walmart-case-study

September 18, 2023

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

[ ]:
```
df=pd.read_csv("/content/drive/MyDrive/walmart_case_study.csv")
```

[ ]:
```
df
```

[ ]:
```
        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
…         …        …    …    …        …            …
550063  1006033  P00372445      M  51-55          13             B
550064  1006035  P00375436      F  26-35           1             C
550065  1006036  P00375436      F  26-35          15             B
550066  1006038  P00375436      F    55+           1             C
550067  1006039  P00371644      F  46-50           0             B

       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
…                             …             …                …       …
550063                          1               1                20       368
550064                          3               0                20       371
550065                         4+               1                20       137
550066                          2               0                20       365
550067                         4+               1                20       490

[550068 rows x 10 columns]
```

[ ]:
```
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.shape
```

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

[ ]:
```
df.size
```

[ ]:
```
5500680
```

[ ]:
```
df.count()
```

[ ]:
```
User_ID                     550068
Product_ID                  550068
Gender                      550068
Age                         550068
Occupation                  550068
City_Category               550068
Stay_In_Current_City_Years  550068
Marital_Status              550068
Product_Category            550068
Purchase                    550068
dtype: int64
```

[ ]:
```
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
```

[ ]:
```
df.tail()
```

[ ]:
```
        User_ID Product_ID Gender    Age  Occupation City_Category  \
550063  1006033  P00372445      M  51-55          13             B
550064  1006035  P00375436      F  26-35           1             C
550065  1006036  P00375436      F  26-35          15             B
550066  1006038  P00375436      F    55+           1             C
550067  1006039  P00371644      F  46-50           0             B

       Stay_In_Current_City_Years  Marital_Status  Product_Category  Purchase
550063                          1               1                20       368
550064                          3               0                20       371
550065                         4+               1                20       137
550066                          2               0                20       365
550067                         4+               1                20       490
```

[ ]:
```
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
```

[ ]:
```
df.describe(include='all')
```

[ ]:
```
              User_ID Product_ID  Gender     Age     Occupation City_Category  \
count   5.500680e+05     550068  550068  550068  550068.000000         550068
unique           NaN       3631       2       7            NaN              3
top              NaN  P00265242       M   26-35            NaN              B
freq             NaN       1880  414259  219587            NaN         231173
mean    1.003029e+06        NaN     NaN     NaN       8.076707            NaN
std     1.727592e+03        NaN     NaN     NaN       6.522660            NaN
min     1.000001e+06        NaN     NaN     NaN       0.000000            NaN
25%     1.001516e+06        NaN     NaN     NaN       2.000000            NaN
50%     1.003077e+06        NaN     NaN     NaN       7.000000            NaN
75%     1.004478e+06        NaN     NaN     NaN      14.000000            NaN
max     1.006040e+06        NaN     NaN     NaN      20.000000            NaN

       Stay_In_Current_City_Years  Marital_Status  Product_Category  \
count                      550068   550068.000000     550068.000000
unique                          5             NaN               NaN
top                             1             NaN               NaN
freq                       193821             NaN               NaN
mean                          NaN        0.409653          5.404270
std                           NaN        0.491770          3.936211
min                           NaN        0.000000          1.000000
25%                           NaN        0.000000          1.000000
50%                           NaN        0.000000          5.000000
75%                           NaN        1.000000          8.000000
max                           NaN        1.000000         20.000000

             Purchase
count   550068.000000
unique            NaN
top               NaN
freq              NaN
mean      9263.968713
std       5023.065394
min         12.000000
25%       5823.000000
50%       8047.000000
75%      12054.000000
max      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
```

[ ]:
```
df.describe(include='number')
```

[ ]:
```
            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
```

[ ]:
```
df.columns
```

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

[ ]:
```
type(df)
```

[ ]:
```
pandas.core.frame.DataFrame
```

[ ]:
```
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["Product_Category"].unique()  #Masked data
```

[ ]:
```
array([ 3,  1, 12,  8,  5,  4,  2,  6, 14, 11, 13, 15,  7, 16, 18, 10, 17,
        9, 20, 19])
```

[ ]:
```
df["Occupation"].unique()   #Masked data
```

[ ]:
```
array([10, 16, 15,  7, 20,  9,  1, 12, 17,  0,  3,  4, 11,  8, 19,  2, 18,
        5, 14, 13,  6])
```

[ ]:
```
df.isna().sum()  #There are no null Values
```

[ ]:
```
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
```

[ ]:
```
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
```

[ ]:
```
len(df)
```

[ ]:
```
550068
```

[ ]:
```
df.isnull().sum()/len(df)
```

```
User_ID                       0.0
Product_ID                    0.0
Gender                        0.0
Age                           0.0
Occupation                    0.0
City_Category                 0.0
Stay_In_Current_City_Years    0.0
Marital_Status                0.0
Product_Category              0.0
Purchase                      0.0
dtype: float64
```

```
df.columns
```

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

```
df["User_ID"].nunique()   #Total 5891 unique customers
```

```
5891
```

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

```
array(['P00069042', 'P00248942', 'P00087842', …, 'P00370293',
       'P00371644', 'P00370853'], dtype=object)
```

```
df["Product_ID"].nunique()  #Total 3631 different products got sold out in one␣
↪day
```

```
3631
```

```
df["Product_ID"].value_counts().sort_values(ascending=False)
```

```
P00265242    1880
P00025442    1615
P00110742    1612
P00112142    1562
P00057642    1470
             …
P00335642       1
P00341542       1
P00077242       1
P00315142       1
P00066342       1
Name: Product_ID, Length: 3631, dtype: int64
```

```
[ ]: df["Occupation"].nunique()  #there are 21 occupations.
```

```
[ ]: 21
```

```
[ ]: df["Occupation"].value_counts().sort_values(ascending=False) # 4,0,7,1 are the␣
     ↪occupations the customers have are more likely to shop in walmart than any␣
     ↪other customers
```

```
[ ]: 4     72308
     0     69638
     7     59133
     1     47426
     17    40043
     20    33562
     12    31179
     14    27309
     2     26588
     16    25371
     6     20355
     3     17650
     10    12930
     5     12177
     15    12165
     11    11586
     19     8461
     13     7728
     18     6622
     9      6291
     8      1546
     Name: Occupation, dtype: int64
```

```
[ ]: df["City_Category"].unique()    # there are 3 different city categories
```

```
[ ]: array(['A', 'C', 'B'], dtype=object)
```

```
[ ]: df["City_Category"].nunique()
```

```
[ ]: 3
```

```
[ ]: df["City_Category"].value_counts().sort_values(ascending=False)  # We have more␣
     ↪customers from the city 'B'
```

```
[ ]: B    231173
     C    171175
     A    147720
     Name: City_Category, dtype: int64
```

```
[ ]: df["City_Category"].value_counts(normalize=True)  # 42% customers are from B␣
      ↪city,31% are from C city,27% are from A city
```

```
[ ]: B    0.420263
     C    0.311189
     A    0.268549
     Name: City_Category, dtype: float64
```

```
[ ]: df["Stay_In_Current_City_Years"].unique()
```

```
[ ]: array(['2', '4+', '3', '1', '0'], dtype=object)
```

```
[ ]: df["Stay_In_Current_City_Years"].value_counts(normalize=True)  # 35% customers␣
      ↪are staying for 1 year,18% customers are staying for 2 years,17% customers␣
      ↪are staying for 3 years.
```

```
[ ]: 1     0.352358
     2     0.185137
     3     0.173224
     4+    0.154028
     0     0.135252
     Name: Stay_In_Current_City_Years, dtype: float64
```

```
[ ]: df["Product_Category"].nunique()   # there are total 20 product categories
```

```
[ ]: 20
```

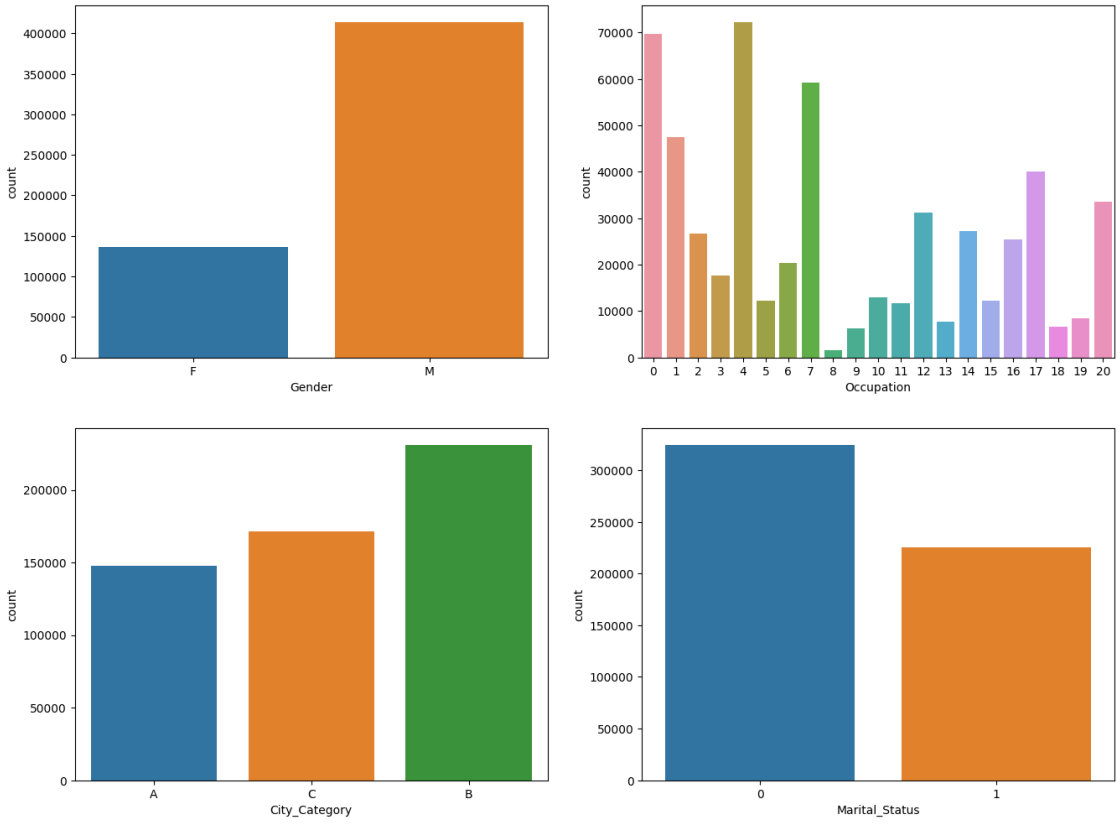
```
[ ]: df["Product_Category"].value_counts().sort_values(ascending=False)  #Customers␣
      ↪are buying more from the product categories 5,1,8,11
```

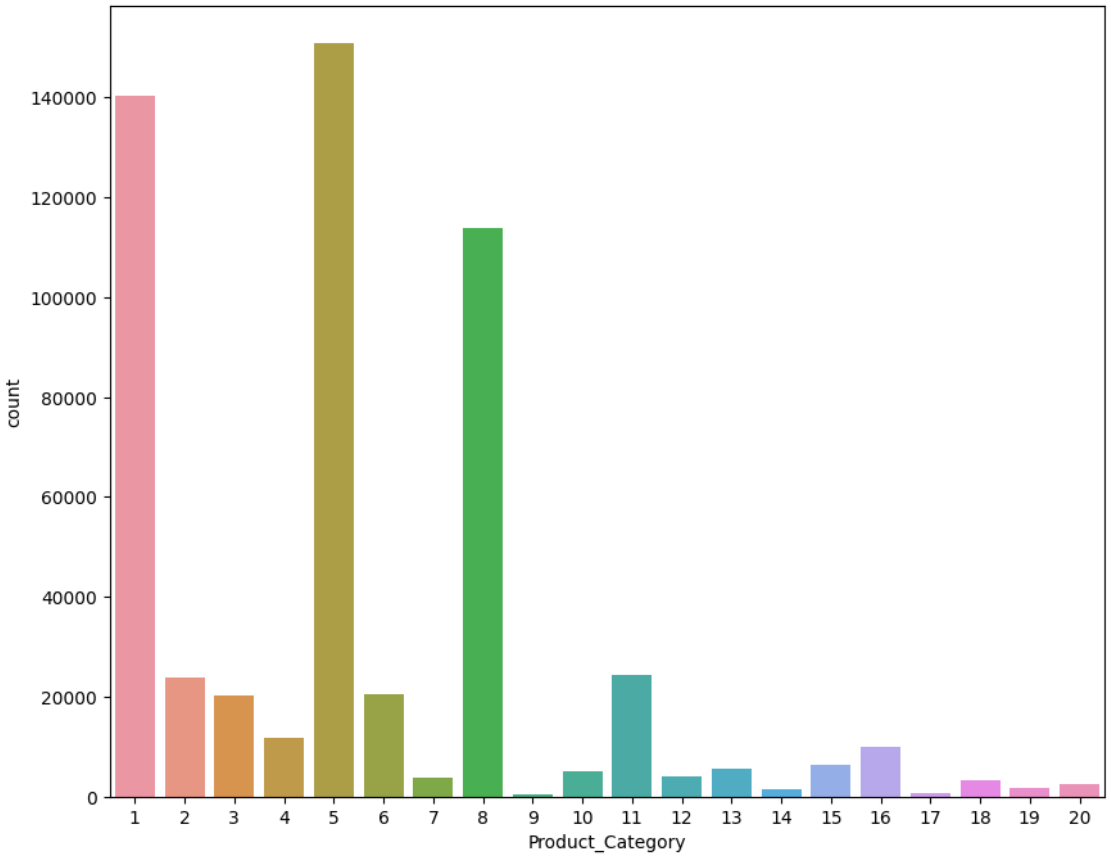
```
[ ]: 5     150933
     1     140378
     8     113925
     11     24287
     2      23864
     6      20466
     3      20213
     4      11753
     16      9828
     15      6290
     13      5549
     10      5125
     12      3947
     7       3721
     18      3125
     20      2550
     19      1603
```

```
14      1523
17       578
9        410
Name: Product_Category, dtype: int64
```

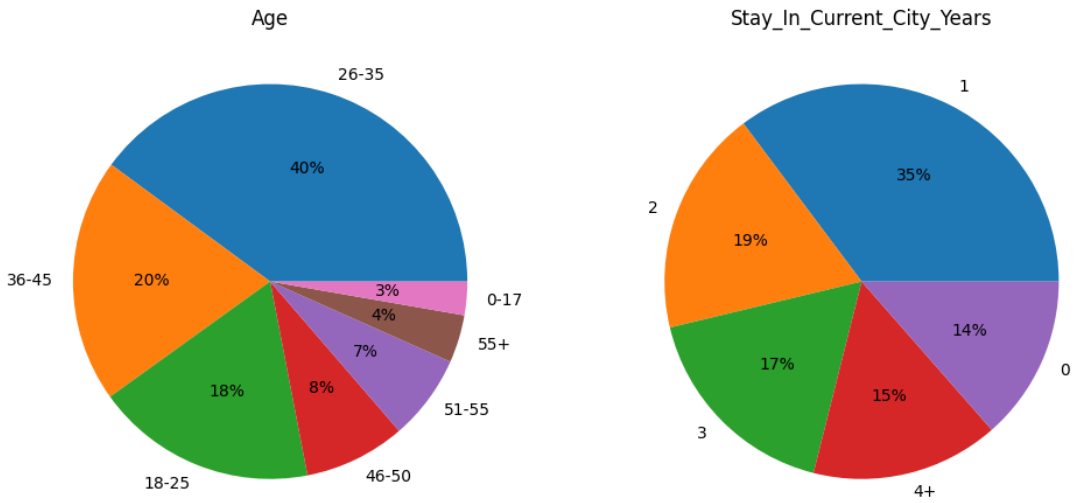
Uni-Variate Analysis

[ ]:
```
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()
```





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

data = df['Age'].value_counts(normalize=True)*100
axs[0].pie(x=data.values, labels=data.index,autopct='%.0f%%')
axs[0].set_title("Age")

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

axs[1].pie(x=data.values, labels=data.index,autopct='%.0f%%')
axs[1].set_title("Stay_In_Current_City_Years")


plt.show()
```



- 35% customers are staying for 1 year,19% customers are staying for 2 years,17% customers are staying for 3 years in the current city.
- There are total 20 product categories.
- Customers are buying more from the product categories 5,1,8,11.
- We have more male customers.
- We have more cusomters from the unmarried than married.
- We have more customers from the age group 26-35 and then 36-45.

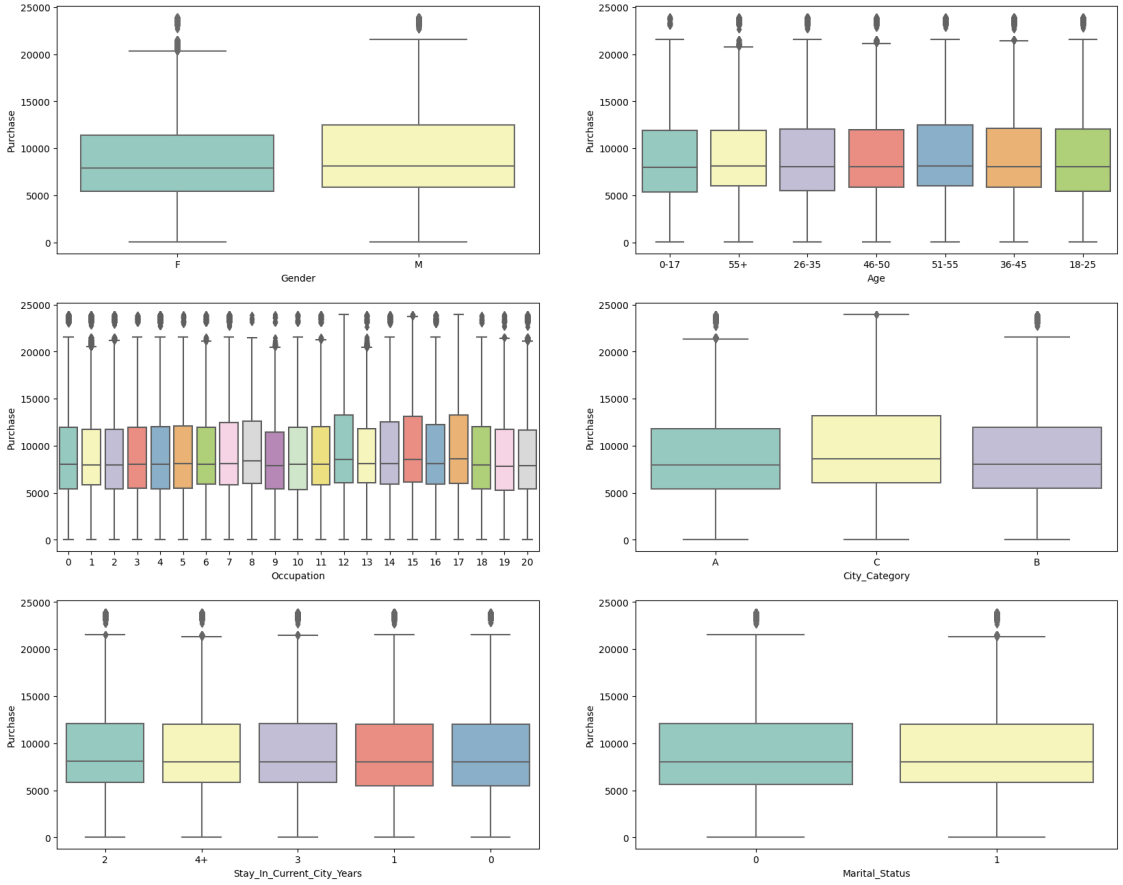
[ ]:
```
data = df['Age'].value_counts(normalize=True)*100
data
```

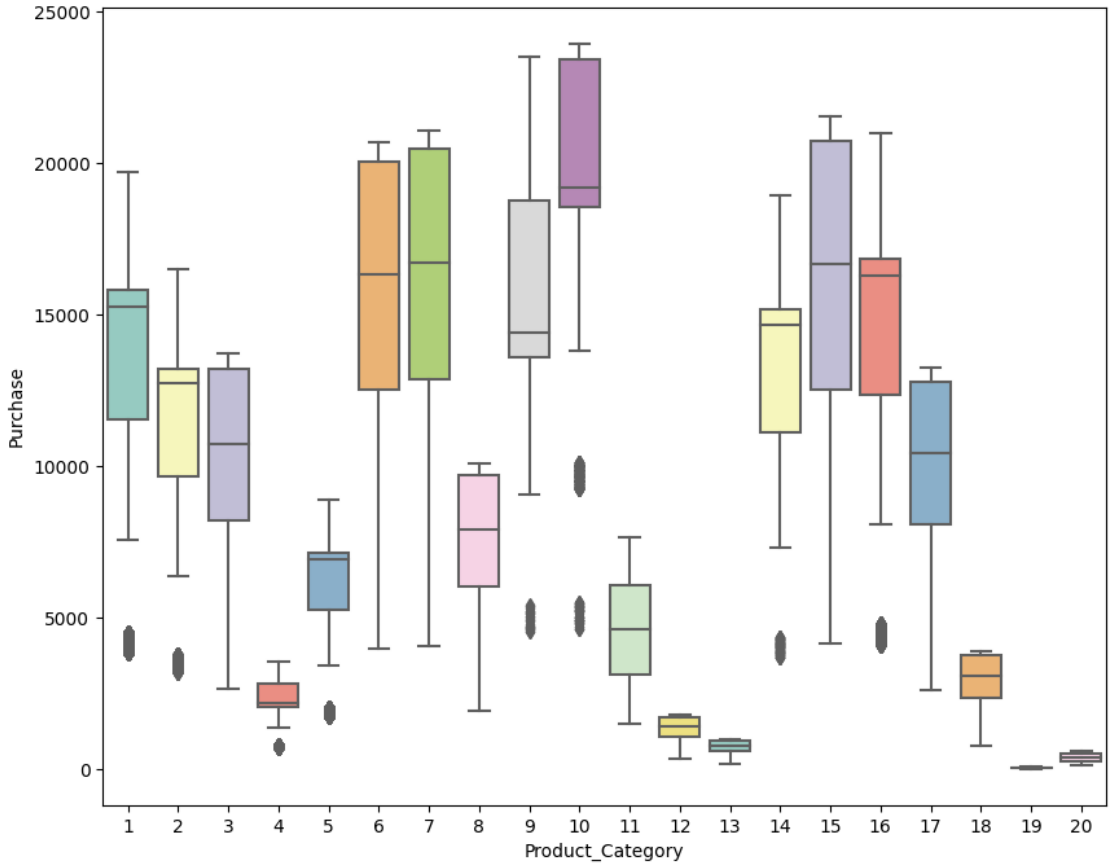
[ ]:
```
26-35    39.919974
36-45    19.999891
18-25    18.117760
46-50     8.308246
51-55     6.999316
55+       3.909335
0-17      2.745479
Name: Age, dtype: float64
```

Bi-Variate Analysis

[ ]:
```
columns = ['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=columns[count], ax=axs[row, col],␣
 ↪palette='Set3')
        #axs[row,col].set_title(f"Purchase vs {columns[count]}", pad=12,␣
 ↪fontsize=13)
        count += 1
plt.show()

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



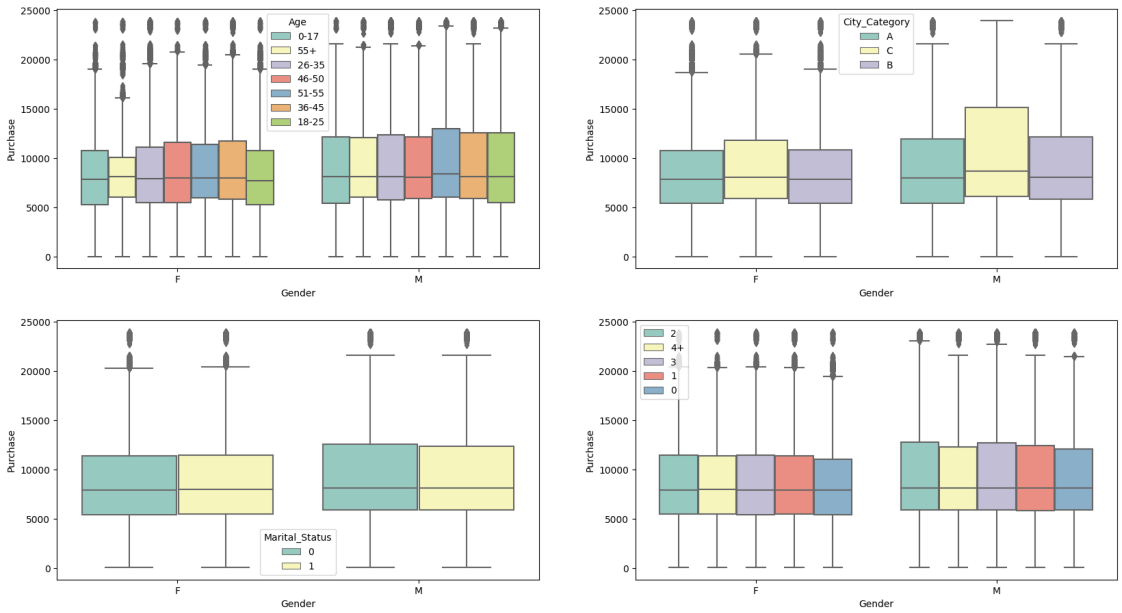


- We got more revenue from 6,7,10,15,16 product_categories.
- We are getting almost same revenue from the people who are staying in the current city irrespective the time they have been staying in the current city.
- We are geting more revenue from the unmarried people
- People are spending more money from the c city categery.
- People from the occupationes 8,12,15,17 are making more purchases.
- Males are making more bill than females.

Multi-Variate 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()
```



- In females between the age group 46-50 are making more revenue and in males between the age group 51-55 are making more revenue
- From c city category we got more revenue.

Finding Mean and CI for Gender(M,F)

[ ]:

[ ]:
```
df["Gender"].value_counts()
```

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

[ ]:
```
df["Gender"].unique()
```

[ ]:
```
array(['F', 'M'], dtype=object)
```

[ ]:
```
df["Gender"].nunique()
```

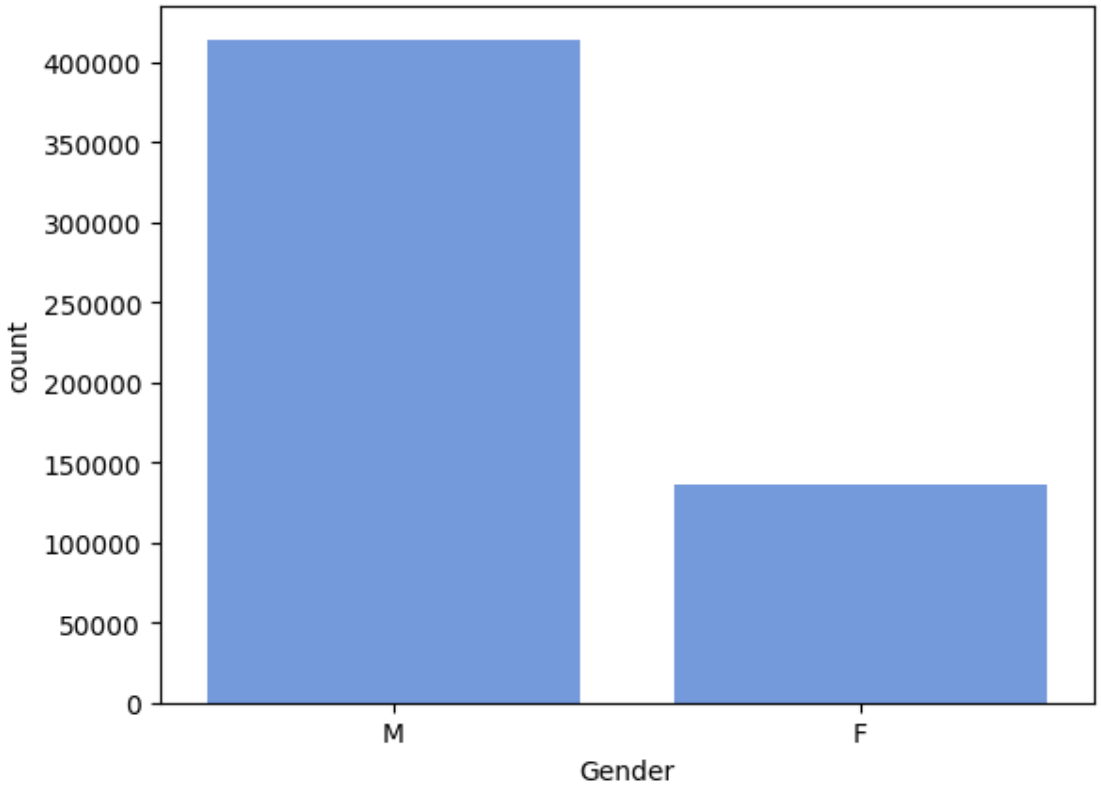
[ ]: 2

[ ]:
```
df["Gender"].value_counts(normalize=True)  #We have 75.3% of the male customers␣
 ↪and 24.6% percent of female customers
```

[ ]:
```
M    0.753105
F    0.246895
Name: Gender, dtype: float64
```

[ ]:
```
sns.countplot(x = 'Gender', data = df, order=df['Gender'].value_counts().index,␣
 ↪color='cornflowerblue')
plt.xticks(rotation=0)
```

[ ]:
```
(array([0, 1]), [Text(0, 0, 'M'), Text(1, 0, 'F')])
```

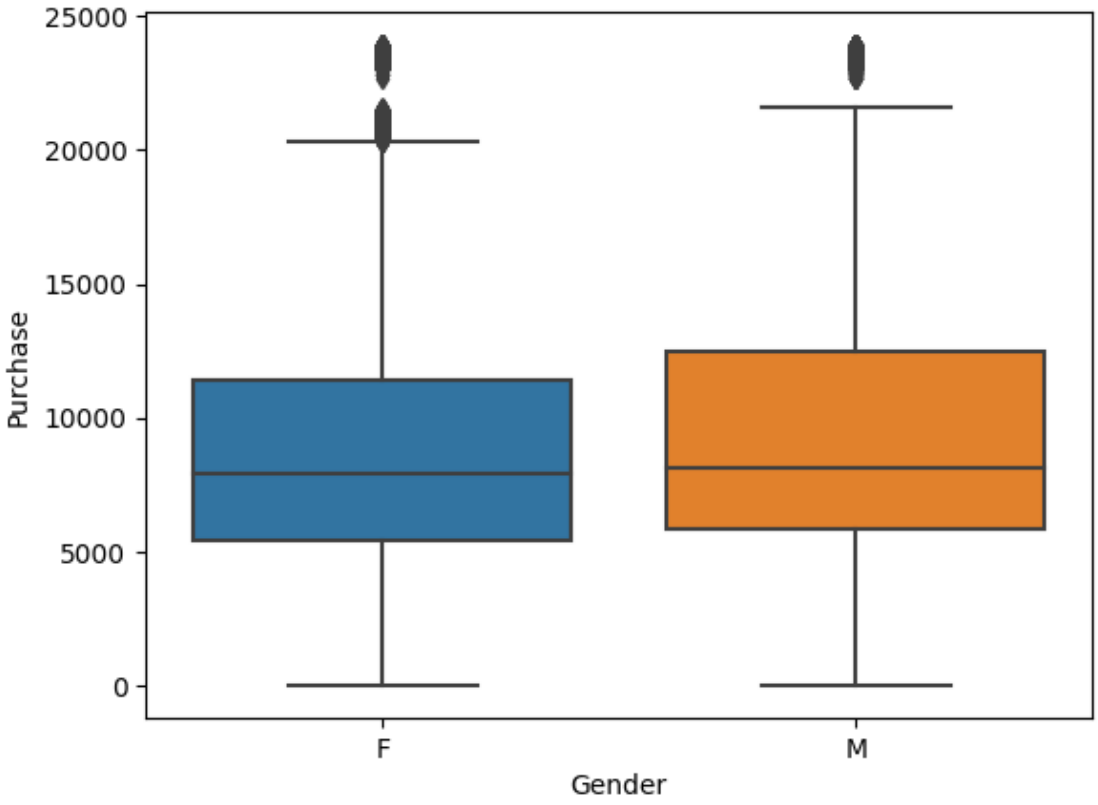


[ ]:
```
df.groupby(["Gender"])["User_ID"].nunique()
```

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

[ ]:
```
sns.boxplot(x = "Gender", y = "Purchase", data = df)  #Male customers make the␣
↪revenue more for the company than the female customers.
```

[ ]:
```
<Axes: xlabel='Gender', ylabel='Purchase'>
```



[ ]:
```
df.groupby(["Gender"])["Purchase"].describe()
```

[ ]:

Gender	count	mean	std	min	25%	50%	75%	max
F	135809.0	8734.565765	4767.233289	12.0	5433.0	7914.0	11400.0	23959.0
M	414259.0	9437.526040	5092.186210	12.0	5863.0	8098.0	12454.0	23961.0

[ ]:
```
sample_df=df.sample(300)
sample_df.groupby(["Gender"])["Purchase"].describe()
```

[ ]:
```
         count         mean          std    min     25%     50%      75%  \
Gender
F         77.0  8784.285714  4377.503599  380.0  6063.0  7971.0   9927.0
M        223.0  8904.394619  5305.559078  363.0  5313.5  7918.0  12109.0

            max
Gender
F       20603.0
M       23123.0
```

[ ]:
```
sample_df=df.sample(300)
sample_df.groupby(["Gender"])["Purchase"].describe()
```

[ ]:
```
         count         mean          std    min      25%     50%       75%  \
Gender
F         72.0  9548.097222  4462.088014   48.0  7051.75  8848.5  12018.50
M        228.0  9517.236842  5553.565638   38.0  5307.00  8603.5  13601.75

            max
Gender
F       19559.0
M       23766.0
```

Taking the sample size of 300

[ ]:
```
sample_size = 300
iterations = 1000
male_sample_df = [ df[df["Gender"] == "M"].sample(sample_size, replace =␣
 ↪True)["Purchase"].mean()  for i in range(iterations)]
```

[ ]:
```
sample_size = 300
iterations = 1000
female_sample_df = [ df[df["Gender"] == "F"].sample(sample_size, replace =␣
 ↪True)["Purchase"].mean()  for i in range(iterations)]
```

[ ]:
```
np.mean(male_sample_df)
```

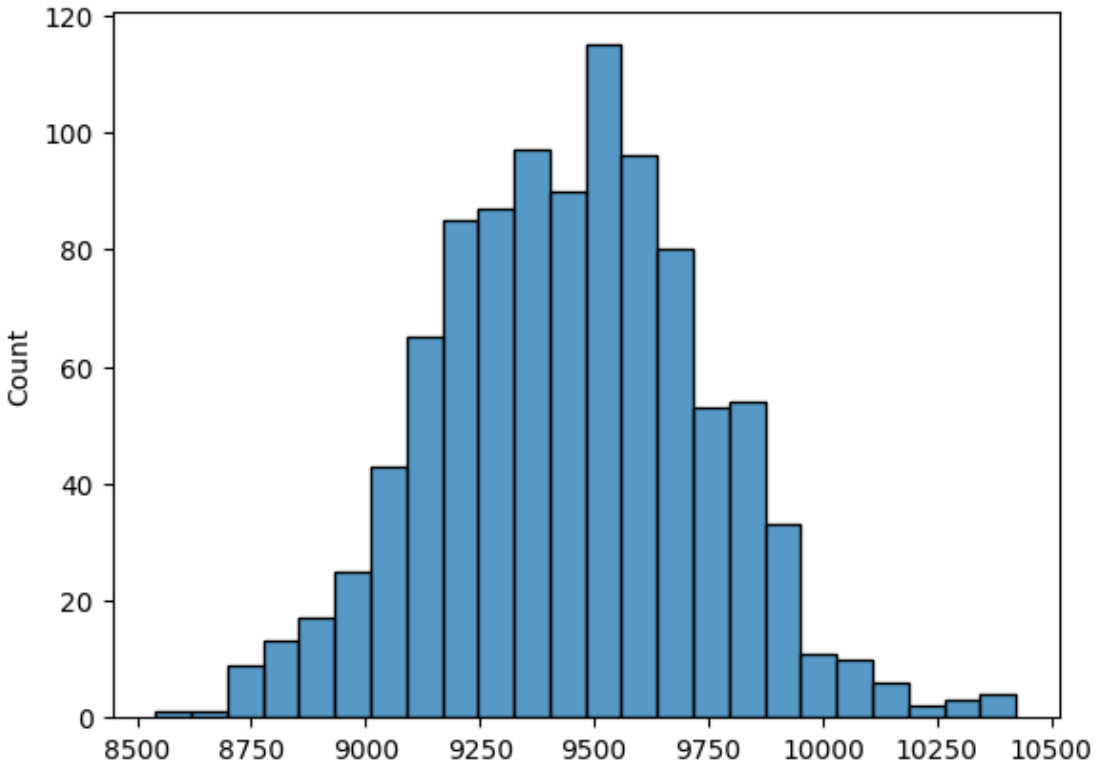
[ ]:
```
9447.26861
```

[ ]:
```
np.mean(female_sample_df)
```

[ ]:
```
8727.179536666665
```

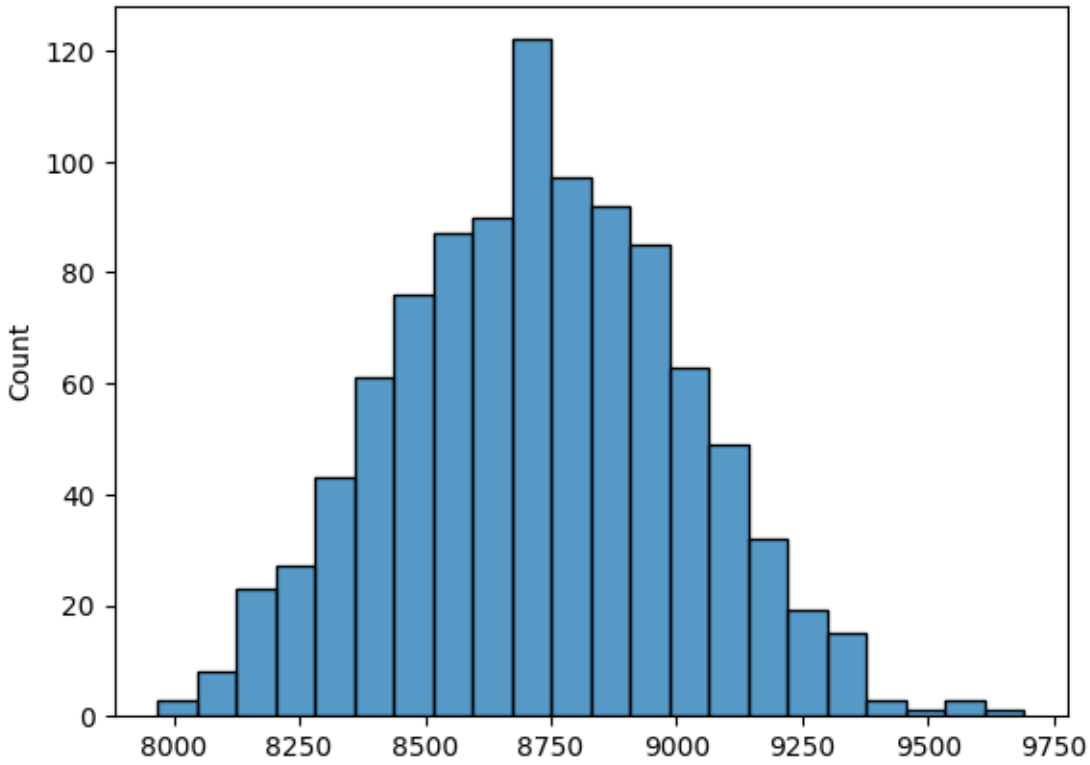
[ ]:
```
sns.histplot(male_sample_df)
```

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



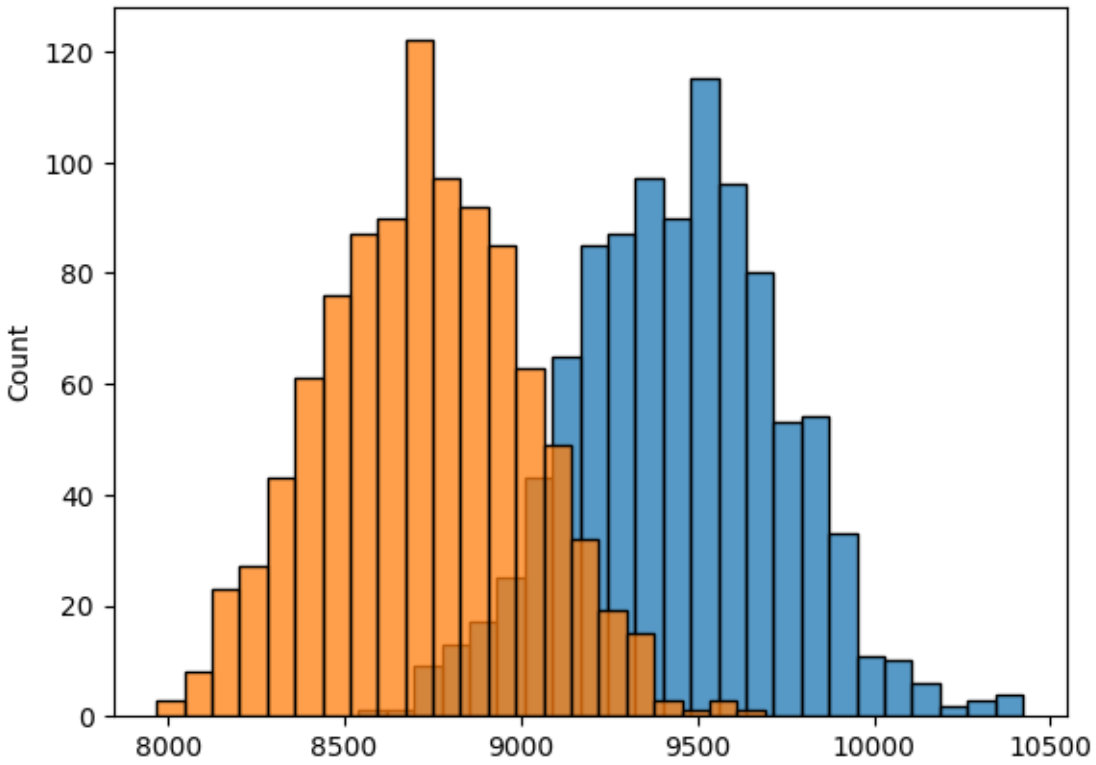
[ ]:
```
sns.histplot(female_sample_df)
```

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



[ ]:
```
sns.histplot(male_sample_df,label="male")
sns.histplot(female_sample_df,label="female")
```

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




```
[ ]: male_confidence_interval = np.percentile(male_sample_df, [2.5 , 97.5])
     male_confidence_interval
```

```
[ ]: array([ 8877.20425, 10010.25925])
```

```
[ ]: female_confidence_interval = np.percentile(female_sample_df, [2.5 , 97.5])
     female_confidence_interval
```

```
[ ]: array([8173.87275   , 9289.24733333])
```

Mean and CI for Gender:

Sample size=300

Mean of the sample means of males:9447.26861

Mean of the sample means of females:8727.179536666665

Male_CI: [8877.20425, 10010.25925]

Female_CI: [8173.87275 , 9289.24733333] with 95% confidence.

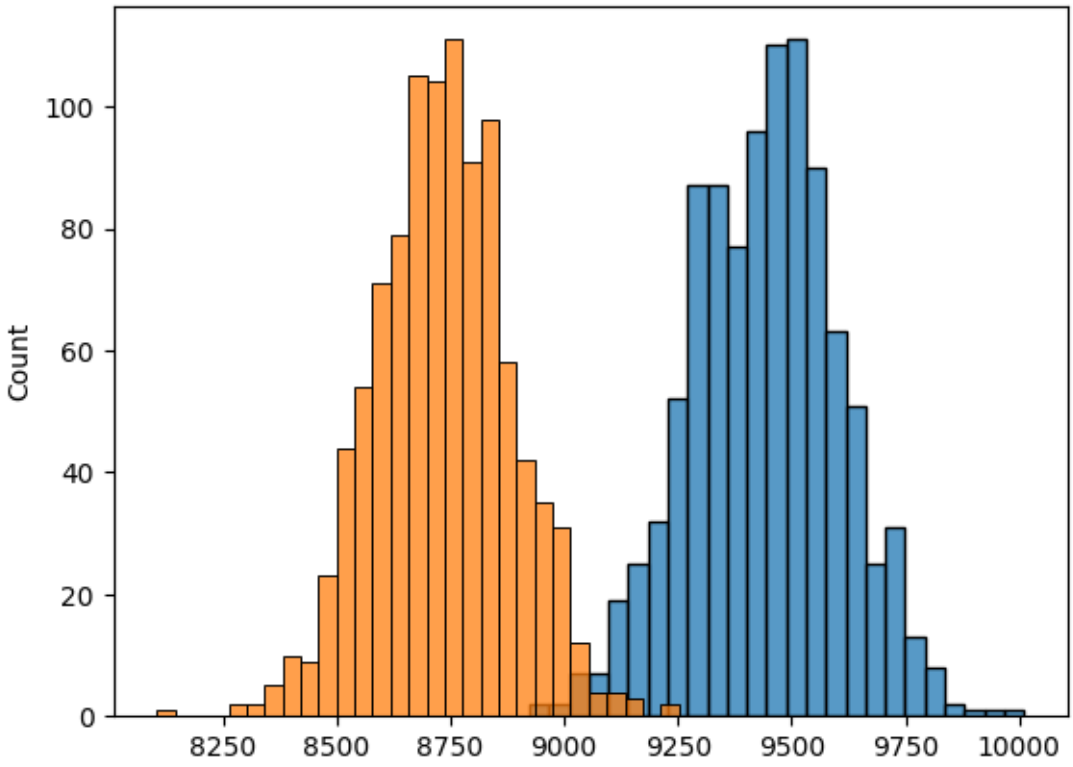
Increasing the sample size to 1000

[ ]:
```
sample_size = 1000
iterations = 1000
male_sample2_df = [ df[df["Gender"] == "M"].sample(sample_size, replace =␣
 ↪True)["Purchase"].mean()  for i in range(iterations)]
```

[ ]:
```
sample_size = 1000
iterations = 1000
female_sample2_df = [ df[df["Gender"] == "F"].sample(sample_size, replace =␣
 ↪True)["Purchase"].mean()  for i in range(iterations)]
```

[ ]:
```
sns.histplot(male_sample2_df,label='male')
sns.histplot(female_sample2_df,label='female')
```

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



[ ]:
```
male_confidence_interval2 = np.percentile(male_sample2_df, [2.5 , 97.5])
male_confidence_interval2
```

[ ]:
```
array([9116.263275, 9750.865575])
```

[ ]:
```
female_confidence_interval2 = np.percentile(female_sample2_df, [2.5 , 97.5])
female_confidence_interval2
```

[ ]:
```
array([8447.346325, 9012.0006  ])
```

when we take the sample size is very low,the confidence intervals are overlapping.When we take the sample size 1000,Overlapping got decreased.

Finding mean and CI for Marital_status

[ ]:
```
df.columns
```

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

[ ]:
```
Marital_df=df.groupby(["User_ID","Marital_Status"])["Purchase"].sum()
Marital_df=Marital_df.reset_index()
Marital_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]

[ ]:
```
Marital_df["Marital_Status"].value_counts()  #We have more number of single␣
↪customers when compared to married customers
```

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

[ ]:
```
Marital_df["Marital_Status"].value_counts(normalize=True)  #We have 60% of␣
↪singles customers and 40% of married customers
```

[ ]:
```
0    0.580037
1    0.419963
Name: Marital_Status, dtype: float64
```

[ ]:
```
df["User_ID"].nunique()  #total number of unique users
```

[ ]: 5891

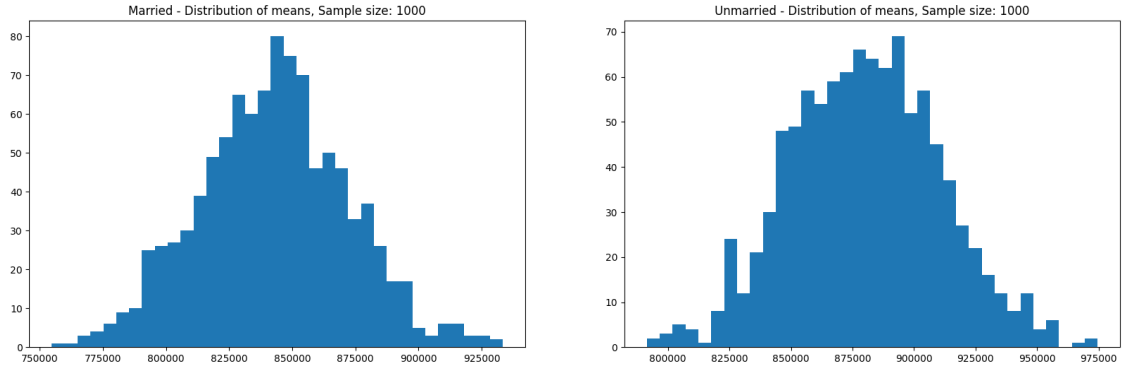
```
[ ]: married_mean=[Marital_df[Marital_df["Marital_Status"]==1].
      ↪sample(1000,replace=True)["Purchase"].mean() for i in range(1000)]
     np.mean(married_mean) #Married customers purchase mean is 842681.378939
```

[ ]: 842681.378939

```
[ ]: singles_mean=[Marital_df[Marital_df["Marital_Status"]==0].
      ↪sample(1000,replace=True)["Purchase"].mean() for i in range(1000)]
     np.mean(singles_mean)  #Single customers purchase mean is 881224.383053
```

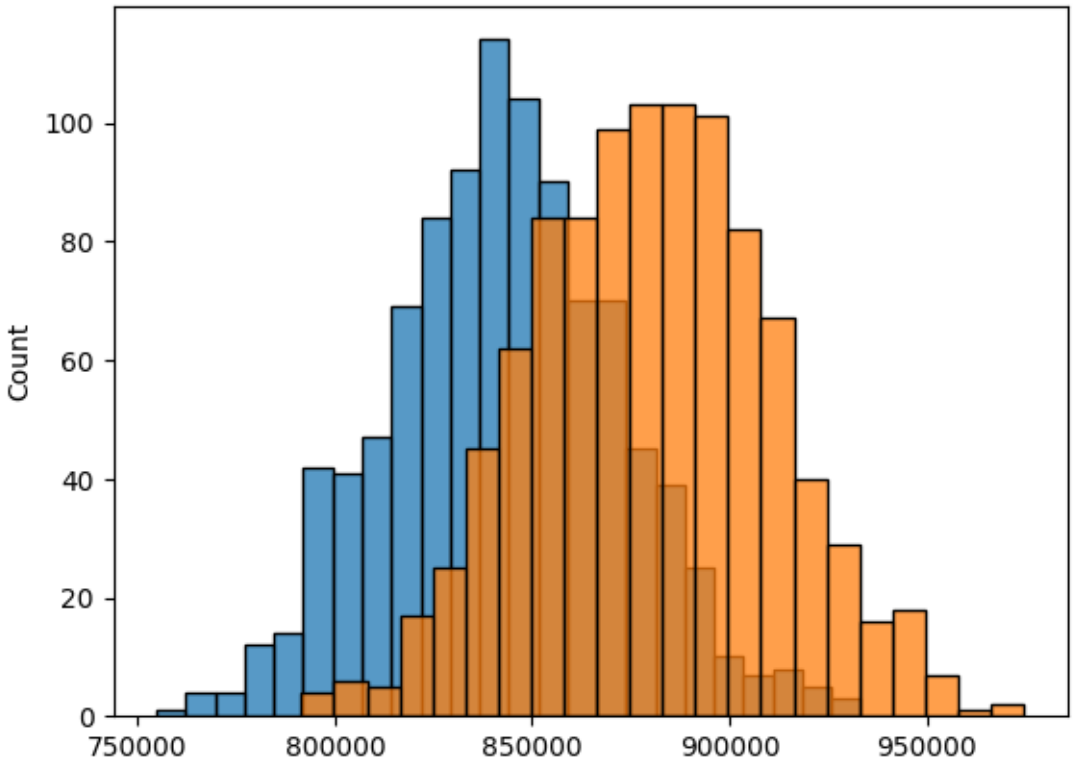
[ ]: 881224.383053

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

     axis[0].hist(married_mean, bins=35)
     axis[1].hist(singles_mean, bins=35)
     axis[0].set_title("Married - Distribution of means, Sample size: 1000")
     axis[1].set_title("Unmarried - Distribution of means, Sample size: 1000")

     plt.show()
```




```
[ ]: sns.histplot(married_mean)
     sns.histplot(singles_mean)
     plt.legend
```

[ ]: <function matplotlib.pyplot.legend(*args, **kwargs)>



[ ]:
```
married_confidence_interval = np.percentile(married_mean, [2.5 , 97.5])
married_confidence_interval   #95% percent of the times purchase mean interval␣
 ↪lies between [785767.584425, 900315.960525] for married customers
```

[ ]:
```
array([785767.584425, 900315.960525])
```

[ ]:
```
singles_confidence_interval = np.percentile(singles_mean, [2.5 , 97.5])
singles_confidence_interval  #95% percent of the times purchase mean interval␣
 ↪lies between [823664.0091 , 942652.68145] for unmmarried customers
```

[ ]:
```
array([823664.0091 , 942652.68145])
```

Mean and CI for Marital status:

Sample size=1000

95% percent of the times purchase mean interval lies between [785767.584425, 900315.960525] for married customers

95% percent of the times purchase mean interval lies between [823664.0091 , 942652.68145] for unmmarried customers

Married customers purchase mean is 842681.378939

Single customers purchase mean is 881224.383053

[ ]:
```
df.columns
```

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

Finding Mean and CI for 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 x 3 columns]
```

[ ]:
```
Age_df["Age"].unique()
```

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

[ ]:
```
Age_df["Age"].value_counts()   #We have more customers from age group 26-35
```

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

[ ]:
```
Age_df["Age"].value_counts(normalize=True)
```

[ ]:
```
26-35    0.348498
36-45    0.198099
18-25    0.181463
46-50    0.090137
51-55    0.081650
55+      0.063147
0-17     0.037006
Name: Age, dtype: float64
```

[ ]:
```
Age_df["Age"]=Age_df["Age"].astype("object")
Age_df["Purchase"]=Age_df["Purchase"].astype("object")
Age_df.dtypes
```

[ ]:
```
User_ID      int64
Age         object
Purchase    object
dtype: object
```

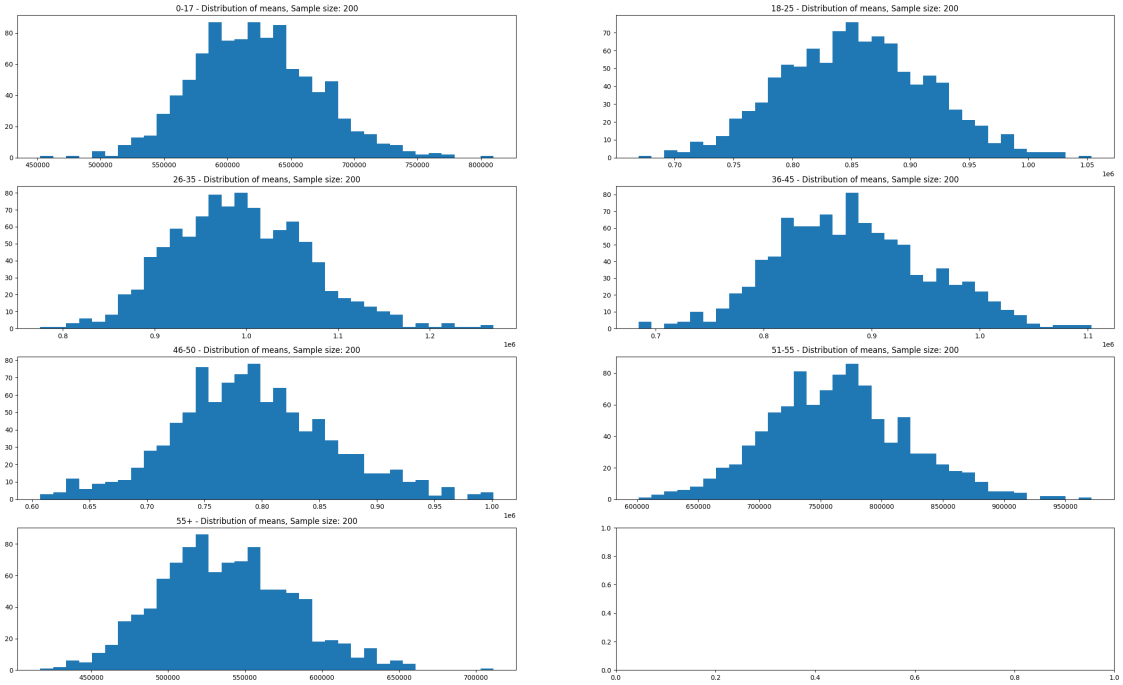
[ ]:
```
All_means={}
for i in ['0-17','18-25','26-35','36-45','46-50','51-55','55+']:
  All_means[i]=[Age_df[Age_df["Age"]==i].sample(200,replace=True)["Purchase"].
 ↪mean() for j in range(1000)]
```

[ ]:
```
for i in ['0-17','18-25','26-35','36-45','46-50','51-55','55+']:
  print("mean of sample means of age groups between" ,i, " is ", np.
 ↪mean(All_means[i])) #mean of purchase sum is more for the people who is in␣
 ↪the agre group of 26-35
```

```
mean of sample means of age groups between 0-17  is  621541.746485
mean of sample means of age groups between 18-25  is  853556.254885
mean of sample means of age groups between 26-35  is  992959.3818
mean of sample means of age groups between 36-45  is  882188.09254
mean of sample means of age groups between 46-50  is  790983.7316400001
mean of sample means of age groups between 51-55  is  763616.6394949999
mean of sample means of age groups between 55+  is  537298.3849249999
```

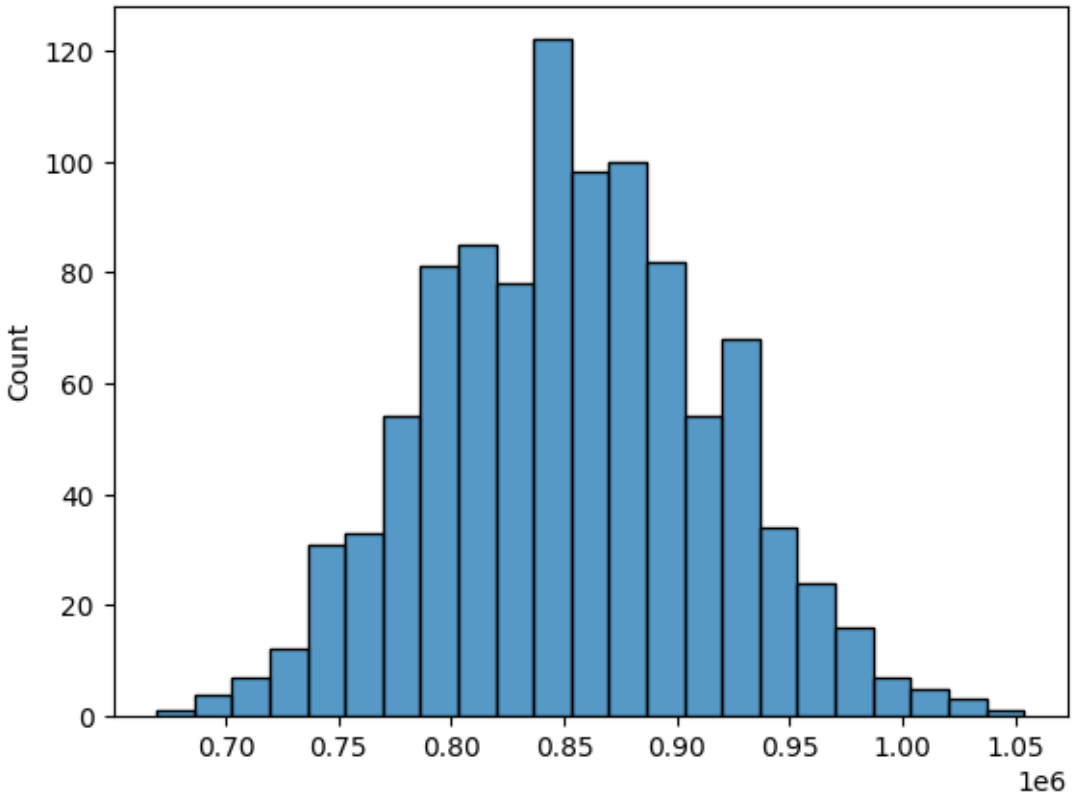
[ ]:
```
fig, axis = plt.subplots(nrows=4, ncols=2, figsize=(30, 18))

axis[0,0].hist(All_means['0-17'], bins=35)
axis[0,1].hist(All_means['18-25'], bins=35)
axis[1,0].hist(All_means['26-35'],bins=35)
axis[1,1].hist(All_means['36-45'],bins=35)
axis[2,0].hist(All_means['46-50'], bins=35)
axis[2,1].hist(All_means['51-55'], bins=35)
axis[3,0].hist(All_means['55+'], bins=35)
axis[0,0].set_title("0-17 - Distribution of means, Sample size: 200")
axis[0,1].set_title("18-25 - Distribution of means, Sample size: 200")
```

```
axis[1,0].set_title("26-35 - Distribution of means, Sample size: 200")
axis[1,1].set_title("36-45 - Distribution of means, Sample size: 200")
axis[2,0].set_title("46-50 - Distribution of means, Sample size: 200")
axis[2,1].set_title("51-55 - Distribution of means, Sample size: 200")
axis[3,0].set_title("55+ - Distribution of means, Sample size: 200")

plt.show()
```



[ ]:
```
sns.histplot(All_means['18-25'])
```

[ ]: <Axes: ylabel='Count'>



[ ]:
```
np.percentile(All_means['0-17'],[2.5,97.5])
```

[ ]:
```
array([530913.26    , 720979.819625])
```

[ ]:
```
print("mean purchase of customers with age group 55+ with 95% CI",np.
 ↪percentile(All_means['55+'],[2.5,97.5]))
print("mean purchase of customers with age group 18-25 with 95% CI",np.
 ↪percentile(All_means['18-25'],[2.5,97.5]))
print("mean purchase of customers with age group 26-35 with 95% CI",np.
 ↪percentile(All_means['26-35'],[2.5,97.5]))
print("mean purchase of customers with age group 36-45 with 95% CI",np.
 ↪percentile(All_means['36-45'],[2.5,97.5]))
print("mean purchase of customers with age group 46-50 with 95% CI",np.
 ↪percentile(All_means['46-50'],[2.5,97.5]))
print("mean purchase of customers with age group 51-55 with 95% CI",np.
 ↪percentile(All_means['51-55'],[2.5,97.5]))
```

```
mean purchase of customers with age group 55+ with 95% CI [459040.321375
628215.104875]
mean purchase of customers with age group 18-25 with 95% CI [736626.711
979450.047125]
```

```
mean purchase of customers with age group 26-35 with 95% CI [ 863230.307875
1150554.275375]
mean purchase of customers with age group 36-45 with 95% CI [ 757658.619375
1021616.731875]
mean purchase of customers with age group 46-50 with 95% CI [657028.662625
935704.366375]
mean purchase of customers with age group 51-55 with 95% CI [655572.472625
877550.170875]
```

#Insights:

We have total 5891 cutsomers on that Black friday.

The customers bought 3631 different types of products on that day.

Total number of occupations are 21.

4,0,7,1 are the occupations we have more customers from.

From B city we have more customers.

42% customers are from B city category,31% are from C city category,27% are from A city category.

35% customers are staying for 1 year,19% customers are staying for 2 years,17% customers are staying for 3 years in the current city.

There are total 20 product categories.

Customers are buying more from the product categories 5,1,8,11.

We have more male customers.

We have more cusomters from the unmarried than married.

We have 60% of singles customers and 40% of married customers.

We have more customers from the age group 26-35 and then 36-45.

We got more revenue from 6,7,10,15,16 product_categories.

We are getting almost same revenue from the people who are staying in the current city irrespective of the time they have been staying in the current city.

We are geting more revenue from the unmarried people

People are spending more money from the c city categery.

People from the occupationes 8,12,15,17 are making more purchases.

Males are making more bill than females.

In females between the age group 46-50 are making more revenue and in males between the age group 51-55 are making more revenue.

From c city category we got more revenue.

We have 75.3% of the male customers and 24.6% percent of female customers

Mean and CI for Gender:

Sample size=300

Mean of the sample means of males:9447.26861

Mean of the sample means of females:8727.179536666665

Male_CI: [8877.20425, 10010.25925]

Female_CI: [8173.87275 , 9289.24733333] with 95% confidence.

when we take the sample size is very low,the confidence intervals are overlapping.When we take the sample size 1000,Overlapping got decreased.

Mean and CI for Marital status:

Sample size=1000

95% percent of the times purchase mean interval lies between [785767.584425, 900315.960525] for married customers

95% percent of the times purchase mean interval lies between [823664.0091 , 942652.68145] for unmmarried customers

Married customers purchase mean is 842681.378939 Single customers purchase mean is 881224.383053

Mean and CI for Age:

Sample size=200

mean of sample means of age groups between 0-17 is 621541.746485

mean of sample means of age groups between 18-25 is 853556.254885

mean of sample means of age groups between 26-35 is 992959.3818

mean of sample means of age groups between 36-45 is 882188.09254

mean of sample means of age groups between 46-50 is 790983.7316400001

mean of sample means of age groups between 51-55 is 763616.6394949999

mean of sample means of age groups between 55+ is 537298.3849249999

mean purchase of customers with age group 55+ with 95% CI [459040.321375 628215.104875]

mean purchase of customers with age group 18-25 with 95% CI [736626.711 979450.047125]

mean purchase of customers with age group 26-35 with 95% CI [863230.307875 1150554.275375]

mean purchase of customers with age group 36-45 with 95% CI [757658.619375 1021616.731875]

mean purchase of customers with age group 46-50 with 95% CI [657028.662625 935704.366375]

mean purchase of customers with age group 51-55 with 95% CI [655572.472625 877550.170875]

#Recommendations:

From the Occupations 4,0,7,1 we got more customers and 8,12,15,17 are spending more money.So the company should concentrate on the needs of these customers and understand their lifestyle so that there are chances we get more purchases from these customers.

We have more purchases from men than women.So we have to try retaining male cusomters more.

We have purchases from the age group 18-50,so we need to target the needs of these customers more.

By our analysis the people from the c city able to spend much.So we need to keep different types of products in C city.So that we can get ore revenue.

Unmarried people tend to do more shopping than married people.So we need to attarct the people between the age group 20-35 more.

6,7,10,15,16 are product categories we got more revenue for.So we need to keep the more products like these in c city.Because the people from there able to keep more money on the things.

Customers are intersted in product categories like 5,1,8,11.We can try getting more of these types of products.

[ ]:

[ ]:

[ ]:

[ ]: