Advanced Data Analytics
(CSE4029)

Black Friday Sales Analysis
Project Report
(Phase - II)

Varad Vijay Temghare
19BCI7045

Submitted to :
Dr. Nitesh Funde

Introduction:-

For this assignment, we have chosen the Black Friday sales dataset of a certain company. The dataset has shared a purchase summary of various customers for selected high-volume products from last month. The data set also contains customer demographics (age, gender, marital status, city_type, stay_in_current_city), product details (product_id and product category), and Total purchase_amount from last month.

Dataset Link :- [Here]

Dimensions of the above dataset:- We have 5,50,069 rows and 12 columns .

Code Link:- [Here]

Data Preprocessing :

```
[1] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive
```

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

```
[100] data = pd.read_csv("/content/drive/MyDrive/ADA LAB/mgt.csv")
```

```
[101] data.head()
```

	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

```
[102] data.shape

(550068, 12)
```

[103] data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 12 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   User_ID                     550068 non-null  int64  
 1   Product_ID                  550068 non-null  object 
 2   Gender                      550068 non-null  object 
 3   Age                         550068 non-null  object 
 4   Occupation                  550068 non-null  int64  
 5   City_Category               550068 non-null  object 
 6   Stay_In_Current_City_Years  550068 non-null  object 
 7   Marital_Status              550068 non-null  int64  
 8   Product_Category_1          550068 non-null  int64  
 9   Product_Category_2          376430 non-null  float64
 10  Product_Category_3          166821 non-null  float64
 11  Purchase                    550068 non-null  int64  
dtypes: float64(2), int64(5), object(5)
memory usage: 50.4+ MB
```

Checking Null values

[104] data.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            173638
Product_Category_3            383247
Purchase                           0
dtype: int64
```

Null Value in percentage

[105] data.isnull().sum()/data.shape[0]*100

```
User_ID                        0.000000
Product_ID                     0.000000
Gender                         0.000000
Age                            0.000000
Occupation                     0.000000
City_Category                  0.000000
Stay_In_Current_City_Years     0.000000
Marital_Status                 0.000000
Product_Category_1             0.000000
Product_Category_2            31.566643
Product_Category_3            69.672659
Purchase                       0.000000
dtype: float64
```

There are 31% null values in the `Product_Category_2` and 69% null values in the `Product_Category_3`

Unique elements in each attributes

```
[106] data.nunique()

User_ID                       5891
Product_ID                    3631
Gender                           2
Age                              7
Occupation                      21
City_Category                    3
Stay_In_Current_City_Years       5
Marital_Status                   2
Product_Category_1              20
Product_Category_2              17
Product_Category_3              15
Purchase                     18105
dtype: int64
```

```
[107] data['Product_Category_2'] =data['Product_Category_2'].fillna(int(data["Product_Category_2"].mean())).astype('int64')
      data['Product_Category_3'] =data['Product_Category_3'].fillna(int(data["Product_Category_3"].mean())).astype('int64')
```

```
data.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
```

We did a Descriptive Analysis for the above dataset and Below are the observations which we have made from the data visualization done as part of the Data Understanding process.

- Approximately, 75% of the number of purchases are made by Male users and rest of the 25% is done by female users. This tells us the Male consumers are the major contributors to the number of sales for the retail store.On average the male gender spends more money on purchase contrary to female, and it is possible to also observe this trend by adding the total value of purchase.
- When we combined Purchase and Marital_Status for analysis, we came to know that Single Men spend the most during the Black Friday. It also tells that Men tend to spend less once they are married. It maybe because of the added responsibilities.
- For Age feature, we observed the consumers who belong to the age group 25-40 tend to spend the most.
- There is an interesting column Stay_In_Current_City_Years, after analyzing this column

we came to know the people who have spent 1 year in the city tend to spend the most. This is understandable as, people who have spent more than 4 years in the city are generally well settled and are less interested in buying new things as compared to the people new to the city, who tend to buy more.

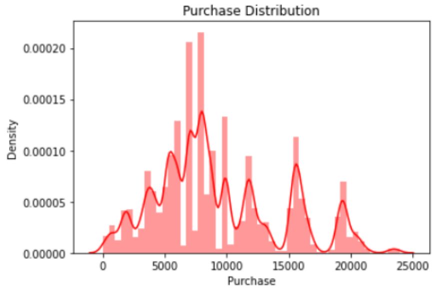
- When examining which city the product was purchased to our surprise, even though the city B is majorly responsible for the overall sales income, but when it comes to the above product, it majorly purchased in the city C.

EDA

Target Variable Purchase

```
[109] sns.distplot(data["Purchase"],color='r')
      plt.title("Purchase Distribution")
      plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version.
  warnings.warn(msg, FutureWarning)
```

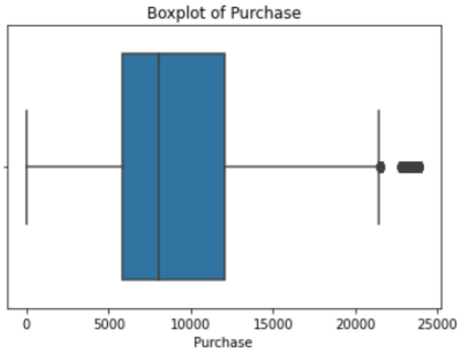


We can observe that purchase amount is repeating for many customers.This may be because on Black Friday many are buying discounted products in large numbers and kind of follows a Gaussian Distribution.

We can observe that purchase amount is repeating for many customers.This may be because on Black Friday many are buying discounted products in large numbers and kind of follows a Gaussian Distribution.

[110]
```
sns.boxplot(data["Purchase"])
plt.title("Boxplot of Purchase")
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x.
  FutureWarning
```



[111]
```
data["Purchase"].skew()
```

```
0.6001400037087128
```

[112]
```
data["Purchase"].kurtosis()
```

```
-0.3383775655851702
```

```
[113] data["Purchase"].describe()
```

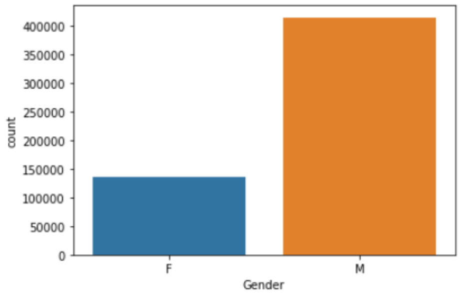
```
count    550068.000000
mean       9263.968713
std        5023.065394
min          12.000000
25%        5823.000000
50%        8047.000000
75%       12054.000000
max       23961.000000
Name: Purchase, dtype: float64
```

The purchase is right skewed and we can observe multiple peaks in the distribution we can do a log transformation for the purchase.

Gender

```
sns.countplot(data['Gender'])
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12,
  FutureWarning
```



```
F     24.689493
Name: Gender, dtype: float64
```

There are more males than females

```
[116] data.groupby("Gender").mean()["Purchase"]
```

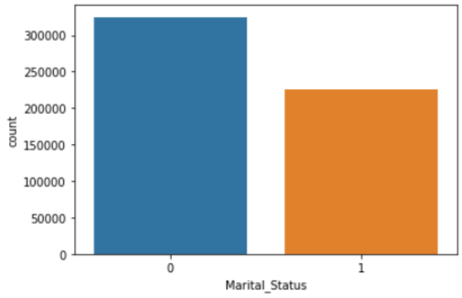
```
Gender
F     8734.565765
M     9437.526040
Name: Purchase, dtype: float64
```

On average the male gender spends more money on purchase contrary to female, and it is possible to also observe this trend by adding the total value of purchase.

Marital Status

```
[117] sns.countplot(data['Marital_Status'])
      plt.show()
```

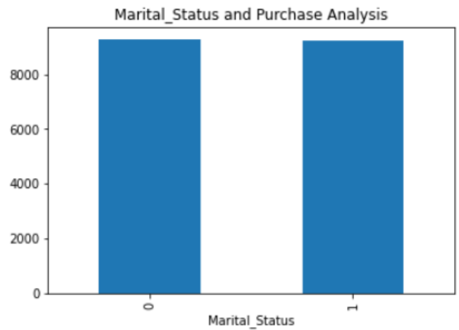
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, t
  FutureWarning
```




```
data.groupby("Marital_Status").mean()["Purchase"]
```

```
Marital_Status
0     9265.907619
1     9261.174574
Name: Purchase, dtype: float64
```

```
data.groupby("Marital_Status").mean()["Purchase"].plot(kind='bar')
plt.title("Marital_Status and Purchase Analysis")
plt.show()
```

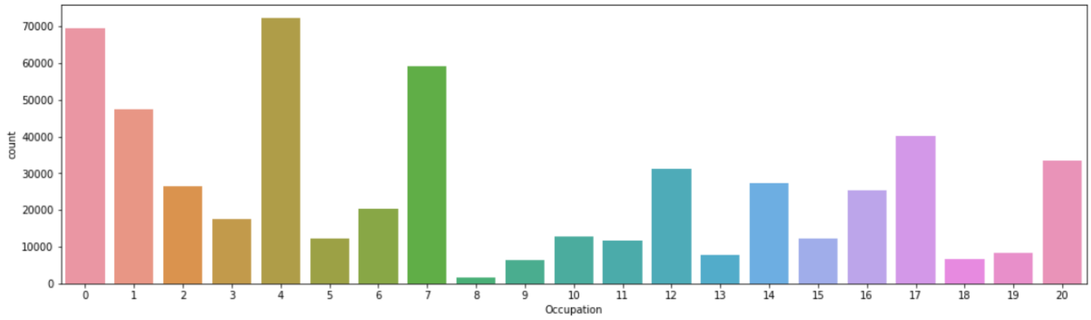


This is interesting though unmarried people spend more on purchasing, the average purchase amount of married and unmarried people are the same.

Occupation

```
[120] plt.figure(figsize=(18,5))
      sns.countplot(data['Occupation'])
      plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid posit
  FutureWarning
```

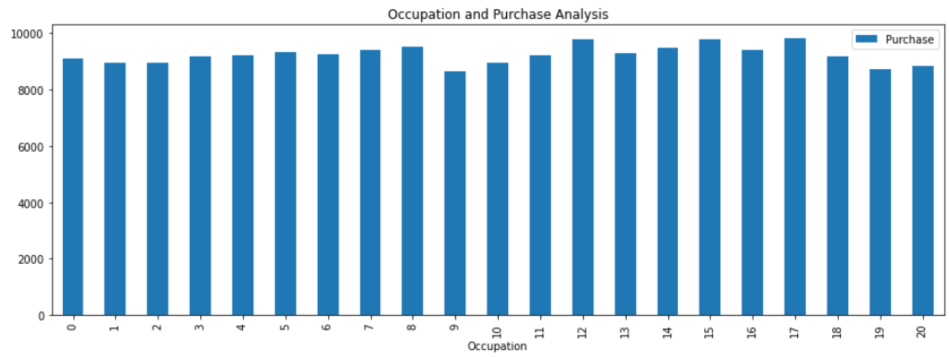


Occupation has at least 20 different values. Since we do not known to each occupation each number corresponds, is difficult to make any analysis. Furthermore, it seems we have no alternative but to use since there is no way to reduce this number

```
[121] occup = pd.DataFrame(data.groupby("Occupation").mean()["Purchase"])
      occup
```

Occupation	Purchase
0	9124.428588
1	8953.193270
2	8952.481683
3	9178.593088
4	9213.980251
5	9333.149298
6	9256.535691
7	9425.728223
8	9532.592497
9	8637.743761
10	8959.355375
11	9213.845848
12	9796.640239
13	9306.351061
14	9500.702772
15	9778.891163
16	9394.464349
17	9821.478236
18	9169.655844
19	8710.627231

```
[122] occup.plot(kind='bar',figsize=(15,5))
      plt.title("Occupation and Purchase Analysis")
      plt.show()
```

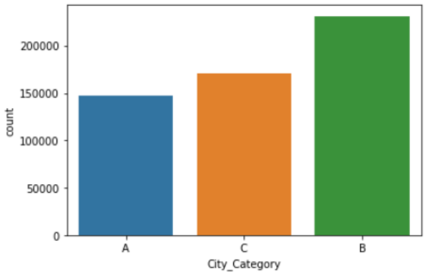


Although there are some occupations which have higher representations, it seems that the amount each user spends on average is more or less the same for all occupations. Of course, in the end, occupations with the highest representations will have the highest amounts of purchases.

City_Category

```
[123] sns.countplot(data['City_Category'])
      plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valic
  FutureWarning
```



It is observed that city category B has made the most number of puchases.

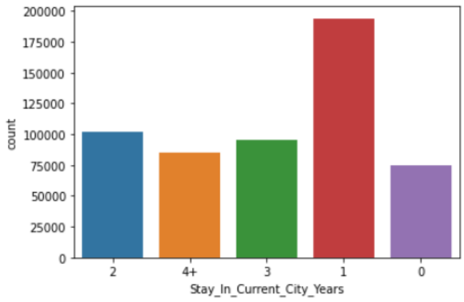
```
[124] data.groupby("City_Category").mean()["Purchase"].plot(kind='bar')
      plt.title("City Category and Purchase Analysis")
      plt.show()
```

However, the city whose buyers spend the most is city type ‘C’.

Stay_In_Current_City_Years

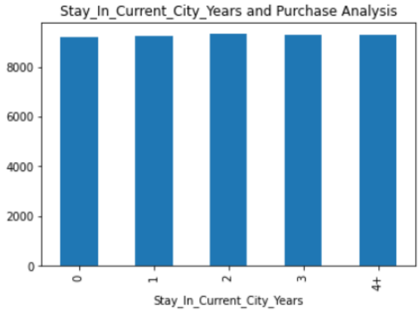
```
[125] sns.countplot(data['Stay_In_Current_City_Years'])
      plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12,
  FutureWarning
```



It looks like the longest someone is living in that city the less prone they are to buy new things. Hence, if someone is new in town and needs a great number of new things for their house that they'll take advantage of the low prices in Black Friday to purchase all the things needed.

```
[126] data.groupby("Stay_In_Current_City_Years").mean()["Purchase"].plot(kind='bar')
      plt.title("Stay_In_Current_City_Years and Purchase Analysis")
      plt.show()
```

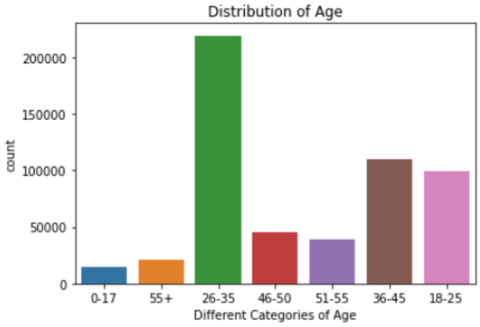


We see the same pattern seen before which show that on average people tend to spend the same amount on purchases regardeless of their group. People who are new in city are responsible for the higher number of purchase, however looking at it individually they tend to spend the same amount independently of how many years the have lived in their current city.

Age

```
[127] sns.countplot(data['Age'])
      plt.title('Distribution of Age')
      plt.xlabel('Different Categories of Age')
      plt.show()
```

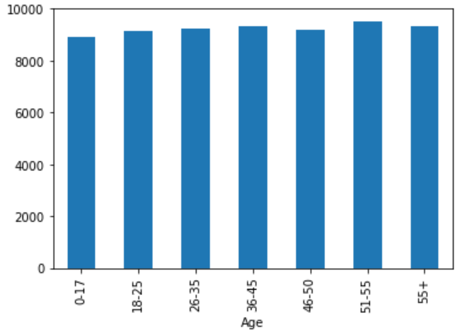
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version
  FutureWarning
```



Age 26-35 Age group makes the most no of purchases in the age group.

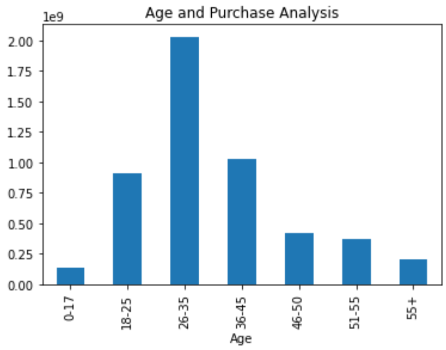
```
[128] data.groupby("Age").mean()["Purchase"].plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95b7472d50>
```



Mean puchase rate between the age groups tends to be the same except that the 51-55 age group has a little higher average purchase amount

```
data.groupby("Age").sum()['Purchase'].plot(kind="bar")
plt.title("Age and Purchase Analysis")
plt.show()
```

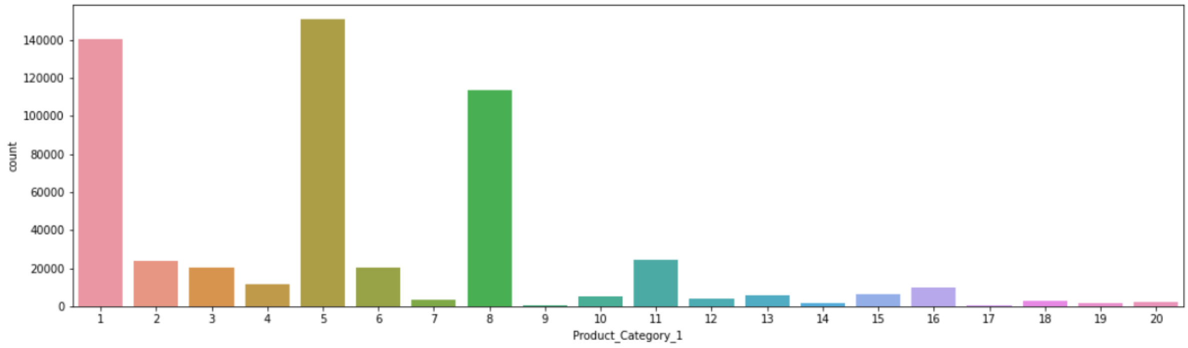


Total amount spent in purchase is in accordance with the number of purchases made, distributed by age.

Product_Category_1

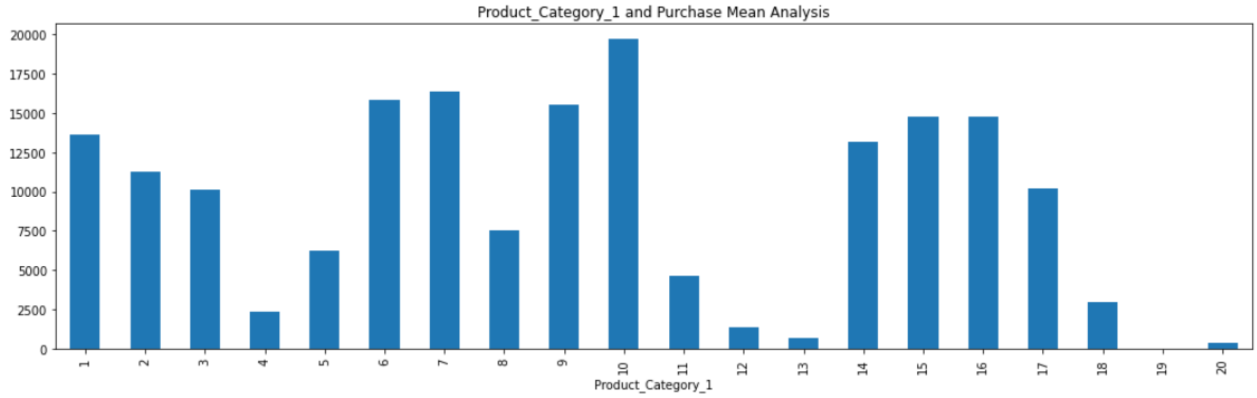
```
[130] plt.figure(figsize=(18,5))
      sns.countplot(data['Product_Category_1'])
      plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the or
  FutureWarning
```



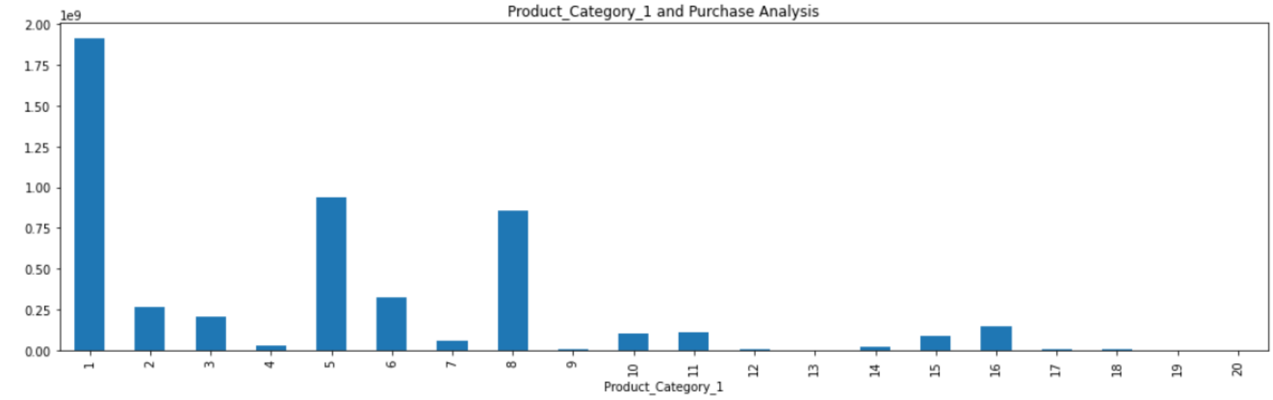
It is clear that `Product_Category_1` numbers 1,5 and 8 stand out. Unfortunately we don't know which product each number represents as it is masked.

```
[131] data.groupby('Product_Category_1').mean()['Purchase'].plot(kind='bar',figsize=(18,5))
      plt.title("Product_Category_1 and Purchase Mean Analysis")
      plt.show()
```



If you see the value spent on average for Product_Category_1 you see that although there were more products bought for categories 1,5,8 the average amount spent for those three is not the highest. It is interesting to see other categories appearing with high purchase values despite having low impact on sales number.

```
[132] data.groupby('Product_Category_1').sum()['Purchase'].plot(kind='bar',figsize=(18,5))
      plt.title("Product_Category_1 and Purchase Analysis")
      plt.show()
```

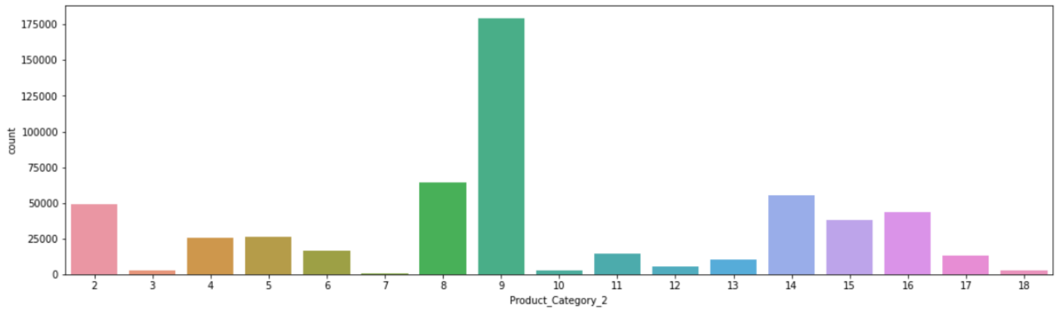


The distribution that we saw for this predictor previously appears here. For example, those three products have the highest sum of sales since their were three most sold products.

Product_Category_2

```
[133] plt.figure(figsize=(18,5))
      sns.countplot(data['Product_Category_2'])
      plt.show()
```

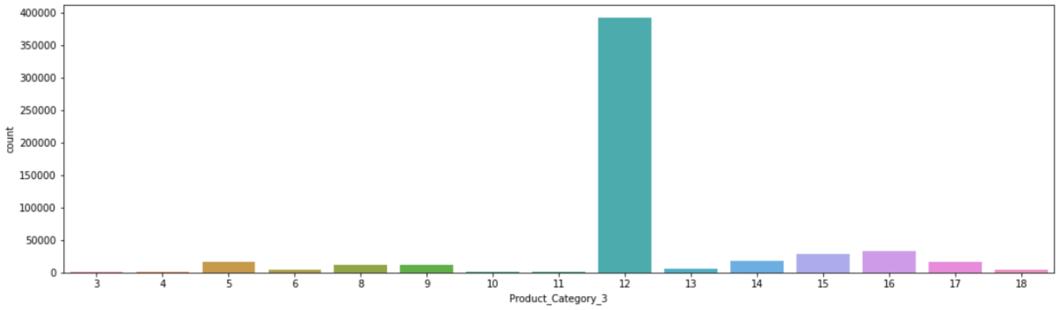
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional
  FutureWarning
```



Product_Category_3

```
[134] plt.figure(figsize=(18,5))
      sns.countplot(data['Product_Category_3'])
      plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional
  FutureWarning
```



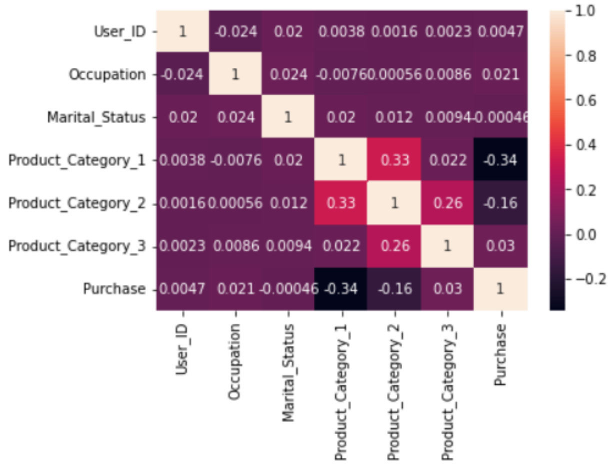

```
[135] data.corr()
```

	User_ID	Occupation	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
User_ID	1.000000	-0.023971	0.020443	0.003825	0.001644	0.002291	0.004716
Occupation	-0.023971	1.000000	0.024280	-0.007618	0.000557	0.008584	0.020833
Marital_Status	0.020443	0.024280	1.000000	0.019888	0.011526	0.009374	-0.000463
Product_Category_1	0.003825	-0.007618	0.019888	1.000000	0.331691	0.022191	-0.343703
Product_Category_2	0.001644	0.000557	0.011526	0.331691	1.000000	0.259891	-0.156676
Product_Category_3	0.002291	0.008584	0.009374	0.022191	0.259891	1.000000	0.029984
Purchase	0.004716	0.020833	-0.000463	-0.343703	-0.156676	0.029984	1.000000

HeatMap

[136]

```
sns.heatmap(data.corr(),annot=True)
plt.show()
```



There is a some corellation between the product category groups.