In [9]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [14]:
```
# import csv file
df = pd.read_csv('C:\\Users\\hp\\Downloads\\sales.data.csv', encoding= 'unicode_escape')
```

In [15]:
```
df.shape
```

Out[15]:
```
(11251, 15)
```

In [16]:
```
df.head()
```

Out[16]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanasini	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237942	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN

In [17]:
```
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
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

In [18]:
```
#drop unrelated/blank columns
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

In [19]:
```
#check for null values
pd.isnull(df).sum()
```

Out[19]:
```
User_ID         0
Cust_name       0
Product_ID      0
Gender          0
Age Group       0
Age             0
Marital_Status  0
State           0
Zone            0
Occupation      0
Product_Category  0
Orders          0
Amount         12
dtype: int64
```

In [20]:
```
# drop null values
df.dropna(inplace=True)
```

In [21]:
```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

In [22]:
```
df.columns
```

Out[22]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

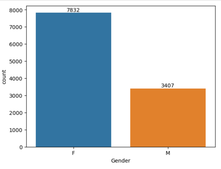
In [23]:
```
df.describe()
```

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%	1.001402e+06	27.000000	0.000000	2.000000	5443.000000
50%	1.003064e+06	33.000000	0.000000	2.000000	8108.000000
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

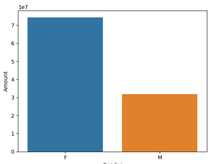
In [24]:
```
#EDA    GENDER
```

In [25]:
```
# plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

for bars in ax.containers:
    ax.bar_label(bars)
```



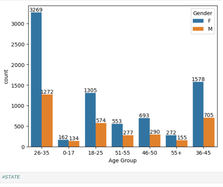
In [26]:
```
# plotting a bar chart for gender vs total amount

sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gen)
```

Out[26]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



In [27]:
```
#AGE
```

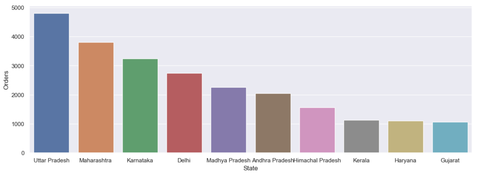
In [28]:
```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

for bars in ax.containers:
    ax.bar_label(bars)
```



In [29]:
```
#STATE
```

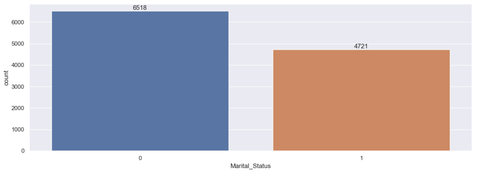
In [30]:
```
# total number of orders from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Orders')
```

Out[30]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



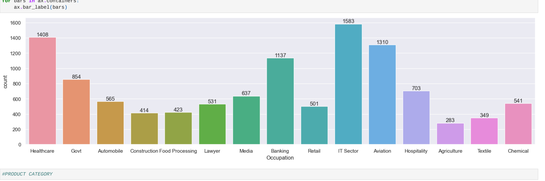
In [31]:
```
#MARITAL STATUS
```

In [32]:
```
ax = sns.countplot(data = df, x = 'Marital_Status')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



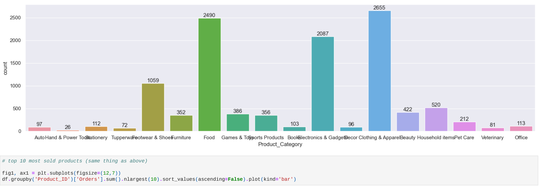
In [33]:
```
# OCCUPATION
```

In [34]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

for bars in ax.containers:
    ax.bar_label(bars)
```



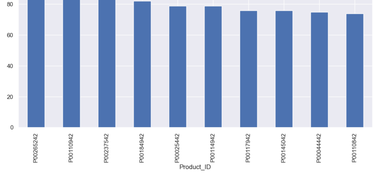
In [35]:
```
#PRODUCT CATEGORY
```

In [36]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

for bars in ax.containers:
    ax.bar_label(bars)
```



In [37]:
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')
```

Out[37]:
```
<Axes: xlabel='Product_ID'>
```



In [ ]:
```
#CONCLUSION

# Married women age 26-35 years from U.P , maharashtra and Karnataka working in IT , Healthcare and aviation are more likely to buy products from food clothing and electronics.

THANK YOU !!!!!!!!!
```