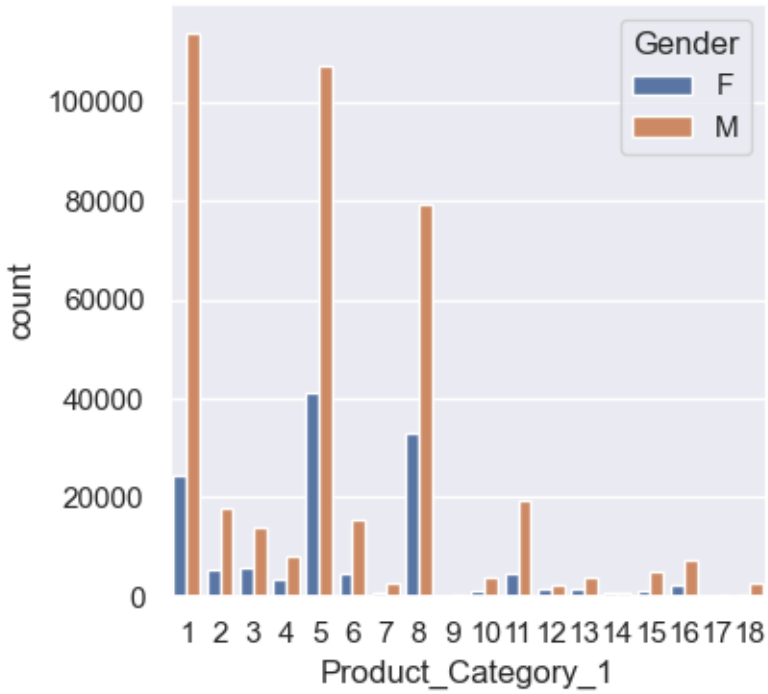
[92]:
```
<Axes: xlabel='Product_ID'>
```



[98]:
```
## Gender ratio in Product_categroy_1
```

[94]:
```
sb.countplot(x = 'Product_Category_1', hue = 'Gender', data = df)
```

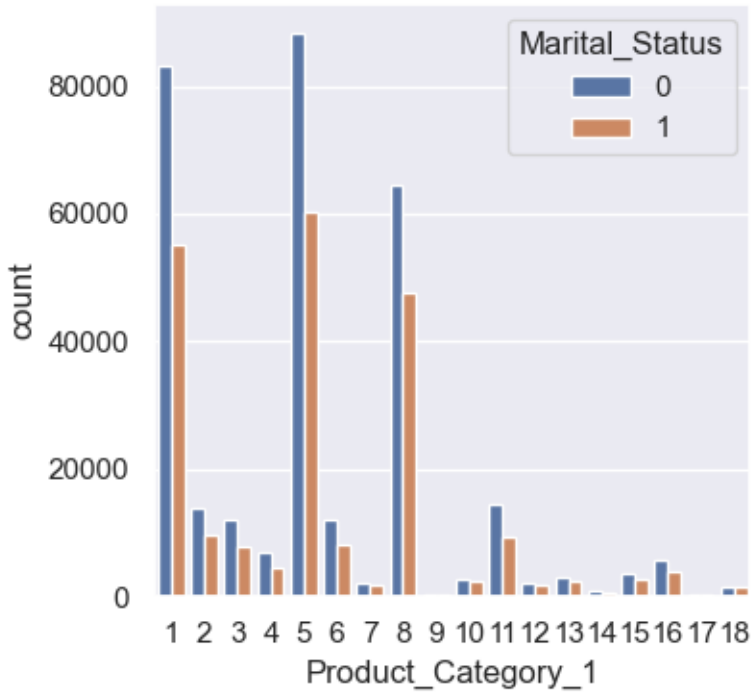
[94]:
```
<Axes: xlabel='Product_Category_1', ylabel='count'>
```



most of men in product_Cat_1

[96]:
```
sb.countplot(x = 'Product_Category_1', hue = 'Marital_Status', data = df)
```

[96]:
```
<Axes: xlabel='Product_Category_1', ylabel='count'>
```



most of the unmaaried/bachelor are buying pr_cat_1

10 Combining Gender & Marital_Status

[99]:
```
lst=[]
for i in range(len(df)):
    lst.append(df['Gender'][i]+'_'+str(df['Marital_Status'][i]))

df['MaritalGender']=lst
```

[100]:
```
df.head(5)
```

[100]:

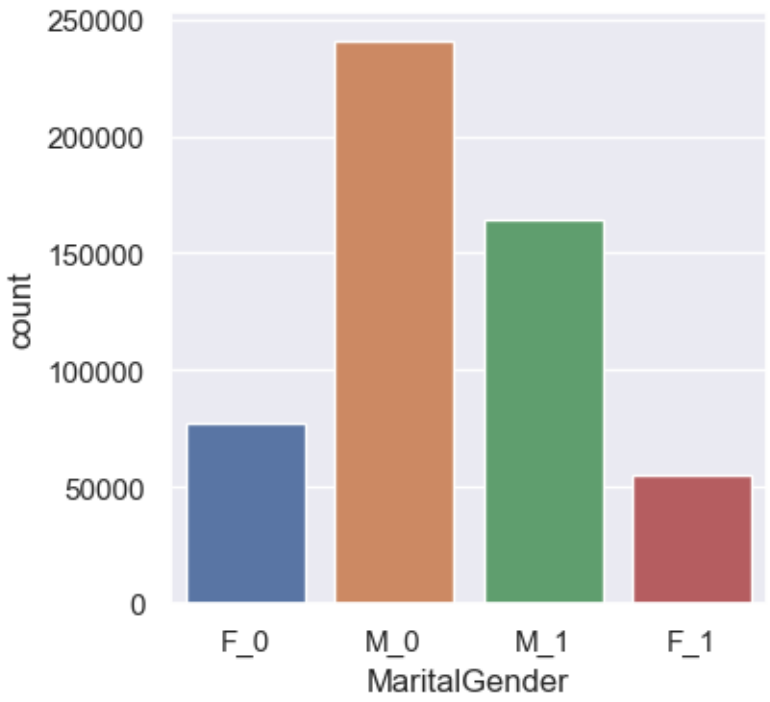
	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A				
4	1000002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

```
3                          2                0                  12       1057
4                         4+                0                   8       7969

  MaritalGender
0           F_0
1           F_0
2           F_0
3           F_0
4           M_0
```

[101]:
```
sb.countplot(x=df['MaritalGender'])
```

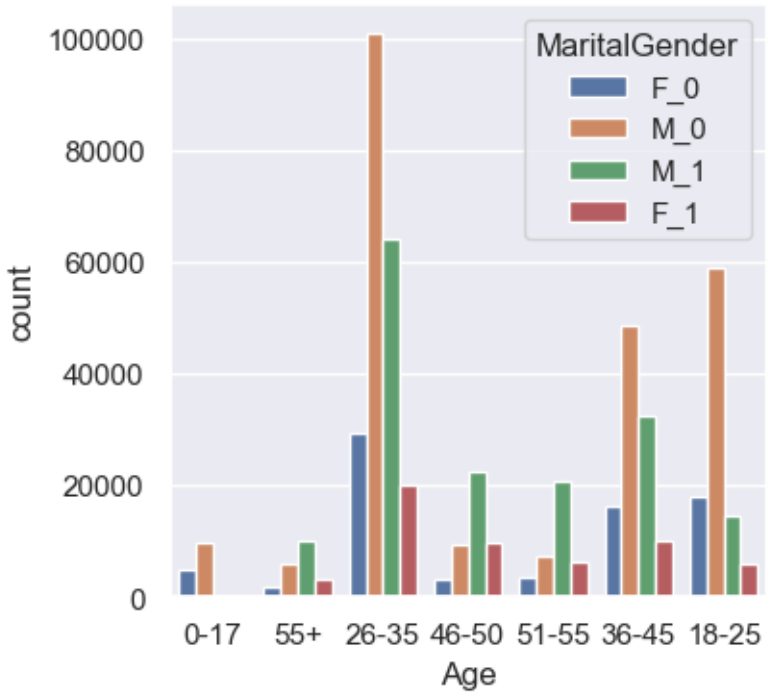
[101]:
```
<Axes: xlabel='MaritalGender', ylabel='count'>
```



10.1 On Age parameter, MaritalGender is

[102]:
```
sb.countplot(x='Age', hue='MaritalGender', data=df)
```

[102]:
```
<Axes: xlabel='Age', ylabel='count'>
```

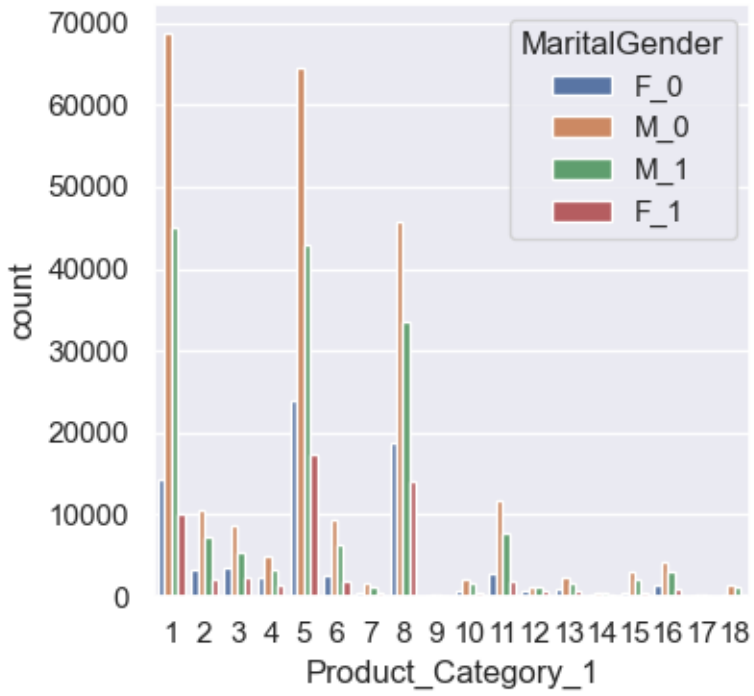


Most of men(unmarried) within the age:26-35 are the buying in sale

10.1.1 On Product_Category_1 , the MaritalGender is

[103]:
```
sb.countplot(x='Product_Category_1', hue='MaritalGender', data=df)
```

[103]:
```
<Axes: xlabel='Product_Category_1', ylabel='count'>
```



[ ]:
```
# Staying in city on MaritalGender
```

Prodict 1,5,8,11 are being most by men(unmarried)

[104]:
```
sb.countplot(x = df['Stay_In_Current_City_Years'], hue = df['MaritalGender'])
```

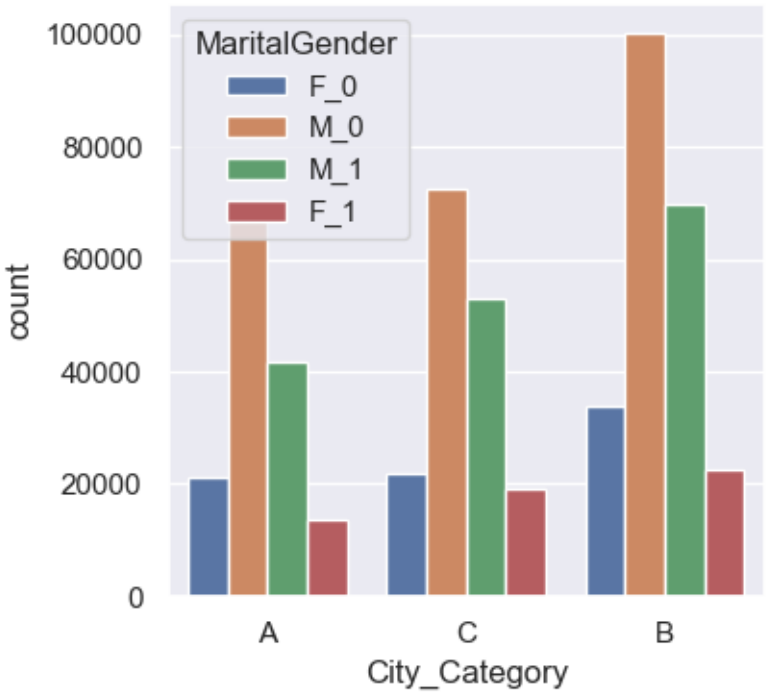
[104]:
```
<Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>
```

Staying in cuurent city years in highest ascending order are : 1>Men(unmarried) 2>Men(married) 3>Female(Unmarried) 4>Femel(married)

10.1.2 City category vs Marital gender

[106]:
```
sb.countplot(x='City_Category', hue='MaritalGender', data=df)
```

[106]:
```
<Axes: xlabel='City_Category', ylabel='count'>
```


MaritalGender
F_0
M_0
M_1
F_1
100000
80000
60000
40000
20000
0
count
A
C
B
City_Category

City B with men(unmarried)