Walmart Business Case Study

Problem Statement

- Walmart is an American multinational retail corporation that operates a chain of supercenters, discount departmental stores, and grocery stores from the United States.
- This data is collected on black friday which is a colloquial term for the Friday after Thanksgiving in the United States. It traditionally marks the start of the Christmas shopping season in the United States
- 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).

In [245…

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm
```

Analysing basic metrics

In [2]:

```
df = pd.read_csv('C:/Users/hp/OneDrive/Desktop/walmart.csv')
```

```
df.shape # There are 10 features and 550068 rows for customer data.
```

In [3]:

Out[3]:
```
(550068, 10)
```

In [4]:
```python
df.head()
```

Out[4]:

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

In [5]:
```python
df.columns # Columns present in data
```

Out[5]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

In [6]:
```python
df.dtypes # Data type of each column
```

Out[6]:
```
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
```

In [7]:
```python
df.info() # Basic information about dataset
```

```
<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
```

In [8]:
```python
df.describe()  # Analysis on statistical information about dataset
```

	User_ID	Occupation	Marital_Status	Product_Category	Purchase

Out[8]:

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

In [9]:
```
df.describe(include='object') # Checking data for categorical columns
```

Out[9]:

	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

Checking is there any missing values(null values) in given data

In [10]:
```
np.any(df.isna())
```

Out[10]:
```
False
```

Checking is there any missing values(null values) in given data

In [11]:
```
np.any(df.duplicated())
```

Out[11]:
```
False
```

In [12]:
```
df['User_ID'].nunique()
```

Out[12]:
```
5891
```

In [13]:
```
df['Marital_Status'].value_counts()
```

Out[13]:
```
0    324731
1    225337
Name: Marital_Status, dtype: int64
```

In [14]:
```
np.round(df['Stay_In_Current_City_Years'].value_counts(normalize=True)*100 ,2)
```

Out[14]:
```
1     35.24
2     18.51
3     17.32
4+    15.40
0     13.53
Name: Stay_In_Current_City_Years, dtype: float64
```

In [15]:
```
df['Gender'].value_counts(normalize=True)*100
```

Out[15]:
```
M    75.310507
F    24.689493
```

```
Name: Gender, dtype: float64
```

In [16]:
```
print(df['Product_Category'].nunique())
print(df['Occupation'].nunique())
```

```
20
21
```

Primary Analysis on data

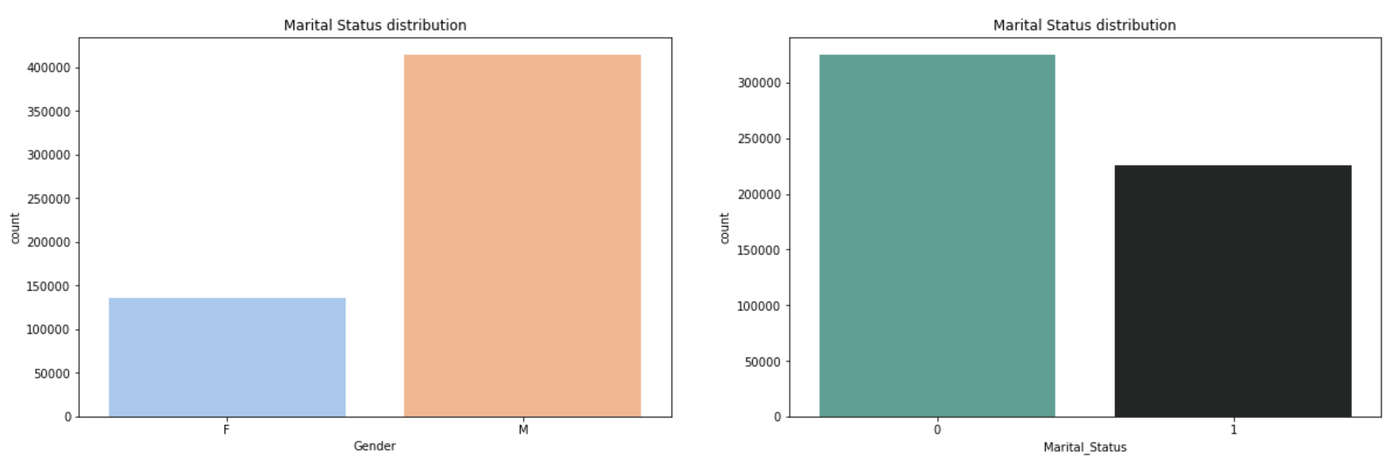
1. The given dataset has total 10 features and 550068 rows for customer data.
2. There are no null values and duplicates present in dataset.
3. This data includes 3631 unique products and product id is most occuring which occurs 1880 times in data.
4. There are 5891 unique customers.
5. Data includes more unmaried customers than married.
6. Purchase is target variable.

Visual Analysis - Univariate & Bivariate

Univariate Analysis

In [17]:
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
sns.countplot(data=df,x='Gender',palette='pastel',ax=axs[0]).set(title='Marital Status d
sns.countplot(data=df, x='Marital_Status',palette='dark:#5A9_r',ax=axs[1]).set(title='Ma
```

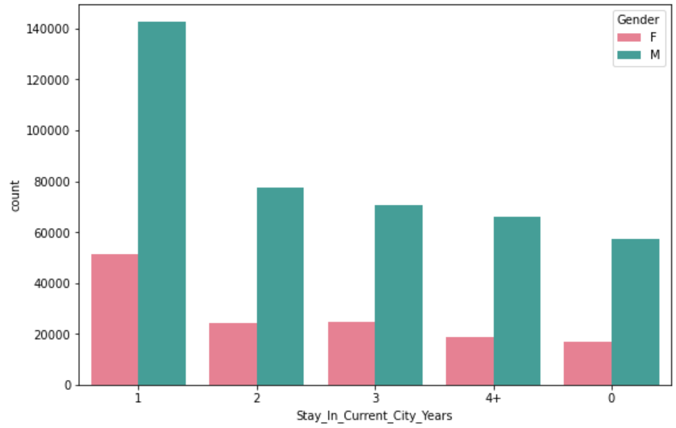
Out[17]:
```
[Text(0.5, 1.0, 'Marital Status distribution')]
```

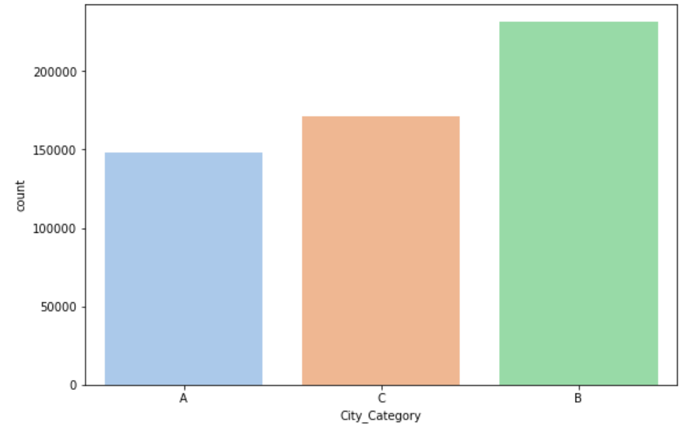


- Above graphs displays there are more number of male present in data
- Martial status graph shows, unmarried people have done more shopping on black friday

In [18]:
```
fig, ax = plt.subplots(nrows=1, ncols=2,figsize=(20,6))
sns.countplot(data=df,x='Stay_In_Current_City_Years',order=df['Stay_In_Current_City_Year
sns.countplot(data=df,x='City_Category',palette='pastel',ax=ax[1])
```

Out[18]:
```
<AxesSubplot:xlabel='City_Category', ylabel='count'>
```

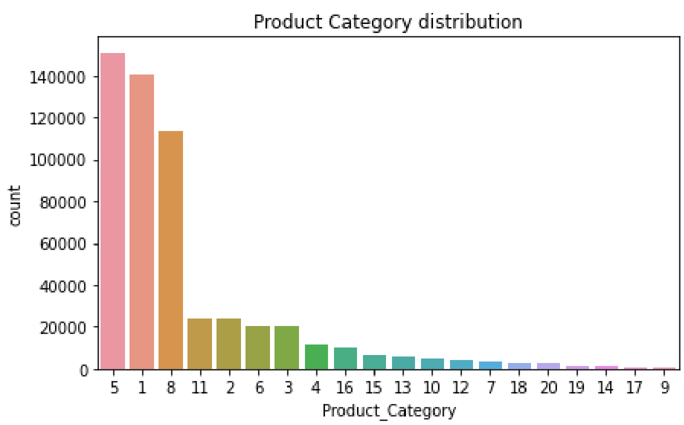


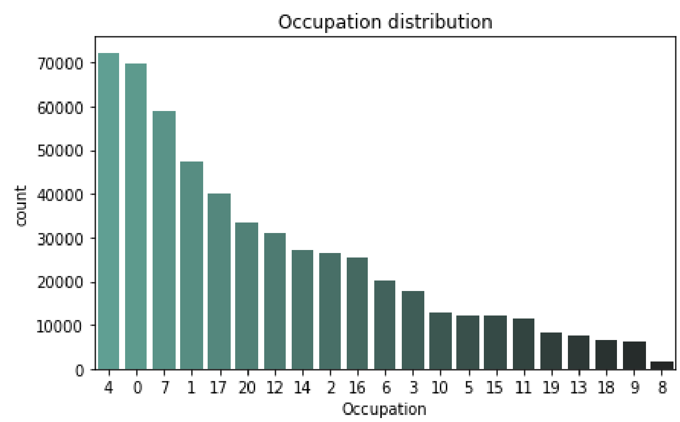


- Above graphs displays irrespective of gender, people shopping more have 1 year stay in current city.
- City category B has more records followed by C and A.

In [19]:
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 4))
sns.countplot(data=df, x='Product_Category',order=df['Product_Category'].value_counts(as
sns.countplot(data=df, x='Occupation',palette='dark:#5A9_r',order=df['Occupation'].value
```

Out[19]:
```
[Text(0.5, 1.0, 'Occupation distribution')]
```



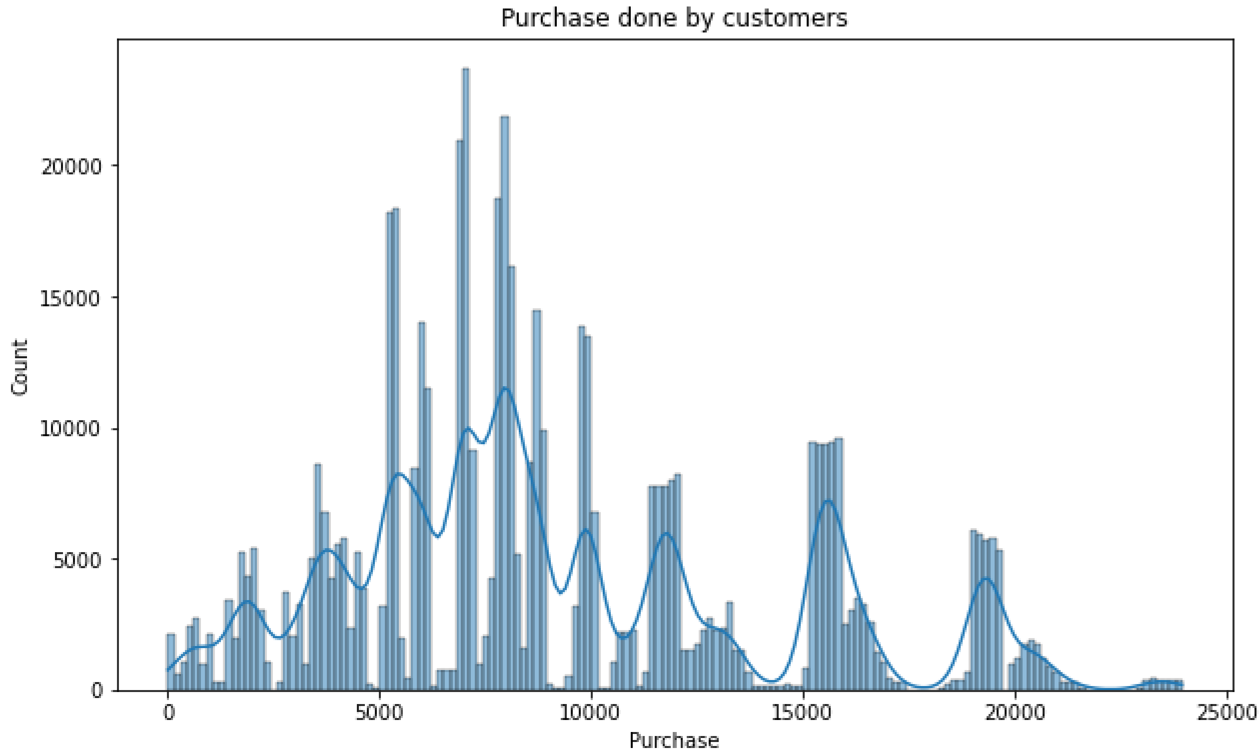


- There are 20 product categories with product category 5,1,8 having more purchasing counts.
- We have 21 occupations categories. Occupation category 4 are with higher number of purchases and category 8 with the lowest number of purchaes.

In [20]:
```
print('Purchase mean: ',round(df['Purchase'].mean()))
print('Purchase meadian: ',round(df['Purchase'].median()))
```

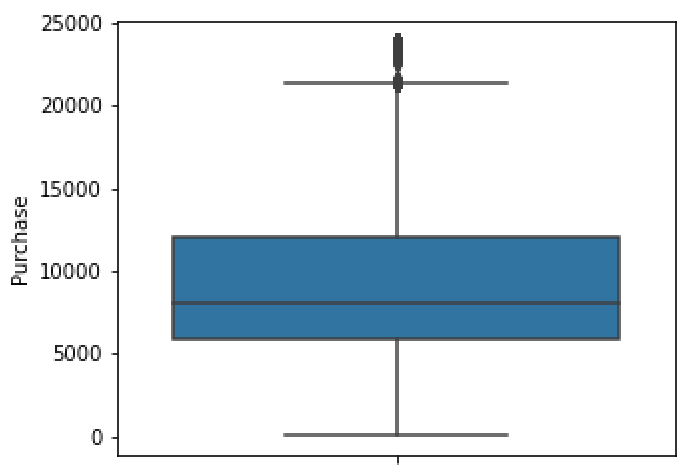
```
Purchase mean:  9264
Purchase meadian:  8047
```

In [21]:
```
fig, ax = plt.subplots(figsize=(10, 6))
r = sns.histplot(df['Purchase'],kde=True).set(title='Purchase done by customers')
plt.show()
```



- From above plot we can see most of data between 5000 to 10000.
- Mean and median for purchase are 9264 and 8047 respectively.

In [22]:
```
plt.figure(figsize=(5, 4))
sns.boxplot(data=df, y='Purchase')
plt.show()
```



- Outliers are present in purchase column

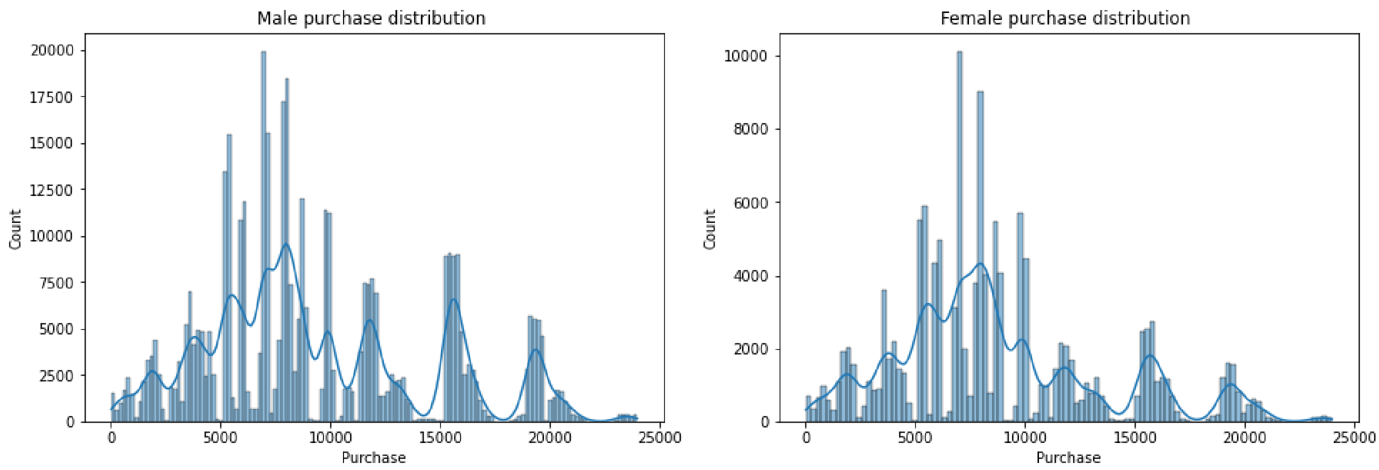
Bivariate Analysis

- We will separate male and female data in two data frame for gender wise analysis

In [23]:
```
Male_Data = df.loc[df['Gender'] == 'M']
Female_Data = df.loc[df['Gender'] == 'F']
```

In [24]:
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(16,5))
sns.histplot(Male_Data['Purchase'], ax=axs[0],kde=True).set_title("Male purchase distrib
```

```
sns.histplot(Female_Data['Purchase'], ax=axs[1],kde=True).set_title("Female purchase dis
plt.show()
```



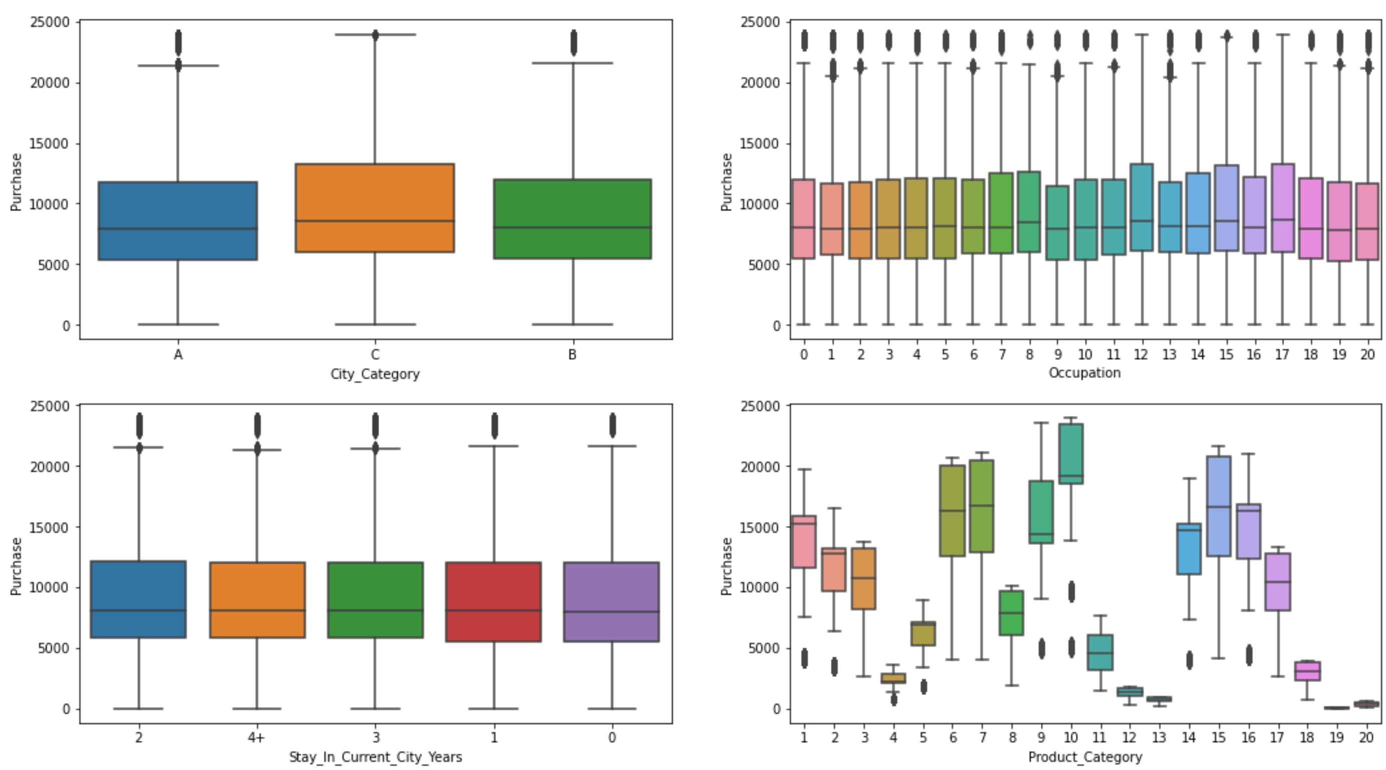
- It can be seen that for female and male customers data lies between 5000 - 10000.
- So spending behaviour is very much similar in nature for both males and females
- The purchase count are more in case of males.

In [244…

```
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(18, 10))
sns.boxplot(data=df,x='City_Category', y='Purchase',ax=axs[0,0])
sns.boxplot(data=df,x='Occupation', y='Purchase',ax=axs[0,1])
sns.boxplot(data=df,x='Stay_In_Current_City_Years', y='Purchase',ax=axs[1,0])
sns.boxplot(data=df,x='Product_Category', y='Purchase',ax=axs[1,1])
```

Out[244]:

```
<AxesSubplot:xlabel='Product_Category', ylabel='Purchase'>
```



- Observation

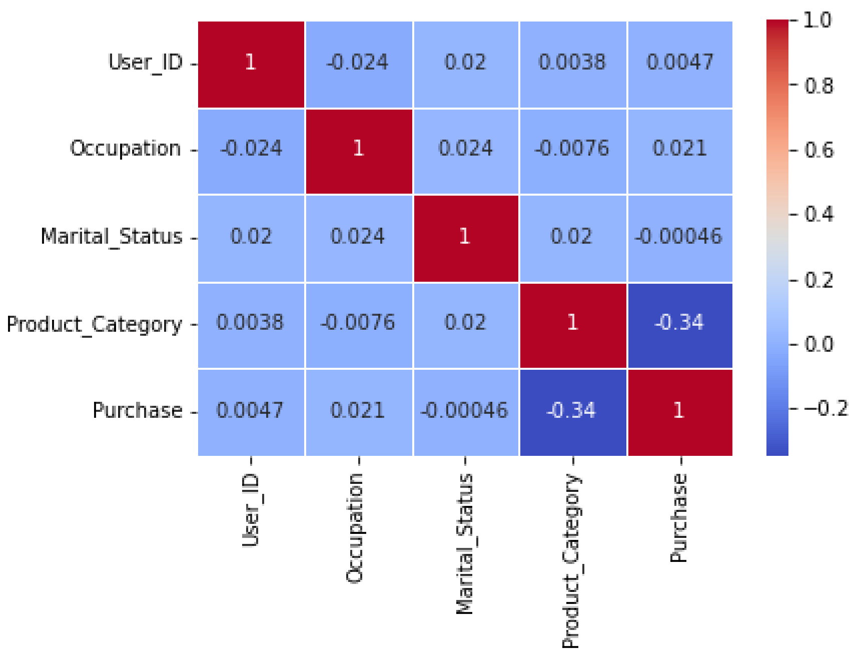
1. Age categories, we see similar purchase behaviour. For all age groups, most of the purchases are of the values between 5k to 15k with all have some outliers.
2. Occupation as well, we see similar purchasing behaviour.

3. City category, stay in current city years, marital status - we see the users spends mostly in the range of 5k to 12k.
4. Product category 10 products are having highest cost. while product 4,13,19,20 have very less amount.

In [ ]:

In [28]:
```
tc = df.corr()
sns.heatmap(tc, annot=True, cmap="coolwarm", linewidth=.5)
```

Out[28]:
```
<AxesSubplot:>
```



From correlation plot, correlation is significant between categorical variables and values are less than 0.

4. Data exploration and Answering questions

Question 4.0.

Are women spending more money per transaction than men?Why or Why not? Average amount spends per customer for Male and Female

In [72]:
```
avg_amount_df = df.groupby(['User_ID', 'Gender'])[['Purchase']].sum()
avg_amount_df = avg_amount_df.reset_index()
avg_amount_df
```

Out[72]:

	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

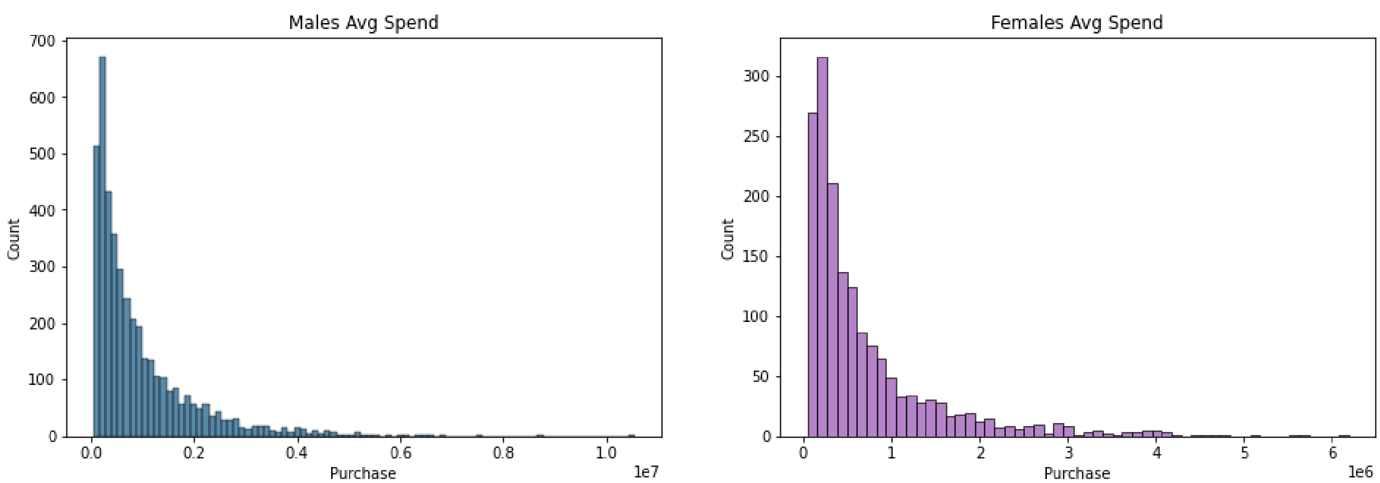
5891 rows × 3 columns

In [73]:
```
avg_amount_df['Gender'].value_counts()
```

Out[73]:
```
M    4225
F    1666
Name: Gender, dtype: int64
```

In [77]:
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(16,5))

sns.histplot(data=avg_amount_df[avg_amount_df['Gender']=='M']['Purchase'],color='#21618C
sns.histplot(data=avg_amount_df[avg_amount_df['Gender']=='F']['Purchase'], color='#9B59B
```

Out[77]:
```
Text(0.5, 1.0, 'Females Avg Spend')
```



In [85]:
```
print('Average amount spent by males: ',avg_amount_df[avg_amount_df['Gender']=='M']['Pur
print('Average amount spent by females: ',avg_amount_df[avg_amount_df['Gender']=='F']['P
```

```
Average amount spent by males:  925344.4023668639
Average amount spent by females:  712024.3949579832
```

Answer

Average amount spend by males are higher than females.

Question 4.1.

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

In [ ]:
```
# We will check Finding the sample(sample size=1000) for avg purchase amount for males a
```

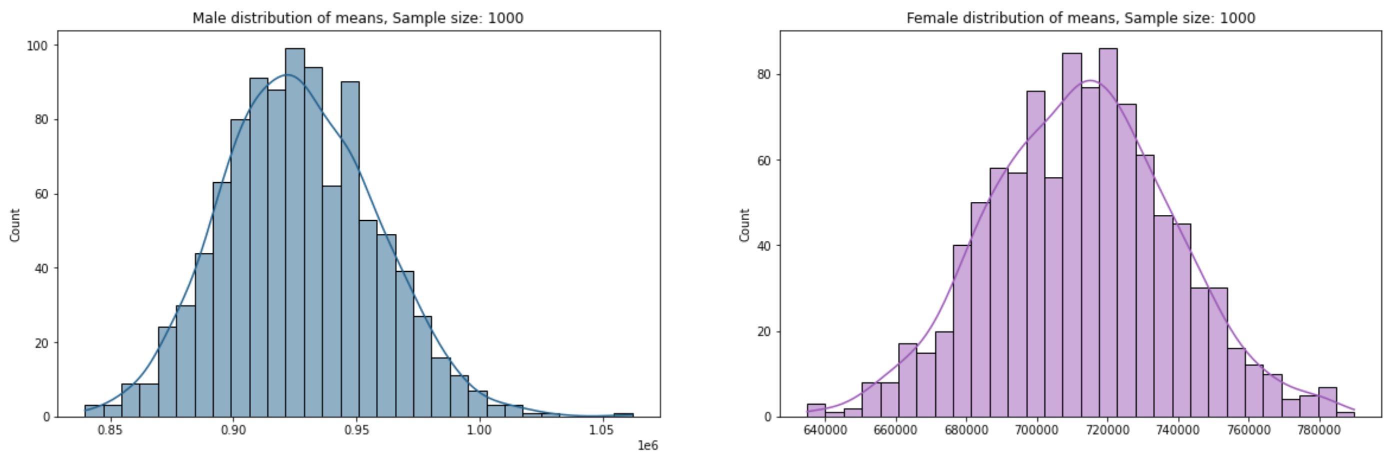
In [68]:
```
avgamt_male = avgamt_gender[avgamt_gender['Gender']=='M']
avgamt_female = avgamt_gender[avgamt_gender['Gender']=='F']
```

In [69]:
```
fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
```

```
n  = 1000
bootstrapped_mean_male_data = []
bootstrapped_mean_female_data = []

for reps in range(1000):

    bootstrapped_samples_male = np.random.choice(avgamt_male['Purchase'], size=n)
    bootstrapped_mean_male = bootstrapped_samples_male.mean()
    bootstrapped_mean_male_data.append(bootstrapped_mean_male)


    bootstrapped_sample_female = np.random.choice(avgamt_female['Purchase'], size=n)
    bootstrapped_mean_female = bootstrapped_sample_female.mean()
    bootstrapped_mean_female_data.append(bootstrapped_mean_female)

sns.histplot(bootstrapped_mean_male_data,bins=30,ax = axis[0],color='#21618C',kde=True).
sns.histplot(bootstrapped_mean_female_data,bins=30,ax = axis[1],kde=True,color='#9B59B6'
```

Out[69]:

```
Text(0.5, 1.0, 'Female distribution of means, Sample size: 1000')
```



In [159…

```
print("Population avg spent amount for Male: ",avgamt_male['Purchase'].mean())
print("Population avg spent amount for Female: ",avgamt_female['Purchase'].mean())
print('-'*60)
print("Sample avg spent amount for Male: ",np.mean(bootstrapped_mean_male_data))
print("Sample avg spent amount for Female: ",np.mean(bootstrapped_mean_female_data))
```

```
Population avg spent amount for Male:   925344.4023668639
Population avg spent amount for Female:   712024.3949579832
------------------------------------------------------------
Sample avg spent amount for Male:  926736.1635769999
Sample avg spent amount for Female:  712461.2448580001
```

Answer

- Central Limit Theorem for the population we can say that:

1. Average amount spend by male customers is 926736
2. Average amount spend by female customers is 712461

Question 4.2.

Are confidence intervals of average male and female spending overlapping?

Sample size = 1000 Calculating 90% confidence interval

```
In [162...: #Z values at 90%,
z_90=1.645 #90% Confidence Interval
```

```
In [157...: print("Sample devation for Male: ",np.std(bootstrapped_mean_male_data))
print("Sample devation for Female: ",np.std(bootstrapped_mean_female_data))
print('-'*50)
print("Sample standard error for Male: ",np.std(bootstrapped_mean_male_data)/np.sqrt(100
print("Sample std error for Female: ",np.std(bootstrapped_mean_female_data)/np.sqrt(1000
print('-'*50)
sample_mean_male=np.mean(bootstrapped_mean_male_data)
sample_std_male=np.std(bootstrapped_mean_male_data)
sample_std_error_male=sample_std_male/np.sqrt(1000)
Upper_Limit_male=z_90*sample_std_error_male + sample_mean_male
Lower_Limit_male=sample_mean_male - z_90*sample_std_error_male

sample_mean_female=np.mean(bootstrapped_mean_female_data)
sample_std_female=np.std(bootstrapped_mean_female_data)
sample_std_error_female=sample_std_female/np.sqrt(1000)
Upper_Limit_female=z_90*sample_std_error_female + sample_mean_female
Lower_Limit_female=sample_mean_female - z_90*sample_std_error_female

print("Male limit: ",[round(Lower_Limit_male),round(Upper_Limit_male)])
print("Female limit: ",[round(Lower_Limit_female),round(Upper_Limit_female)])
```

```
Sample devation for Male:  31055.939936976774
Sample devation for Female:  25970.098586465687
--------------------------------------------------
Sample standard error for Male:  982.0750507823265
Sample std error for Female:  821.2466259235085
--------------------------------------------------
Male limit:  [925121, 928352]
Female limit:  [711110, 713812]
```

Answer

Confidence interval at 90%:

Average amount spend by **male** customers lie in the range 925121 - 928352

Average amount spend by **female** customers lie in range 711110 - 713812

Sample size = 1000 Calculating 95% confidence interval

```
In [ ]: z_95 = 1.960 #95% Confidence Interval
```

```
In [161...: sample_mean_male=np.mean(bootstrapped_mean_male_data)
sample_std_male=np.std(bootstrapped_mean_male_data)
sample_std_error_male=sample_std_male/np.sqrt(1000)
Upper_Limit_male=z_95*sample_std_error_male + sample_mean_male
Lower_Limit_male=sample_mean_male - z_95*sample_std_error_male

sample_mean_female=np.mean(bootstrapped_mean_female_data)
sample_std_female=np.std(bootstrapped_mean_female_data)
sample_std_error_female=sample_std_female/np.sqrt(1000)
Upper_Limit_female=z_95*sample_std_error_female + sample_mean_female
Lower_Limit_female=sample_mean_female - z_95*sample_std_error_female

print("Male limit: ",[round(Lower_Limit_male),round(Upper_Limit_male)])
print("Female limit: ",[round(Lower_Limit_female),round(Upper_Limit_female)])
```

```
Sample devation for Male:  31055.939936976774
Sample devation for Female:  25970.098586465687
--------------------------------------------------
```

```
Sample standard error for Male:  982.0750507823265
Sample std error for Female:  821.2466259235085
--------------------------------------------------
Male limit:  [924811, 928661]
Female limit:  [710852, 714071]
```

Answer

Confidence interval at 95%:

Average amount spend by **male** customers lie in the range 924811 - 928661

Average amount spend by **female** customers lie in range 710852 - 714071

Sample size = 1000 Calculating 99% confidence interval

In [ ]:

```
z_99=2.576 # 99% Confidence Interval
```

In [163…

```
sample_mean_male=np.mean(bootstrapped_mean_male_data)
sample_std_male=np.std(bootstrapped_mean_male_data)
sample_std_error_male=sample_std_male/np.sqrt(1000)
Upper_Limit_male=z_99*sample_std_error_male + sample_mean_male
Lower_Limit_male=sample_mean_male - z_99*sample_std_error_male

sample_mean_female=np.mean(bootstrapped_mean_female_data)
sample_std_female=np.std(bootstrapped_mean_female_data)
sample_std_error_female=sample_std_female/np.sqrt(1000)
Upper_Limit_female=z_99*sample_std_error_female + sample_mean_female
Lower_Limit_female=sample_mean_female - z_99*sample_std_error_female

print("Male limit: ",[round(Lower_Limit_male),round(Upper_Limit_male)])
print("Female limit: ",[round(Lower_Limit_female),round(Upper_Limit_female)])
```

```
Male limit:  [924206, 929266]
Female limit:  [710346, 714577]
```

Answer

Confidence interval at 99%:

Average amount spend by **male** customers lie in the range 924811 - 929266

Average amount spend by **female** customers lie in range 710346 - 714577

Now we will increase sample size to 1500 and 95% confidence interval

In [165…

```
z_95 = 1.960
print("Population avg spent amount for Male: ",avgamt_male['Purchase'].mean())
print("Population avg spent amount for Female: ",avgamt_female['Purchase'].mean())
print('-'*60)
print("Sample devation for Male: ",np.std(bootstrapped_mean_male_data))
print("Sample devation for Female: ",np.std(bootstrapped_mean_female_data))
print('-'*50)
print("Sample standard error for Male: ",np.std(bootstrapped_mean_male_data)/np.sqrt(150
print("Sample std error for Female: ",np.std(bootstrapped_mean_female_data)/np.sqrt(1500
print('-'*50)
sample_mean_male=np.mean(bootstrapped_mean_male_data)
sample_std_male=np.std(bootstrapped_mean_male_data)
sample_std_error_male=sample_std_male/np.sqrt(1500)
Upper_Limit_male=z_95*sample_std_error_male + sample_mean_male
Lower_Limit_male=sample_mean_male - z_95*sample_std_error_male
```

```
sample_mean_female=np.mean(bootstrapped_mean_female_data)
sample_std_female=np.std(bootstrapped_mean_female_data)
sample_std_error_female=sample_std_female/np.sqrt(1500)
Upper_Limit_female=z_95*sample_std_error_female + sample_mean_female
Lower_Limit_female=sample_mean_female - z_95*sample_std_error_female

print("Male limit: ",[round(Lower_Limit_male),round(Upper_Limit_male)])
print("Female limit: ",[round(Lower_Limit_female),round(Upper_Limit_female)])
```

```
Population avg spent amount for Male:  925344.4023668639
Population avg spent amount for Female:  712024.3949579832
-------------------------------------------------------------
Sample devation for Male:  31055.939936976774
Sample devation for Female:  25970.098586465687
---------------------------------------------------
Sample standard error for Male:  801.8609211781925
Sample std error for Female:  670.5450621649759
---------------------------------------------------
Male limit:  [925165, 928308]
Female limit:  [711147, 713776]
```

Answer

Confidence interval at 95% and large sample size:
As we increase sample size spread of data will be less and confidence interval is will be more close to population mean.

Average amount spend by **male** customers lie in the range 925165 - 928308

Average amount spend by **female** customers lie in range 711147 - 713776

Question 4.3:

Results when the same activity is performed for Married vs Unmarried

In [192…

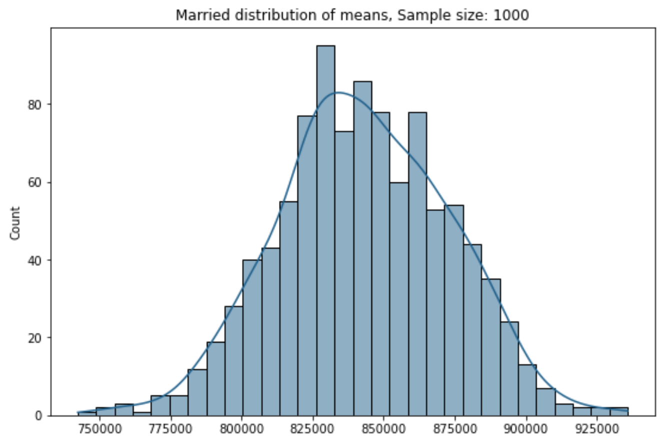
```
average_marital_df = df.groupby(['User_ID','Marital_Status'])['Purchase'].sum()
average_marital_df = average_marital_df.reset_index()

average_married = average_marital_df[average_marital_df['Marital_Status']==1]
average_single = average_marital_df[average_marital_df['Marital_Status']==0]
```

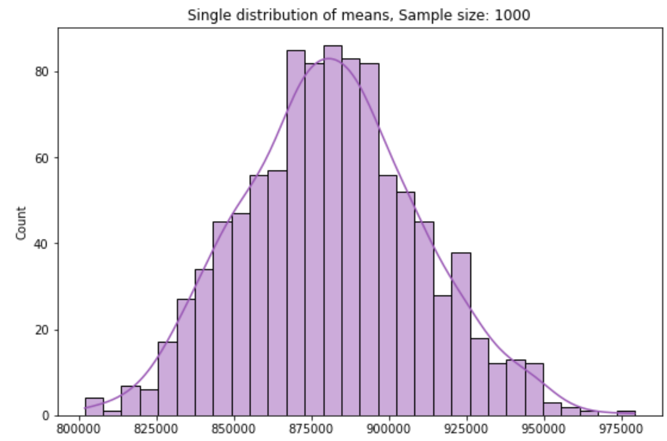
In [195…

```
fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
n  = 1000
married_sample_mean = []
single_sample_mean = []


for reps in range(1000):

    avg_married = average_married[average_married['Marital_Status']==1].sample(1000, rep
    avg_single = average_single[average_single['Marital_Status']==0].sample(1000, replac

    married_sample_mean.append(avg_married)
    single_sample_mean.append(avg_single)

sns.histplot(married_sample_mean,bins=30,ax = axis[0],color='#21618C',kde=True).set_titl
sns.histplot(single_sample_mean,bins=30,ax = axis[1],kde=True,color='#9B59B6').set_title
```

Out[195]:

```
Text(0.5, 1.0, 'Single distribution of means, Sample size: 1000')
```





Answer

- The means sample seems to be normally distributed for both married and singles.
- Also, we can see the mean of the sample means are closer to the population mean as per central limit theorem.

In [202…

```
z_95 = 1.960

print("Population avg spent amount for married: ",average_married['Purchase'].mean())
print("Population avg spent amount for single: ",average_single['Purchase'].mean())
print('-'*60)
print("Sample avg spent amount for married: ",np.mean(married_sample_mean))
print("Sample avg spent amount for single: ",np.mean(single_sample_mean))
print('-'*60)
print("Sample standard devation for married: ",np.std(married_sample_mean))
print("Sample standard for single: ",np.std(single_sample_mean))
print('-'*50)
print("Sample standard error for married: ",np.std(married_sample_mean)/np.sqrt(1500))
print("Sample std error for single: ",np.std(single_sample_mean)/np.sqrt(1500))
print('-'*50)

sample_mean_married=np.mean(married_sample_mean)
sample_std_married=np.std(married_sample_mean)
sample_std_error_married=sample_std_married/np.sqrt(1500)

Upper_Limit_male=z_95*sample_std_error_married + sample_mean_married
Lower_Limit_male=sample_mean_married - z_95*sample_std_error_married

sample_mean_single=np.mean(single_sample_mean)
sample_std_single=np.std(sample_mean_single)
sample_std_error_single=sample_std_single/np.sqrt(1500)
Upper_Limit_female=z_95*sample_std_error_single + sample_mean_single
Lower_Limit_female=sample_mean_single - z_95*sample_std_error_single

print("Married limit: ",[round(Lower_Limit_male),round(Upper_Limit_male)])
print("Single limit: ",[round(Lower_Limit_female),round(Upper_Limit_female)])
```

```
Population avg spent amount for married:  843526.7966855295
Population avg spent amount for single:  880575.7819724905
------------------------------------------------------------
Sample avg spent amount for married:  842974.5403770001
Sample avg spent amount for single:  881296.329447
------------------------------------------------------------
Sample standard devation for married:  30026.12390220903
Sample standard for single:  28641.786985609273
--------------------------------------------------
Sample standard error for married:  775.2711854961069
Sample std error for single:  739.5277600059003
--------------------------------------------------
```

```
Married limit:  [841455, 844494]
Single limit:  [881296, 881296]
```

Answer

- Average amount spend by married customers lie in the range: [841455, 844494]
- Average amount spend by unmarried customers lie in the range: [881296, 881296]

Question 4.4

Results when the same activity is performed for Age

In [203…

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

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

Out[203]:

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

In [232…

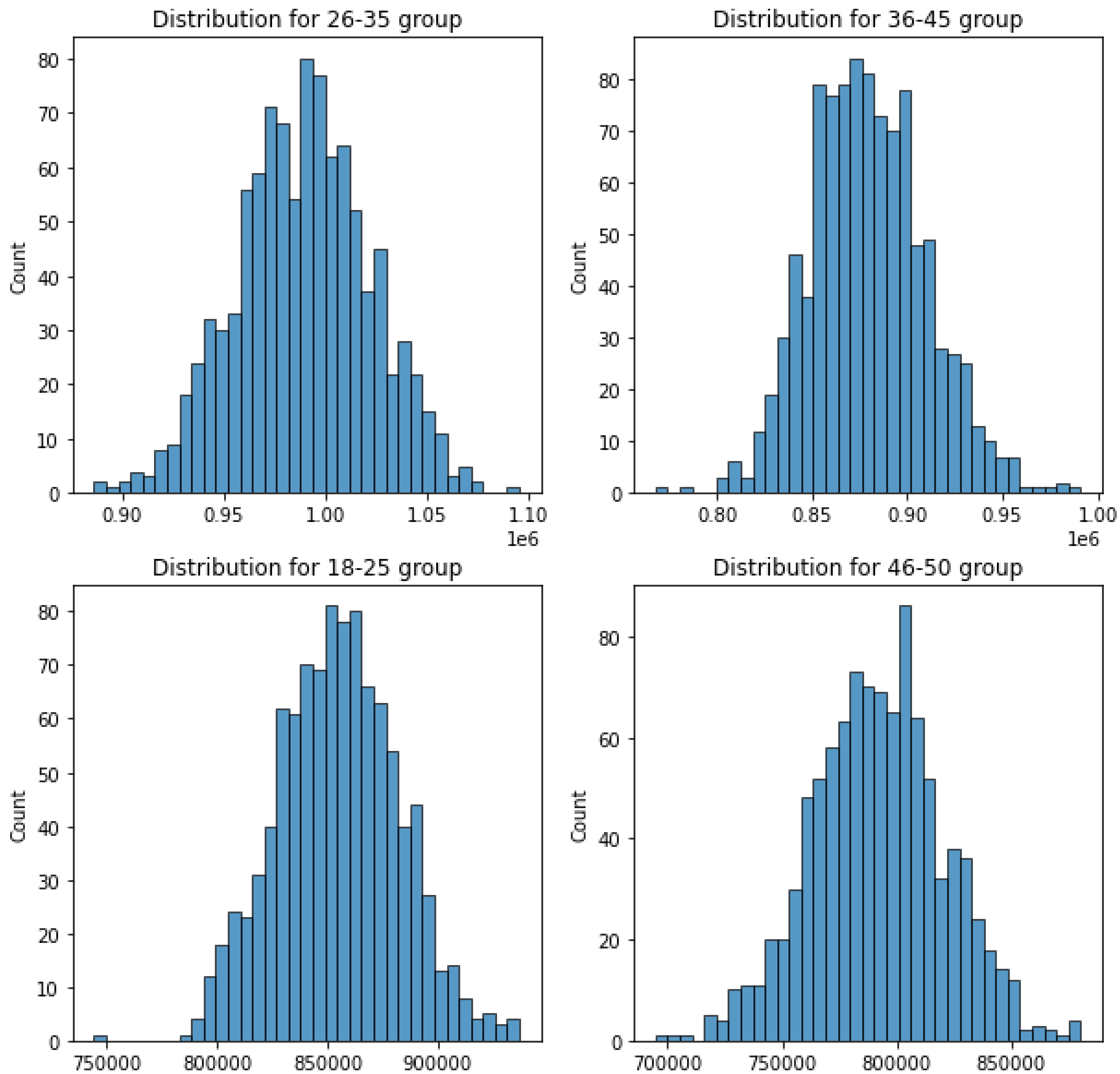
```
sample_size = 1000
num_repitions = 1000
sample_26_35=[]
sample_36_45=[]
sample_18_25=[]
sample_46_50=[]

for i in range(1000):
    mean_1 = average_age[average_age['Age']=='26-35'].sample(sample_size, replace=True)[
    mean_2 = average_age[average_age['Age']=='36-45'].sample(sample_size, replace=True)[
    mean_3 = average_age[average_age['Age']=='18-25'].sample(sample_size, replace=True)[
    mean_4 = average_age[average_age['Age']=='46-50'].sample(sample_size, replace=True)[
    sample_26_35.append(mean_1)
    sample_36_45.append(mean_2)
    sample_18_25.append(mean_3)
    sample_46_50.append(mean_4)
```

In [235…

```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(10,10))
sns.histplot(sample_26_35,bins=35,ax=axis[0,0]).set_title('Distribution for 26-35 group'
sns.histplot(sample_36_45,bins=35,ax=axis[0,1]).set_title('Distribution for 36-45 group'
sns.histplot(sample_18_25,bins=35,ax=axis[1,0]).set_title('Distribution for 18-25 group'
sns.histplot(sample_46_50,bins=35,ax=axis[1,1]).set_title('Distribution for 46-50 group'
```

Out[235]:

```
Text(0.5, 1.0, 'Distribution for 46-50 group')
```



Observations-

- For sample size of 1000 all means sample seems to be normally distributed for all age groups
- Also, we can see the mean of the sample means are closer to the population mean as per central limit theorem.

In [216…

```
print('Population mean for age group 26-35:',average_age[average_age['Age']=='26-35']['P
print('Population mean for age group 36-45:',average_age[average_age['Age']=='36-45']['P
print('Population mean for age group 18-25:',average_age[average_age['Age']=='18-25']['P
print('Population mean for age group 51-55:',average_age[average_age['Age']=='51-55']['P
print('Population mean for age group 55+:',average_age[average_age['Age']=='55+']['Purch
print('Population mean for age  group 0-17:',average_age[average_age['Age']=='0-17']['Pu
```

```
Population mean for age group 26-35: 989659.3170969313
Population mean for age group 36-45: 879665.7103684661
Population mean for age group 18-25: 854863.119738073
Population mean for age group 51-55: 763200.9230769231
Population mean for age group 55+: 539697.2446236559
Population mean for age  group 0-17: 618867.8119266055
```

- Calculating 95% confidence interval for avg expenses for different age groups for sample size 1000

In [207…

```
z_95=1.960
sample_size = 1000
num_repitions = 1000
```

```
age_intervals = ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']
all_means = {}


for i in age_intervals:
    all_means[i] = []
#print(all_means)
for i in age_intervals:
    for j in range(num_repitions):
        mean = average_age[average_age['Age']==i].sample(sample_size, replace=True)['Pur
        all_means[i].append(mean)

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

    new_df = average_age[average_age['Age']==val]
    std_error = z_95*new_df['Purchase'].std()/np.sqrt(len(new_df))
    sample_mean = new_df['Purchase'].mean()
    lower_lim = sample_mean - std_error
    upper_lim = sample_mean + std_error

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

```
{'26-35': [], '36-45': [], '18-25': [], '46-50': [], '51-55': [], '55+': [], '0-17': []}
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)
```

Answer

We can see the sample means are closer to the population mean for the differnt age groups.

5. Insights

1. 75% of the number of purchases are made by Male users and rest of the 25% is done by female users.This tells us the Male consumers are the major contributors to the number of sales for the retail store
2. When we combined Purchase and Marital_Status for analysis, we came to know that Single Men spend the most during the Black Friday.
3. For Age feature, we observed the consumers who belong to the age group 25-40 tend to spend the most.
4. Stay_In_Current_City_Years column, after analyzing this column we came to know the people who have spent 1 year in the city tend to spend the most.
5. Product_Category - 1, 5, 8, & 11 have highest purchasing frequency.
6. More users belong to B City_Category
7. Age 26-35 category is most occuring.
8. 53.75% purchase are made by customers who are staying in city for 1 and 2 years.
9. There are 20 product categories in total.
10. There are 21 different types of occupations in the city.
11. Average amount spend by Male customers: 925344.40
12. Average amount spend by Female customers: 712024.39

Confidence Interval by Gender

- Average amount spend by male customers lie in the range 924811 - 928661
- Average amount spend by female customers lie in range 710852 - 714071

Confidence Interval by Marital_Status

- Average amount spend by married customers lie in the range: [841455, 844494]
- Average amount spend by unmarried customers lie in the range: [881296, 881296]

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)

6. Recommendations

- For top purchasing customers company should offer discounts and benefits.
- To attract female customers in city company should offer discounts on beauty products and cloths.
- Men spent more money than women, So company should focus on retaining the male customers and getting more male customers.
- Company can focus on selling more of these products in Category - 1, 5, 8, & 11.
- 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. Many customers are falling under adult category.
- We have highest frequency of purchase order between 5k and 10k, company can focus more on these mid range products to increase the sales.
- Some of the Product category like 19,20,13 have very less purchase. Company can think of dropping it