Importing Packages

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

Data Preprocessing

```
df=pd.read_csv(r"C:\Users\Arigala.Adarsh\Downloads\train (23).csv")
```

```
df.head()
```

```
   User_ID Product_ID Gender   Age  Occupation City_Category  \
0  1000001  P00069042      F  0-17        10.0             A   
1  1000001  P00248942      F  0-17        10.0             A   
2  1000001  P00087842      F  0-17        10.0             A   
3  1000001  P00085442      F  0-17        10.0             A   
4  1000002  P00285442      M   55+        16.0             C   

  Stay_In_Current_City_Years  Marital_Status  Product_Category_1  \
0                          2             0.0                 3.0   
1                          2             0.0                 1.0   
2                          2             0.0                12.0   
3                          2             0.0                12.0   
4                         4+             0.0                 8.0   

   Product_Category_2  Product_Category_3  Purchase  
0                 NaN                 NaN    8370.0  
1                 6.0                14.0   15200.0  
2                 NaN                 NaN    1422.0  
3                14.0                 NaN    1057.0  
4                 NaN                 NaN    7969.0  
```

```
df.shape
```

```
(263015, 12)
```

Exploratory Data Analysis(EDA)

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 263015 entries, 0 to 263014
Data columns (total 12 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   User_ID                     263015 non-null  int64  
 1   Product_ID                  263014 non-null  object 
 2   Gender                      263014 non-null  object 
 3   Age                         263014 non-null  object 
 4   Occupation                  263014 non-null  float64
 5   City_Category               263014 non-null  object 
 6   Stay_In_Current_City_Years  263014 non-null  object 
 7   Marital_Status              263014 non-null  float64
 8   Product_Category_1          263014 non-null  float64
 9   Product_Category_2          181501 non-null  float64
 10  Product_Category_3          80582 non-null   float64
 11  Purchase                    263014 non-null  float64
dtypes: float64(6), int64(1), object(5)
memory usage: 24.1+ MB
```

```
df.dtypes
```

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

```
df.describe()
```

```
            User_ID     Occupation  Marital_Status  Product_Category_1  
\
count  2.630150e+05  263014.000000   263014.000000       263014.000000   

mean   1.002941e+06       8.083558        0.408685            5.291099   

std    2.593126e+03       6.524052        0.491592            3.745722   

min    1.000000e+01       0.000000        0.000000            1.000000   

25%    1.001457e+06       2.000000        0.000000            1.000000   

50%    1.002972e+06       7.000000        0.000000            5.000000   

75%    1.004335e+06      14.000000        1.000000            8.000000   
```

```
max    1.006040e+06      20.000000        1.000000           18.000000


       Product_Category_2  Product_Category_3       Purchase
count       181501.000000        80582.000000  263014.000000
mean             9.844756           12.658298    9319.305269
std              5.086696            4.129156    4970.152966
min              2.000000            3.000000     185.000000
25%              5.000000            9.000000    5863.000000
50%              9.000000           14.000000    8060.000000
75%             15.000000           16.000000   12059.000000
max             18.000000           18.000000   23961.000000
```

```
df.isnull().sum()
```

```
User_ID                             0
Product_ID                          1
Gender                              1
Age                                 1
Occupation                          1
City_Category                       1
Stay_In_Current_City_Years          1
Marital_Status                      1
Product_Category_1                  1
Product_Category_2              81514
Product_Category_3             182433
Purchase                            1
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df.shape
```

```
(80582, 12)
```

```
df.columns
```

```
Index(['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'],
      dtype='object')
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['User_ID'] = le.fit_transform(df['User_ID'])
df['Product_ID'] = le.fit_transform(df['Product_ID'])
df['Gender'] = le.fit_transform(df['Gender'])
df['Age'] = le.fit_transform(df['Age'])
```

```
df['City_Category'] = le.fit_transform(df['City_Category'])
```

```
df.dtypes
```

```
User_ID                          int64
Product_ID                       int32
Gender                           int32
Age                              int32
Occupation                     float64
City_Category                    int32
Stay_In_Current_City_Years      object
Marital_Status                 float64
Product_Category_1             float64
Product_Category_2             float64
Product_Category_3             float64
Purchase                       float64
dtype: object
```

```
df
```

```
        User_ID  Product_ID  Gender  Age  Occupation  City_Category  \
1             0         391       0    0        10.0              0
6             3         284       1    4         7.0              1
13            4         211       1    2        20.0              0
14            5         363       0    5         9.0              0
16            5         517       0    5         9.0              0
...         ...         ...     ...  ...         ...            ...
262997     4232         402       0    4        16.0              1
263001     4232         337       0    4        16.0              1
263003     4232         480       0    4        16.0              1
263006     4232          49       0    4        16.0              1
263011     4233         159       1    3         1.0              1

       Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
1                               2             0.0                 1.0

6                               2             1.0                 1.0

13                              1             1.0                 1.0

14                              1             0.0                 5.0

16                              1             0.0                 2.0

...                           ...             ...                 ...

262997                          0             1.0                 1.0

263001                          0             1.0                 1.0
```

```
263003                             0           1.0                1.0

263006                             0           1.0                8.0

263011                             3           0.0                1.0


        Product_Category_2  Product_Category_3  Purchase
1                      6.0                14.0   15200.0
6                      8.0                17.0   19215.0
13                     2.0                 5.0   15665.0
14                     8.0                14.0    5378.0
16                     3.0                 4.0   13055.0
...                    ...                 ...       ...
262997                11.0                16.0   15175.0
263001                11.0                15.0   15430.0
263003                 2.0                15.0   15387.0
263006                13.0                16.0    5861.0
263011                 8.0                17.0   19253.0

[80582 rows x 12 columns]
```

```python
df[df["Purchase"]==19060]
```

```
        User_ID  Product_ID  Gender  Age  Occupation  City_Category  \
301          48           2       1    1         4.0              2
18116      2695          71       1    2         0.0              1
30806      4448         145       1    1        20.0              2
34234      4980         149       1    1         4.0              2
63488      3547         149       0    1         1.0              1
99973      3305         107       1    2         3.0              0
106330     4139         371       0    2         5.0              2
113162     5157         145       1    3         1.0              2
118000      192         138       1    1         4.0              1
122437      874         284       0    6        16.0              2
130088     1871          33       1    2        14.0              0
163877     1237         159       1    4         7.0              1
202902     1226         179       1    2        14.0              2
216176     3214           2       1    2        17.0              2

       Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
301                             0             0.0                 1.0

18116                           2             0.0                 1.0

30806                           1             0.0                 1.0

34234                           3             0.0                 1.0
```

```
63488                                 4+              0.0                    1.0

99973                                  2              0.0                    1.0

106330                                 3              0.0                    1.0

113162                                 0              1.0                    1.0

118000                                 1              0.0                    1.0

122437                                4+              1.0                    1.0

130088                                 3              0.0                    1.0

163877                                 3              1.0                    1.0

202902                                 1              1.0                    1.0

216176                                 1              0.0                    1.0
```

```
        Product_Category_2  Product_Category_3  Purchase
301                    6.0                16.0   19060.0
18116                  2.0                15.0   19060.0
30806                  2.0                 8.0   19060.0
34234                  2.0                14.0   19060.0
63488                  2.0                14.0   19060.0
99973                 15.0                17.0   19060.0
106330                15.0                16.0   19060.0
113162                 2.0                 8.0   19060.0
118000                 2.0                15.0   19060.0
122437                 8.0                17.0   19060.0
130088                 6.0                15.0   19060.0
163877                 8.0                17.0   19060.0
202902                 2.0                15.0   19060.0
216176                 6.0                16.0   19060.0
```

```
df.isnull().sum()
```

```
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            0
Product_Category_3            0
```

```
Purchase                      0
dtype: int64
```

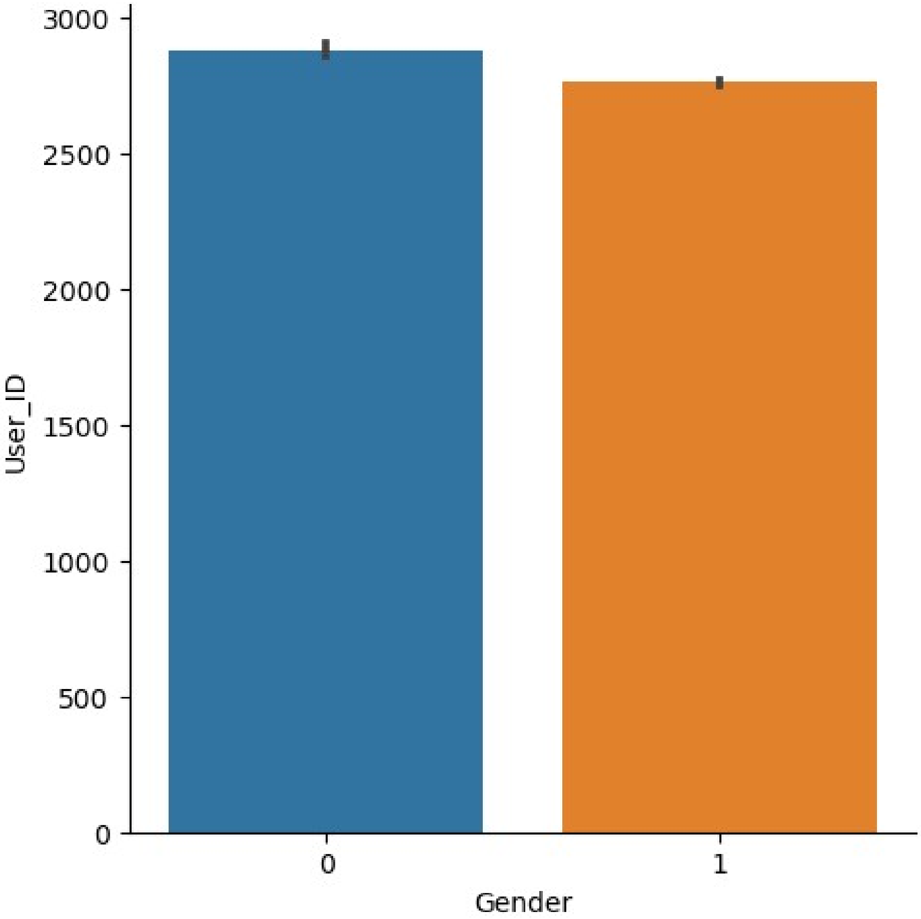
```
df['Stay_In_Current_City_Years'].unique()
```

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

```
df['Stay_In_Current_City_Years']=df['Stay_In_Current_City_Years'].repl
ace('4+','4')
```

```
#changing the datatype from string to integer
df['Stay_In_Current_City_Years'] =
df['Stay_In_Current_City_Years'].astype(int)
```

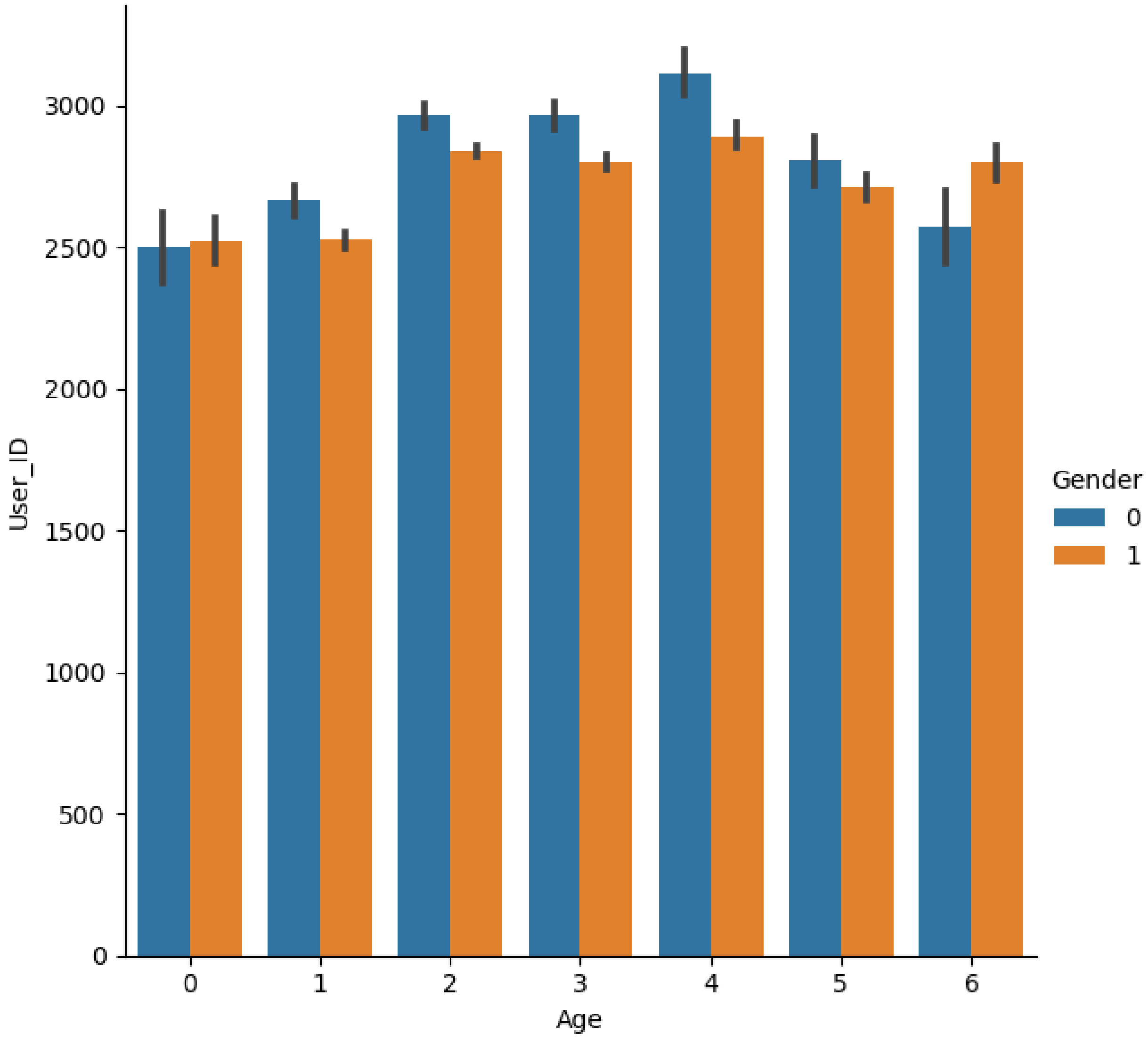
```
sns.catplot(data=df, x='Gender', y='User_ID', kind='bar')
plt.show()
```



```
df.Gender.value_counts()
```

```
1    62549
0    18033
Name: Gender, dtype: int64
```

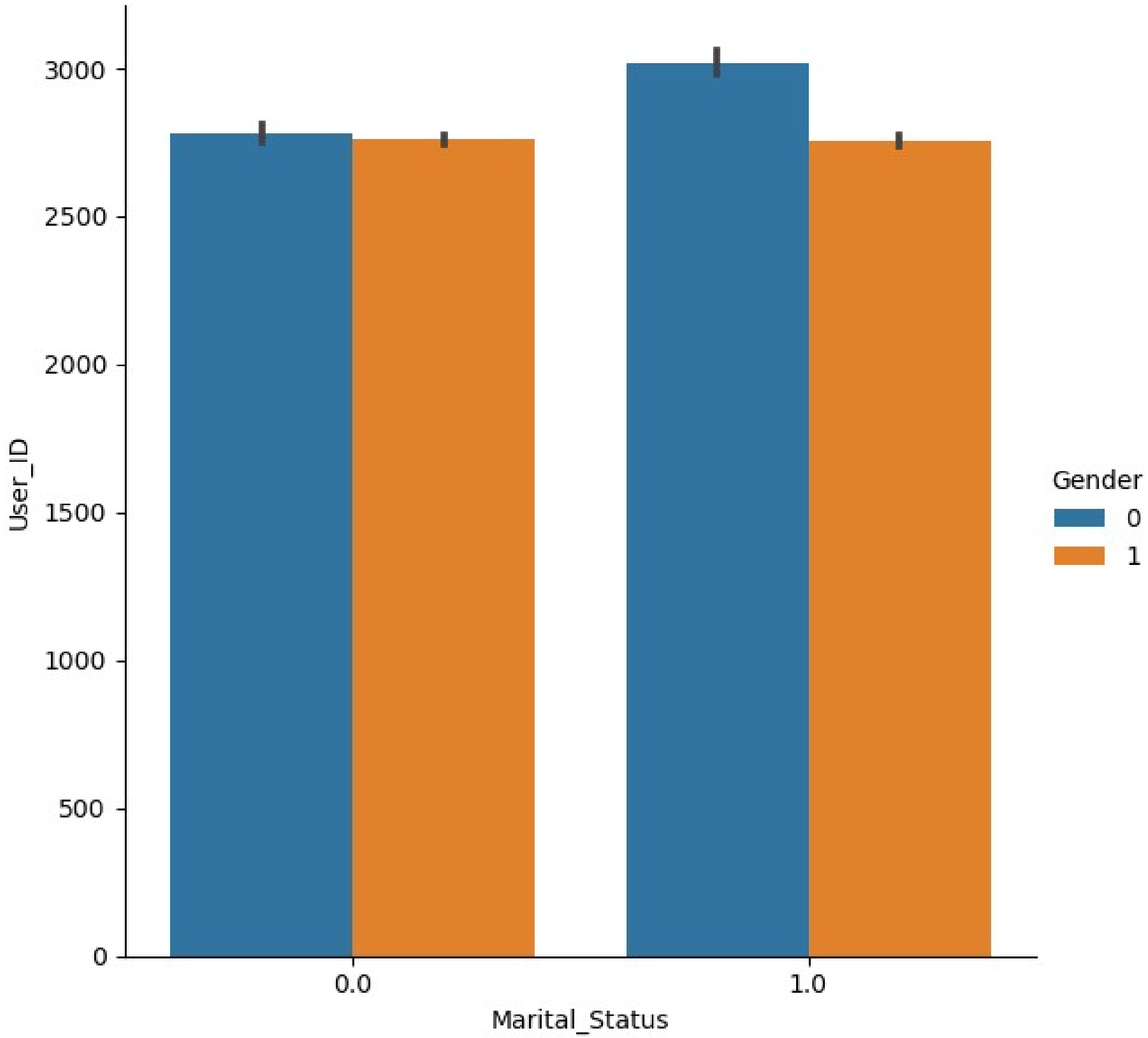
```
sns.catplot(data=df, x='Age', y='User_ID', hue='Gender', kind='bar',
height=6)
plt.show()
```



Inferences from the Plot:

1. The most frequent purchases came from the females having user ids that belong to age category 4(36-45)
2. The least frequent purchases came from the males of the age category 1(18-25), and females of age category 0(0-17).

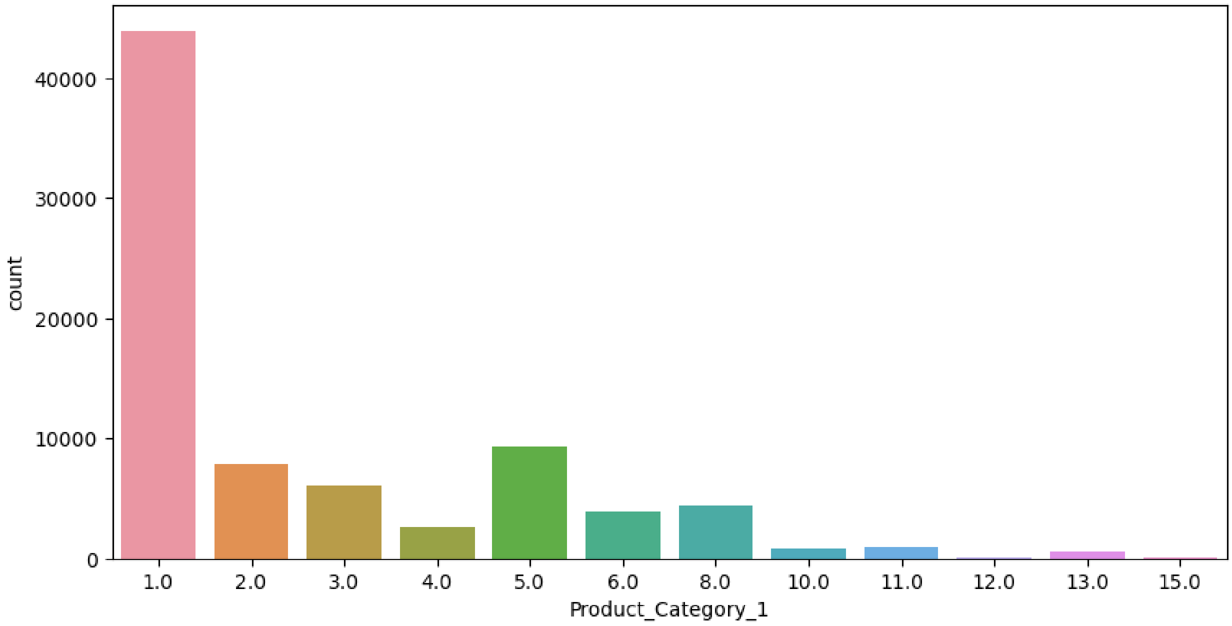
```
sns.catplot(data=df, x='Marital_Status', y='User_ID', hue='Gender',
kind='bar', height=6)
plt.show()
```



the gender ratio in unmarried customers is almost similar, whereas the married customers have a sightly higher number of females in the lot.

```
plt.figure(figsize=(10,5))
sns.countplot(df['Product_Category_1'])
plt.show()
```

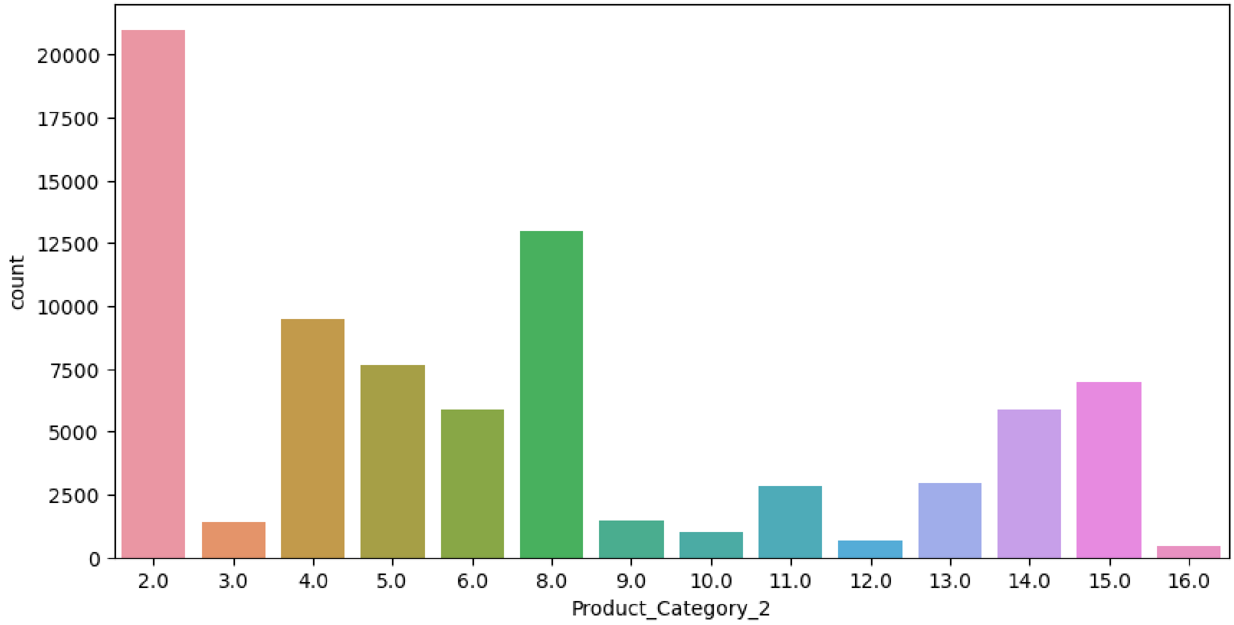
```
C:\Users\Arigala.Adarsh\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



The product_category_1 sees a great rise of product category 1 and diminishes with the other products. The other considerable categories are 5, 2, 3, 6, 8, etc.

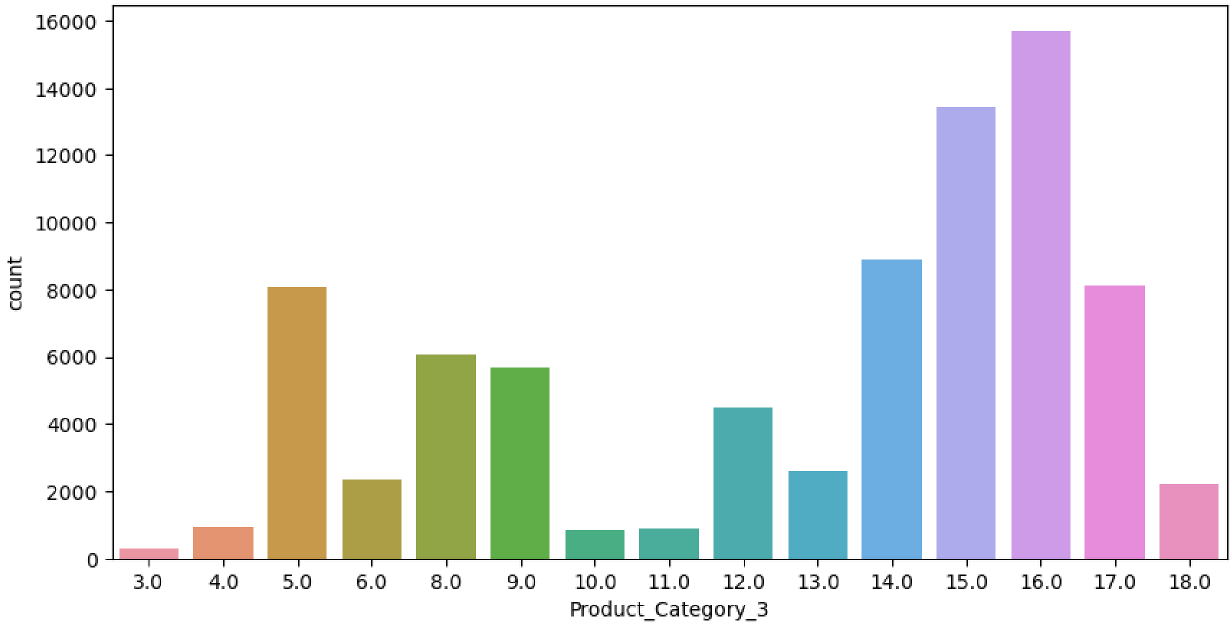
```
plt.figure(figsize=(10,5))
sns.countplot(df['Product_Category_2'])
plt.show()
```

```
C:\Users\Arigala.Adarsh\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



Product_Category_2 sees a considerate balance among categories. With category 2 topping the charts, and other considerable categories are 8, ,4,5,6,14,15,etc.

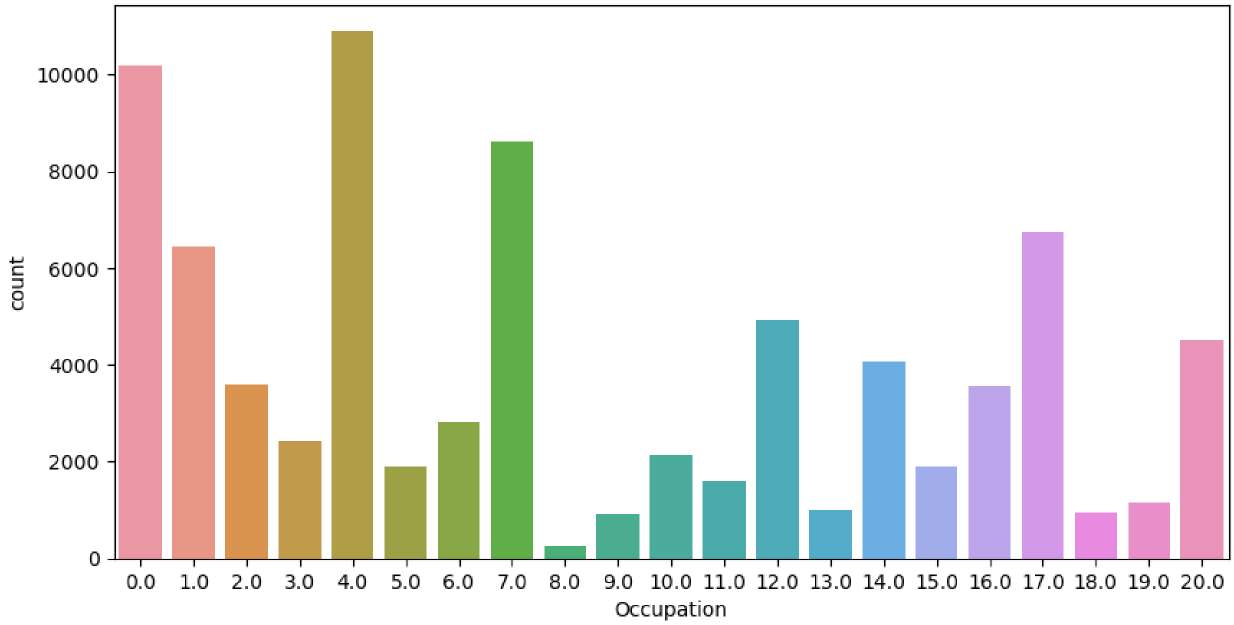
```
plt.figure(figsize=(10,5))
sns.countplot(df['Product_Category_3'])
plt.show()

C:\Users\Arigala.Adarsh\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



For product_category_3, the balance is towards the end with 16 topping the charts, and other considerable categories are 15, 14, 5, 8, 9, 17, etc.

```
plt.figure(figsize=(10,5))
sns.countplot(df['Occupation'])
plt.show()
```

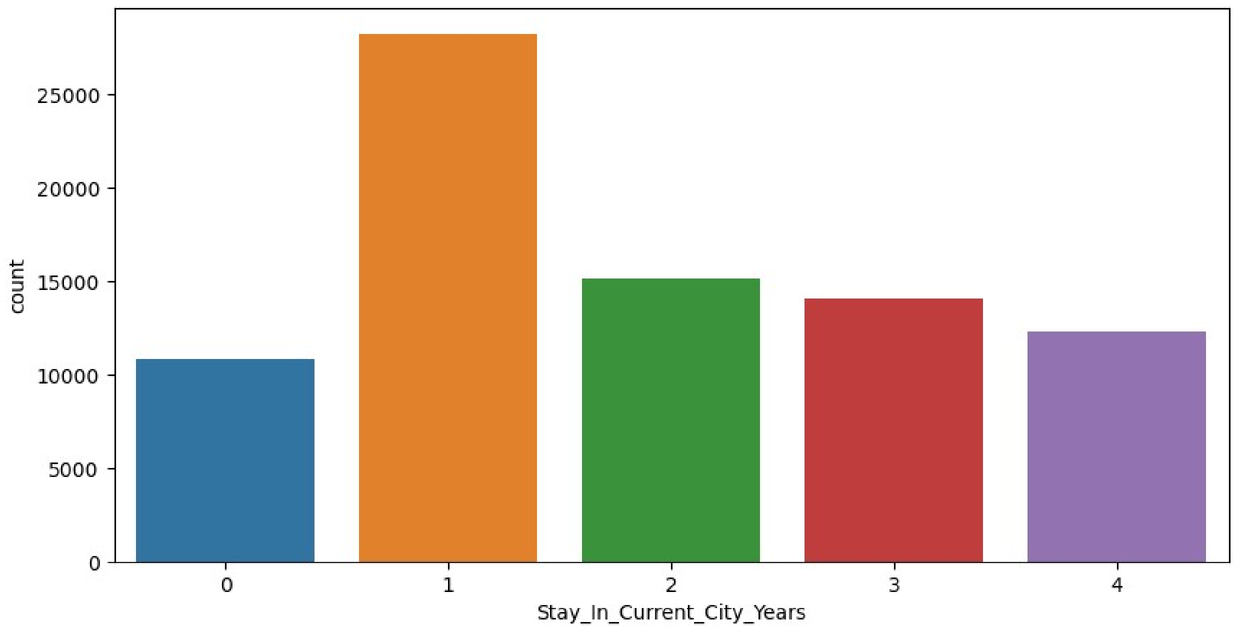
```
C:\Users\Arigala.Adarsh\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



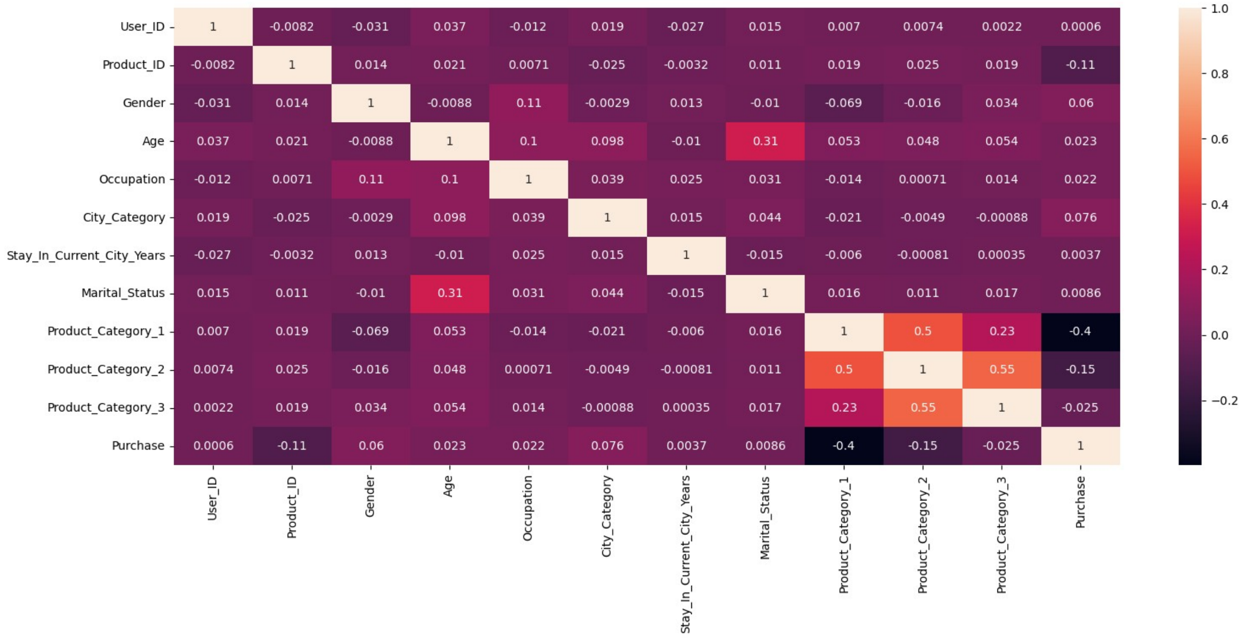
occupation also sees a constant balance with 4 topping the chart, and other categories in the considerable amount with 0,1,2,7,12,17,20.

```
plt.figure(figsize=(10,5))
sns.countplot(df['Stay_In_Current_City_Years'])
plt.show()
```

```
C:\Users\Arigala.Adarsh\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



```
plt.figure(figsize=(18,7))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



Statistical Analysis

A) It was observed that the average purchase made by the Men of the age 18-25 was 10000. Is it still the same?

- One Sample Test For Mean

```
new_data = df.loc[(df['Age'] == 1) & df['Gender'] == 1]
```

```
new_data.shape
```

```
(11904, 12)
```

```
new_data
```

```
        User_ID  Product_ID  Gender  Age  Occupation  City_Category  \
98           20         487       1    1        15.0              0
103          20         402       1    1        15.0              0
111          20         358       1    1        15.0              0
127          23         332       1    1         4.0              2
128          23          71       1    1         4.0              2
...         ...         ...     ...  ...         ...            ...
262758     4206         180       1    1        18.0              1
262759     4206         394       1    1        18.0              1
262760     4206          23       1    1        18.0              1
262764     4206          75       1    1        18.0              1
262952     4225         209       1    1         4.0              0

        Stay_In_Current_City_Years  Marital_Status  Product_Category_1
\
98                               4             0.0                 1.0

103                              4             0.0                 1.0

111                              4             0.0                 2.0

127                              4             0.0                 1.0

128                              4             0.0                 1.0

...                            ...             ...                 ...

262758                           1             1.0                 1.0

262759                           1             1.0                 1.0

262760                           1             1.0                 1.0

262764                           1             1.0                 5.0

262952                           1             1.0                 2.0
```

```
        Product_Category_2  Product_Category_3  Purchase
98                     8.0                17.0   12099.0
103                   11.0                16.0   12098.0
111                    4.0                15.0    9564.0
127                    5.0                 9.0   15361.0
128                    2.0                15.0   15770.0
...                    ...                 ...       ...
262758                 2.0                15.0   11512.0
262759                15.0                18.0   11521.0
262760                 2.0                13.0    3988.0
262764                 8.0                17.0    5444.0
262952                 3.0                 4.0    3240.0

[11904 rows x 12 columns]
```

```
sample_size = 1000
sample = new_data.sample(sample_size, random_state=0)
```

```
new_data.mean()
```

```
User_ID                           2525.070733
Product_ID                         222.067624
Gender                               1.000000
Age                                  1.000000
Occupation                           7.042255
City_Category                        1.063172
Stay_In_Current_City_Years           1.866683
Marital_Status                       0.194892
Product_Category_1                   2.484207
Product_Category_2                   6.584677
Product_Category_3                  12.471522
Purchase                         11785.188172
dtype: float64
```

```
new_data.Purchase.std()
```

```
5080.322126868783
```

```
pos_mean=11785
```

```
#one sample t-test
from scipy.stats import ttest_1samp
t_stat, p_value = ttest_1samp(sample['Purchase'], pos_mean)
print(t_stat, p_value)
```

```
-0.21540497598886765 0.8294955587149927
```

```
 # P-value is less than 0.05, reject the null hypothesis.
# Null Hypothesis will be accepted
# therefore, the mean purchase for men aged 18-25 is 10000.
```

B) It was observed that the percentage of women of the age that spend more than 10000 was 35%. Is it still the same?

- One Sample Test for Proportion

```
#null hypothesis - proportion is 35%.
#alternate hypothesis - proportion is not 35%.

new_data = df.loc[(df['Purchase'] > 10000)]

new_data.mean()

User_ID                           2777.960883
Product_ID                         221.135756
Gender                               0.793987
Age                                  2.459020
Occupation                           8.283276
City_Category                        1.125491
Stay_In_Current_City_Years           1.861065
Marital_Status                       0.403786
Product_Category_1                   1.808018
Product_Category_2                   6.055254
Product_Category_3                  12.360397
Purchase                         14982.258210
dtype: float64

new_data.Purchase.std()

3040.3040968756654

#hypothesised value
po=0.35

#number of observations

nobs = len(new_data['Gender'])

#number of women observations
count =new_data['Gender'].value_counts()[0]

from statsmodels.stats.proportion import proportions_ztest
z_stat, p_val = proportions_ztest(count=count,
                                  nobs=nobs,
                                  value=po,
                                  alternative="two-sided",
                                  prop_var=False)
print(z_stat, p_val)
```

```
-79.12020590883206 0.0

#p-value is less than 0.05, reject the null hypothesis.
#the proportion of women spending more than 10000, is not 35%.
```

c. Is the average purchase made by men and women of the age 18-25 same?

- Two Sample test for Means

```
#null hypothesis - average spends are equal
#alternate hypothesis - average spends are not equal
men = df.loc[(df['Gender'] == 1)& (df['Age'] == 1)]
women = df.loc[(df['Gender'] == 0) & (df['Age'] == 1)]

#creating samples
data_men_sample = men.sample(500, random_state=0)
data_women_sample = women.sample(500, random_state=0)

#checking variances of the two samples
print(data_men_sample.Purchase.var())
print(data_women_sample.Purchase.var())

25403579.49849695
26680870.93292181

#two sample t-test for unequal variances
from scipy.stats import ttest_ind

t_stat_2, p_val_2 = ttest_ind(data_men_sample.Purchase,
data_women_sample.Purchase, equal_var=False)
print(t_stat_2, p_val_2)

3.4922719108842966 0.0004999285373589167
```

#we can reject the null hypothesis using the test statistic and since p-value is less than 0.05.

#the average purchases are not the same.

D.Is the percentage of men who have spend more than 10000 same for the ages 18-25 and 26-35

- Two Sample test for Proportion

```
data_age1 = df.loc[(df['Age'] == 1) & (df['Purchase'] > 10000)]
data_age2 = df.loc[(df['Age'] == 2) & (df['Purchase'] > 10000)]

data_age1_sample = data_age1.sample(1000, random_state=0)
data_age2_sample = data_age2.sample(1000, random_state=0)

count = [(data_age1_sample['Gender'] == 1).sum(),
(data_age2_sample['Gender'] == 1).sum()]

nobs = [(len(data_age1_sample)), len(data_age2_sample)]

from statsmodels.stats.proportion import proportions_ztest
stat_2sample, p_value_2sample = proportions_ztest(count=count,
                                                  nobs=nobs,
                                                  value=0,
                                                  alternative='two-
sided',
                                                  prop_var=False)

print(stat_2sample, p_value_2sample)

-0.759111307093946 0.44778597581119517
```

#p value is more than 0.05, cannot reject the null hypthesis. Null hypothesis is accepted

#therefore, Percentage of the men in both the age groups who have spent more than 10000 is same