

```
In [1]: # import python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: # import csv file
df = pd.read_csv('Diwali Sales Data.csv', encoding='unicode_escape')

In [3]: df.shape
Out[3]: (11251, 15)

In [4]: df.head()
Out[4]:
   User_ID  Cust_name  Product_ID  Gender  Age Group  Age  Marital_Status  State  Zone  Occupation  Product_Category  Orders  Amount  Status  unnamed1
0  1002903  Sanskriti  P00125942  F      26-35    28      0      Maharashtra  Western  Healthcare  Auto              1  23952.0  NaN      NaN
1  1000732  Karik      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  Sudeshi    P00237842  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 [5]: df.info()
Out[5]:
<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 [6]: #drop unrelated/blank columns
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)

In [7]: #check for null values
pd.isnull(df).sum()
Out[7]:
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 [8]: # drop null values
df.dropna(inplace=True)

In [9]: # change data type
df['Amount'] = df['Amount'].astype('int')

In [10]: df['Amount'].dtypes
Out[10]: dtype('int32')

In [11]: df.columns
Out[11]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age', 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category', 'Orders', 'Amount'],
      <dtype='object'>

In [12]: #rename column
df.rename(columns={'Marital_Status':'Shaadi'})

Out[12]:
   User_ID  Cust_name  Product_ID  Gender  Age Group  Age  Shaadi  State  Zone  Occupation  Product_Category  Orders  Amount
0  1002903  Sanskriti  P00125942  F      26-35    28      0      Maharashtra  Western  Healthcare  Auto              1  23952
1  1000732  Karik      P00110942  F      26-35    35      1      Andhra Pradesh  Southern  Govt         Auto              3  23934
2  1001990  Bindu      P00118542  F      26-35    35      1      Uttar Pradesh  Central   Automobile  Auto              3  23924
3  1001425  Sudeshi    P00237842  M      0-17    16      0      Karnataka    Southern  Construction  Auto              2  23912
4  1000588  Joni       P00057942  M      26-35    28      1      Gujarat       Western  Food Processing  Auto              2  23877
...
11246  1000695  Manning  P00296942  M      18-25    19      1      Maharashtra  Western  Chemical      Office              4  370
11247  1004089  Reichenbach  P00171342  M      26-35    33      0      Haryana       Northern  Healthcare     Veterinary          3  367
11248  1001209  Oshin     P00201342  F      36-45    40      0      Madhya Pradesh  Central   Textile        Office              4  213
11249  1004023  Noonan    P00059442  M      36-45    37      0      Karnataka    Southern  Agriculture     Office              3  206
11250  1002744  Burnley    P00281742  F      18-25    19      0      Maharashtra  Western  Healthcare     Office              3  188

11239 rows x 13 columns
```

```
In [13]: # describe() method returns description of the data in the DataFrame (i.e. count, mean, std, etc)
df.describe()
Out[13]:
   User_ID      Age  Marital_Status  Orders  Amount
count  1.123090e+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 [14]: # use describe() for specific columns
df[['Age', 'Orders', 'Amount']].describe()
Out[14]:
   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
max     92.000000     4.000000 23952.000000
```

Exploratory Data Analysis

Gender

```
In [15]: # 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)

Out[15]:
8000
7000
6000
5000
4000
3000
2000
1000
0
F M
Gender
7832 3407
```

From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

Age

```
In [17]: ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')
for bars in ax.containers:
    ax.bar_label(bars)

Out[17]:
3269 1272 162 134 1305 574 553 277 693 290 272 155 1578 705
26-35 0-17 18-25 51-55 46-50 55+ 36-45
Age Group
Gender
F M
```

From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

State

```
In [19]: # 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.barplot(data = sales_state, x = 'State', y = 'Orders')

Out[19]:
5000
4000
3000
2000
1000
0
Uttar Pradesh Maharashtra Karnataka Delhi Madhya Pradesh Andhra PradeshHimachal Pradesh Kerala Haryana Gujarat
State
Orders
```

```
In [20]: # total amount/sales from top 10 states
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State', y = 'Amount')

Out[20]:
200
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Uttar Pradesh Maharashtra Karnataka Delhi Madhya Pradesh Andhra PradeshHimachal Pradesh Haryana Bihar Gujarat
State
Amount
```

From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

Marital Status

```
In [21]: ax = sns.countplot(data = df, x = 'Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)

Out[21]:
6000
5000
4000
3000
2000
1000
0
0 1
Marital_Status
6518 4721
```

```
In [22]: sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Marital_Status', y = 'Amount', hue='Gender')

Out[22]:
1e7
4
3
2
1
0
0 1
Marital_Status
Gender
F M
```

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

Occupation

```
In [23]: sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')
for bars in ax.containers:
    ax.bar_label(bars)

Out[23]:
1600
1400
1200
1000
800
600
400
200
0
Healthcare Govt Automobile ConstructionFood Processing Lawyer Media Banking Occupation Retail IT Sector Aviation Hospitality Agriculture Textile Chemical
Occupation
count
```

```
In [24]: sales_state = 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_state, x = 'Occupation', y = 'Amount')

Out[24]:
1e7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
IT Sector Healthcare Aviation Banking Govt Hospitality Media Automobile Chemical Lawyer Retail Food Processing Construction Textile Agriculture
Occupation
Amount
```

From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

Product Category

```
In [25]: sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')
for bars in ax.containers:
    ax.bar_label(bars)

Out[25]:
2500
2000
1500
1000
500
0
Food Games & Toys Sports Products Electronics & GadgetsBeauty Household ItemsPet Care Veterinary Office
Product_Category
count
```

```
In [26]: sales_state = 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_state, x = 'Product_Category', y = 'Amount')

Out[26]:
1e7
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Food Clothing & Apparel Electronics & Gadgets Footwear & Shoes Furniture Games & Toys Sports Products Beauty Auto Stationery
Product_Category
Amount
```

From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

```
In [27]: sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,7)})
sns.barplot(data = sales_state, x = 'Product_ID', y = 'Orders')

Out[27]:
120
100
80
60
40
20
0
P00265042 P00110942 P00237542 P00184942 P00114942 P0025442 P00117942 P00145042 P0004442 P00110842
Product_ID
Orders
```

```
In [28]: # 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[28]:
<Axes: xlabel='Product_ID'>
```



Conclusion:

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