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.

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

import os
for dirname, _, filenames in
os.walk('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/
000/001/293/original/walmart_data.csv?1641285094'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

df =
pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/asset
s/000/001/293/original/walmart_data.csv?1641285094")
df.head()
```

```
   User_ID Product_ID Gender   Age  Occupation City_Category  \
0  1000001  P00069042      F  0-17          10             A
1  1000001  P00248942      F  0-17          10             A
2  1000001  P00087842      F  0-17          10             A
3  1000001  P00085442      F  0-17          10             A
4  1000002  P00285442      M   55+          16             C

  Stay_In_Current_City_Years  Marital_Status  Product_Category
Purchase
0                          2               0                 3
8370
1                          2               0                 1
```

```
15200
2                           2               0                12
1422
3                           2               0                12
1057
4                          4+               0                 8
7969
```

```
print(f"Number of rows: {df.shape[0]:,} \nNumber of columns:
{df.shape[1]}")
```

```
Number of rows: 550,068
Number of columns: 10
```

```
df.dtypes
```

```
User_ID                      int64
Product_ID                  object
Gender                      object
Age                         object
Occupation                   int64
City_Category               object
Stay_In_Current_City_Years  object
Marital_Status               int64
Product_Category             int64
Purchase                     int64
dtype: object
```

```
df.memory_usage()
```

```
Index                          128
User_ID                    4400544
Product_ID                 4400544
Gender                     4400544
Age                        4400544
Occupation                 4400544
City_Category              4400544
Stay_In_Current_City_Years 4400544
Marital_Status             4400544
Product_Category           4400544
Purchase                   4400544
dtype: int64
```

```
df.describe()
```

```
            User_ID     Occupation  Marital_Status
Product_Category  \
count  5.500680e+05  550068.000000   550068.000000     550068.000000

mean   1.003029e+06       8.076707        0.409653          5.404270

std    1.727592e+03       6.522660        0.491770          3.936211
```

```
min    1.000001e+06       0.000000        0.000000          1.000000

25%    1.001516e+06       2.000000        0.000000          1.000000

50%    1.003077e+06       7.000000        0.000000          5.000000

75%    1.004478e+06      14.000000        1.000000          8.000000

max    1.006040e+06      20.000000        1.000000         20.000000
```

```
            Purchase
count  550068.000000
mean     9263.968713
std      5023.065394
min        12.000000
25%      5823.000000
50%      8047.000000
75%     12054.000000
max     23961.000000
```

Observations

- There are no missing values in the dataset.
- Purchase amount might have outliers.

```
# 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
```

How many users are there in the dataset?

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

```
5891
```

How many products are there?

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

```
3631
```

Value_counts for the following:

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

```
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)
```

variable	value	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

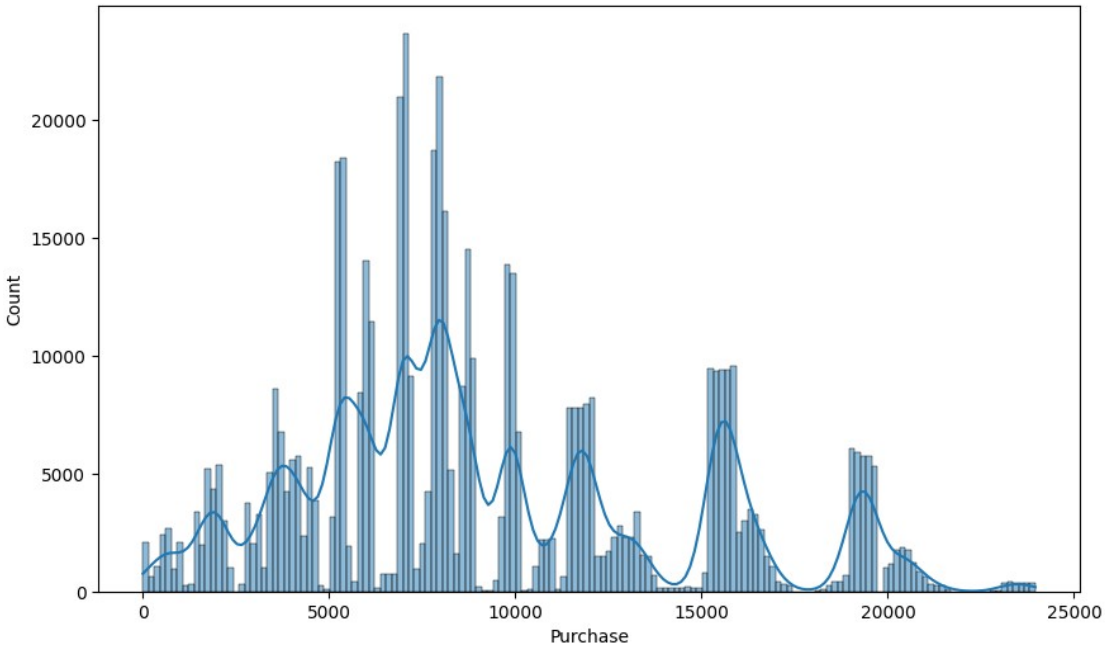
```
                           15     0.022115
                           16     0.046123
                           17     0.072796
                           18     0.012039
                           19     0.015382
                           20     0.061014
Product_Category           1      0.255201
                           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
                           15     0.011435
                           16     0.017867
                           17     0.001051
                           18     0.005681
                           19     0.002914
                           20     0.004636
Stay_In_Current_City_Years 0      0.135252
                           1      0.352358
                           2      0.185137
                           3      0.173224
                           4+     0.154028
```

Observations

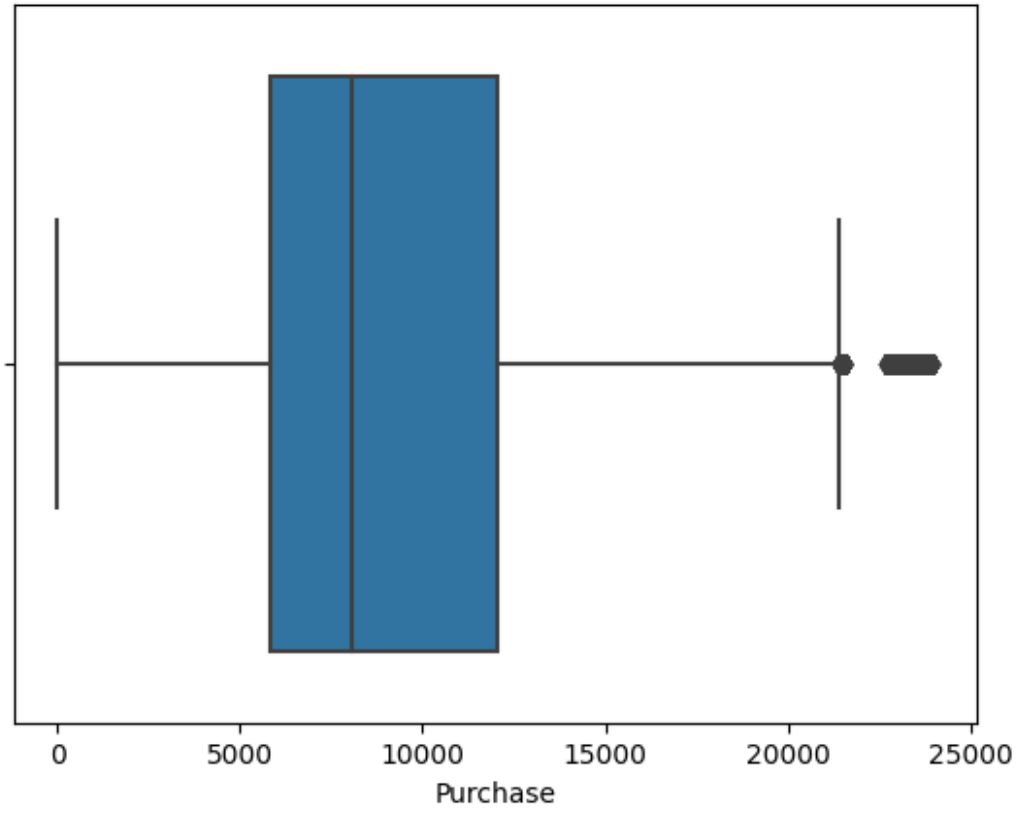
- ~ 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45)
- 75% of the users are Male and 25% are Female
- 60% Single, 40% Married
- 35% Staying in the city from 1 year, 18% from 2 years, 17% from 3 years
- Total of 20 product categories are there
- There are 20 differnent types of occupations in the city

Univariate Analysis

```
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='Purchase', kde=True)
plt.show()
```




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



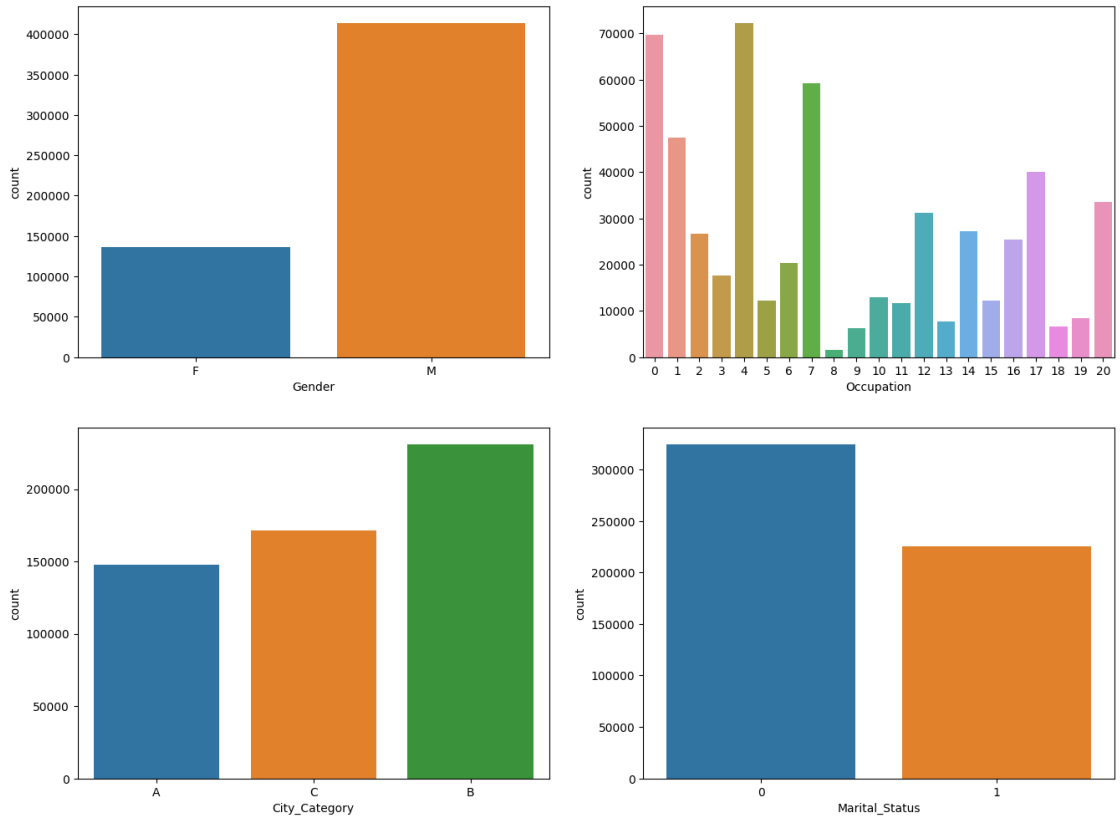
Observation

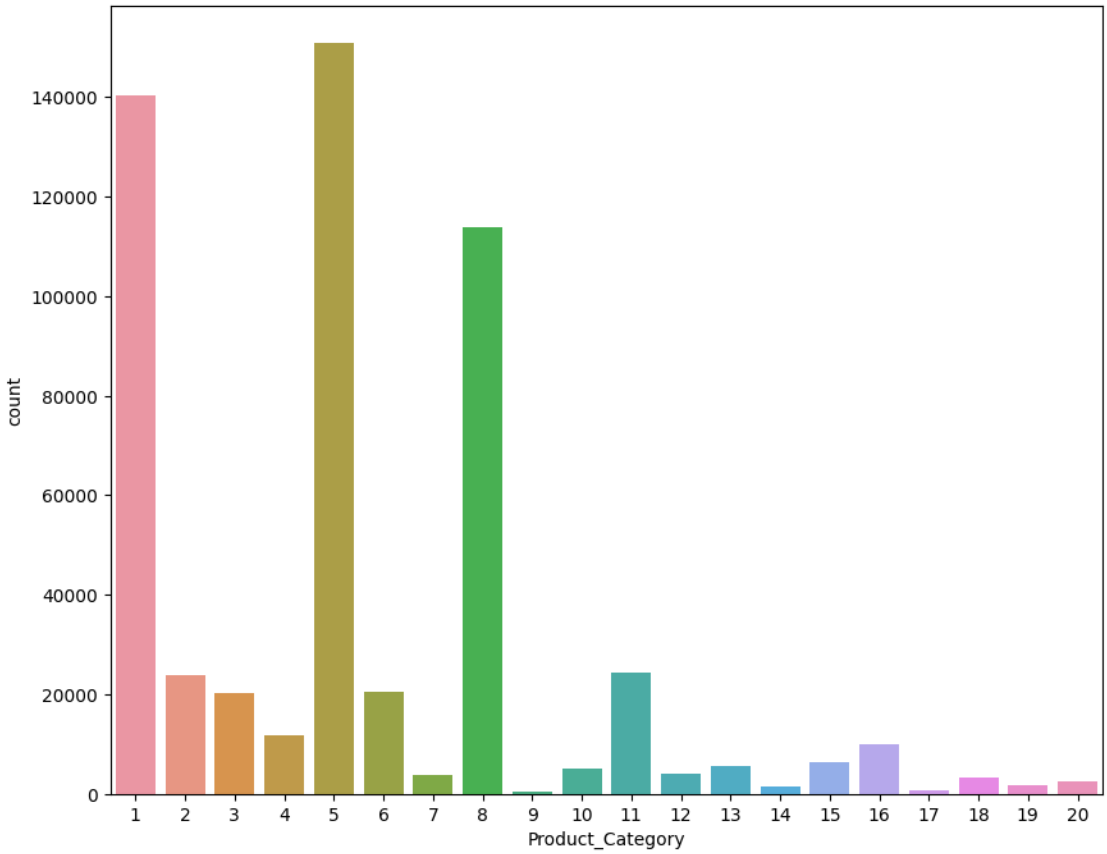
- Purchase is having outliers

Understanding the distribution of data for the categorical variables

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

```
categorical_cols = ['Gender',
'Occupation','City_Category','Marital_Status','Product_Category']

fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))
sns.countplot(data=df, x='Gender', ax=axs[0,0])
sns.countplot(data=df, x='Occupation', ax=axs[0,1])
sns.countplot(data=df, x='City_Category', ax=axs[1,0])
sns.countplot(data=df, x='Marital_Status', ax=axs[1,1])
plt.show()

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



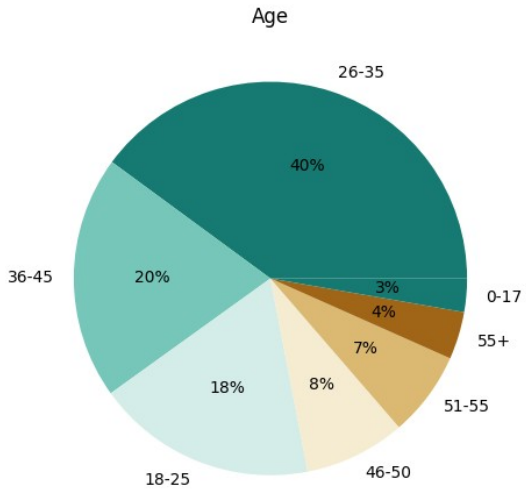


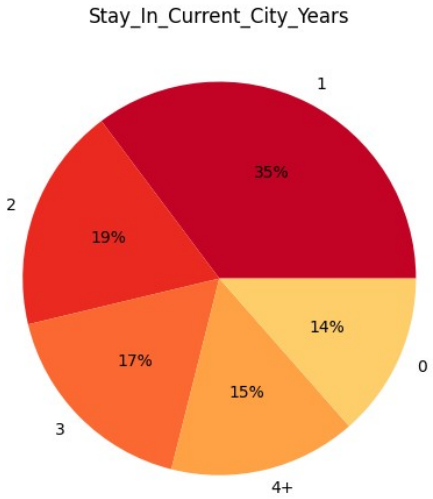
Observations

- 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.

```
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()
```

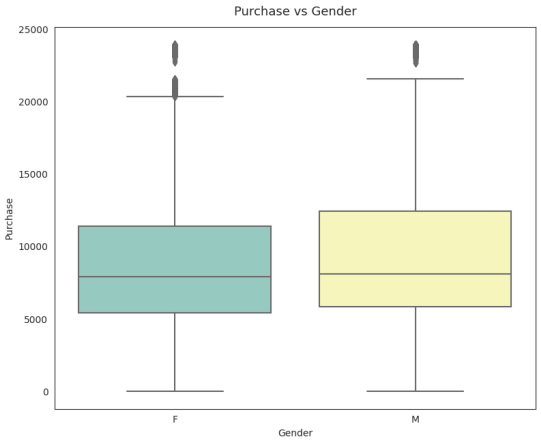


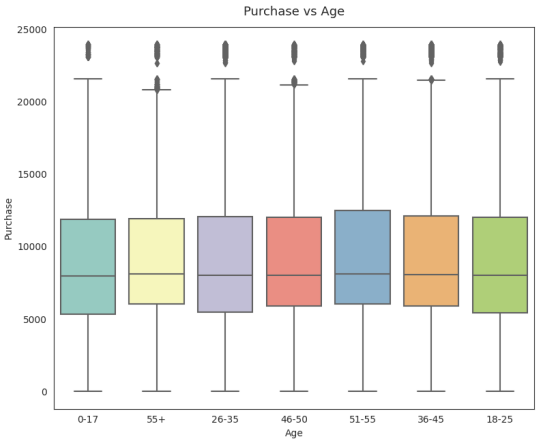


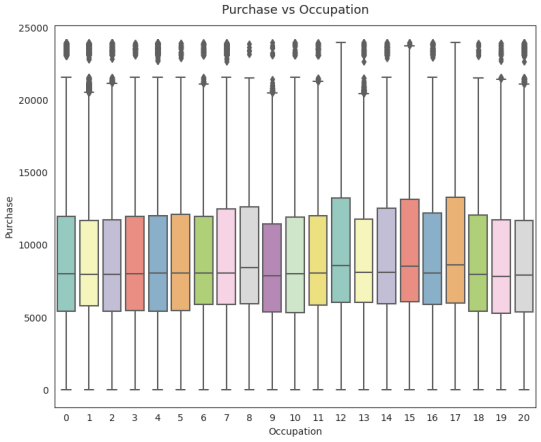
Upper two graphs are self-explanatory.

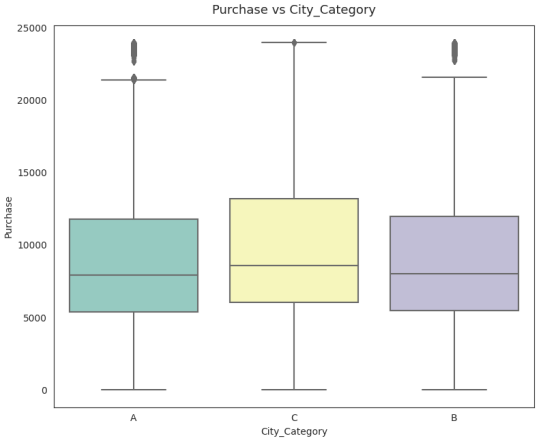
Bi-variate Analysis

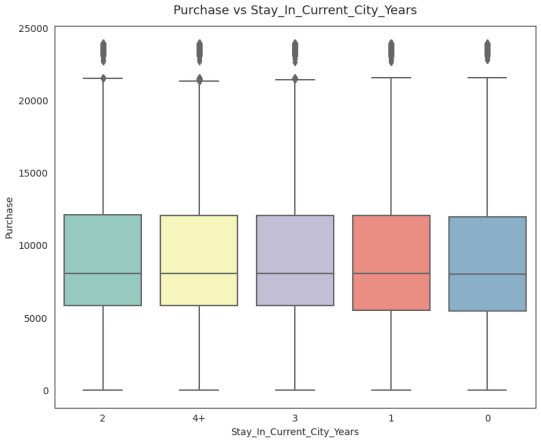
```
attrs = ['Gender', 'Age', 'Occupation', 'City_Category',
'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
sns.set_style("white")

fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(20, 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()

plt.figure(figsize=(10, 8))
sns.boxplot(data=df, y='Purchase', x=attrs[-1], palette='Set3')
plt.show()
```

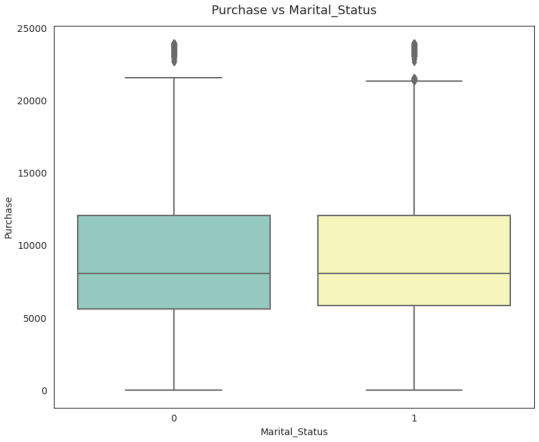

Purchase vs Gender
Purchase
Gender
F
M

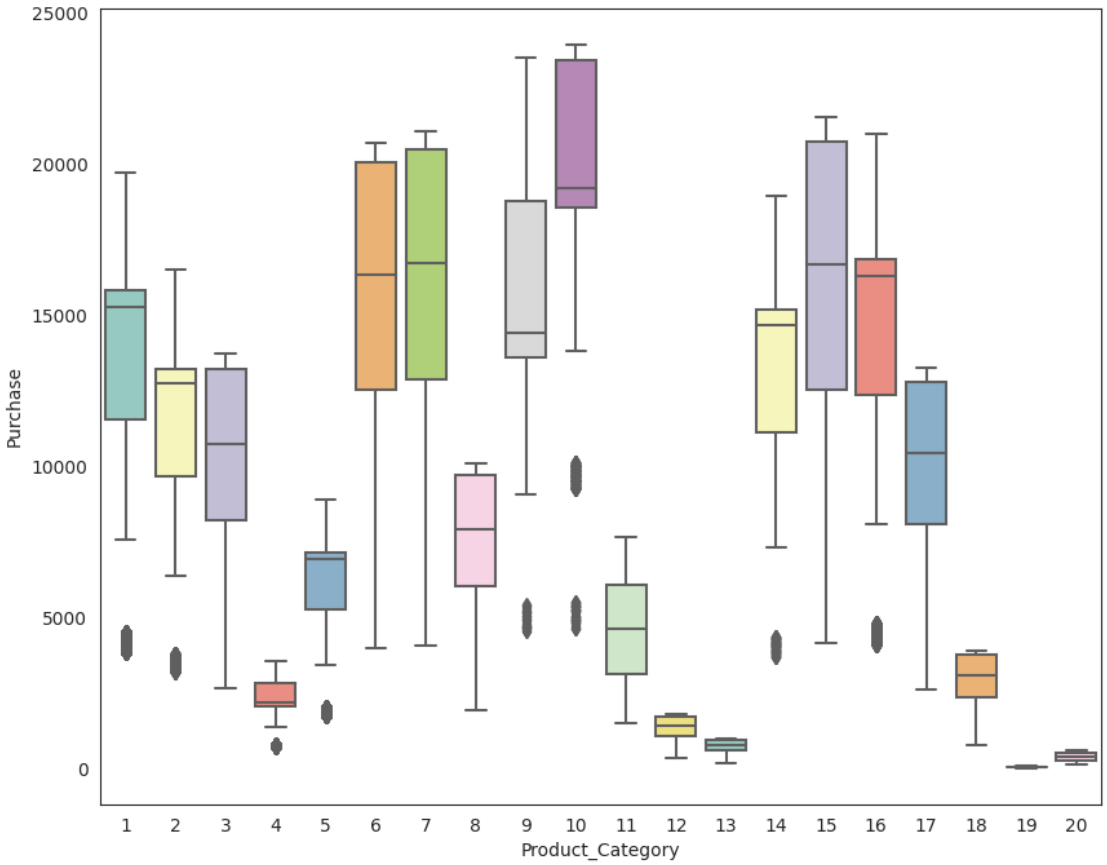

Purchase vs Age
Purchase
Age
0-17
55+
26-35
46-50
51-55
36-45
18-25


Purchase vs Occupation
Purchase
Occupation


Purchase vs City_Category
Purchase
City_Category
A
C
B

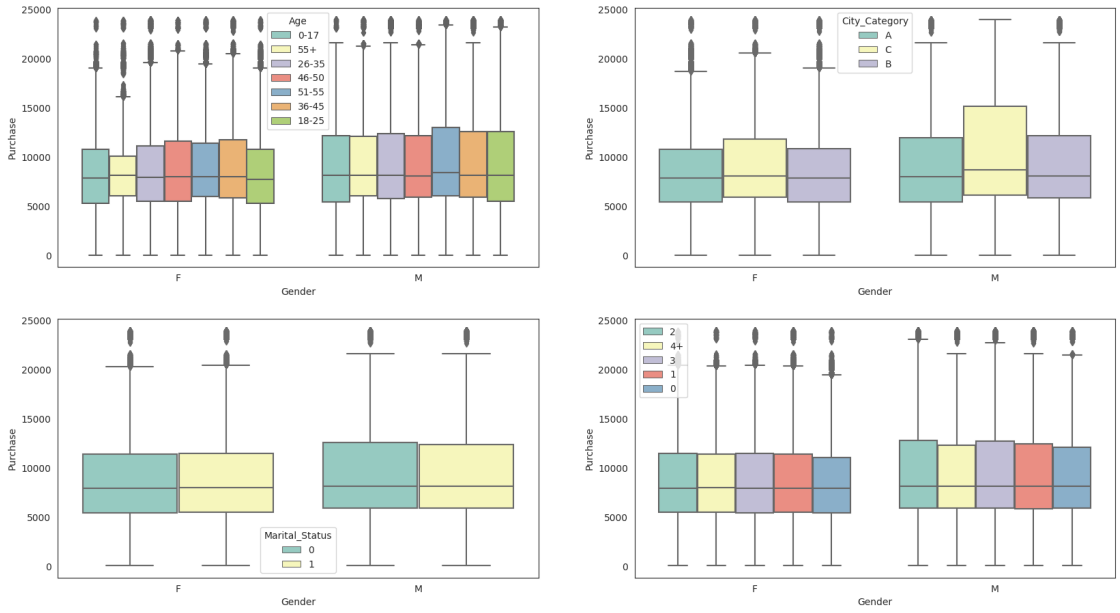

Purchase vs Stay_In_Current_City_Years
Purchase
Stay_In_Current_City_Years
2
4+
3
1
0


Purchase vs Marital_Status
Purchase
Marital_Status



Multivariate Analysis

```
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(20, 6))
fig.subplots_adjust(top=1.5)
sns.boxplot(data=df, y='Purchase', x='Gender', hue='Age',
palette='Set3', ax=axs[0,0])
sns.boxplot(data=df, y='Purchase', x='Gender', hue='City_Category',
palette='Set3', ax=axs[0,1])

sns.boxplot(data=df, y='Purchase', x='Gender', hue='Marital_Status',
palette='Set3', ax=axs[1,0])
sns.boxplot(data=df, y='Purchase', x='Gender',
hue='Stay_In_Current_City_Years', palette='Set3', ax=axs[1,1])
axs[1,1].legend(loc='upper left')

plt.show()
```




```
df.head(10)
```

```
   User_ID Product_ID Gender    Age  Occupation City_Category  \
0  1000001  P00069042      F   0-17          10             A
1  1000001  P00248942      F   0-17          10             A
2  1000001  P00087842      F   0-17          10             A
3  1000001  P00085442      F   0-17          10             A
4  1000002  P00285442      M    55+          16             C
5  1000003  P00193542      M  26-35          15             A
6  1000004  P00184942      M  46-50           7             B
7  1000004  P00346142      M  46-50           7             B
8  1000004   P0097242      M  46-50           7             B
9  1000005  P00274942      M  26-35          20             A

  Stay_In_Current_City_Years  Marital_Status  Product_Category
Purchase
0                          2               0                 3
8370
1                          2               0                 1
15200
2                          2               0                12
1422
3                          2               0                12
1057
4                         4+               0                 8
7969
5                          3               0                 1
15227
6                          2               1                 1
19215
7                          2               1                 1
15854
```

```
8                           2               1                   1
15686
9                           1               1                   8
7871
```

Average amount spend per customer for Male and Female

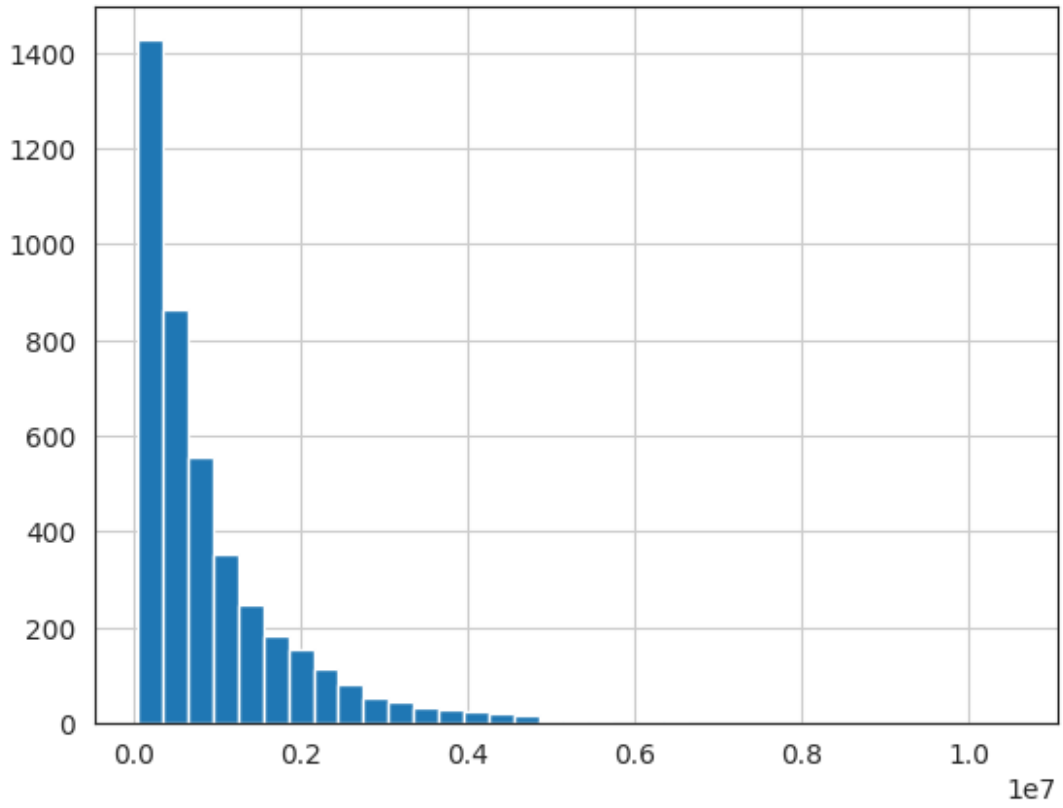
```
amt_df = df.groupby(['User_ID', 'Gender'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
```

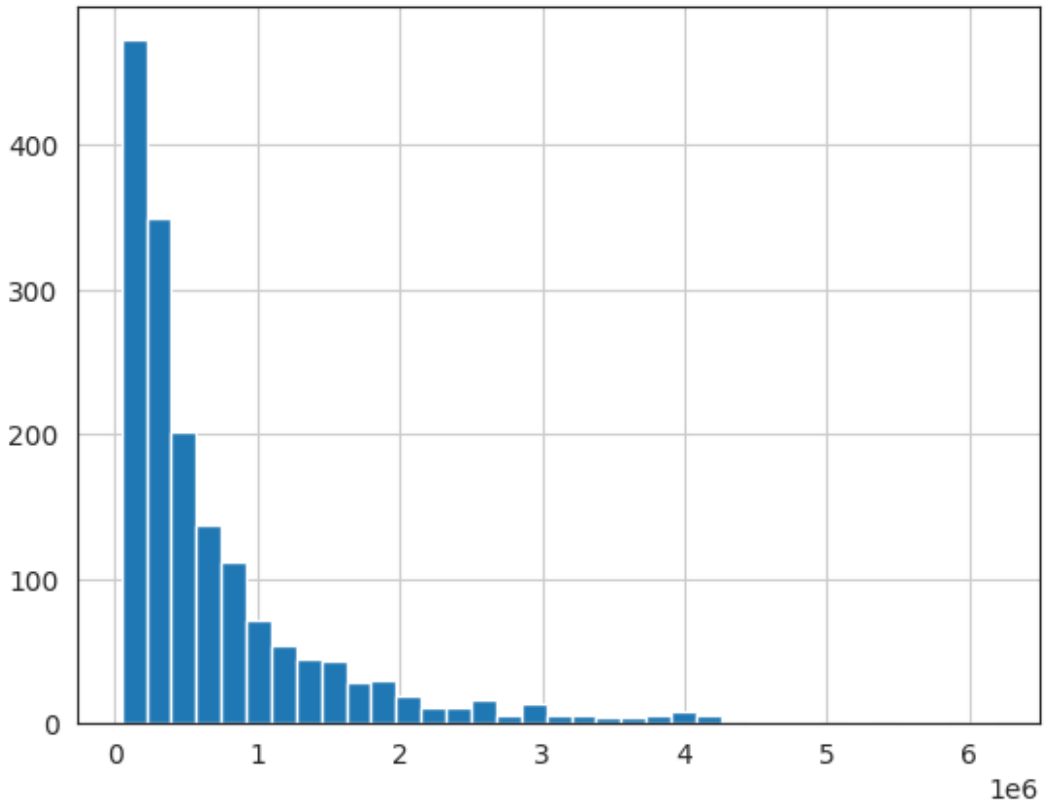
```
      User_ID Gender  Purchase
0     1000001      F    334093
1     1000002      M    810472
2     1000003      M    341635
3     1000004      M    206468
4     1000005      M    821001
...       ...    ...       ...
5886  1006036      F   4116058
5887  1006037      F   1119538
5888  1006038    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

[5891 rows x 3 columns]
```

```
# Gender wise value counts in avg_amt_df
df['Gender'].value_counts()
```

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

```
# histogram of average amount spend for each customer - Male & Female
amt_df[amt_df['Gender']=='M']['Purchase'].hist(bins=35)
plt.show()

amt_df[amt_df['Gender']=='F']['Purchase'].hist(bins=35)
plt.show()
```


1400
1200
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
1e7


400
300
200
100
0
0
1
2
3
4
5
6
1e6

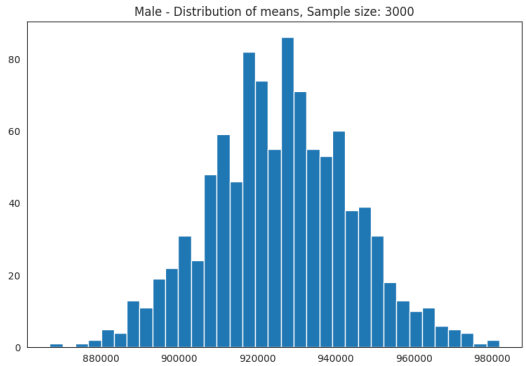
```
male_avg = amt_df[amt_df['Gender']=='M']['Purchase'].mean()
female_avg = amt_df[amt_df['Gender']=='F']['Purchase'].mean()

print("Average amount spend by Male customers:
{:.2f}".format(male_avg))
print("Average amount spend by Female customers:
{:.2f}".format(female_avg))
```

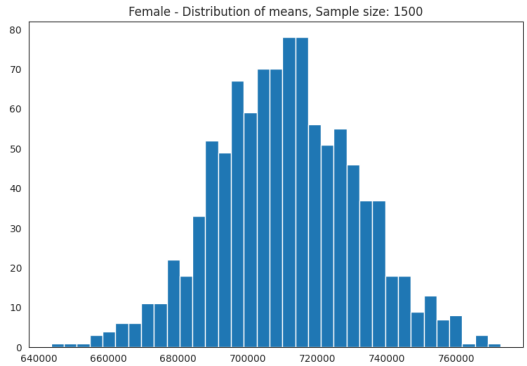
```
Average amount spend by Male customers: 925344.40
Average amount spend by Female customers: 712024.39
```

Observation

- Male customers spend more money than female customers

```
male_df = amt_df[amt_df['Gender']=='M']
female_df = amt_df[amt_df['Gender']=='F']

genders = ["M", "F"]

male_sample_size = 3000
female_sample_size = 1500
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=35)
axis[1].hist(female_means, bins=35)
axis[0].set_title("Male - Distribution of means, Sample size: 3000")
axis[1].set_title("Female - Distribution of means, Sample size: 1500")

plt.show()
```






```
print("Population mean - Mean of sample means of amount spend for
Male: {:.2f}".format(np.mean(male_means)))
print("Population mean - Mean of sample means of amount spend for
Female: {:.2f}".format(np.mean(female_means)))

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

```
Population mean - Mean of sample means of amount spend for Male:
925877.07
Population mean - Mean of sample means of amount spend for Female:
710935.46

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

Observation

Now using the **Central Limit Theorem** for the population we can say that:

- Average amount spend by male customers is **9,26,341.86**
- Average amount spend by female customers is **7,11,704.09**

```
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))
```

```
Male confidence interval of means: (895617.83, 955070.97)
Female confidence interval of means: (673254.77, 750794.02)
```

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

- Average amount spend by male customer will lie in between: **(895617.83, 955070.97)**
- Average amount spend by female customer will lie in between: **(673254.77, 750794.02)**

Doing the same activity for married vs unmarried

```
amt_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 x 3 columns]
```

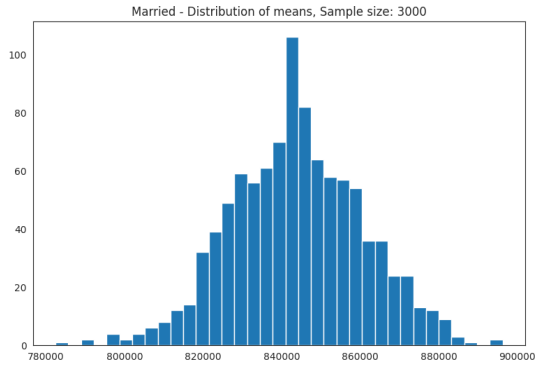
```
amt_df = df.groupby(['User_ID', 'Marital_Status'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
```

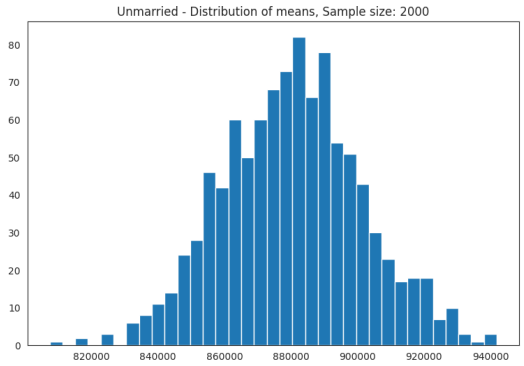
```
      User_ID  Marital_Status  Purchase
0     1000001               0    334093
1     1000002               0    810472
2     1000003               0    341635
3     1000004               1    206468
4     1000005               1    821001
...       ...             ...       ...
5886  1006036   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

[5891 rows x 3 columns]
```

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

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

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

for _ in range(num_repitions):
    marid_mean =
amt_df[amt_df['Marital_Status']==1].sample(marid_samp_size,
replace=True)['Purchase'].mean()
    unmarid_mean =
amt_df[amt_df['Marital_Status']==0].sample(unmarid_sample_size,
replace=True)['Purchase'].mean()

    marid_means.append(marid_mean)
    unmarid_means.append(unmarid_mean)


fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))

axis[0].hist(marid_means, bins=35)
axis[1].hist(unmarid_means, bins=35)
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(marid_means)))
print("Population mean - Mean of sample means of amount spend for
Unmarried: {:.2f}".format(np.mean(unmarid_means)))

print("\nMarried - Sample mean: {:.2f} Sample std:
{:.2f}".format(amt_df[amt_df['Marital_Status']==1]['Purchase'].mean(),
amt_df[amt_df['Marital_Status']==1]['Purchase'].std()))
print("Unmarried - Sample mean: {:.2f} Sample std:
{:.2f}".format(amt_df[amt_df['Marital_Status']==0]['Purchase'].mean(),
amt_df[amt_df['Marital_Status']==0]['Purchase'].std()))
```






```
Population mean - Mean of sample means of amount spend for Married:
843731.50
Population mean - Mean of sample means of amount spend for Unmarried:
880327.02

Married - Sample mean: 843526.80 Sample std: 935352.12
Unmarried - Sample mean: 880575.78 Sample std: 949436.25
```

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

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

    new_df = amt_df[amt_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))
```

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

Calculating the average amount spent by Age

```
amt_df = df.groupby(['User_ID', 'Age'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_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 x 3 columns]
```

```
amt_df['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 = 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 = amt_df[amt_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 = amt_df[amt_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_lim = sample_mean - margin_of_error_clt
    upper_lim = sample_mean + margin_of_error_clt

    print("For age {} --> confidence interval of means: ({:.2f},
{:.2f})".format(val, lower_lim, upper_lim))
```

```
For age 26-35 --> confidence interval of means: (945034.42,
1034284.21)
```

```
For age 36-45 --> confidence interval of means: (823347.80, 935983.62)
For age 18-25 --> confidence interval of means: (801632.78, 908093.46)
For age 46-50 --> confidence interval of means: (713505.63, 871591.93)
For age 51-55 --> confidence interval of means: (692392.43, 834009.42)
For age 55+ --> confidence interval of means: (476948.26, 602446.23)
For age 0-17 --> confidence interval of means: (527662.46, 710073.17)
```

Insights

- ~ 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% Single, 40% Married
- 35% Staying in the city from 1 year, 18% from 2 years, 17% from 3 years
- Total of 20 product categories are there
- There are 20 differnent 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: **925344.40**
- Average amount spend by Female customers: **712024.39**

Confidence Interval by Gender

Now using the **Central Limit Theorem** for the **population**:

- Average amount spend by male customers is **9,26,341.86**
- Average amount spend by female customers is **7,11,704.09**

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

- Average amount spend by male customer will lie in between:**(895617.83, 955070.97)**
- Average amount spend by female customer will lie in between: **(673254.77, 750794.02)**

Confidence Interval by Marital_Status

- Married confidence interval of means: **(806668.83, 880384.76)**
- Unmarried confidence interval of means: **(848741.18, 912410.38)**

Confidence Interval by Age

- For age **26-35** --> confidence interval of means: **(945034.42, 1034284.21)**
- For age **36-45** --> confidence interval of means: **(823347.80, 935983.62)**
- For age **18-25** --> confidence interval of means: **(801632.78, 908093.46)**
- For age **46-50** --> confidence interval of means: **(713505.63, 871591.93)**
- For age **51-55** --> confidence interval of means: **(692392.43, 834009.42)**
- For age **55+** --> confidence interval of means: **(476948.26, 602446.23)**
- For age **0-17** --> confidence interval of means: **(527662.46, 710073.17)**

Recommendations

- Men spent more money than women, So company should focus on retaining the male customers and getting more male customers.
- **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.
- **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.