```
# What good looks like?

# Import the dataset and do usual data analysis steps like checking the structure & character
# Detect Null values & Outliers (using boxplot, “describe” method by checking the difference
# Do some data exploration steps like:
# Tracking the amount spent per transaction of all the 50 million female customers, and all t
# Inference after computing the average female and male expenses.
# Use the sample average to find out an interval within which the population average will lie
# Use the Central limit theorem to compute the interval. Change the sample size to observe th
# The interval that you calculated is called Confidence Interval. The width of the interval i
# Conclude the results and check if the confidence intervals of average male and female spend
# Perform the same activity for Married vs Unmarried and Age
# For Age, you can try bins based on life stages: 0-17, 18-25, 26-35, 36-50, 51+ years.
# Give recommendations and action items to Walmart.

# Evaluation Criteria

# Defining Problem Statement and Analyzing basic metrics (10 Points)
# Observations on shape of data, data types of all the attributes, conversion of categorical
# Non-Graphical Analysis: Value counts and unique attributes
# Visual Analysis - Univariate & Bivariate
# For continuous variable(s): Distplot, countplot, histogram for univariate analysis
# For categorical variable(s): Boxplot
# For correlation: Heatmaps, Pairplots
# Missing Value & Outlier Detection (10 Points)
# Business Insights based on Non- Graphical and Visual Analysis (10 Points)
# Comments on the range of attributes
# Comments on the distribution of the variables and relationship between them
# Comments for each univariate and bivariate plot
# Answering questions (50 Points)
# Are women spending more money per transaction than men? Why or Why not? (10 Points)
# Confidence intervals and distribution of the mean of the expenses by female and male custom
# Are confidence intervals of average male and female spending overlapping? How can Walmart l
# Results when the same activity is performed for Married vs Unmarried (10 Points)
# Results when the same activity is performed for Age (10 Points)
# Final Insights (10 Points) - Illustrate the insights based on exploration and CLT
# Comments on the distribution of the variables and relationship between them
# Comments for each univariate and bivariate plots
# Comments on different variables when generalizing it for Population
# Recommendations (10 Points)
# Actionable items for business. No technical jargon. No complications. Simple action items t
```

```
!gdown "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walma
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/wa
To: /content/walmart_data.csv?1641285094
100% 23.0M/23.0M [00:00<00:00, 63.1MB/s]
```

Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset.

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

```
walmart_df = pd.read_csv("/content/walmart_data.csv?1641285094")
```

```
walmart_df.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Ye
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	

```
print("Number of Rows:",walmart_df.shape[0],"Number of Columns: ",walmart_df.shape[1])
```

```
Number of Rows: 550068 Number of Columns:  10
```

```
walmart_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
```

```
walmart_df.describe()
```

```
walmart_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

```
walmart_df.columns
```

```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

Detect Null values & Outliers (using boxplot, “describe” method by checking the difference between mean and median, isnull etc.)

```
walmart_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
```

As we can see from above query no null values are present.

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

```
3631
```

```
cols = ['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
```

```
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase']
for i in cols:
  print(i,":",walmart_df[i].nunique())
```

```
User_ID : 5891
Product_ID : 3631
Gender : 2
Age : 7
Occupation : 21
City_Category : 3
Stay_In_Current_City_Years : 5
Marital_Status : 2
Product_Category : 20
Purchase : 18105
```

From above total number of Unique values from each columns, we can define that

Categorical Columns are:

1. Gender
2. Age
3. Occupation
4. City_Category
5. Stay_In_Current_City_Years
6. Marital_Status
7. Product_Category

Genral Unique Values from each Columns:

Quick Age-Group Spending Analysis:

```
age_dist = walmart_df['Age'].unique()
age_dist
```

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

```
age_counts =walmart_df['Age'].value_counts()
age_counts
```

```
26-35    219587
36-45    110013
18-25     99660
46-50     45701
51-55     38501
```

```
55+      21504
0-17     15102
Name: Age, dtype: int64
```

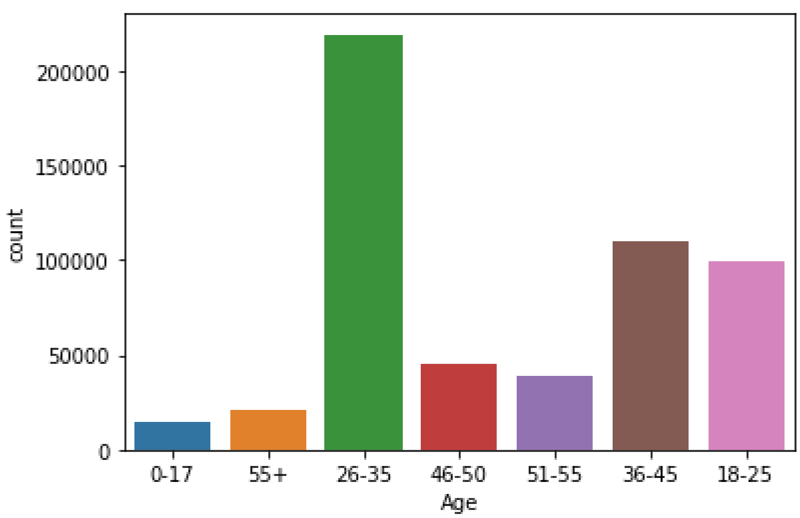
```
# z=[]
for i in age_counts:
    print(np.round(i/len(walmart_df['Age'])*100,2))
```

```
39.92
20.0
18.12
8.31
7.0
3.91
2.75
```

1. **26-35** -----> 219587 -----> 39.92
2. **36-45** -----> 110013 -----> 20.0
3. **18-25** -----> 99660 -----> 18.12
4. **46-50** -----> 45701 -----> 8.31
5. **51-55** -----> 38501 -----> 7.0
6. **55+** --------> 21504 -----> 3.91
7. **0-17** -------> 15102 -----> 2.75

```
sns.countplot(x=walmart_df['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95a4d76c90>
```



As we can see from the age-group distribution, the most enthusiastic age groups are between 26 to 45 aged peoples. We can see around 60% are from this age groups.

They have frequent amount of purchases and active buying can be seen from this age persons.

On the other hand, teenagers and the persons from the 55 years and above might be seen less active buyers, as we can see that there purchases might be done from the above group ages or there might be less population present at this age-groups.

Which Gender is Dominated?

```
gender_counts =walmart_df['Gender'].value_counts()
gender_counts
```

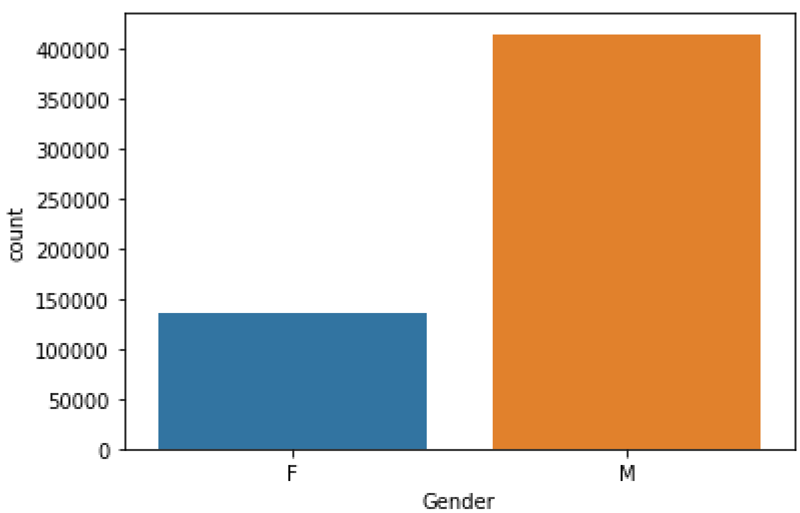
```
M    414259
F    135809
Name: Gender, dtype: int64
```

```
# z=[]
for i in gender_counts:

    print(np.round(i/len(walmart_df['Gender'])*100,2))
```

```
75.31
24.69
```

```
sns.countplot(x=walmart_df['Gender'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95a4d30350>
```



We can see that registered users Gender percentage in which Male is around 75% and female is 25%.

We can infer that the most product purchased is by Male. So from it, we can infer that more number of Offers, Days should be celebrated in which Female Product purchase should be encourage.

Some General Analysis:

```
cols = ['Occupation','City_Category','Stay_In_Current_City_Years','Marital_Status','Product_C
occ_counts =walmart_df['City_Category'].value_counts()
occ_counts
for i in occ_counts:
    print(np.round(i/len(walmart_df['City_Category'])*100,2))
```

```
42.03
31.12
26.85
```

```
dist=[]
for i in cols:
  dist.append(walmart_df[i].unique())
dist
```

```
[array([10, 16, 15,  7, 20,  9,  1, 12, 17,  0,  3,  4, 11,  8, 19,  2, 18,
         5, 14, 13,  6]),
 array(['A', 'C', 'B'], dtype=object),
 array(['2', '4+', '3', '1', '0'], dtype=object),
 array([0, 1]),
 array([ 3,  1, 12,  8,  5,  4,  2,  6, 14, 11, 13, 15,  7, 16, 18, 10, 17,
         9, 20, 19])]
```

Rest of Categorical Columns Insights:

```
cols = ['Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Produ
walmart_df[cols].melt().groupby(['variable', 'value'])[['value']].count()/len(walmart_df)
```

variable	value	value
City_Category	**A**	0.268549
	B	0.420263
	C	0.311189
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
	19	0.015382
	20	0.061014
Product_Category	**1**	0.255201
	2	0.043384
	3	0.036746

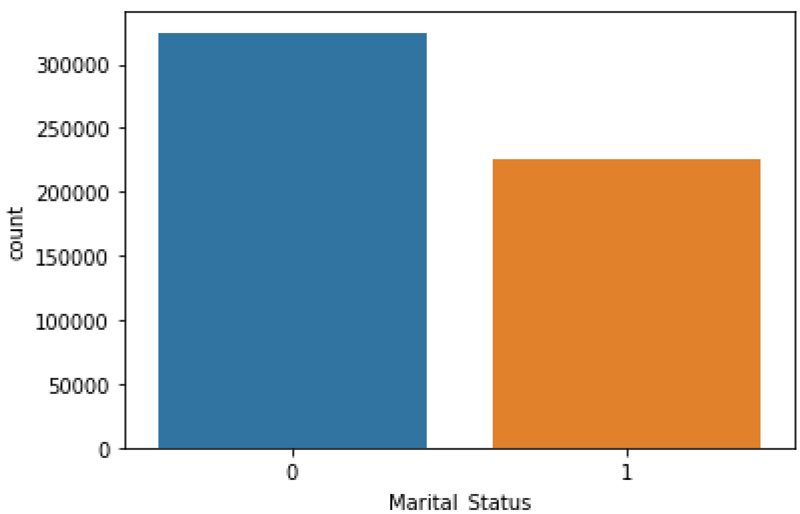
4	0.021366
5	0.274390
6	0.037206
7	0.006765
8	0.207111
9	0.000745
10	0.009317
11	0.044153
12	0.007175
13	0.010088
14	0.002769

Uni-Variate Analysis:

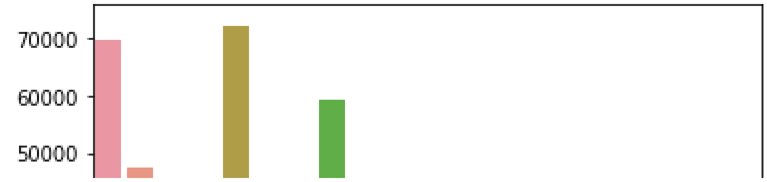
16	0.017867

```
sns.countplot(x=walmart_df['Marital_Status'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95a4cf79d0>
```

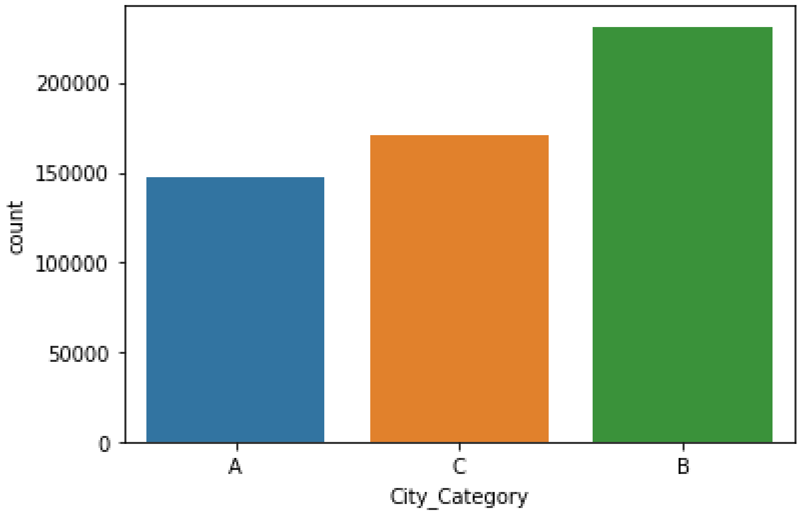



```
sns.countplot(x=walmart_df['Occupation'])
plt.show()
```



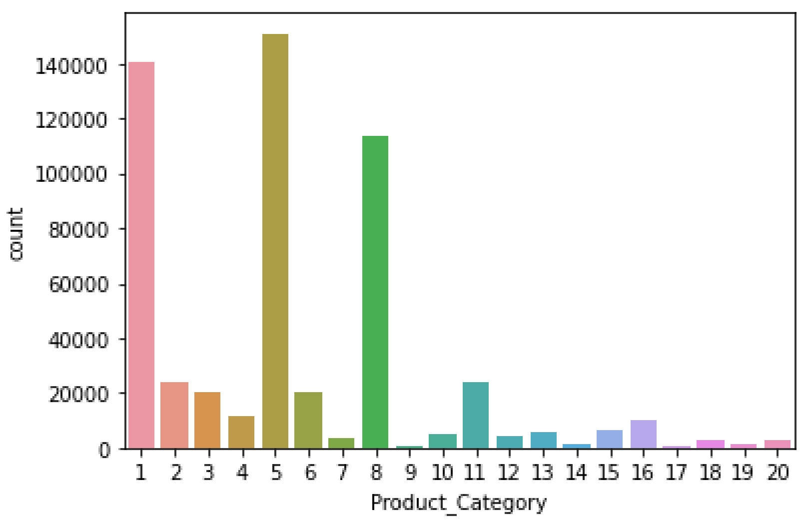

```
sns.countplot(x=walmart_df['City_Category'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95a52011d0>
```




```
sns.countplot(walmart_df['Product_Category'])
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass tl
  FutureWarning
```



From above output we can observe the insights:

1. In City_Category column, we have three unique features named as A,B,C. In which Category-B is the most active in Purchasing and generating revenue and following with Category-C then Category-A.

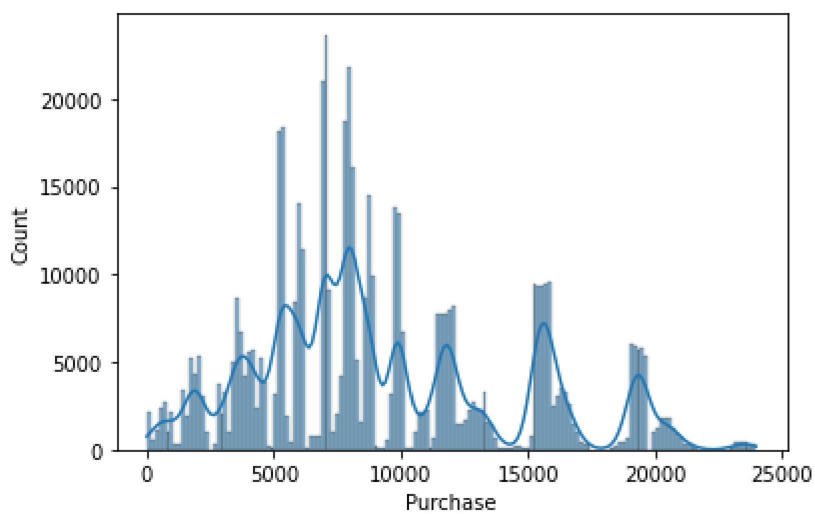
> Category-A performance is below PAR, so need to introduce something new offers,products or should be working very closely towards the reviews and services for better enhancements.

2. Around 60% customers are single and 40% are couple customers. It is a good ratio of the column, as we can see they are almost equally balanced.
3. There are around 20-product Categories available in the stores. Products from Category 5,1 and 8 are most purchased.
4. The customers mostly are belonging from the same region of the store most of the customer are being living over there for around 1+ years are more.

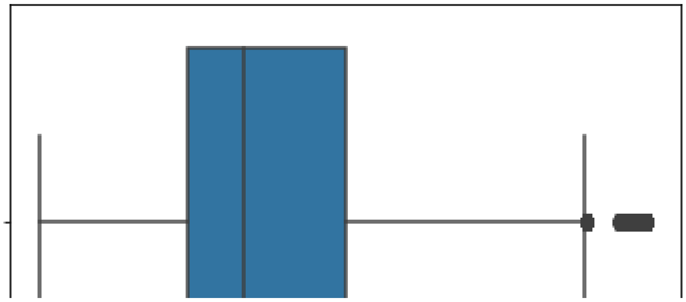
```
walmart_df.columns
```

```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

```
sns.histplot(data=walmart_df,x='Purchase',kde=True)
plt.show()
```

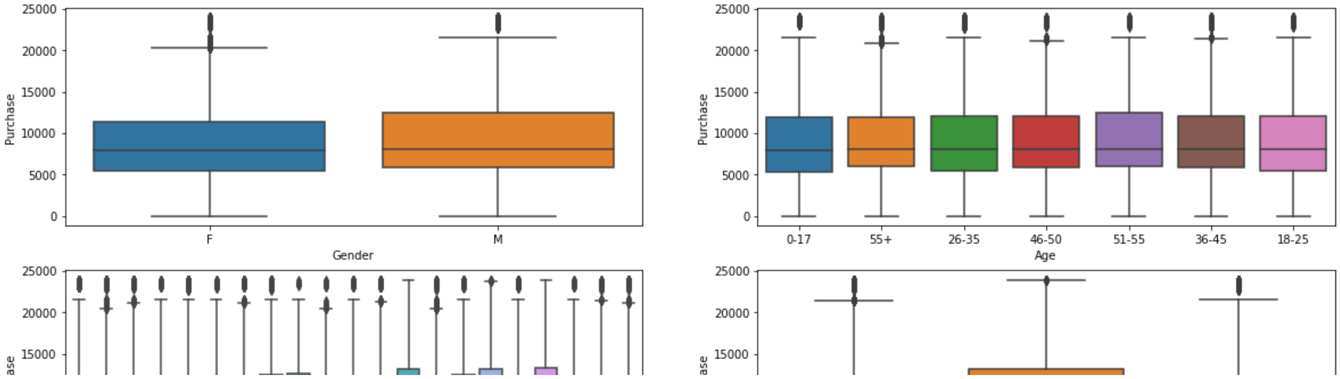



```
sns.boxplot(data=walmart_df,x='Purchase')
plt.show()
```



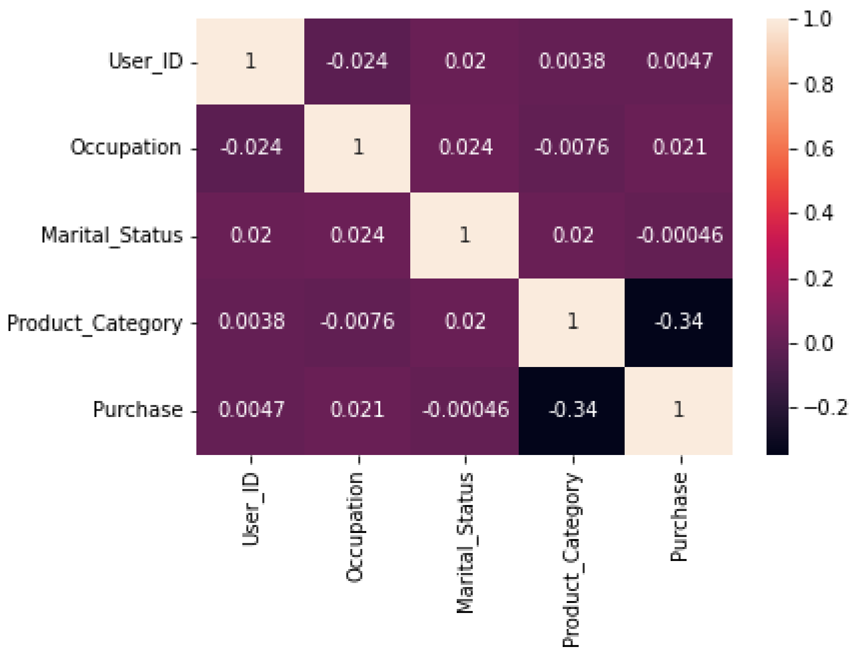
Bi-Variate Analysis

```
attrs = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marit
fig, axs = plt.subplots(nrows=3, ncols=2,figsize=(20,12))

sns.boxplot(data=walmart_df, y='Purchase', x='Gender',ax=axs[0,0])
sns.boxplot(data=walmart_df, y='Purchase', x='Age',ax=axs[0,1])
sns.boxplot(data=walmart_df, y='Purchase', x='Occupation',ax=axs[1,0])
sns.boxplot(data=walmart_df, y='Purchase', x='City_Category',ax=axs[1,1])
sns.boxplot(data=walmart_df, y='Purchase', x='Marital_Status',ax=axs[2,0])
sns.boxplot(data=walmart_df, y='Purchase', x='Stay_In_Current_City_Years',ax=axs[2,1])

plt.show()
```



From above Bi-Variate Analysis w.r.t to Purchase column, we can clearly see that every column has some amount of Outliers present in that.


```
sns.heatmap(walmart_df.corr(),annot=True)
plt.show()
```



We can see that there is only self-correlation between columns are observed. Other than that, no columns are correlated with other columns.

Questions:

1. Are women spending more money per transaction than men? Why or Why not?
2. Confidence intervals and distribution of the mean of the expenses by female and male customers.
3. Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?

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

MEN Spending VS WOMEN Spending

```
spending_df = walmart_df.groupby(['User_ID','Gender'])[['Purchase']].sum()
spending_df = spending_df.reset_index()
```

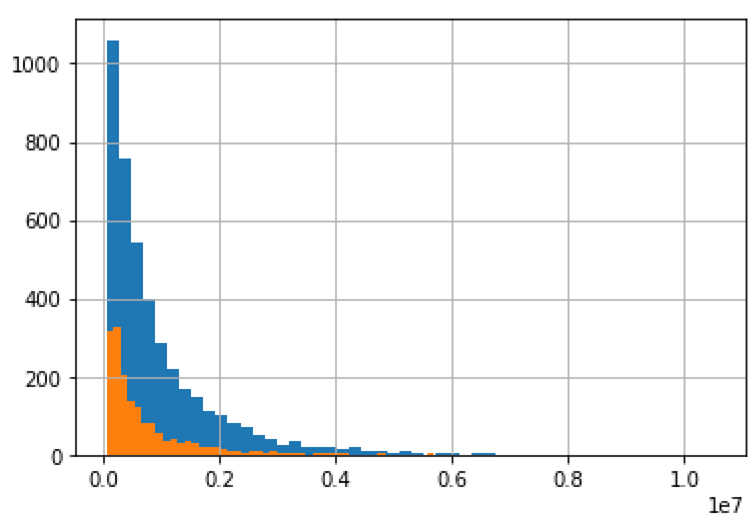
```
spending_df.head()
```

	User_ID	Gender	Purchase
0	1000001	F	334093
1	1000002	M	810472
2	1000003	M	341635
3	1000004	M	206468
4	1000005	M	821001

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

```
M    4225
F    1666
Name: Gender, dtype: int64
```

```
spending_df[spending_df['Gender']=='M']['Purchase'].hist(bins=50)
spending_df[spending_df['Gender']=='F']['Purchase'].hist(bins=50)
plt.show()
```




```
spend_male_avg = spending_df[spending_df['Gender']=='M']['Purchase'].mean()
```

```
spend_woman_avg = spending_df[spending_df['Gender']=='F']['Purchase'].mean()
print("Average Spending by Males: ",spend_male_avg,"Average Spending by Women: ",spend_woman_
```

```
Average Spending by Males:  925344.4023668639 Average Spending by Women:  712024.394957
```

As we can see that Male Average Spending is approximately 58% and Female average Spending is 42%.

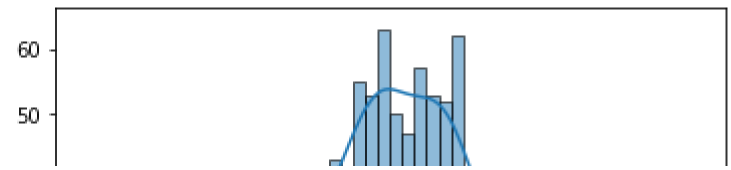
Partioning the Data into two groups: Male and Female

```
male_walmart_df = spending_df[spending_df['Gender']=='M']
female_walmart_df = spending_df[spending_df['Gender']=='F']
```

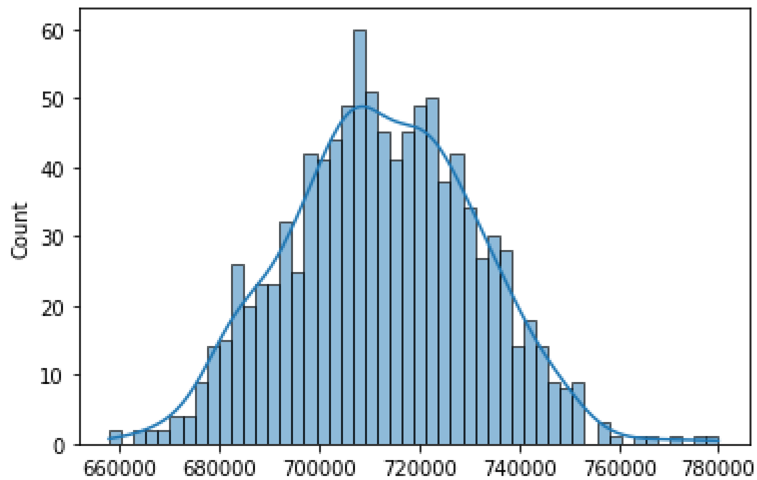
```
genders = ["M", "F"]

male_sample_size = 4000
female_sample_size = 2000
iterations = 1000
male_means = []
female_means = []

for i in range(iterations):
    male_mean = male_walmart_df.sample(male_sample_size, replace=True)['Purchase'].mean()
    female_mean = female_walmart_df.sample(female_sample_size, replace=True)['Purchase'].mean

    male_means.append(male_mean)
    female_means.append(female_mean)


print("Population mean - sample means of amount spend for Male: ",np.mean(male_means))
print("Population mean - sample means of amount spend for Female: ",np.mean(female_means))
```

```
Population mean - sample means of amount spend for Male:  924759.6809645
Population mean - sample means of amount spend for Female:  712569.8390939999
```

```
sns.histplot(male_means, kde=True,bins=50)
plt.show()
```



```
sns.histplot(female_means, kde=True,bins=50)
plt.show()
```



```
print("Male - Sample mean: ",(male_walmart_df['Purchase'].mean()))
print("Female - Sample mean: ",(female_walmart_df['Purchase'].mean()))
```

```
Male - Sample mean:  925344.4023668639
Female - Sample mean:  712024.3949579832
```

```
print("Male - Sample std deviation: ",male_walmart_df['Purchase'].std())
print("Female - Sample std deviation: ",female_walmart_df['Purchase'].std())
```

```
Male - Sample std deviation:  985830.1007953875
Female - Sample std deviation:  807370.7261464577
```

According to Central Limit Theorem for Population we can say that:

1. Average Male Spending is 925344.402
2. Average Female Spending is 712024.39

```
male_margin_of_error_clt = 1.96*male_walmart_df['Purchase'].std()/np.sqrt(len(male_walmart_df
male_sample_mean = male_walmart_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_walmart_df['Purchase'].std()/np.sqrt(len(female_walm
female_sample_mean = female_walmart_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: ",(male_lower_lim, male_upper_lim))
print("Female confidence interval of means: ",(female_lower_lim, female_upper_lim))
```

```
Male confidence interval of means:  (895617.8331736492, 955070.9715600787)
Female confidence interval of means:  (673254.7725364959, 750794.0173794704)
```

Confidence Interval for the Population that 95% of the times:

1. Average amount spend by male customer will lie in between: (895617.83, 955070.97)
2. Average amount spend by female customer will lie in between: (673254.77, 750794.02)

Finding CLT and CI for Married vs Un-Married:

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

```
mar_df.head()
```

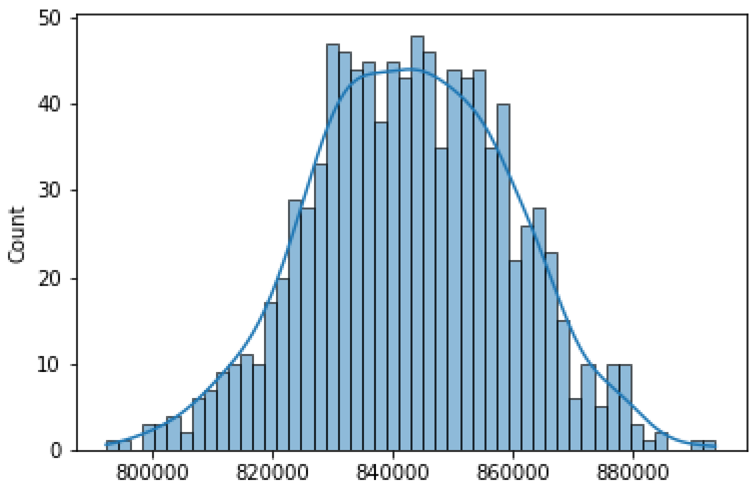
	User_ID	Marital_Status	Purchase
0	1000001	0	334093
1	1000002	0	810472
2	1000003	0	341635
3	1000004	1	206468
4	1000005	1	821001

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

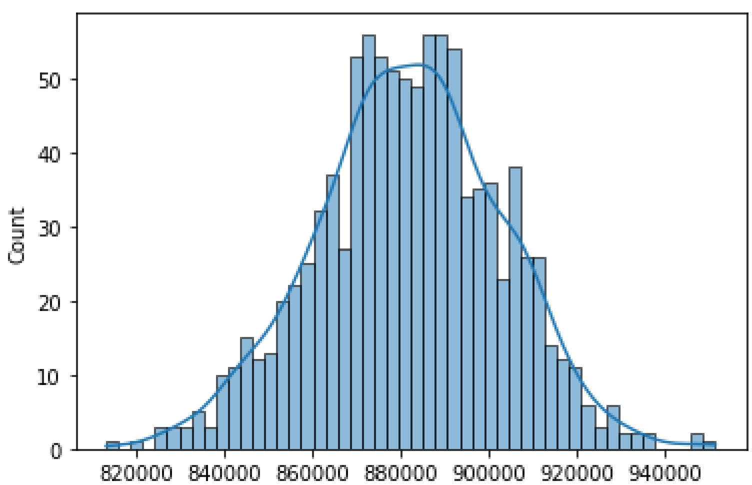
```
0    3417
1    2474
Name: Marital_Status, dtype: int64
```

```
marid_samp_size = 3000
unmarid_sample_size = 2000
iterations = 1000
marid_means = []
unmarid_means = []

for i in range(iterations):
    marrid_mean = mar_df[mar_df['Marital_Status']==1].sample(marid_samp_size, replace=True)['
    unmarrid_mean = mar_df[mar_df['Marital_Status']==0].sample(unmarid_sample_size, replace=T
```

```
    marid_means.append(marrid_mean)
    unmarid_means.append(unmarrid_mean)

sns.histplot(marid_means, kde=True,bins=50)
plt.show()
```




```
sns.histplot(unmarid_means, kde=True,bins=50)
plt.show()
```




```
print("Marrid - Population mean: ",np.mean(marid_means))
print("Unmarrid - Population mean: ",np.mean(unmarid_means))
```

```
Marrid - Population mean:  842984.3003496667
Unmarrid - Population mean:  881780.9986360001
```

```
print("Married - Sample mean:  ",mar_df[mar_df['Marital_Status']==1]['Purchase'].mean())
print("Unmarried - Sample mean: ",mar_df[mar_df['Marital_Status']==0]['Purchase'].mean())
```

```
Married - Sample mean:   843526.7966855295
Unmarried - Sample mean:  880575.7819724905
```

```
print("Married - Sample Standard Deviation:  ",mar_df[mar_df['Marital_Status']==1]['Purchase'
print("Unmarried - Sample Standard Deviation: ",mar_df[mar_df['Marital_Status']==0]['Purchase
```

```
Married - Sample Standard Deviation:   935352.1158252305
Unmarried - Sample Standard Deviation:  949436.2495552396
```

According to Central Limit Theorem for Population we can say that:

1. Average Married Couple Spending is 844411.2102
2. Average Un-Married Couple Spending is 880546.1318

```
for val in ["Married", "Unmarried"]:

    new_val = 1 if val == "Married" else 0

    new_df = mar_df[mar_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(val," CI :- ",(lower_lim, upper_lim))
```

```
Married  CI :-  (806668.8313977643, 880384.7619732948)
Unmarried  CI :-  (848741.1824337273, 912410.3815112537)
```

Confidence Interval:

From above data we can say that, Married Couples Confidence Interval levels are (806668.8313977643, 880384.7619732948).

And Similarly for Singles, the Confidence Interval ranges are (848741.1824337273, 912410.3815112537)

Age Analysis : - Spending Amounts in Different Age groups

```
age_grp = walmart_df.groupby(['User_ID','Age'])[['Purchase']].sum()
age_grp = age_grp.reset_index()
age_grp.head(5)
```

	User_ID	Age	Purchase
0	1000001	0-17	334093
1	1000002	55+	810472
2	1000003	26-35	341635

```
age_grp['Age'].unique()
```

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

```
age_grp['Age'].value_counts()
```

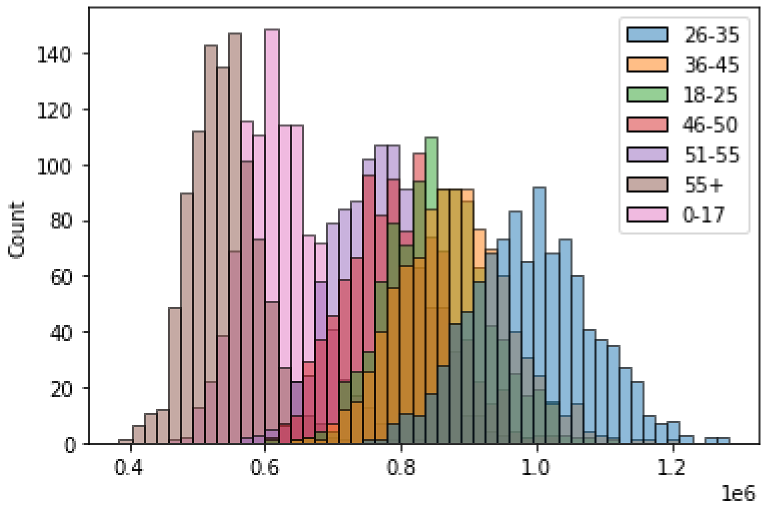
```
26-35    2053
36-45    1167
18-25    1069
46-50     531
51-55     481
55+       372
0-17      218
Name: Age, dtype: int64
```

```
sample_size = 150
repitions = 1000

all_means = {}

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

for age_interval in age_distributions:
    for _ in range(repitions):
        mean = age_grp[age_grp['Age']==age_interval].sample(sample_size, replace=True)['Purch
        all_means[age_interval].append(mean)
```

```
for val in age_distributions:

    new_df = age_grp[age_grp['Age']==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("For age: ",val," CI :- ",(lower_lim, upper_lim))
```

```
For age:  26-35  CI :-  (945034.4236487859, 1034284.2105450766)
For age:  36-45  CI :-  (823347.8021361914, 935983.6186007408)
For age:  18-25  CI :-  (801632.7751885153, 908093.4642876306)
For age:  46-50  CI :-  (713505.6344444095, 871591.9286441028)
For age:  51-55  CI :-  (692392.4251764436, 834009.4209774026)
For age:  55+  CI :-  (476948.2595905849, 602446.2296567269)
For age:  0-17  CI :-  (527662.4567141125, 710073.1671390985)
```

```
sns.histplot(all_means,bins=50)
plt.show()
```



Observations:

1. The Central Limit Theorem for Spending,

   Male Spending 925356

   Female spending 710536.

2. Confidence Intervals:

   Male Spending will lies in between (895917.83, 957070).

   Female Spending will lies in between (673254, 750794).

   Married confidence interval of means: (806968, 880384).

   Unmarried confidence interval of means: (848741.18, 912410.38).

   For age: 26-35 :- (945034.4236487859, 1034284.2105450766)

   For age: 36-45 :- (823347.8021361914, 935983.6186007408)

   For age: 18-25 :- (801632.7751885153, 908093.4642876306)

   For age: 46-50 :- (713505.6344444095, 871591.9286441028)

   For age: 51-55 :- (692392.4251764436, 834009.4209774026)

For age: 55+ :- (476948.2595905849, 602446.2296567269)

For age: 0-17 :- (527662.4567141125, 710073.1671390985)

Recommendations:

1. MALE customers spent more than FEMALE customers, so Company might try customer acquisition by introducing Offers related to Females so that they encouragely purchase product.
2. In the Marital Status, around 60% spent was done by Singles group, so company may try Customer Acquisition of Couples group as the difference between revenue generated by the both groups are similar.
3. In age-group, we can see that the range of years are between 25-45 years, this group acquires almost 60% of whole customers, so in this case instead of customer acquisition, customer retention is important as most of the revenue is generated from this age-group only. So Product of this age-group might be more saleable.
4. From Product Category we can see that Product Category - 1, 8 and 11 is more purchased or liked by customers, so company can introduce new products similar to this categories to increase Revenue and can also provide variations for customers. And might due to this the customer acquisition can also be increased.