1: Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
```

2: Load the Data

In [3]:
```
pj=pd.read_csv('Diwali_Sales_Data.csv',encoding='unicode_escape')
```

In [4]:
```
# FIND OUT TOTAL NUMBER OF ROWS AND COLUMNS PRESENT
pj.shape
```

Out[4]:
```
(11251, 15)
```

In [5]:
```
pj.head()
```

Out[5]:

	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	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	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 [6]:
```
#INFORMATION PRESENT IN THE DATA
pj.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
```

3: Data Cleaning

In [9]:
```
#DROP UNRELATED OR BLANK COLUMNS
pj.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

In [11]:
```
#CHECK FOR NULL VALUES
pd.isnull(pj).sum()
```

Out[11]:
```
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 [12]:
```
#DROP NULL VALUES
pj.dropna(inplace=True)
```

4: Data Type Conversion

In [13]:
```
#CHANGE 'AMOUNT' COLUMN TO INTEGER
pj['Amount'] = pj['Amount'].astype('int')
```

In [14]:
```
pj['Amount'].dtypes
```

Out[14]:
```
dtype('int32')
```

In [15]:
```
pj.columns
```

Out[15]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

5: Data Description

In [16]:
```
# DESCRIBE THE ENTIRE DATASET
pj.describe()
```

Out[16]:

	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 [17]:
```
# DESCRIBE SPECIFIC COLUMNS
pj[['Age','Orders','Amount']].describe()
```

Out[17]:

	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

6: GENDER ANALYSIS (DATA PREPARATION)

A: Gender Count

In [18]:
```
# COUNT DATA FOR EACH GENDER
gender_count=pj['Gender'].value_counts().reset_index()
gender_count.columns=['Gender','Count']
```

B: Gender VS Total Sales Amount

In [19]:
```
# TOTAL SALES AMOUNT BY GENDER
sales_gender=pj.groupby('Gender')['Amount'].sum().sort_values(ascending=False).reset_index()
```

7: AGE GROUP ANALYSIS (DATA PREPARATION)

A: Age Group and Gender Distribution

In [20]:
```
# COUNT DATA FOR EACH AGE GROUP WITH GENDER
age_gender_count=pj.groupby(['Age Group','Gender']).size().reset_index(name='Count')
```

B: Total Sales Amount by Age Group

In [21]:
```
# TOTAL SALES AMOUNT BY AGE GROUP
sales_age=pj.groupby('Age Group')['Amount'].sum().sort_values(ascending=False).reset_index()
```

8: STATE ANALYSIS (DATA PREPARATION)

A: Top 10 States by Number of Orders

In [23]:
```
# TOP 10 STATES BY NUMBER OF ORDERS
sales_states_orders=pj.groupby('State')['Orders'].sum().sort_values(ascending=False).head(10).reset_index()
```

B: Top 10 States by Sales Amount

In [24]:
```
sales_state_amount=pj.groupby('State')['Amount'].sum().sort_values(ascending=False).head(10).reset_index()
```

9: MARITAL STATUS ANALYSIS (DATA PREPARATION)

A: Marital Status Distribution

In [25]:
```
#COUNT DATA FOR MARITAL STATUS
marital_status_count=pj['Marital_Status'].value_counts().reset_index()
marital_status_count.columns=['Marital_Status','Count']
```

B: Marital Status VS Total Sales Amount by Gender

In [26]:
```
#TOTAL SALES AMOUNT BY MARITAL STATUS AND GENDER
sales_marital_status=pj.groupby(['Marital_Status','Gender'])['Amount'].sum().sort_values(ascending=False).reset_index()
```

10: OCCUPATION ANALYSIS (DATA PREPARATION)

A: Occupation Distribution

In [28]:
```
#COUNT DATA FOR EACH OCCUPATION
occupation_count=pj['Occupation'].value_counts().reset_index()
occupation_count.columns=['Occupation','Count']
```

B: Total Sales Amount by Occupation

In [29]:
```
#TOTAL SALES AMOUNT BY OCCUPATION
sales_occupation=pj.groupby('Occupation')['Amount'].sum().sort_values(ascending=False).reset_index()
```

11: PRODUCT CATEGORY ANALYSIS (DATA PREPARATION)

A: Product Category Distribution

In [30]:
```
# COUNT DATA FOR EACH PRODUCT CATEGORY
product_category_count=pj['Product_Category'].value_counts().reset_index()
product_category_count.columns = ['Product_Category', 'Count']
```

B: Top 10 Product Categories by Sales Amount

In [31]:
```
# TOP 10 PRODUCT CATEGORIES BY TOTAL SALES AMOUNT
sales_product_category=pj.groupby('Product_Category')['Amount'].sum().sort_values(ascending=False).head(10).reset_index()
```

C: Top 10 Most Sold Products

In [32]:
```
# TOP 10 MOST SOLD PRODUCTS BY ORDERS
sales_product=pj.groupby('Product_ID')['Orders'].sum().sort_values(ascending=False).head(10).reset_index()
```

DATA VISUALIZATION

In [35]:
```
# Save the processed data for use in Power BI
gender_count.to_csv('gender_count.csv', index=False)
sales_gender.to_csv('sales_gender.csv', index=False)
age_gender_count.to_csv('age_gender_count.csv', index=False)
sales_age.to_csv('sales_age.csv', index=False)
sales_states_orders.to_csv('sales_state_orders.csv', index=False)
sales_state_amount.to_csv('sales_state_amount.csv', index=False)
marital_status_count.to_csv('marital_status_count.csv', index=False)
sales_marital_status.to_csv('sales_marital_status.csv', index=False)
occupation_count.to_csv('occupation_count.csv', index=False)
sales_occupation.to_csv('sales_occupation.csv', index=False)
product_category_count.to_csv('product_category_count.csv', index=False)
sales_product_category.to_csv('sales_product_category.csv', index=False)
sales_product.to_csv('sales_product.csv', index=False)
```

In [38]:
```
# Export data to CSV files in D:/csv files folder
pj.to_csv('D:/csv files/Cleaned_Diwali_Sales_Data.csv', index=False)
sales_gender.to_csv('D:/csv files/Gender_Sales_Data.csv', index=False)
sales_age.to_csv('D:/csv files/Age_Group_Sales_Data.csv', index=False)
sales_states_orders.to_csv('D:/csv files/Top_States_Orders_Data.csv', index=False)
sales_state_amount.to_csv('D:/csv files/Top_States_Amount_Data.csv', index=False)
sales_marital_status.to_csv('D:/csv files/Marital_Status_Sales_Data.csv', index=False)
sales_occupation.to_csv('D:/csv files/Occupation_Sales_Data.csv', index=False)
sales_product_category.to_csv('D:/csv files/Top_Product_Category_Sales_Data.csv', index=False)
sales_product.to_csv('D:/csv files/Top_Product_ID_Orders_Data.csv', index=False)
```

In [ ]:
```

```