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

```
df = pd.read_csv("Diwali Sales Dataset.csv")
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unn
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	
1	1003733	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	

```
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
```

```
df.describe()
```

	User_ID	Age	Marital_Status	Orders	Amount	Status	unnamed1
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000	0.0	0.0
mean	1.003004e+06	35.421820	0.420318	2.489290	9453.610858	NaN	NaN
std	1.716039e+03	12.754122	0.493632	1.115047	5222.355869	NaN	NaN
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000	NaN	NaN
25%	1.001492e+06	27.000000	0.000000	1.000000	5443.000000	NaN	NaN
50%	1.003064e+06	33.000000	0.000000	2.000000	8109.000000	NaN	NaN
75%	1.004426e+06	43.000000	1.000000	3.000000	12675.000000	NaN	NaN
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000	NaN	NaN

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

```
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
Status          11251
unnamed1        11251
dtype: int64
```

Data Cleaning:

Step to DROP Blank/unwanted columns:

```
df.drop(['Status', 'unnamed1'], axis = 1, inplace = True)
```

```
df
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0
1	1003733	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000995	Manning	P00266942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0
11247	1004089	Reichenbach	P00273442	M	55+	61	0	Haryana	Northern	Healthcare	Veterinary	3	367.0
11248	1001209	Oshin	P00031342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0
11249	1004023	Noonan	P00228442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0
11250	1002744	Brumley	P00057942	M	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0

11251 rows × 13 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 13 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
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

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

```
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
```

```
df.shape
```

```
(11251, 13)
```

DROP null values

```
df.dropna(inplace = True)
```

```
df.shape
```

```
(11239, 13)
```

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

```
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              0
dtype: int64
```

```
df.shape
```

```
(11239, 13)
```

Exploratory Data Analysis

Gender

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

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

	Gender	Amount
0	F	74335853.0
1	M	31913276.0

```
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)
```

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

Age

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

Total Amount vs Age Group

```
sales_age = df.groupby(['Age Group'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending = False)
sns.barplot(x = 'Age Group', y = 'Amount', data = sales_age)
```

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

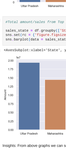
State

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' : (16,5)})
sns.barplot(data = sales_state, x = 'State', y = 'Orders')
```

```
#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' : (16,5)})
sns.barplot(data = sales_state, x = 'State', y = 'Amount')
```



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

Martial Status

```
ax = sns.countplot(data = df, x = 'Marital_Status')
sns.set(rc = {'figure.figsize' : (5,5)})
```

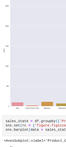
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending = False)
sns.set(rc = {'figure.figsize' : (15,5)})
sns.barplot(data = sales_state, x = 'Marital_Status', y = 'Amount',hue = 'Gender')
```

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

Occupation

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

```
sales_state = df.groupby(['Occupation'], 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 = 'Occupation', y = 'Amount')
```

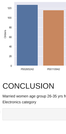


Insights: From above graphs we can see that most of the buyers are working in IT,Aviation and Healthcare sector

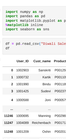
Product Category

```
ax = sns.countplot(data = df, x = 'Product_Category')
sns.set(rc = {'figure.figsize' : (30,5)})
```

```
sales_state = df.groupby(['Product_Category'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending = False).head(10)
sns.set(rc = {'figure.figsize' : (30,5)})
sns.barplot(data = sales_state, x = 'Product_Category', y = 'Amount')
```



```
sales_state = df.groupby(['Product_ID'], as_index = False)['Orders'].sum().sort_values(by = 'Orders', ascending = False).head(10)
sns.set(rc = {'figure.figsize' : (20,5)})
sns.barplot(data = sales_state, x = 'Product_ID', y = 'Orders')
```



CONCLUSION

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