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

```
df=pd.read_csv('diwali_sale_data.csv')  #dataset form of diwali salecase as a dataframe
```

```
df
```

	Unnamed: 0	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000995	Manning	P00296942	M	18-25	19	0	Maharashtra	Western	Chemical	Office	4	370.0	NaN
11247	1004008	Reichenbach	P00157142	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0	NaN
11248	1001209	Oshin	P00125142	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0	NaN
11249	1004023	Noonan	P00289442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0	NaN
11250	1001754	Brumley	P00251742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0	NaN

11251 rows × 15 columns

```
df.shape
```

```
(11251, 15)
```

```
df.head(10)
```

	Unnamed: 0	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	2	23877.00	NaN
6	1001132	Balk	P00291452	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN
7	1003052	Shivangi	P00073442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	NaN	NaN
8	1003224	Kushal	P00258942	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN
9	1003090	Ginny	P00051142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23798.99	NaN

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      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  unnamed1        0 non-null      float64
 14  unnamed2        0 non-null      float64
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

```
#drop blank values
df.drop('Status', 'unnamed1'),axis=1,inplace=True)
```

```
#check null values
pd.isnull(df)
```

	Unnamed: 0	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

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

```
Unnamed: 0          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
```

```
#drop null values
df.dropna(inplace=True)
```

```
df.shape
```

```
(11239, 15)
```

```
# change the data type
df['Amount'].astype('int')
```

```
df['Amount'].dtypes
```

```
dtype('int32')
```

```
df.columns
```

```
Index(['Unnamed: 0', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
#describe method returns description of the data in the DataFrame
df.describe()
```

	Unnamed: 0	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	1.114782	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%	1.004604e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

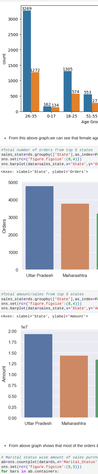
```
# Work_describe method use for only this columns
df[['Age','Orders','Amount']].describe()
```

	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114782	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
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis

```
sns.countplot(x='Gender',data=ds)

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



- From above graph,we can see that most of the buyers are female,even the purchasing power of females are greater then males

```
ax=sns.countplot(x='Age Group',data=ds,hue='Gender')

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

- From this above graph,we can see that female age group between 26-35years are the most buyers group.

```
#total number of orders from top 5 states
sales_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(5)
sns.set(rc={'figure.figsize':(8,4)})
sns.barplot(data=sales_state,x='State',y='Orders')
```

```
<Axes: xlabel='State', ylabel='Orders'>
```


```
#total amount/sales from top 5 states
sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(5)
sns.set(rc={'figure.figsize':(8,4)})
sns.barplot(data=sales_state,x='State',y='Amount')
```

```
<Axes: xlabel='State', ylabel='Amount'>
```

- From above graph shows that most of the orders & total amount are come from Uttar Pradesh, Maharashtra and Karnataka respectively.

```
# Marital status wise amount of sales purchasing value
sns.set(rc={'figure.figsize':(6.41)})
sns.set(rc={'figure.figsize':(6,4)})
for bars in ax.containers:
    ax.bar_label(bars)
```



```
#total number of orders from top 5 states
sales_ms=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(6,4)})
sns.barplot(data=sales_ms,x='Marital_Status',y='Amount',hue='Gender')
```

```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```


- From above graphs, we can see that most of the buyers are married women and they have high purchasing power .

```
# Occupation wise sales
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data=ds,x='Occupation')
for bars in ax.containers:
    ax.bar_label(bars)
```



```
#Purchasing amount occupation wise
sales_occ=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_occ,x='Occupation',y='Amount')
```

```
<Axes: xlabel='Occupation', ylabel='Amount'>
```


- From above graphs show that most of the buyers are working in IT,Healthcare and Aviation sector.

```
# Product Category
sns.set(rc={'figure.figsize':(12,5)})
ax=sns.countplot(data=ds,x='Product_Category')

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



```
#Product category wise purchasing value
sales_prods_groups=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_prods_groups,x='Product_Category',y='Amount')
```

```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```


- From above graphs show that most of the sold products are from Food,Clothing & Apparel,Electronics & Gadgets categories.

```
#Total number of orders according to product id
sales_prods_groups=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_prods_groups,x='Product_ID',y='Orders')
```

```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```


Conclusion -

26-35 years age group married women from Uttar Pradesh, Maharashtra & Karnataka and mostly working in IT, Healthcare ,Aviation are more likely to buy products from Food, Clothing and Electronics categories.

```
```