

About Walmart:

Walmart is an American multinational retail corporation that operates a chain of supercenters, discount departmental stores, and grocery stores from the United States. Walmart has more than 100 million customers worldwide.

Business Problem:

The Management team at Walmart Inc. wants to analyze the customer purchase behavior (specifically, purchase amount) against the customer's gender and the various other factors to help the business make better decisions. They want to understand if the spending habits differ between male and female customers: Do women spend more on Black Friday than men? (Assume 50 million customers are male and 50 million are female).

1.Defining Problem Statement and Analyzing basic metrics

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

df = pd.read_csv('Walmart_BusinessCase_data.csv')
df.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00005410	F	0-	10	A	2	

```
df.size

5500680

df.shape

(550068, 10)

df.ndim

2

len(df)

550068

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   User_ID                               550068 non-null  int64
1   Product_ID                           550068 non-null  object
2   Gender                                550068 non-null  object
3   Age                                   550068 non-null  object
4   Occupation                            550068 non-null  int64
5   City_Category                         550068 non-null  object
6   Stay_In_Current_City_Years            550068 non-null  object
7   Marital_Status                        550068 non-null  int64
8   Product_Category                      550068 non-null  int64
9   Purchase                             550068 non-null  int64
```

dtypes: int64(5), object(5)
memory usage: 42.0+ MB

Conversion of categorical attributes to 'category' :

```
columns = ['Occupation','Marital_Status','Product_Category']  
df[columns] = df[columns].astype(object)  
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 550068 entries, 0 to 550067  
Data columns (total 10 columns):  
#   Column                                Non-Null Count  Dtype  
---  ---                                -  
0   User_ID                             550068 non-null int64  
1   Product_ID                         550068 non-null object  
2   Gender                             550068 non-null object  
3   Age                                550068 non-null object  
4   Occupation                         550068 non-null object  
5   City_Category                     550068 non-null object  
6   Stay_In_Current_City_Years        550068 non-null object  
7   Marital_Status                    550068 non-null object  
8   Product_Category                  550068 non-null object  
9   Purchase                          550068 non-null int64  
dtypes: int64(2), object(8)  
memory usage: 42.0+ MB
```

Statistical Summary

```
df.describe(include = 'all').T
```

	count	unique	top	freq	mean	std	min
User_ID	550068.0	NaN	NaN	NaN	1003028.842401	1727.591586	1000001.0
Product_ID	550068	3631	P00265242	1880	NaN	NaN	NaN
Gender	550068	2	M	414259	NaN	NaN	NaN
Age	550068	7	26-35	219587	NaN	NaN	NaN
Occupation	550068.0	21.0	4.0	72308.0	NaN	NaN	NaN
City_Category	550068	3	B	231173	NaN	NaN	NaN
Stay_In_Current_City_Years	550068	5	1	193821	NaN	NaN	NaN
Marital_Status	550068.0	2.0	0.0	324731.0	NaN	NaN	NaN
Product_Category	550068.0	20.0	5.0	150933.0	NaN	NaN	NaN
Purchase	550068.0	NaN	NaN	NaN	9263.968713	5023.065394	12.0

Checking Null values

```
df.isnull().sum()  
  
User_ID                0  
Product_ID            0  
Gender                0  
Age                  0  
Occupation            0  
City_Category         0  
Stay_In_Current_City_Years  0  
Marital_Status        0  
Product_Category      0  
Purchase              0  
dtype: int64
```

2. Non-Graphical Analysis: Value counts and unique attributes

```
#Unique Attributes - User_ID column
```

```
df['User_ID'].nunique()
```

```
5891
```

```
#Unique Attributes - Product_ID column
```



```
df['Product_ID'].nunique()
```

```
3631
```

Value_counts for the following:

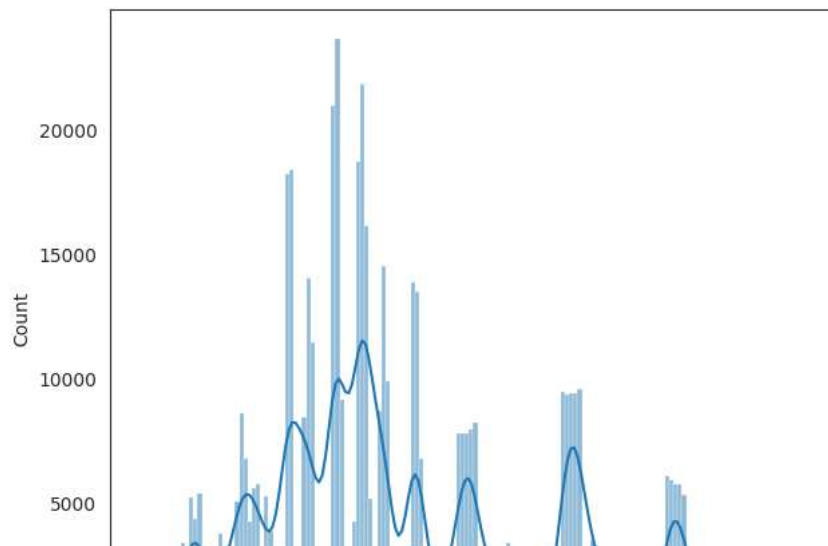
- Gender
- Age
- Occupation
- City_Category
- Stay_In_Current_City_Years
- Marital_Status
- Product_Category
- Age

```
categorical_cols = ['Gender', 'Age', 'Occupation', 'City_Category',  
'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']  
df[categorical_cols].melt().groupby(['variable', 'value'])['value'].count()/len(df)
```

		value	
variable	value		
Age	0-17	0.027455	
	18-25	0.181178	
	26-35	0.399200	
	36-45	0.199999	
	46-50	0.083082	
	51-55	0.069993	
	55+	0.039093	
City_Category	A	0.268549	
	B	0.420263	
	C	0.311189	
Gender	F	0.246895	
	M	0.753105	
Marital_Status	0	0.590347	
	1	0.409653	
Occupation	0	0.126599	
	1	0.086218	
	2	0.048336	
	3	0.032087	
	4	0.131453	
	5	0.022137	
	6	0.037005	
	7	0.107501	
	8	0.002811	
	9	0.011437	
	10	0.023506	
	11	0.021063	
	12	0.056682	
	13	0.014049	
	14	0.049647	
	15	0.022115	
	16	0.046123	
	17	0.072796	
	18	0.012039	

3.Visual Analysis - Univariate & Bivariate

```
20 0.001014
For continuous variable(s): countplot, histogram for univariate analysis
2 0.043384
#Histogram for Univariate variable
plt.figure(figsize=(7,6))
sns.histplot(data = df, x = 'Purchase', kde = True)
plt.show()
```



#Countplot for Univariate variable

```
plt.figure(figsize=(7,6))
sns.countplot(data = df, x = 'Purchase')
plt.show()
```

#Boxplot for Univariate ()

```
sns.boxplot(data = df, x = 'Purchase')
plt.show
```

Observations :

- Purchase is having outliers
- 25% of data shows 6000 purchases
- 50% of data shows 9000 purchases
- 75% of data shows 13500 purchases

For Categorical variable(s): countplot for univariate analysis

```
#Countplot for Univariate ()

plt.figure(figsize=(10, 8))
sns.countplot(data=df, x='Product_Category')
plt.show()
```

Observations:

- Product category 5 is sold the highest followed by Product category 1 and 8 respectively

```
#pie charts for 'Age' and Stay in Current year'
```

```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(12, 8))
```

```
data = df['Age'].value_counts(normalize=True)*100
```

```

palette_color = sns.color_palette('BrBG_r')
axs[0].pie(x=data.values, labels=data.index, autopct='%0f%%',
           colors=palette_color)
axs[0].set_title("Age")
data = df['Stay_In_Current_City_Years'].value_counts(normalize=True)*100
palette_color = sns.color_palette('YlOrRd_r')
axs[1].pie(x=data.values, labels=data.index, autopct='%0f%%',
           colors=palette_color)
axs[1].set_title("Stay_In_Current_City_Years")

plt.show()

```

Bivariate Analysis :

```

# Analysis of purchase V/s each categorical field

attrs = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
sns.set_style("white")
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(30, 16))
fig.subplots_adjust(top=1.3)
count = 0
for row in range(3):
    for col in range(2):
        sns.boxplot(data=df, y='Purchase', x=attrs[count], ax=axs[row,col], palette='Set3')
        axs[row,col].set_title(f"Purchase vs {attrs[count]}", pad=12,
                               fontsize=13)
        count += 1
plt.show()

```

```
#Purchase analysis for each Gender
```

```
sns.barplot(data = df, x= 'Gender', y = 'Purchase', hue = 'Age')  
plt.show()
```

▼ 2. Missing Value & Outlier Detection

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


The given dataset has no null values

```
#Using pandas describe() to find outliers:  
  
df.describe()
```

▾ Observations:

- **Null values present** : We see that there are no null values/ missing values present in dataset given.
- **Purchase amount might have outliers**: The max Purchase amount is 23961 while its mean is 9263.96. The mean is sensitive to outliers, but the fact the mean is so small compared to the max value indicates the max value is an outlier.

```
sns.boxplot(data=df, x='Purchase', orient='h')  
plt.show()
```

Purchase has outliers

Using the pandas .quantile() function, lets see the outliers behaviour :

```
#create a function to find outliers using IQR  
def find_outliers_IQR(df):  
    q1=df.quantile(0.25)  
    q3=df.quantile(0.75)  
    IQR=q3-q1  
    outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]  
    return outliers  
  
outliers = find_outliers_IQR(df["Purchase"])  
print("number of outliers: "+ str(len(outliers)))  
print("max outlier value:"+ str(outliers.max()))  
print("min outlier value: "+ str(outliers.min()))
```

▾ 3. Business Insights based on Non- Graphical and Visual Analysis

Insights for non graphical Analysis: Comments on range of attributes the distribution of the variables and relationship between them

- The given dataset has total size of 550068. The shape of dataset is (550068, 10) having 2D Dimensional.
- There are total 5891 unique values for user ID
- There are total 3631 unique values for Product ID
- 75% of the users are Male and 25% are Female
- ~ 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45)
- 60% Single, 40% Married
- 35% Staying in the city from 1 year, 18% from 2 years, 17% from 3 years
- Total of 18 product categories are there
- There are 20 different types of occupations in the city

Observations drawn from univariate and bivariate plot :

- Most of the users are Male
- There are 20 different types of Occupation and Product_Category
- More users belong to B City_Category
- More users are Single as compare to Married
- Product_Category - 1, 5, 8, & 11 have highest purchasing frequency.

▼ 4.Answering Questions

1.Are women spending more money per transaction than men? Why or Why not?

```
df['Gender'].value_counts()
```

```
df.groupby(df['Gender'])['Purchase'].sum()
```

```
# histogram of average amount spend for each customer - Male & Female
df[df['Gender']=='M']['Purchase'].hist(bins=35)
plt.xlabel("Purchase")
plt.ylabel("Count")
plt.title("Male Purchase")
plt.show()
df[df['Gender']=='F']['Purchase'].hist(bins=35)
plt.xlabel("Purchase")
plt.ylabel("Count")
plt.title("Female Purchase")
plt.show()
```

```

male_avg = df[df['Gender']=='M']['Purchase'].mean()
female_avg = df[df['Gender']=='F']['Purchase'].mean()

print("Average amount spend by Male customers:",(male_avg))
print("Average amount spend by Female customers:",(female_avg))

```

Observation:

- Male customers spend more money than female customers

2.Confidence intervals and distribution of the mean of the expenses by female and male customers

```

male_df = df[df['Gender']=='M']
female_df = df[df['Gender']=='F']
genders = ["M", "F"]
male_sample_size = 3000
female_sample_size = 3000
num_repitions = 1000

male_means = []
female_means = []
for _ in range(num_repitions):
    male_mean = male_df.sample(male_sample_size,replace=True)['Purchase'].mean()
    female_mean = female_df.sample(female_sample_size,
    replace=True)['Purchase'].mean()
    male_means.append(male_mean)
    female_means.append(female_mean)

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
axis[0].hist(male_means, bins=20)
axis[1].hist(female_means, bins=20)
axis[0].set_title("Male - Distribution of means, Sample size: 3000")
axis[1].set_title("Female - Distribution of means, Sample size: 3000")
plt.show()

```

```

print("Population mean - Mean of sample means of amount spend for Male:",(np.mean(male_means)))
print("Population mean - Mean of sample means of amount spend for Female:",(np.mean(female_means)))
print("")
print("Male - Sample mean: {:.2f} Sample SD : {:.2f}".format(male_df['Purchase'].mean(), male_df['Purchase'].std()))
print("Female - Sample mean: {:.2f} Sample SD : {:.2f}".format(female_df['Purchase'].mean(), female_df['Purchase'].std()))

```

Observations: For the given population using the Central Limit Theorem we can say that:

1. Average amount spend by male customers is **9433.74**
2. Average amount spend by female customers is **8739.67**

3. Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?

```

male_margin_of_error_clt =1.96*male_df['Purchase'].std()/np.sqrt(len(male_df))
male_sample_mean = male_df['Purchase'].mean()
male_lower_lim = male_sample_mean - male_margin_of_error_clt
male_upper_lim = male_sample_mean + male_margin_of_error_clt
female_margin_of_error_clt =1.96*female_df['Purchase'].std()/np.sqrt(len(female_df))
female_sample_mean = female_df['Purchase'].mean()
female_lower_lim = female_sample_mean - female_margin_of_error_clt
female_upper_lim = female_sample_mean + female_margin_of_error_clt
print("Male confidence interval of means: ({:.2f},{:.2f})".format(male_lower_lim, male_upper_lim))
print("Female confidence interval of means: ({:.2f},{:.2f})".format(female_lower_lim, female_upper_lim))

```

Now we can infer about the population that, 95% of the times:

1. Average amount spend by male customer will lie in between: **(9422.02,9453.03)**
2. Average amount spend by female customer will lie in between: **(8709.21,8759.92)**

4.Results when the same activity is performed for Married vs Unmarried

```

data_df = df.groupby(['User_ID', 'Marital_Status'])['Purchase'].sum()
data_df = data_df.reset_index()
data_df['Marital_Status'].value_counts()
mar_samp_size = 3000
unmar_sample_size = 2000
num_repitions = 1000
mar_means = []
unmar_means = []

```

```

for _ in range(num_repitions):
    mar_mean =data_df[data_df['Marital_Status']==1].sample(mar_samp_size,replace=True)['Purchase'].mean()
    unmar_mean = data_df[data_df['Marital_Status']==0].sample(unmar_sample_size,replace=True)['Purchase'].mean()
    mar_means.append(mar_mean)
    unmar_means.append(unmar_mean)

```

```

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(18, 6))
axis[0].hist(mar_means, bins=30)
axis[1].hist(unmar_means, bins=30)
axis[0].set_title("Married - Distribution of means, Sample size: 3000")
axis[1].set_title("Unmarried - Distribution of means, Sample size: 2000")
plt.show()

```

```

print("Population mean - Mean of sample means of amount spend for Married: {:.2f}".format(np.mean(mar_means)))
print("Population mean - Mean of sample means of amount spend for Unmarried: {:.2f}".format(np.mean(unmar_means)))
print("\nMarried - Sample mean: {:.2f} Sample std : {:.2f}".format(data_df[data_df['Marital_Status']==1]['Purchase'].mean(),
data_df[data_df['Marital_Status']==1]['Purchase'].std()))
print("Unmarried - Sample mean: {:.2f} Sample std : {:.2f}".format(data_df[data_df['Marital_Status']==0]['Purchase'].mean(),
data_df[data_df['Marital_Status']==0]['Purchase'].std()))

```

```

for val in ["Married", "Unmarried"]:
    new_val = 1 if val == "Married" else 0
    new_df = data_df[data_df['Marital_Status']==new_val]
    margin_of_error_clt = 1.96*new_df['Purchase'].std()/np.sqrt(len(new_df))
    sample_mean = new_df['Purchase'].mean()
    lower_lim = sample_mean - margin_of_error_clt
    upper_lim = sample_mean + margin_of_error_clt

    print("{} confidence interval of means: {:.2f},{:.2f}".format(val, lower_lim, upper_lim))

```

5. Results when the same activity is performed for Age

```

data_df = df.groupby(['User_ID', 'Age'])['Purchase'].sum()
data_df = data_df.reset_index()
print(data_df)
data_df['Age'].value_counts()
sample_size = 200
num_repitions = 1000
all_means = {}
age_intervals = ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']
for age_interval in age_intervals:
    all_means[age_interval] = []

for age_interval in age_intervals:
    for _ in range(num_repitions):
        mean = data_df[data_df['Age']==age_interval].sample(sample_size, replace=True)['Purchase'].mean()
        all_means[age_interval].append(mean)

for val in ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']:
    new_df = data_df[data_df['Age']==val]
    margin_of_error_clt = 1.96*new_df['Purchase'].std()/np.sqrt(len(new_df))
    sample_mean = new_df['Purchase'].mean()
    lower_limit = sample_mean - margin_of_error_clt
    upper_limit = sample_mean + margin_of_error_clt
    print("For age {} --> confidence interval of means: {:.2f},{:.2f}".format(val, lower_limit, upper_limit))

```

▼ 5. Insights based on exploration and CLT

- ~ 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45)
- 75% of the users are Male and 25% are Female
- 60% are Single and 40% Married
- Total of 20 product categories are there
- There are 20 different types of occupations in the city
- Most of the users are Male
- There are 20 different types of Occupation and Product_Category
- More users belong to B City_Category
- More users are Single as compare to Married
- Product_Category - 1, 5, 8, & 11 have highest purchasing frequency.
- Average amount spend by Male customers: 9433.74
- Average amount spend by Female customers: 8739.67

Confidence Interval by Gender Now using the Central Limit Theorem for the population:

1. Average amount spend by male customers is 9433.74
2. Average amount spend by female customers is 8739.67 Now we can infer about the population that, 95% of the times:
3. Average amount spend by male customer will lie in between: (9422.02,9453.03)
4. Average amount spend by female customer will lie in between: (8709.21,8759.92)

Confidence Interval by Marital_Status

1. Married confidence interval of means: (806668.83,880384.76)
2. Unmarried confidence interval of means: (848741.18,912410.38)

6.Recommendations :

- **1. Product_Category** - 1, 5, 8, & 11 have highest purchasing frequency. it means these are the products in these categories are liked more by customers. Company can focus on selling more of these products or selling more of the products which are purchased less.
- Men spent more money than women, So company should focus on retaining the male customers and getting more male customers.
- Unmarried customers spend more money than married customers, So company should focus on acquisition of Unmarried customers.
- Customers in the age 18-45 spend more money than the others, So company should focus on acquisition of customers who are in the age 18-45
- Male customers living in City_Category C spend more money than other male customers living in B or C, Selling more products in the City_Category C will help the company increase the revenue.

