Business Case: Walmart - Confidence Interval and CLT

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).

Dataset

The company collected the transactional data of customers who purchased products from the Walmart Stores during Black Friday. The dataset has the following features:

- User_ID: User ID
- Product_ID: Product ID
- Gender: Sex of User
- Age: Age in bins
- Occupation: Occupation(Masked)
- City_Category: Category of the City (A,B,C)
- StayInCurrentCityYears: Number of years stay in current city
- Marital_Status: Marital Status
- ProductCategory: Product Category (Masked)
- Purchase: Purchase Amount

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

from scipy.stats import binom
from scipy.stats import norm
```

```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv?1641285094"
```

```
--2023-03-21 03:53:53--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv?164128509
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 99.84.170.67, 99.84.170.22, 99.84.170.176, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|99.84.170.67|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23027994 (22M) [text/plain]
Saving to: ‘walmart_data.csv?1641285094.3’

walmart_data.csv?16 100%[===================>]  21.96M  73.0MB/s    in 0.3s

2023-03-21 03:53:53 (73.0 MB/s) - ‘walmart_data.csv?1641285094.3’ saved [23027994/23027994]
```

```
df = pd.read_csv("walmart_data.csv?1641285094")
df
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marita
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	

```
df.shape
```

```
(550068, 10)
```

```
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
```

```
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
```

There are no missing values present in the data. Data is clean to proceed further.

Non-Graphical Analysis: Value counts and unique attributes

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

```
M    414259
F    135809
Name: Gender, dtype: int64
```

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

```
0    324731
1    225337
Name: Marital_Status, dtype: int64
```

```
df["Age"].unique()
```

```
array(['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25'],
      dtype=object)
```

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

```
5891
```

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

```
3631
```

```
df.describe()
```

	User_ID	Occupation	Marital_Status	Product_Category	Purchase
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.000000
mean	1.003029e+06	8.076707	0.409653	5.404270	9263.968713
std	1.727592e+03	6.522660	0.491770	3.936211	5023.065394
min	1.000001e+06	0.000000	0.000000	1.000000	12.000000
25%	1.001516e+06	2.000000	0.000000	1.000000	5823.000000
50%	1.003077e+06	7.000000	0.000000	5.000000	8047.000000
75%	1.004478e+06	14.000000	1.000000	8.000000	12054.000000
max	1.006040e+06	20.000000	1.000000	20.000000	23961.000000

```
df.describe(include="object")
```

	Product_ID	Gender	Age	City_Category	Stay_In_Current_City_Years
count	550068	550068	550068	550068	550068
unique	3631	2	7	3	5
top	P00265242	M	26-35	B	1
freq	1880	414259	219587	231173	193821

```
cols = ['Gender', 'Age', 'City_Category', 'Stay_In_Current_City_Years','Marital_Status']
df[cols].melt().groupby(['variable', 'value'])[['value']].count()/len(df) * 100
```

		value
variable	**value**	
Age	**0-17**	2.745479
	18-25	18.117760
	26-35	39.919974
	36-45	19.999891
	46-50	8.308246
	51-55	6.999316
	55+	3.909335
City_Category	**A**	26.854862
	B	42.026259
	C	31.118880
Gender	**F**	24.689493
	M	75.310507
Marital_Status	**0**	59.034701
	1	40.965299
Stay_In_Current_City_Years	**0**	13.525237
	1	35.235825
	2	18.513711
	3	17.322404
	4+	15.402823

Observations

1)75% of the users are Male and 25% are Female.

2)60% Single, 40% Married are users.

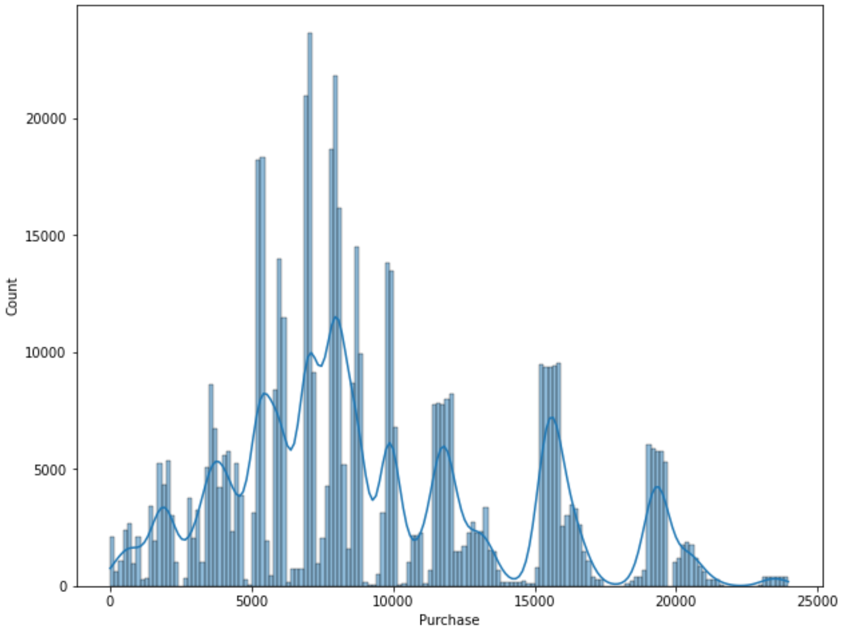
3)35% of users are Staying in the city since 1 year, 18% since 2 years, 17% since 3 years

4)Approx 80% of the users are between 18-45 age (18% -(18-25), 40% - (26-35), 20% - (36-45)).

Understanding on univariant plots on purchase data

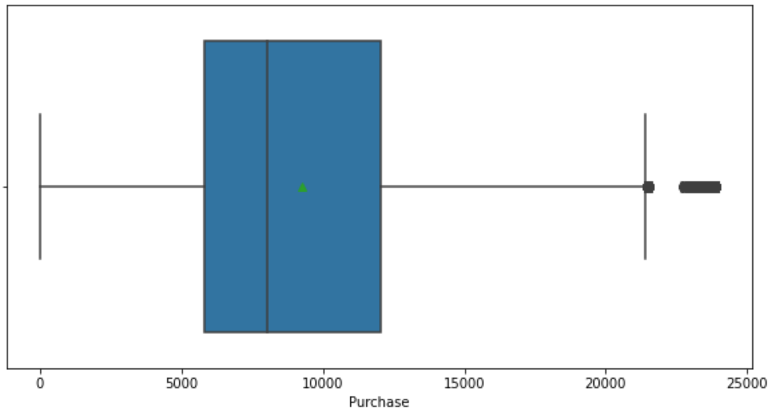
```
plt.figure(figsize=(10, 8))

sns.histplot(data = df , x= "Purchase", kde=True)
plt.show()
```



```
plt.figure(figsize=(10, 5))
sns.boxplot(data = df , x= "Purchase" , showmeans=True)

plt.show()
```

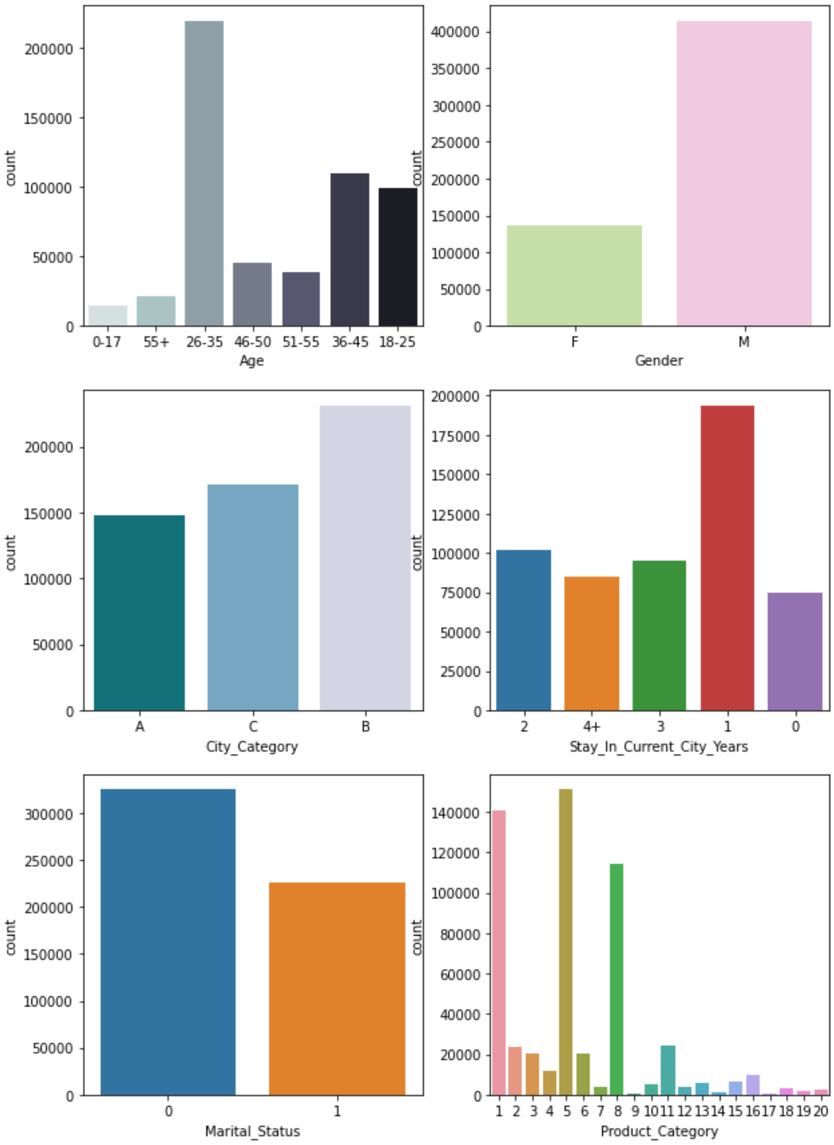


==>Outliers are present in Purchases of users data.

Understanding of categorical data

```
fig,ax = plt.subplots(3,2,figsize=(10,15))
sns.countplot(data= df , x="Age", ax=ax[0,0], palette = "bone_r")
sns.countplot(data= df , x="Gender", ax=ax[0,1], palette = "PiYG_r")
sns.countplot(data= df , x="City_Category", ax=ax[1,0], palette = "PuBuGn_r")
sns.countplot(data= df , x="Stay_In_Current_City_Years", ax=ax[1,1])
sns.countplot(data= df , x="Marital_Status", ax=ax[2,0])
sns.countplot(data= df , x="Product_Category", ax=ax[2,1])
```

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



Observations:

1) Most of the users are males.

2) Most of the users are staying in the current city for ***One*** Year.

3) Users of age between ***26-35*** are the most purchasing users.

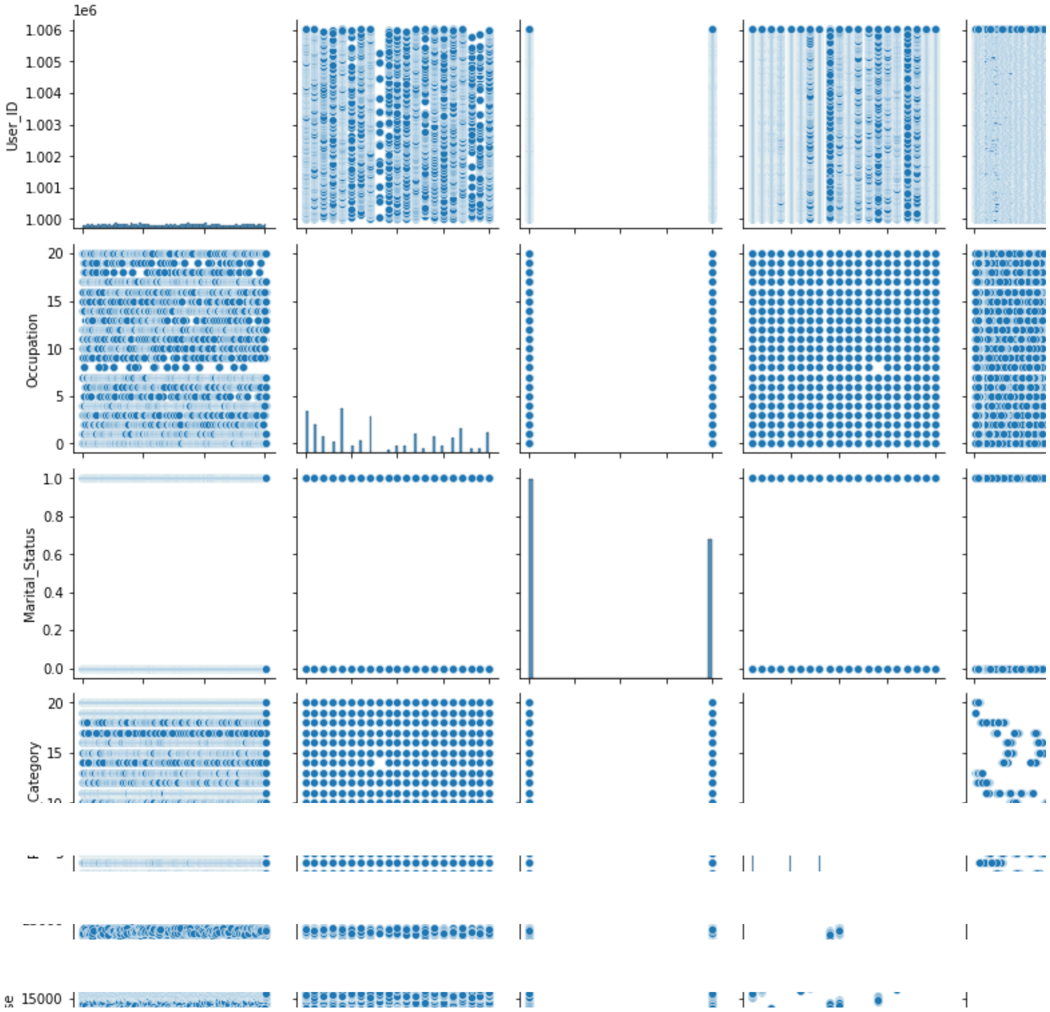
4) More users belong to ***B*** City_Category

5) Most of the users are ***SINGLE***

Pairplots and Heatmaps

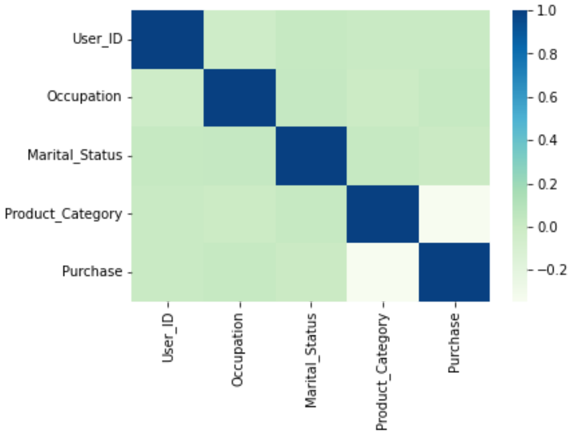
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f0941910fa0>
```



```
sns.heatmap(df.corr(), cmap="GnBu")
```

```
<Axes: >
```

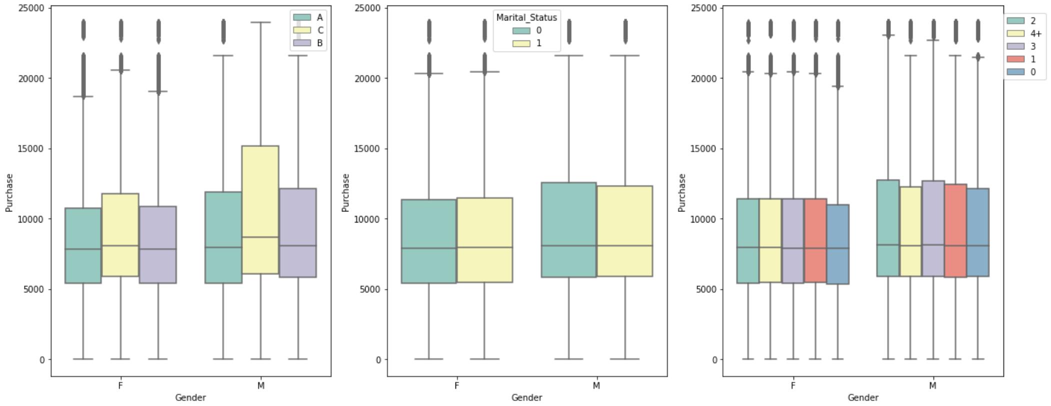


```
fig, axs = plt.subplots(1, 3, figsize=(20, 8))

sns.boxplot(data=df, y='Purchase', x='Gender', hue='City_Category', palette='Set3', ax=axs[0])

sns.boxplot(data=df, y='Purchase', x='Gender', hue='Marital_Status', palette='Set3', ax=axs[1])
sns.boxplot(data=df, y='Purchase', x='Gender', hue='Stay_In_Current_City_Years', palette='Set3', ax=axs[2])

axs[0].legend(loc=("upper right"))
axs[2].legend(loc=(1,0.8))
plt.show()
```



Central Limit Theorem & 95% Confidence intervals

One user buying more than once so group by user_id and Gender/marital status/Age to get purchase sum of each individual accordingly.

Performing CLT & Confidence intervals based on ***Gender***

```
ind_df = df.groupby(['User_ID', 'Gender'])[['Purchase']].sum()
ind_df = ind_df.reset_index()
ind_df
```

	User_ID	Gender	Purchase
0	1000001	F	334093
1	1000002	M	810472
2	1000003	M	341635
3	1000004	M	206468
4	1000005	M	821001
...	...	...	...
5886	1006036	F	4116058
5887	1006037	F	1119538
5888	1006038	F	90034
5889	1006039	F	590319
5890	1006040	M	1653299

5891 rows × 3 columns

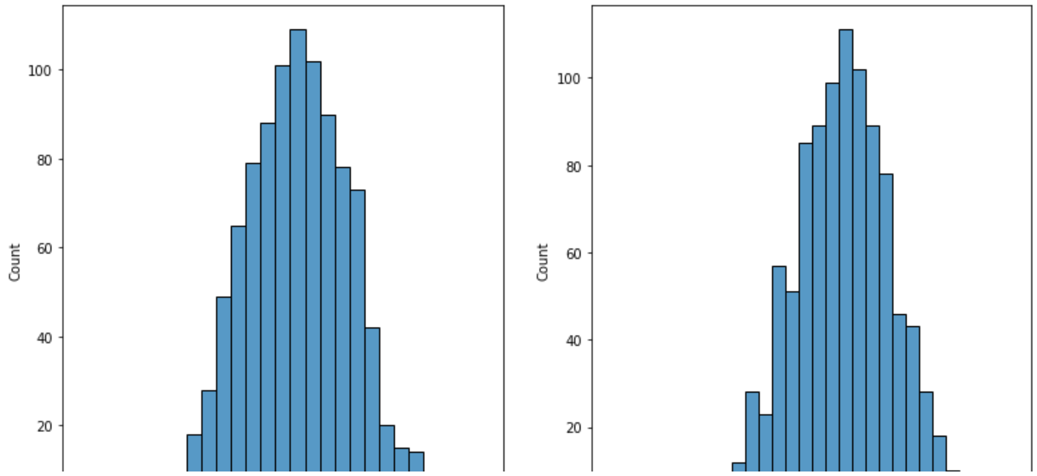
```
male = ind_df[ind_df["Gender"]=="M"]
female = ind_df[ind_df["Gender"]=="F"]
```

```
male_samples = 3000
male_sample_means = []
female_samples = 1500
female_sample_means = []
```

```
for person in range(1000):
  male_means = male.sample(male_samples)['Purchase'].mean()
  female_means = female.sample(female_samples)["Purchase"].mean()
  male_sample_means.append(male_means)
  female_sample_means.append(female_means)
```

```
fig,ax = plt.subplots(1,2,figsize=(13,7))
sns.histplot(male_sample_means , ax= ax[0])
sns.histplot(female_sample_means, ax= ax[1])
```

```
<Axes: ylabel='Count'>
```



```
print("In Population mean, Average amount spent by male users is: {:.2f}".format(np.mean(male_sample_means)))
print("In Population mean, Average amount spent by female users is: {:.2f}".format(np.mean(female_sample_means)))

print("\nMale - Sample mean: {:.2f}, Sample std: {:.2f}".format(male['Purchase'].mean(), male['Purchase'].std()))
print("Female - Sample mean: {:.2f}, Sample std: {:.2f}".format(female['Purchase'].mean(), female['Purchase'].std()))
```

```
In Population mean, Average amount spent by male users is: 925326.82
In Population mean, Average amount spent by female users is: 711982.97

Male - Sample mean: 925344.40, Sample std: 985830.10
Female - Sample mean: 712024.39, Sample std: 807370.73
```

```
#Confidence Levels using Percentile

Male_confidence_level = np.percentile(male_sample_means,[2.5,97.5])
Female_confidence_level = np.percentile(female_sample_means,[2.5,97.5])
print("Male 95% Confidence level: ", Male_confidence_level)
print("Female 95% Confidence level: ", Female_confidence_level)
```

```
Male 95% Confidence level:  [906894.20304167 944148.43895   ]
Female 95% Confidence level:  [698163.61276667 724392.55853333]
```

Insights:

Using the ***Central Limit Theorem*** for the population:

1) Average amount spent by male users is - **925344.40**

2) Average amount spent by female users is - **712024.39**

Using ***Confidence level*** by Percentile about population that, 95% of the times:

3) Average amount spent by male users is between - **(905607.13 - 943847.44)**

4) Average amount spent by female users is between - **(697659.98 - 724021.26)**

Performing CLT & Confidence intervals based on ***Marital Status***

```
mar_df = df.groupby(['User_ID', 'Marital_Status'])[['Purchase']].sum()
mar_df = mar_df.reset_index()
mar_df
```

	User_ID	Marital_Status	Purchase
0	1000001	0	334093
1	1000002	0	810472
3	1000004	1	206468

```
single = mar_df[mar_df["Marital_Status"]==0]
married = mar_df[mar_df["Marital_Status"]==1]
```

```
single_samples = 3000
single_sample_means = []
married_samples = 2000
married_sample_means = []


for person in range(2000):
   single_means = single.sample(single_samples)['Purchase'].mean()
   married_means = married.sample(married_samples)["Purchase"].mean()
   single_sample_means.append(single_means)
   married_sample_means.append(married_means)
```

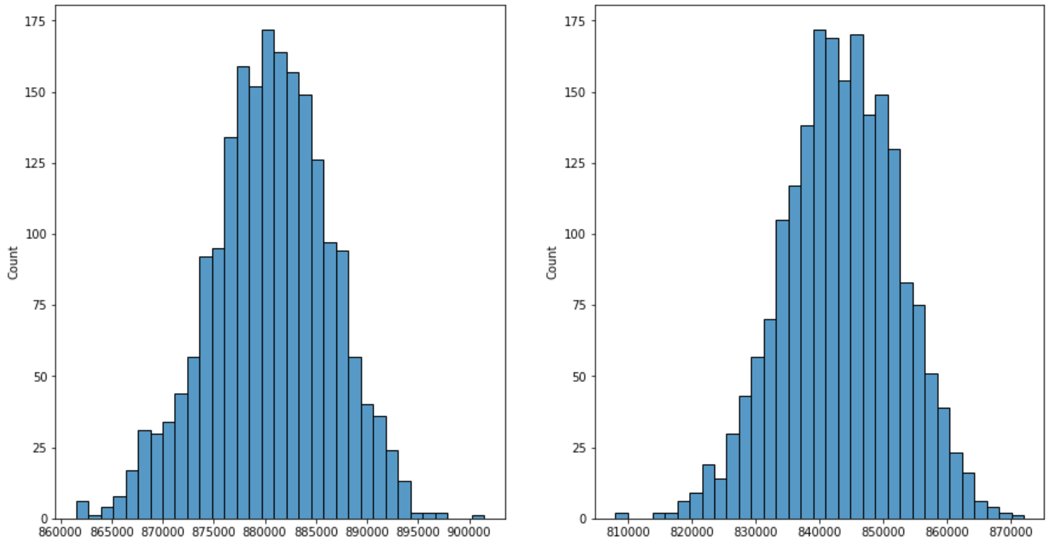
5891 rows × 3 columns

```
print("In Population mean, Average amount spent by unmarried users is: {:.2f}".format(np.mean(single_sample_means)))
print("In Population mean, Average amount spent by married users is: {:.2f}".format(np.mean(married_sample_means)))
```

```
In Population mean, Average amount spent by unmarried users is: 880499.73
In Population mean, Average amount spent by married users is: 843402.14
```

```
fig,ax = plt.subplots(1,2,figsize=(15,8))
sns.histplot(single_sample_means , ax= ax[0])
sns.histplot(married_sample_means, ax= ax[1])
```

```
<Axes: ylabel='Count'>
```



```
#Confidence Levels using Zscore

Error_margin_single = 1.96*single["Purchase"].std()/np.sqrt(len(single))
single_mean = single["Purchase"].mean()

single_lower_limit =  single_mean - Error_margin_single
single_upper_limit =  single_mean + Error_margin_single



Error_margin_married = 1.96*married["Purchase"].std()/np.sqrt(len(married))
married_mean = married["Purchase"].mean()

married_lower_limit = married_mean - Error_margin_married
married_upper_limit = married_mean + Error_margin_married

print("Single confidence interval of means: ({:.2f}, {:.2f})".format(single_lower_limit, single_upper_limit))
print("Married confidence interval of means: ({:.2f}, {:.2f})".format(married_lower_limit, married_upper_limit))
```

```
Single confidence interval of means: (848741.18, 912410.38)
Married confidence interval of means: (806668.83, 880384.76)
```

Insights:

Using the ***Central Limit Theorem*** for the population:

1) Average amount spent by unmarried users is - **880556.48**

2) Average amount spent by married users is - **843025.97**

Using ***Confidence level*** by Percentile about population that, 95% of the times:

3) Average amount spent by unmarried users is between - **(848741.18, 912410.38)**

4) Average amount spent by married users is between - **(806668.83, 880384.76)**

Performing CLT & Confidence intervals based on ***Age***

```
age_df = df.groupby(['User_ID', 'Age'])[['Purchase']].sum()
age_df = age_df.reset_index()
age_df
```

	User_ID	Age	Purchase
0	1000001	0-17	334093
1	1000002	55+	810472
2	1000003	26-35	341635
3	1000004	46-50	206468
4	1000005	26-35	821001
...	...	...	...
5886	1006036	26-35	4116058
5887	1006037	46-50	1119538
5888	1006038	55+	90034
5889	1006039	46-50	590319
5890	1006040	26-35	1653299

5891 rows × 3 columns

```
age_samples = 1000
age_sample_means = {}

age_interval = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55','55+']

for age in age_interval:
  age_sample_means[age] = []

for age in age_interval:
  for x in range(1000):
    mean = age_df[age_df['Age']==age].sample(age_samples,replace=True)['Purchase'].mean()
    age_sample_means[age].append(mean)


for age in age_interval:
  Error_margin_age = []
  age_mean = 0

  new_df = age_df[age_df["Age"]==age]
  Error_margin_age= 1.96*new_df["Purchase"].std()/np.sqrt(len(new_df))
  age_mean = new_df["Purchase"].mean()

  age_lower_limit =  age_mean - Error_margin_age
  age_upper_limit =  age_mean + Error_margin_age

  print("\n Average amount spent by ", age , "is : {:.2f}".format(np.mean(age_sample_means[age])))
  print("95% of the times: Average amount spent by", age , "users is between : ({:.2f}, {:.2f})".format(age_lower_limit, age_upper_limit)
```

```
 Average amount spent by  0-17 is : 619707.22
95% of the times: Average amount spent by 0-17 users is between : (527662.46, 710073.17)

 Average amount spent by  18-25 is : 854195.20
95% of the times: Average amount spent by 18-25 users is between : (801632.78, 908093.46)

 Average amount spent by  26-35 is : 988581.30
95% of the times: Average amount spent by 26-35 users is between : (945034.42, 1034284.21)
```

```
 Average amount spent by  36-45 is : 881150.99
95% of the times: Average amount spent by 36-45 users is between : (823347.80, 935983.62)

 Average amount spent by  46-50 is : 792321.42
95% of the times: Average amount spent by 46-50 users is between : (713505.63, 871591.93)

 Average amount spent by  51-55 is : 762571.55
95% of the times: Average amount spent by 51-55 users is between : (692392.43, 834009.42)

 Average amount spent by  55+ is : 540294.07
95% of the times: Average amount spent by 55+ users is between : (476948.26, 602446.23)
```

Answering questions

- Are women spending more money per transaction than men? Why or Why not? (10 Points)

--> Women are not the most spending individual as 75% of transactions are being done by males.

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

-->No the confidence levels of male and females are not overlapping and Walmart have to improve its sales among females customers.

Insights:

1) 75% of the users are Male and 25% are Female.

2) 60% Single, 40% Married are users.

3) Users of age between 26-35 are the most purchasing users.

4) Overall Median of Purchase is approx 8k and mean is approx 9k and it has outliers

5) Most of the users are males.

6) Most of the users are staying in the current city for One Year.

7) More users belong to B City_Category

8) In Product_Category - 1, 5, 8, 11 are most purchasable product.

==> 95 Percent Confidence Interval by Gender:

Male - **(905607.13 - 943847.44)**

Female - **(697659.98 - 724021.26)**

==> 95 Percent Confidence Interval by Marital_Status:

Unmarried - **(848741.18, 912410.38)**

Married - **(806668.83, 880384.76)**

==> 95 Percent Confidence Interval by Age:

Between 0-17 age CI is : (527662.46, 710073.17)

Between 18-25 age CI is : (801632.78, 908093.46)

Between 26-35 age CI is : (945034.42, 1034284.21)

Between 36-45 age CI is : (823347.80, 935983.62)

Between 46-50 age CI is : (713505.63, 871591.93)

Between 51-55 age CI is : (692392.43, 834009.42)

Above 55+ age CI is : (476948.26, 602446.23)

Recommendations:

1) As most of the users are male , company should focus more on getting female users to buy more on black friday and retaining male customers as is also important.

2) In Product_Category - 1, 5, 8, 11 are most purchasable product. So most of the users are most likely interested in these products so more stock of these products in walmart is recommended. However we should focus more on purchase of least no.of products bought by customers.

3) Users who are unmarried are most purchasing compared to Married , So we have focus on married customers like offering kitchen/house hold gifts to attract them.

4) Users of age group who buy frequently is around 26-35 years. So we have to focus on other age groups to have high sales in company.

✓ 2s completed at 09:32 ● ✕

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.