Importing Lib.

```
In [1]: import numpy as np
   import pandas as pd
   import matplotlib.pyplot as plt # visualizing data
   %matplotlib inline
   import seaborn as sns
```

Import CSV file

```
In [2]: df = pd.read_csv('Sales Data.csv' , encoding= 'unicode_escape')
In [4]:
         df.shape
Out[4]: (11251, 15)
In [5]:
        df.head(10)
Out[5]:
                                                        Age
            User ID Cust name Product ID Gender
                                                                   Marital Status
                                                                                            State
                                                      Group
         0 1002903
                                                       26-35
                                                                               0
                         Sanskriti
                                  P00125942
                                                   F
                                                               28
                                                                                     Maharashtra
                                                       26-35
                                                                                  Andhra Pradesh
         1 1000732
                           Kartik
                                  P00110942
                                                   F
                                                               35
         2 1001990
                           Bindu
                                  P00118542
                                                   F
                                                       26-35
                                                               35
                                                                               1
                                                                                     Uttar Pradesh
            1001425
                          Sudevi
                                  P00237842
                                                        0-17
                                                                               0
                                                                                        Karnataka
                                                   M
                                                               16
           1000588
                            Joni
                                  P00057942
                                                       26-35
                                                               28
                                                                               1
                                                                                          Gujarat
                                                   M
                                                                                        Himachal
         5 1000588
                                  P00057942
                                                       26-35
                                                               28
                                                                               1
                            Joni
                                                   M
                                                                                         Pradesh
         6 1001132
                                  P00018042
                                                       18-25
                                                               25
                                                                                     Uttar Pradesh
                            Balk
                                                   F
                                                                               1
         7 1002092
                         Shivangi
                                  P00273442
                                                   F
                                                         55+
                                                               61
                                                                               0
                                                                                     Maharashtra
            1003224
                          Kushal
                                  P00205642
                                                       26-35
                                                               35
                                                                               0
                                                                                     Uttar Pradesh
                                                   M
         9 1003650
                                  P00031142
                                                   F
                                                       26-35
                                                               26
                                                                               1 Andhra Pradesh
                           Ginny
In [6]:
        df.tail(10)
```

Marital_Status	Age	Age Group	Gender	Product_ID	Cust_name	User_ID	
0	33	26-35	F	P00058042	Matthias	1003032	241
1	27	26-35	F	P00185442	Hildebrand	1004344	1242
0	53	51-55	М	P00297742	Sheetal	1005446	11243
0	53	51-55	М	P00297742	Sheetal	1005446	11244
1	31	26-35	F	P00057442	Bertelson	1004140	11245
1	19	18-25	М	P00296942	Manning	1000695	11246
0	33	26-35	М	P00171342	Reichenbach	1004089	11247
0	40	36-45	F	P00201342	Oshin	1001209	11248
0	37	36-45	М	P00059442	Noonan	1004023	11249
0	19	18-25	F	P00281742	Brumley	1002744	11250
	0 1 0 0 1 1 0 0	33 0 27 1 53 0 53 0 31 1 19 1 33 0 40 0	Group Age Marital_Status 26-35 33 0 26-35 27 1 51-55 53 0 26-35 31 1 18-25 19 1 26-35 33 0 36-45 40 0 36-45 37 0	Group Age Marital_Status F 26-35 33 0 F 26-35 27 1 M 51-55 53 0 M 51-55 53 0 F 26-35 31 1 M 18-25 19 1 M 26-35 33 0 F 36-45 40 0 M 36-45 37 0	Product_ID Genue Group Age Marital_Status P00058042 F 26-35 33 0 P00185442 F 26-35 27 1 P00297742 M 51-55 53 0 P00297742 M 51-55 53 0 P00057442 F 26-35 31 1 P00296942 M 18-25 19 1 P00171342 M 26-35 33 0 P00201342 F 36-45 40 0 P00059442 M 36-45 37 0	Matthias Product_ID Group Age Maintal_Status Matthias P00058042 F 26-35 33 0 Hildebrand P00185442 F 26-35 27 1 Sheetal P00297742 M 51-55 53 0 Sheetal P00297742 M 51-55 53 0 Bertelson P00057442 F 26-35 31 1 Manning P00296942 M 18-25 19 1 Reichenbach P00171342 M 26-35 33 0 Oshin P00201342 F 36-45 40 0 Noonan P00059442 M 36-45 37 0	OSEI_ID Cust_Iname Product_ID Genup Age Mantal_status 1003032 Matthias P00058042 F 26-35 33 0 1004344 Hildebrand P00185442 F 26-35 27 1 1005446 Sheetal P00297742 M 51-55 53 0 1004140 Bertelson P00297742 M 51-55 53 0 1004140 Bertelson P00057442 F 26-35 31 1 1004095 Manning P00296942 M 18-25 19 1 1004089 Reichenbach P00171342 M 26-35 33 0 1001209 Oshin P00201342 F 36-45 40 0 1004023 Noonan P00059442 M 36-45 37 0

```
In [7]: df.info()
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 15 columns):

#	Column	Non-Null Count	Dtype
0	User_ID	11251 non-null	int64
1	Cust_name	11251 non-null	object
2	Product_ID	11251 non-null	object
3	Gender	11251 non-null	object
4	Age Group	11251 non-null	object
5	Age	11251 non-null	int64
6	Marital_Status	11251 non-null	int64
7	State	11251 non-null	object
8	Zone	11251 non-null	object
9	Occupation	11251 non-null	object
10	Product_Category	11251 non-null	object
11	Orders	11251 non-null	int64
12	Amount	11239 non-null	float64
13	Status	0 non-null	float64
14	unnamed1	0 non-null	float64
dtyp	es: float64(3), in	t64(4), object(8)

Drop blank columns

memory usage: 1.3+ MB

```
In [8]: df.drop(['Status' , 'unnamed1'] , axis = 1 , inplace = True)
In [10]: pd.isnull(df).sum()
```

```
Out[10]: User_ID
                               0
                               0
         Cust name
         Product ID
                               0
         Gender
         Age Group
                               0
                               0
         Age
                               0
         Marital Status
         State
                               0
                               0
         Zone
         Occupation
         Product_Category
         0rders
                               0
         Amount
                              12
         dtype: int64
```

Drop null value

```
In [11]: df.dropna(inplace=True)
In [12]: pd.isnull(df).sum()
                              0
Out[12]: User ID
                              0
         Cust name
         Product_ID
                              0
         Gender
                              0
         Age Group
                              0
                              0
         Age
                              0
         Marital Status
         State
                              0
         Zone
                              0
                              0
         Occupation
         Product_Category
                              0
         0rders
                              0
         Amount
                              0
         dtype: int64
```

Change Data Type

```
In [14]: df['Amount']=df['Amount'].astype(int)
In [17]: df.dtypes
```

```
Out[17]: User_ID
                       int64
                       object
       Cust_name
       Product ID
                       object
       Gender
                       object
       Age Group
                       object
                       int64
       Age
       Marital Status
                       int64
       State
                       object
       Zone
                       object
       Occupation
                       object
       Product_Category
                       object
       0rders
                       int64
       Amount
                       int32
       dtype: object
In [19]: df.columns
'Orders', 'Amount'],
            dtype='object')
```

Rename Column

In [22]:	df.ren	name(colu	ımns = {'Mar	ital_Status	s' : 'Vi	vah'})				
Out[22]:		User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Vivah	State	
	0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	
	1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	S
	2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	
	3	1001425	Sudevi	P00237842	М	0-17	16	0	Karnataka	S
	4	1000588	Joni	P00057942	М	26-35	28	1	Gujarat	,
	11246	1000695	Manning	P00296942	М	18-25	19	1	Maharashtra	1
	11247	1004089	Reichenbach	P00171342	М	26-35	33	0	Haryana	1
	11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	
	11249	1004023	Noonan	P00059442	М	36-45	37	0	Karnataka	S
	11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	1

11239 rows × 13 columns

Count, Mean, Std., etc

In [23]:	df.des	scribe()				
Out[23]:		User_ID	Age	Marital_Status	Orders	Amount
	count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
	mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
	std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
	min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
	25%	1.001492e+06	27.000000	0.000000	2.000000	5443.000000
	50%	1.003064e+06	33.000000	0.000000	2.000000	8109.000000
	75%	1.004426e+06	43.000000	1.000000	3.000000	12675.000000
	max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000
In [24]:	df[['	Ane' 'Order	rs' 'Amount	']].describe(1	
	ui[[/				,	
Out[24]:		Age	Orders	Amount		
	count	11239.000000	11239.000000	11239.000000		
	mean	35.410357	2.489634	9453.610553		
	std	12.753866	1.114967	5222.355168		
	min	12.000000	1.000000	188.000000		
	25%	27.000000	2.000000	5443.000000		
	50%	33.000000	2.000000	8109.000000		
	75%	43.000000	3.000000	12675.000000		
	75% max	43.000000 92.000000	3.000000 4.000000	12675.000000 23952.000000		

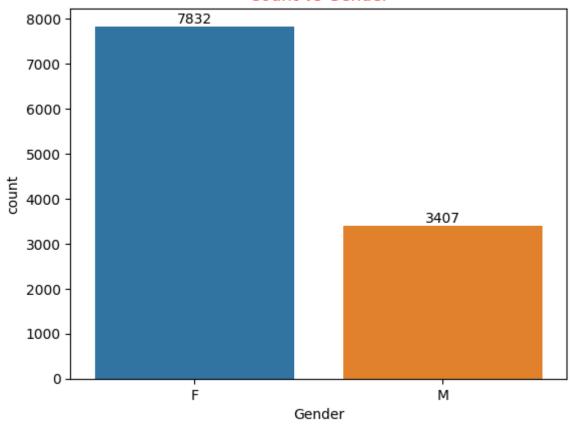
Exploratory Data Analysis

Gender

```
In [28]: ax = sns.countplot(x = 'Gender', data = df)

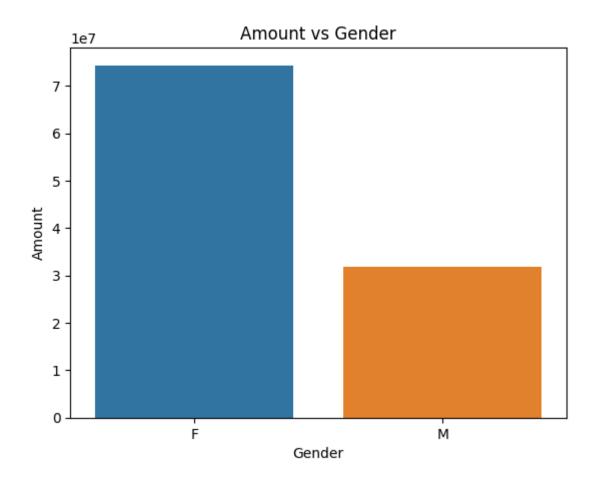
for bars in ax.containers:
    ax.bar_label(bars)
    plt.title('Count vs Gender', color='red')
Loading [MathJax]/extensions/Safe.js
```

Count vs Gender



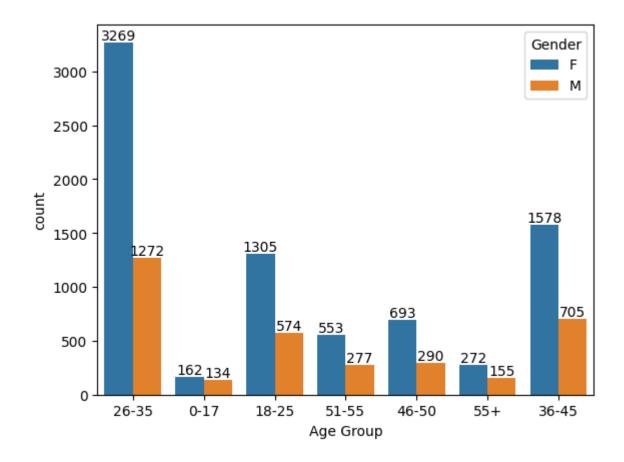
```
In [27]: sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_value
sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gen)
plt.title('Amount vs Gender')
```

Out[27]: Text(0.5, 1.0, 'Amount vs Gender')

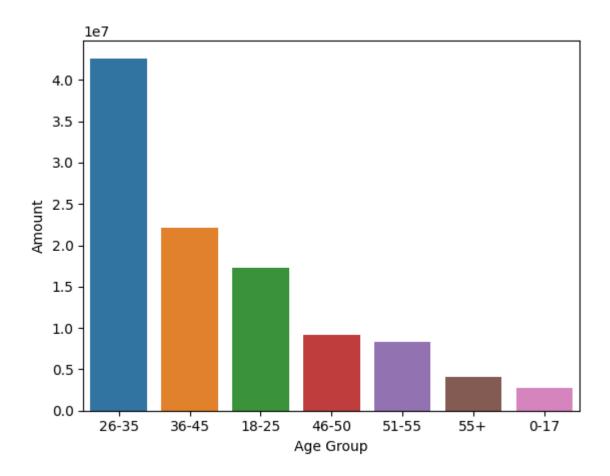


Age Group

```
In [29]: ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



Amount vs Age Group



States

```
In [47]: sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_val
    sns.set(rc={'figure.figsize':(15,5)})
    sns.barplot(data = sales_state, x = 'State',y= 'Orders')
    plt.title('Order vs States')
```

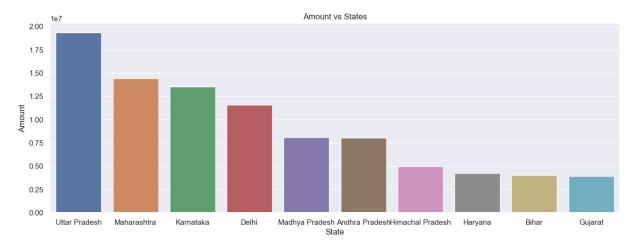
Out[47]: Text(0.5, 1.0, 'Order vs States')



```
In [46]: sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_val
Loading[MathJax]/extensions/Safe.js
```

```
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Amount')
plt.title('Amount vs States')
```

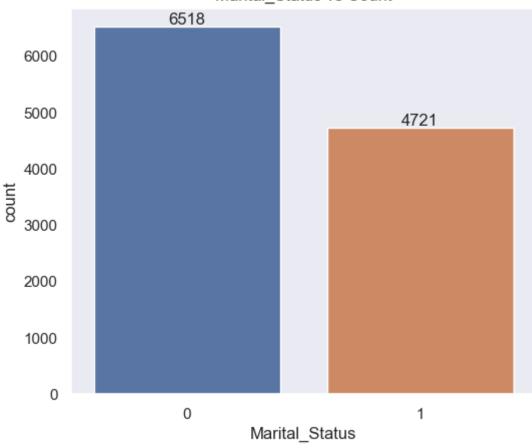
Out[46]: Text(0.5, 1.0, 'Amount vs States')



Marital Status

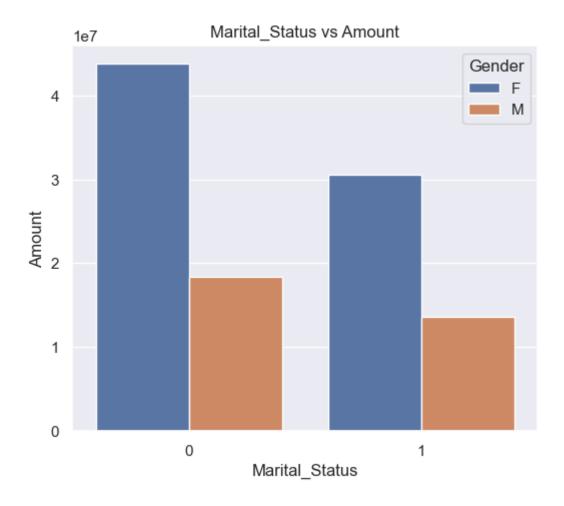
```
In [43]: ax = sns.countplot(data = df, x = 'Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
    ax.grid(False)
    plt.title('Marital_Status vs Count')
```

Marital_Status vs Count



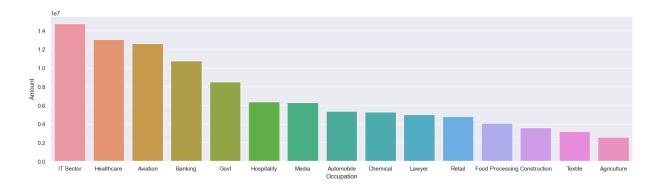
```
In [42]: sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amou
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gende
ax.grid(False)
plt.title('Marital_Status vs Amount')
```

Out[42]: Text(0.5, 1.0, 'Marital Status vs Amount')



Occupations

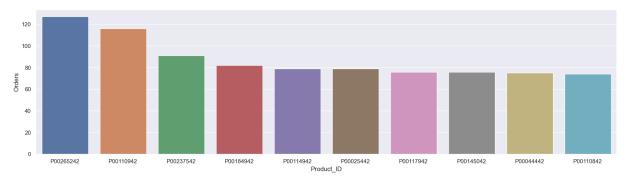
```
In [40]:
         sns.set(rc={'figure.figsize':(20,5)})
          ax = sns.countplot(data = df, x = 'Occupation')
          ax.grid(False)
          for bars in ax.containers:
              ax.bar label(bars)
         1400
         1200
         1000
        800
800
         600
         400
In [50]: sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sor
          sns.set(rc={'figure.figsize':(20,5)})
          sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
          ax.grid(False)
```



Product Category

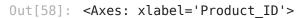
```
In [52]:
               sns.set(rc={'figure.figsize':(20,6)})
               ax = sns.countplot(data = df, x = 'Product_Category')
               for bars in ax.containers:
                    ax.bar_label(bars)
                    ax.grid(False)
               500
                        26
                                                   Food Games & ToSports Products BookSilectronics & GadgetSecor Clothing & ApparelBeauty Household itemsPet Care Product_Category
                   Auto Hand & Power Tocstationery Tupperwaffeotwear & Shoes Furniture
   In [54]: sales state = df.groupby(['Product Category'], as index=False)['Amount'].sum
               sns.set(rc={'figure.figsize':(20,5)})
               sns.barplot(data = sales state, x = 'Product Category',y= 'Amount')
               ax.grid(False)
              3.5
              25
             ± 2.0
                           Clothing & Apparel Electronics & Gadgets Footwear & Shoes
                                                             Product_Category
   In [55]: sales state = df.groupby(['Product ID'], as index=False)['Orders'].sum().sor
               sns.set(rc={'figure.figsize':(20,5)})
Loading [MathJax]/extensions/Safe.js ata = sales_state, x = Product_ID', y = Orders'
```

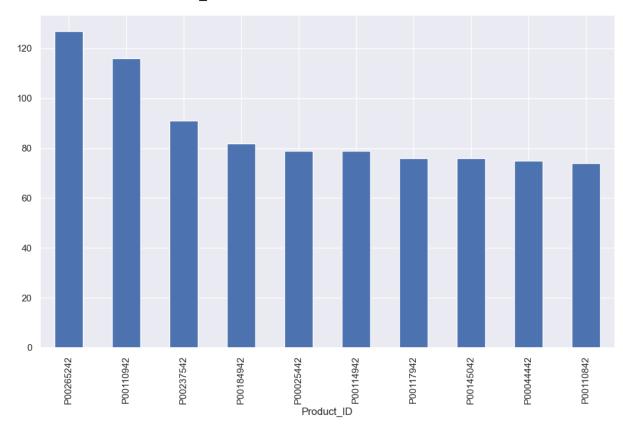
Out[55]: <Axes: xlabel='Product ID', ylabel='Orders'>



Top 10 Most Selling Product

```
In [58]: fig1, ax1 = plt.subplots(figsize=(12,7))
    df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=
```





Conclusion

Married women age group 26-35 yrs from UP, Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothing and Electronics category