Name: Aloysius Lobo

Batch: Morning – DSML BeginnerFeb23

Case-Study: Walmart – Confidence Interval and CLT

IMPORTING BASIC LIBRARIES

import pandas as pd

import numpy as np

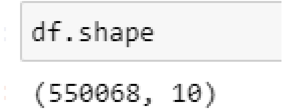
import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import norm, binom, ttest_ind, ttest_1sample

# Applying basic steps

1. Checking total rows and columns in the dataset

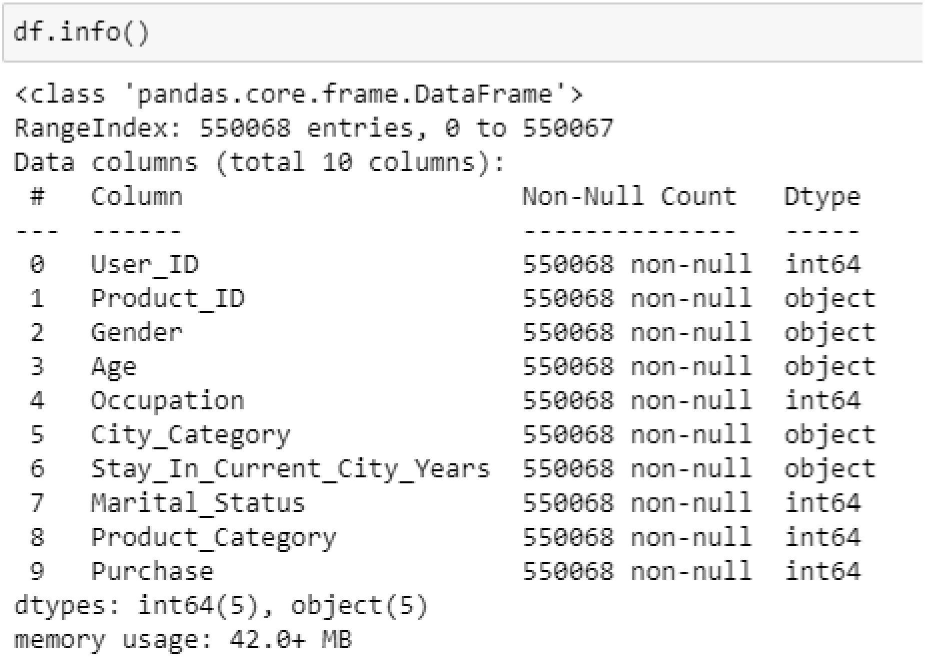


```
df.shape
```

```
(550068, 10)
```

- There are total **550068 rows** and **10 columns** in the entire dataset.

2. Acquiring dataset info like columns and their data type, total null-values in each column, etc.



```
df.info()
```

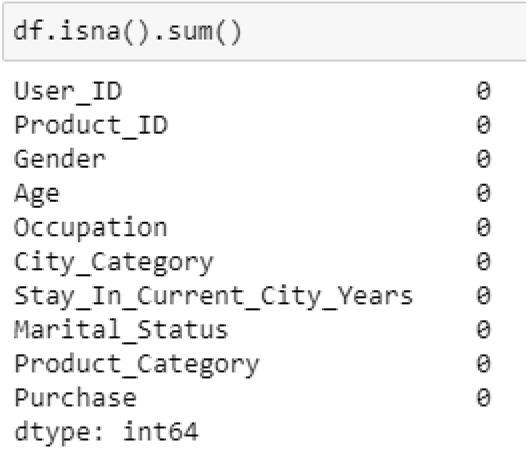
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype 
---  ------                      --------------   ----- 
 0   User_ID                     550068 non-null  int64 
 1   Product_ID                  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
```

Observations:

i. Product_ID, Gender, Age, City_Category and Stay_In_Current_City_Years have 'object' datatype i.e. they have categorical values. Remaining columns have int64 datatype i.e. they have numerical values.

ii. The dataset does not contain any null values.

Missing/Null values



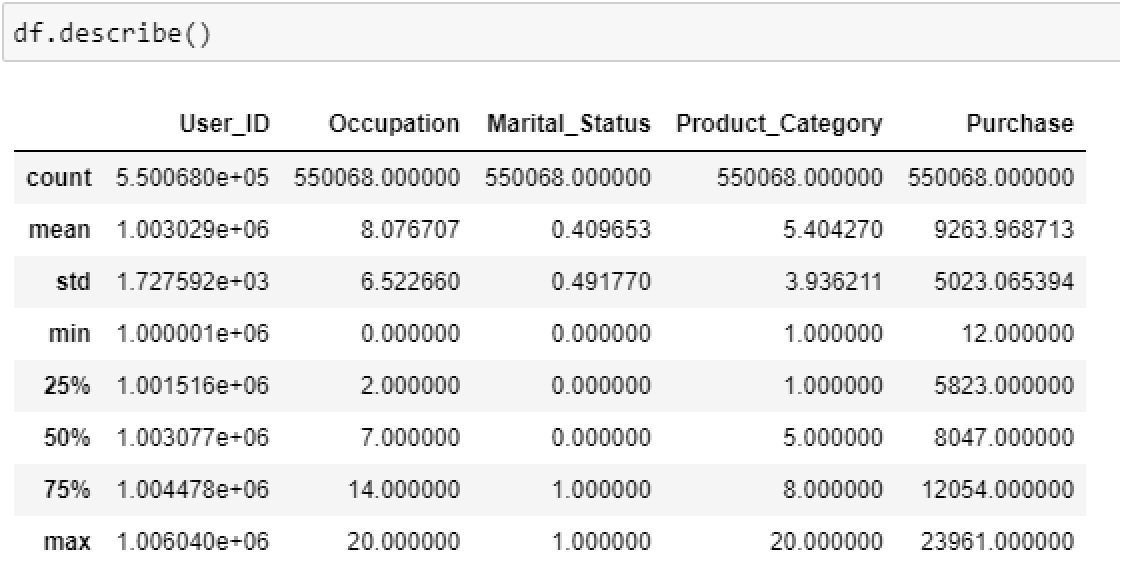
```
df.isna().sum()
```

```
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
```

Observation –

- No missing values found in the dataset.

# Getting statistical info about the dataset

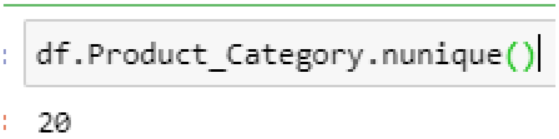


```
df.describe()
```

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

- Average amount of purchase done by a customer is $ 9263.96 where the min purchase is $ 12.00 and max purchase is $ 23961.00.

# Number of unique products in the dataset

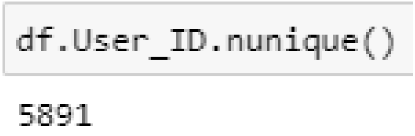


```
df.Product_Category.nunique()
```

```
20
```

There are 20 unique products in the dataset.

# Number of users in the dataset



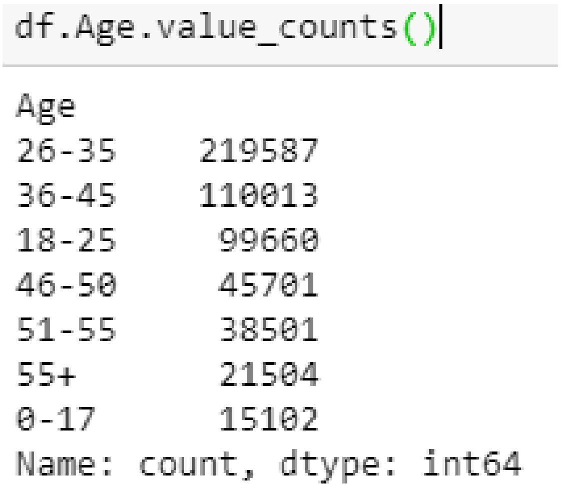
```
df.User_ID.nunique()
```

```
5891
```

There are 5891 unique users in the given Walmart dataset.

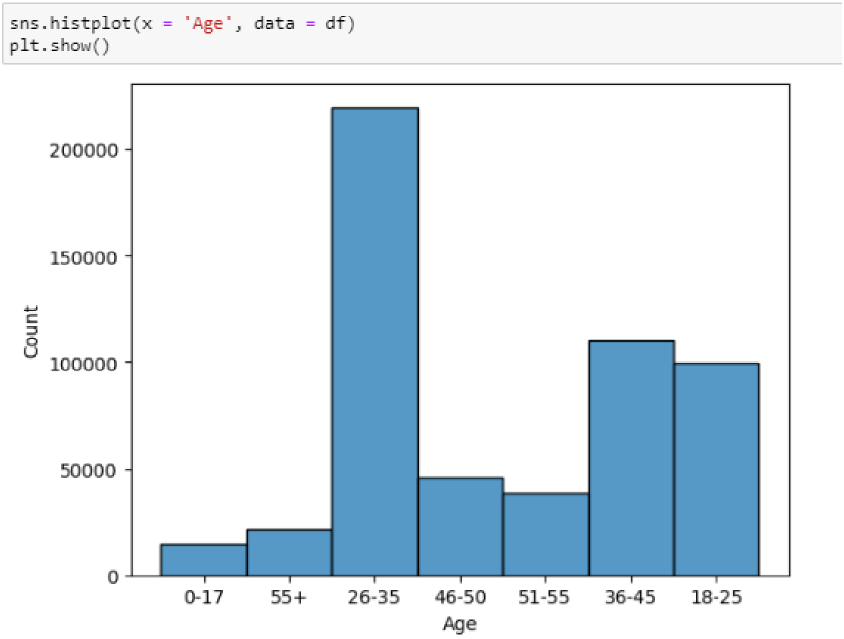
# Calculating value counts of every data

1. Age



```
df.Age.value_counts()
```

```
Age
26-35    219587
36-45    110013
18-25     99660
46-50     45701
51-55     38501
55+       21504
0-17      15102
Name: count, dtype: int64
```

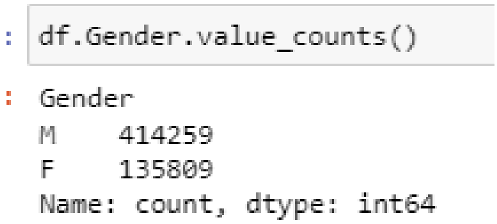


```
sns.histplot(x = 'Age', data = df)
plt.show()
```

Observations –

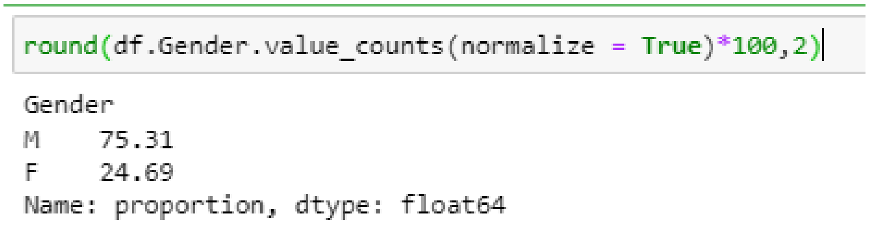
- Age group of 0-17 yrs are the least contributors to sales in Walmart as their total count is 15102 while max purchases are done by age group of 26-35 years.
-

2. Gender – *Customer purchases based on gender*



```
: df.Gender.value_counts()
```

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



```
round(df.Gender.value_counts(normalize = True)*100,2)
```

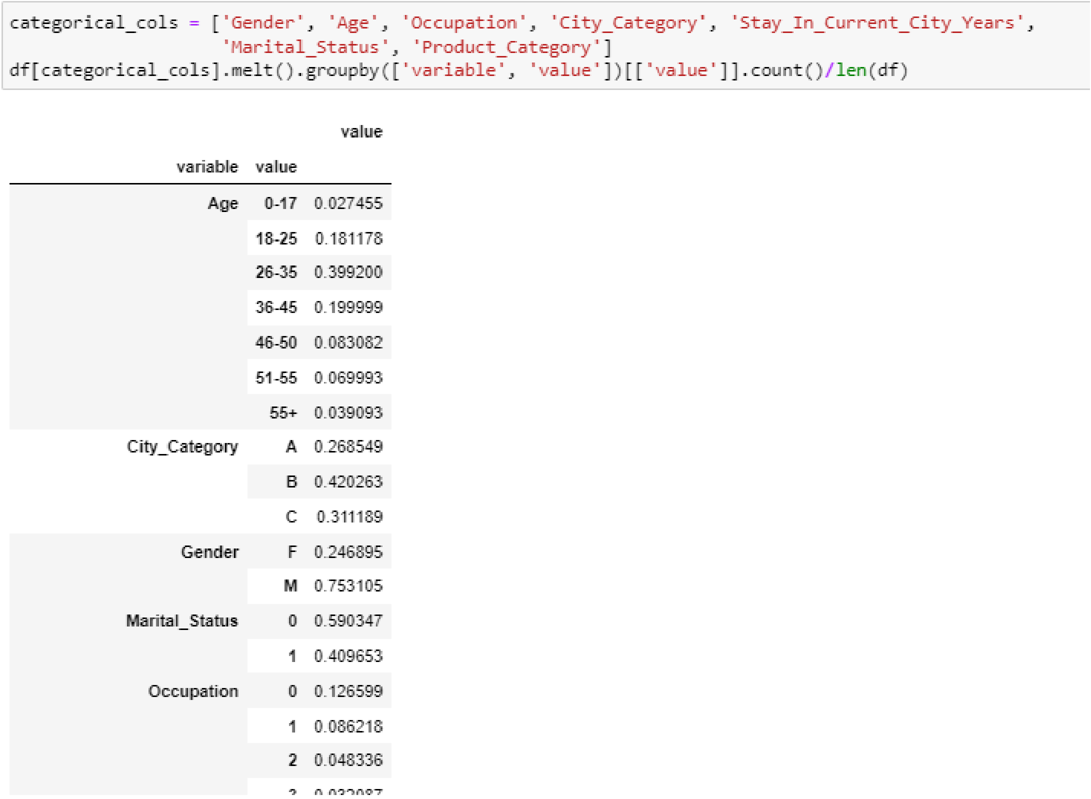
```
Gender
M    75.31
F    24.69
Name: proportion, dtype: float64
```

Observation –

Above analysis show that Male customers purchase more than female customers.

Around 75.31 % Males customers do purchasing in Walmart compared to Female which is 24.69% only.

3. Value_counts



```
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)
```

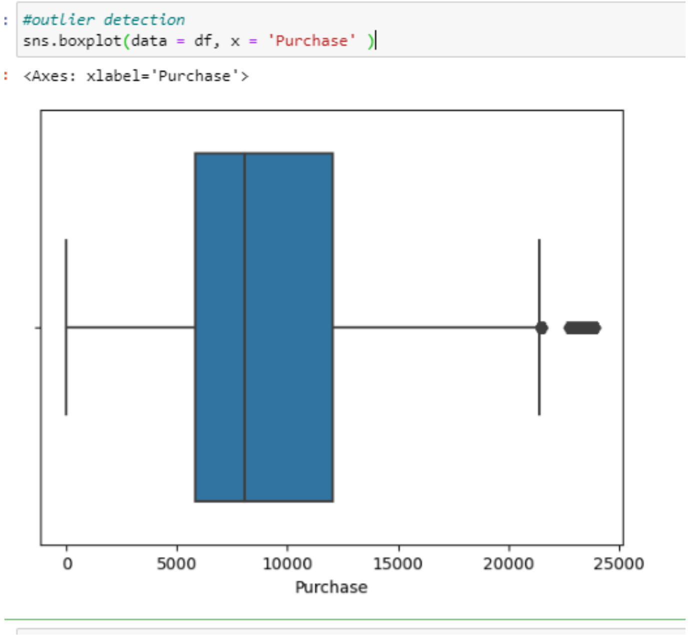
		value
variable	value	
Age	0-17	0.027455
	18-25	0.181178
	26-35	0.399200
	36-45	0.199999
	46-50	0.083082
	51-55	0.069993
	55+	0.039093
City_Category	A	0.268549
	B	0.420263
	C	0.311189
Gender	F	0.246895
	M	0.753105
Marital_Status	0	0.590347
	1	0.409653
Occupation	0	0.126599
	1	0.086218
	2	0.048336
	3	0.032087

Observations –

- ~18% users fall in the age-group 18-25.
- 40% users fall in the age-group 26-35
- 20% users fall in the age-group 36-45
- ~75% purchases are done by male customers
- ~25% purchases are done by female customers
- ~60% purchases are done by married customers
- ~40% purchases are done by unmarried customers
- There are total 20 product categories in the dataset.

# Outlier detection using boxplots

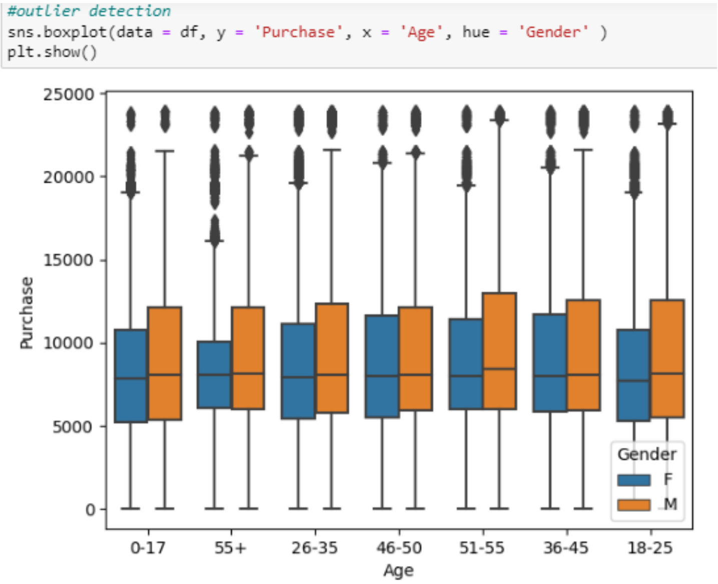
1. Purchase



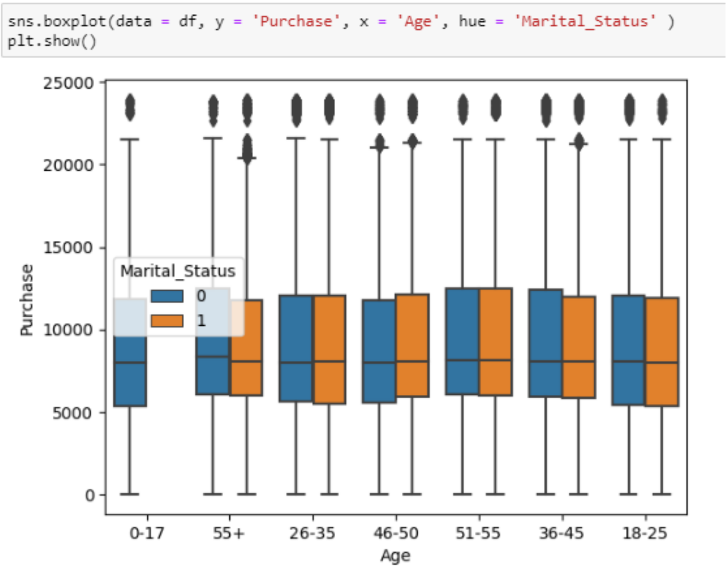
```
#outlier detection
sns.boxplot(data = df, x = 'Purchase' )
```

```
<Axes: xlabel='Purchase'>
```

2. **Purchase vs Age w.r.t Gender and Purchase vs Age w.r.t Marital_Status**



```
#outlier detection
sns.boxplot(data = df, y = 'Purchase', x = 'Age', hue = 'Gender' )
plt.show()
```

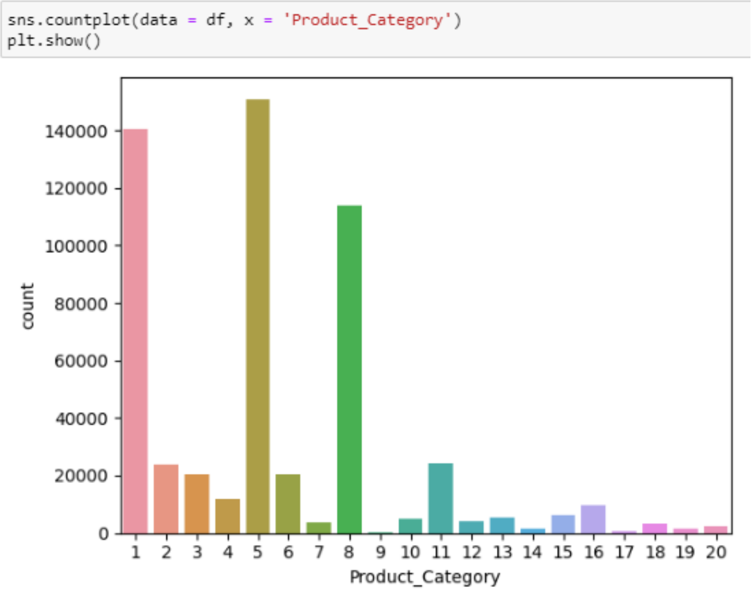


```
sns.boxplot(data = df, y = 'Purchase', x = 'Age', hue = 'Marital_Status' )
plt.show()
```

Observations –

Outliers are seen in both the plots, where maximum outliers are observed for purchases of female customers of the age-group 55+ and the age-group 18-45.

# Frequency of purchases based on Product_Category



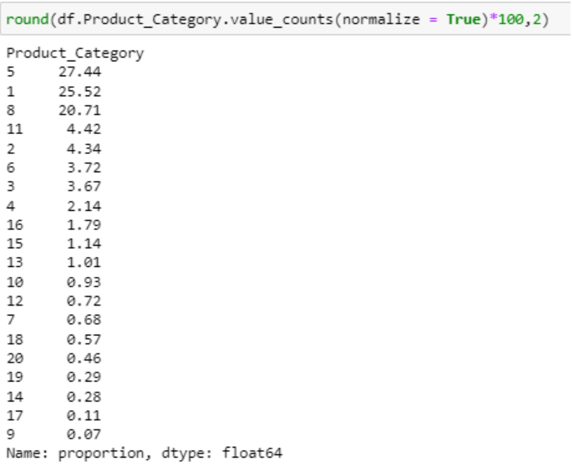
```
sns.countplot(data = df, x = 'Product_Category')
plt.show()
```

- **Categories 1, 5 and 8 have maximum purchases.**

Recommendations –

Walmart must focus on the sales of these product categories.

# No of customer purchases based on Product-type



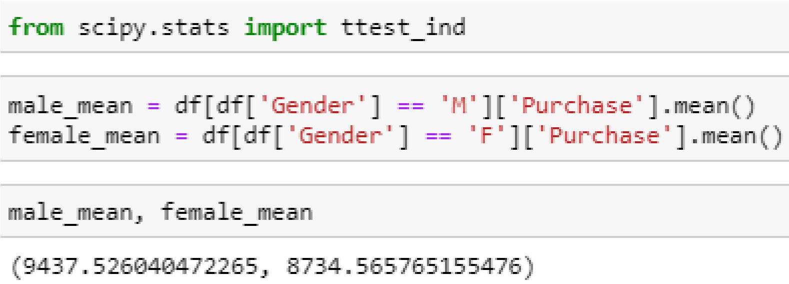
```
round(df.Product_Category.value_counts(normalize = True)*100,2)
```

```
Product_Category
5     27.44
1     25.52
8     20.71
11     4.42
2      4.34
6      3.72
3      3.67
4      2.14
16     1.79
15     1.14
13     1.01
10     0.93
12     0.72
7      0.68
18     0.57
20     0.46
19     0.29
14     0.28
17     0.11
9      0.07
Name: proportion, dtype: float64
```

Insights:

- Product Categories 1,5 and 8 are the maximum purchases.
- Thus, least popular product among the customers is Product_Category 9.

# Average spending of Male and Female customers



```
from scipy.stats import ttest_ind
```

```
male_mean = df[df['Gender'] == 'M']['Purchase'].mean()
female_mean = df[df['Gender'] == 'F']['Purchase'].mean()
```

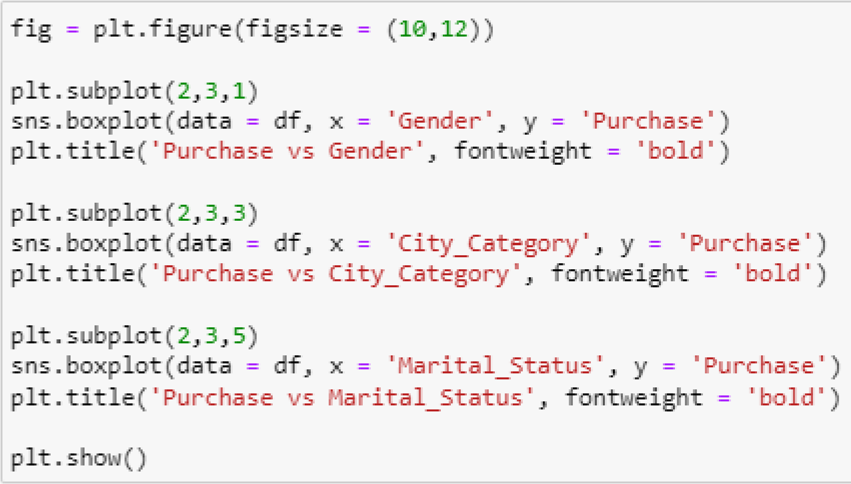
```
male_mean, female_mean
```

```
(9437.526040472265, 8734.565765155476)
```

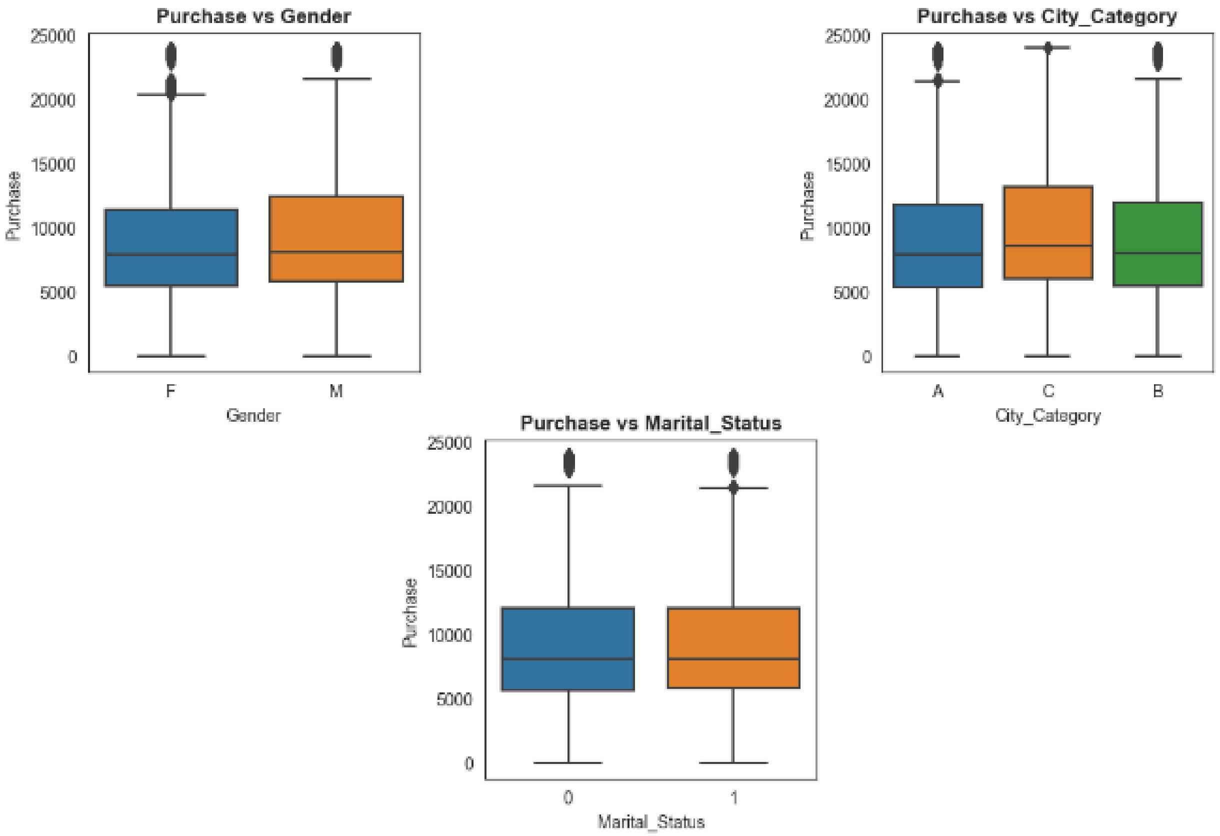
Observations –

The average mean purchase of male customers is large compared to Female customers.

Bivariate Analysis



```
fig = plt.figure(figsize = (10,12))

plt.subplot(2,3,1)
sns.boxplot(data = df, x = 'Gender', y = 'Purchase')
plt.title('Purchase vs Gender', fontweight = 'bold')

plt.subplot(2,3,3)
sns.boxplot(data = df, x = 'City_Category', y = 'Purchase')
plt.title('Purchase vs City_Category', fontweight = 'bold')

plt.subplot(2,3,5)
sns.boxplot(data = df, x = 'Marital_Status', y = 'Purchase')
plt.title('Purchase vs Marital_Status', fontweight = 'bold')

plt.show()
```



Observations –

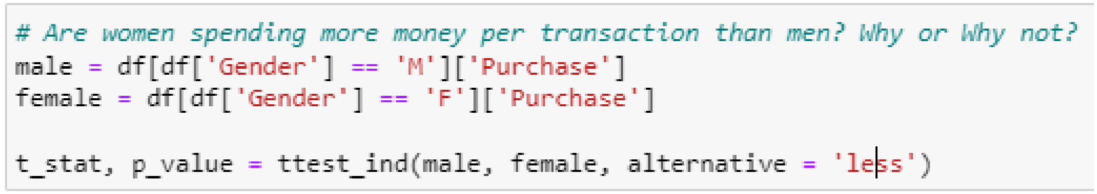
- The above bivariate plots give the spending of customers based on their Gender, City they live and their Marital_Status.
- The median spending done by Male and Female customers are similar. However, male customers are seen to spend more than female customers.
- Also, the city category C has the highest spending compared to other two categories.
- Observations on spending based on Marital_Status infer that spending does not depend on the marital status of the customer i.e. both married and unmarried customers spend equally.

Recommendations –

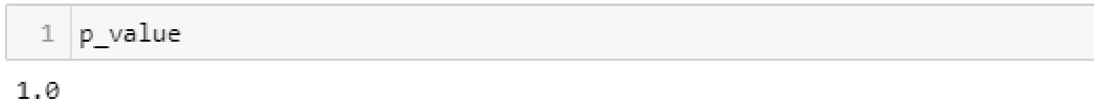
Walmart must focus more over the sales acquired by Male customers in City - C compared to others.

Checking if the data is true using null hypothesis with a 95% confidence interval.

Q. Are women spending more money per transaction than men? Why or why not?



```
# Are women spending more money per transaction than men? Why or Why not?
male = df[df['Gender'] == 'M']['Purchase']
female = df[df['Gender'] == 'F']['Purchase']

t_stat, p_value = ttest_ind(male, female, alternative = 'less')
```



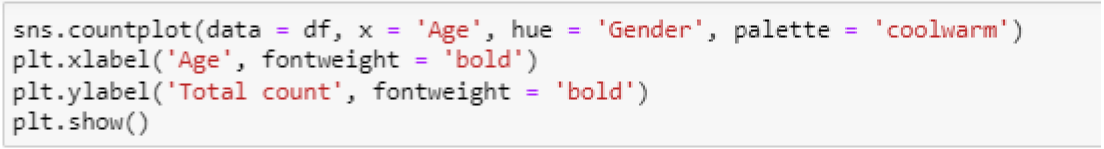
```
p_value
```

```
1.0
```

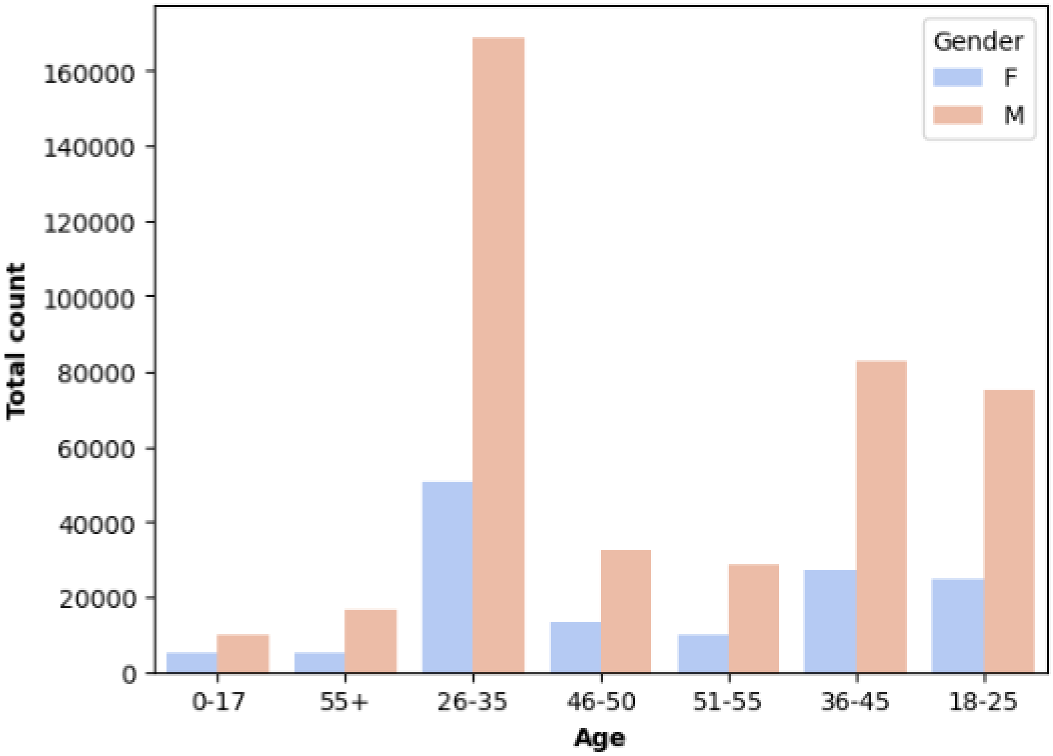
Observations –

As the p_value > significance (0.05), hence we fail to reject the null hypothesis i.e. women are not spending more money than men.

# Customer purchase based on Age and Gender



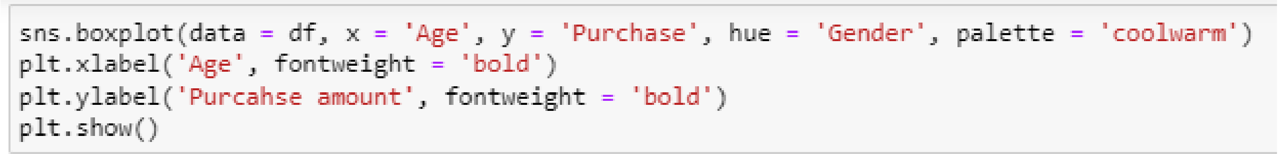
```
sns.countplot(data = df, x = 'Age', hue = 'Gender', palette = 'coolwarm')
plt.xlabel('Age', fontweight = 'bold')
plt.ylabel('Total count', fontweight = 'bold')
plt.show()
```



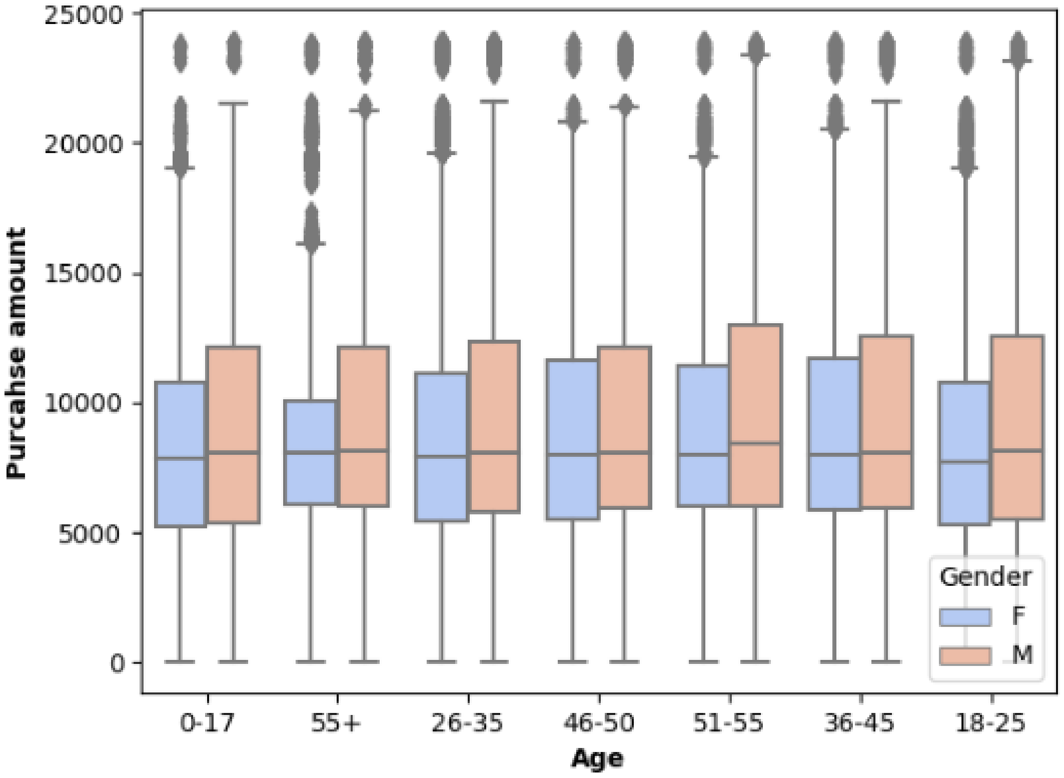
Observations:

- The above plot shows that Male customers of every age group are the maximum spending ones compared to females.
- Maximum purchases are done by age group 26-35.

# Analysis on purchase amount based on age and gender of customers



```
sns.boxplot(data = df, x = 'Age', y = 'Purchase', hue = 'Gender', palette = 'coolwarm')
plt.xlabel('Age', fontweight = 'bold')
plt.ylabel('Purcahse amount', fontweight = 'bold')
plt.show()
```



Observations –

- The box plot shows almost same median in the amount of purchases done by both Male and Female customers of every age group which means that total spending done by male and female are same and in the range upto 22K.
- Some outliers exist in every age group that spend upto 25K max.
- Male customers are seen to have fewer outliers compared to females. i.e. only fewer female customers spend more than 20K.

Recommendation –

- Walmart is doing well as the average spending of male and female are similar. Thus, product category in Walmart are not biased and it caters the need for both male and female.
- Walmart must increase offers and discounts on purchases of most female products to increase their total purchase.

Q] Confidence Interval

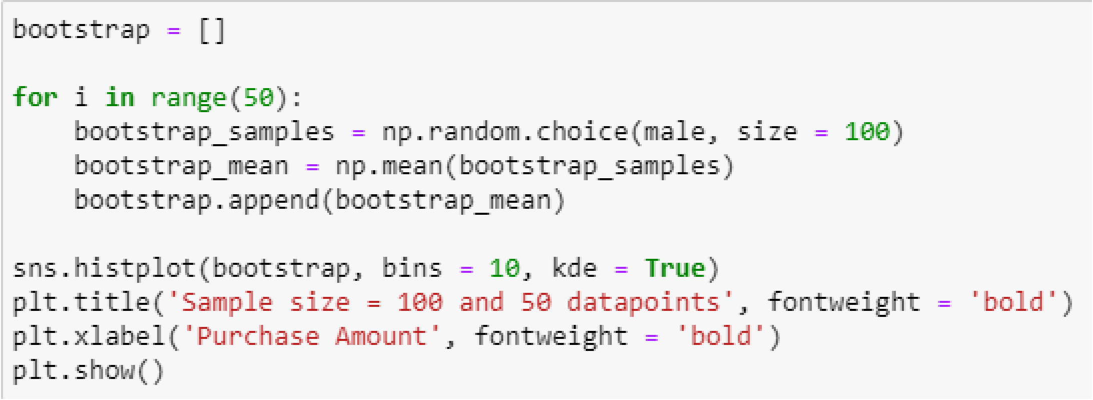
To find the population mean of the expenses for both male and female customers, we take samples from purchase data of both male and female and check whether the population mean lie within the interval for 90%, 95% and 99% confidence level.

Checking the distribution of the mean spending –

1. Based on Gender

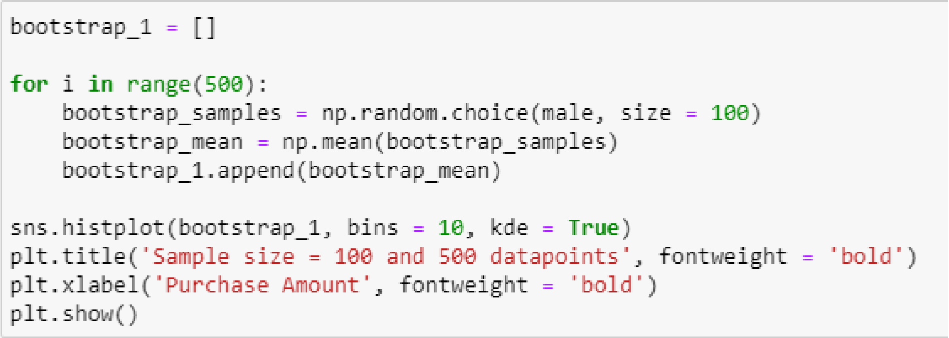
Male customers

1. Taking sample size = 100 and checking for 50 datapoints.



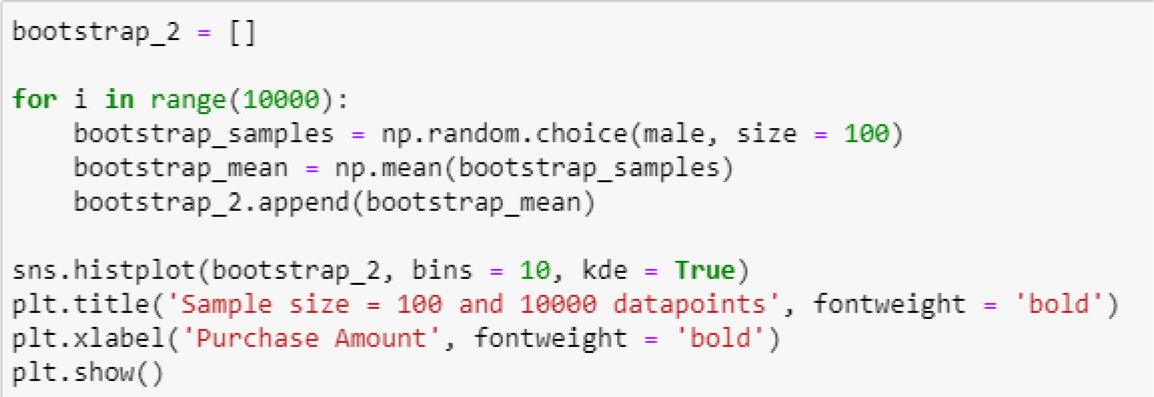
```
bootstrap = []

for i in range(50):
    bootstrap_samples = np.random.choice(male, size = 100)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap.append(bootstrap_mean)

sns.histplot(bootstrap, bins = 10, kde = True)
plt.title('Sample size = 100 and 50 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

2. Taking sample size = 100 and 500 datapoints



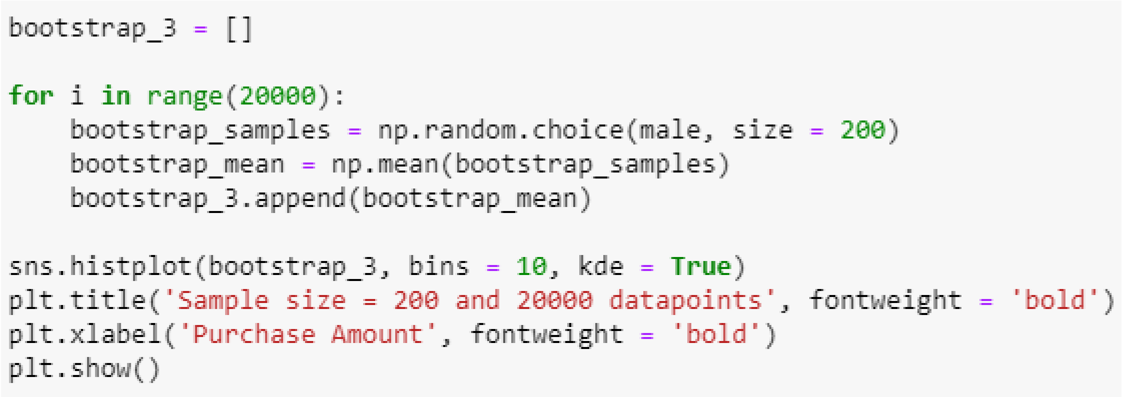
```
bootstrap_1 = []

for i in range(500):
    bootstrap_samples = np.random.choice(male, size = 100)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_1.append(bootstrap_mean)

sns.histplot(bootstrap_1, bins = 10, kde = True)
plt.title('Sample size = 100 and 500 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

3. Taking sample size = 100 and 10,000 datapoints



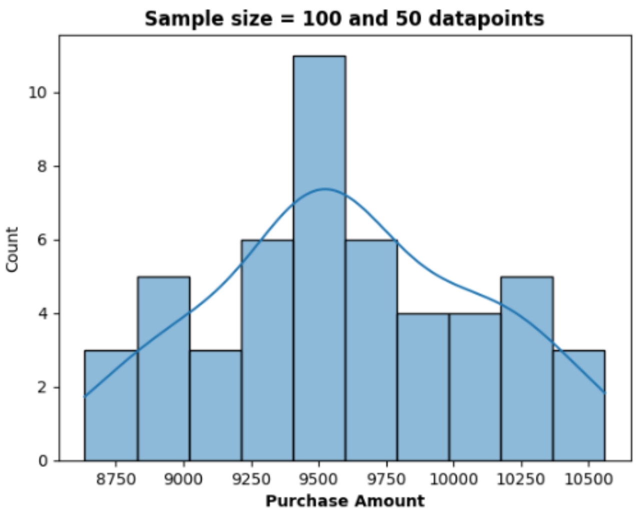
```
bootstrap_2 = []

for i in range(10000):
    bootstrap_samples = np.random.choice(male, size = 100)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_2.append(bootstrap_mean)

sns.histplot(bootstrap_2, bins = 10, kde = True)
plt.title('Sample size = 100 and 10000 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

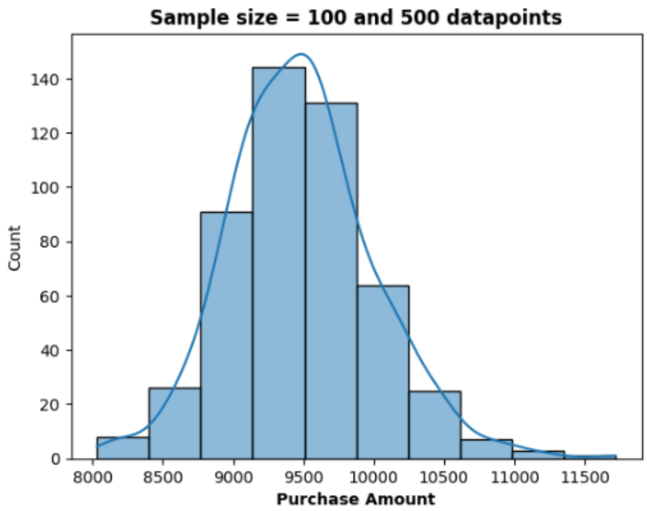
4. Taking sample size = 200 and 20,000 datapoints

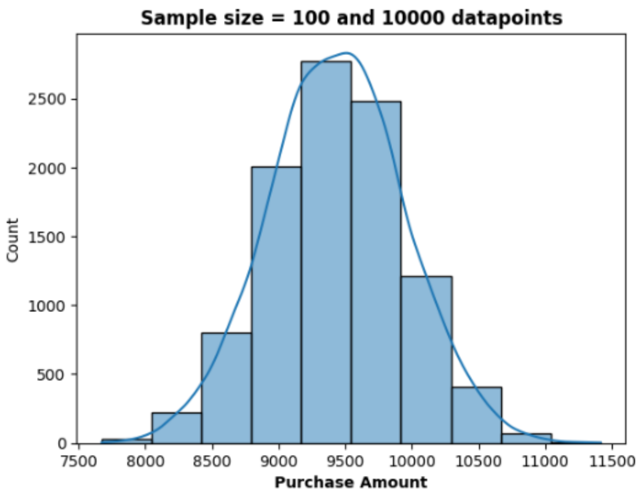


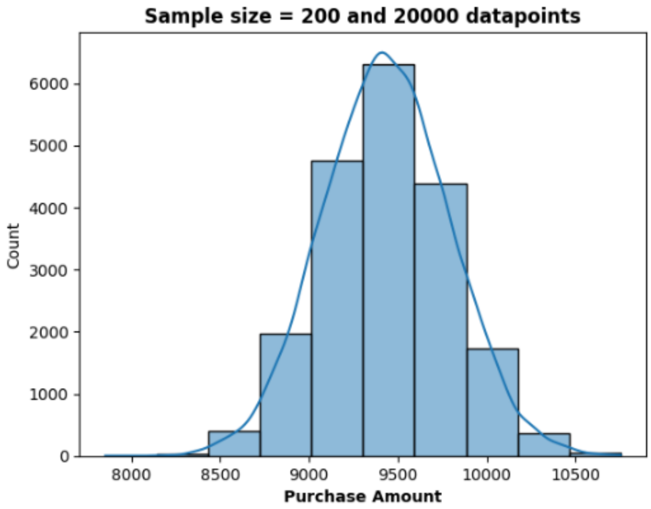
```
bootstrap_3 = []

for i in range(20000):
    bootstrap_samples = np.random.choice(male, size = 200)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_3.append(bootstrap_mean)

sns.histplot(bootstrap_3, bins = 10, kde = True)
plt.title('Sample size = 200 and 20000 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

Visualization of the Distribution of mean spending done by male customers





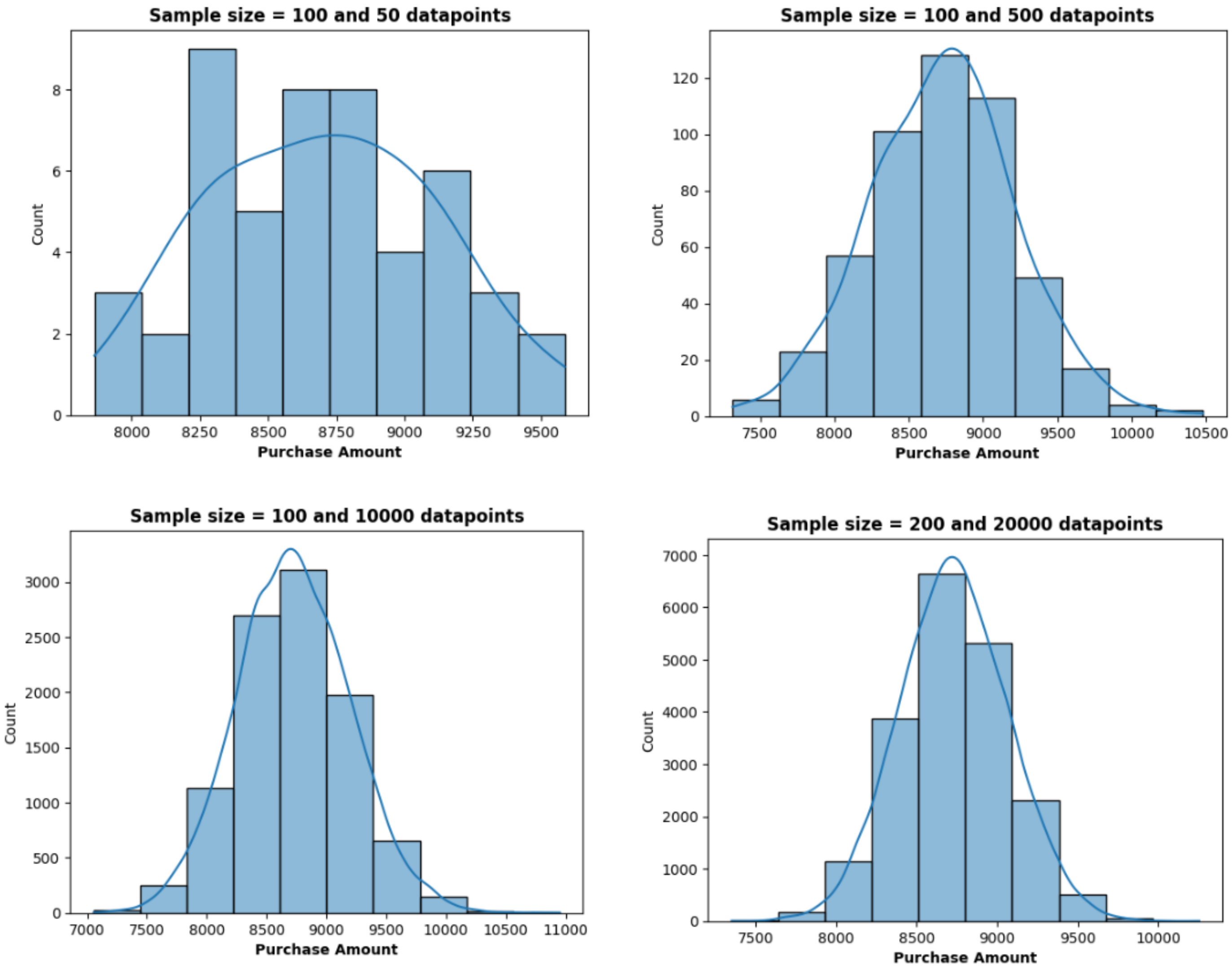




Observations –

Variance (spread of data) decreases as sample size and datapoints increase.

Visualization of the Distribution of mean spending done by female customers

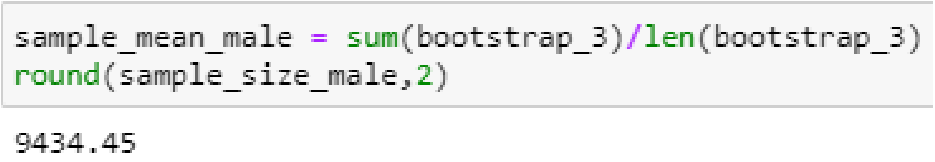


Observations –

Comparison between both the distributions infer that the mean spending of Male customers is more compared to the female customers.

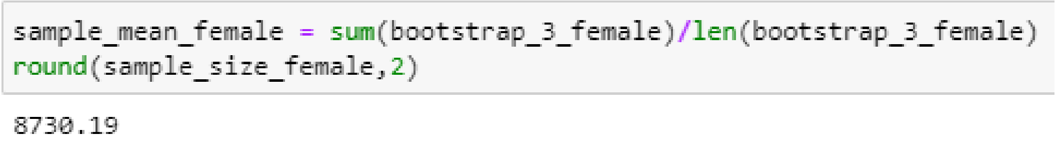
As the variance is less for sample size 200 and 20,000 datapoints, we calculate the confidence intervals for 90%, 95% and 99% confidence levels for the specified sample size and sample mean.

Sample_mean for sample size = 200:



```
sample_mean_male = sum(bootstrap_3)/len(bootstrap_3)
round(sample_size_male,2)
```

```
9434.45
```



```
sample_mean_female = sum(bootstrap_3_female)/len(bootstrap_3_female)
round(sample_size_female,2)
```

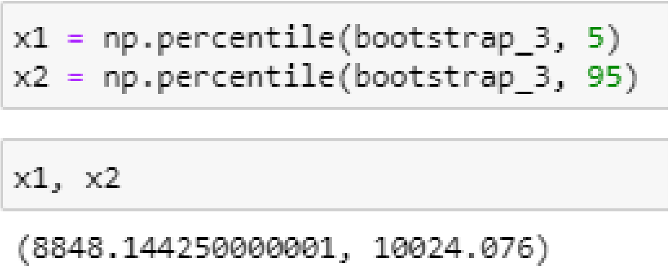
```
8730.19
```

From the above sample means, we can say that the average amount spent by male customers is 9434.45 and the average amount spent by female customers is 8730.19.

Validating the difference in the mean spending with different confidence interval–

For Male customers

1. For Male customer with 90% Confidence Level



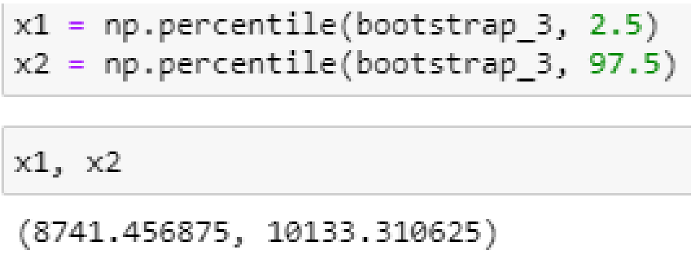
```
x1 = np.percentile(bootstrap_3, 5)
x2 = np.percentile(bootstrap_3, 95)
```

```
x1, x2
```

```
(8848.144250000001, 10024.076)
```

With 90% Confidence Interval, the mean spending of male customers lie in the range (8848.144, 10024.076).

2. For Male customer with 95% Confidence Level



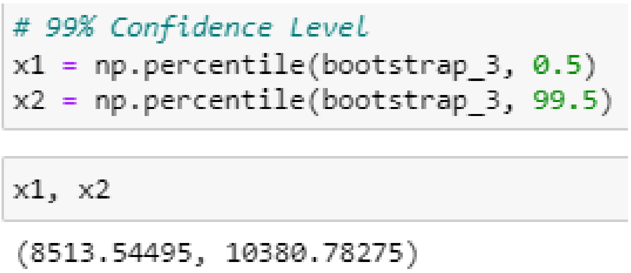
```
x1 = np.percentile(bootstrap_3, 2.5)
x2 = np.percentile(bootstrap_3, 97.5)
```

```
x1, x2
```

```
(8741.456875, 10133.310625)
```

With 95% Confidence Interval, the mean spending of male customers lie in the range (8741.456, 10133.311).

3. For Male customer with 99% Confidence Level



```
# 99% Confidence Level
x1 = np.percentile(bootstrap_3, 0.5)
x2 = np.percentile(bootstrap_3, 99.5)
```

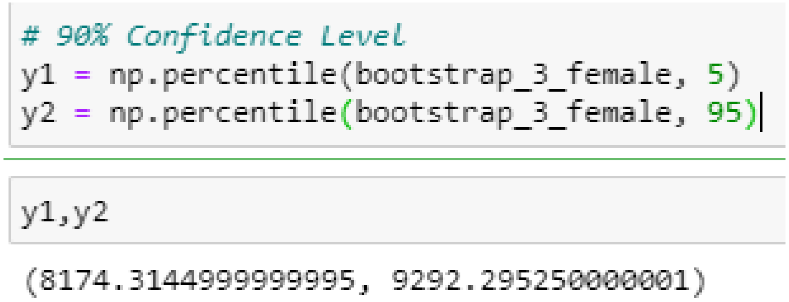
```
x1, x2
```

```
(8513.54495, 10380.78275)
```

With 99% Confidence Interval, the mean spending of male customers lie in the range (8513.545, 10380.782).

For Female customers

1. For Female customer with 90% Confidence Level



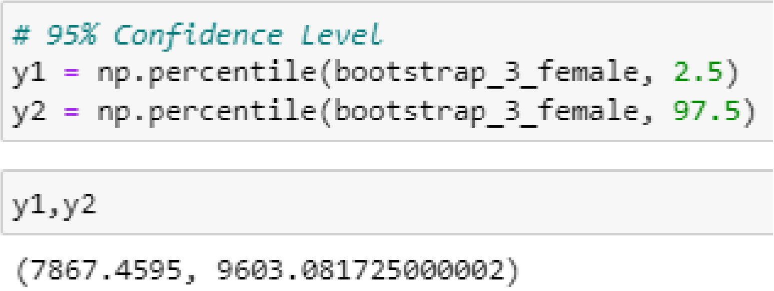
```
# 90% Confidence Level
y1 = np.percentile(bootstrap_3_female, 5)
y2 = np.percentile(bootstrap_3_female, 95)
```

```
y1,y2
```

```
(8174.3144999999995, 9292.295250000001)
```

With 90% Confidence Interval, the mean spending of female customers lie in the range (8174.314, 9292.295).

2. For Female customer with 95% Confidence Level



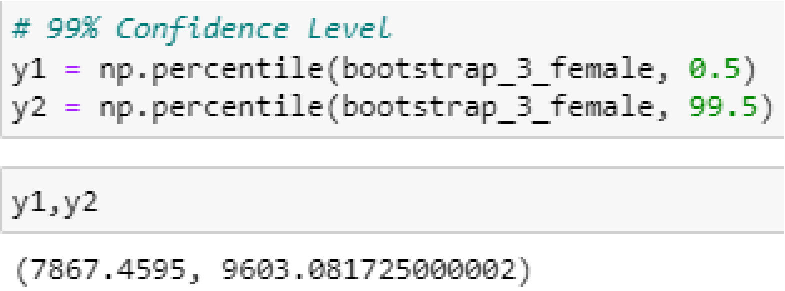
```
# 95% Confidence Level
y1 = np.percentile(bootstrap_3_female, 2.5)
y2 = np.percentile(bootstrap_3_female, 97.5)
```

```
y1,y2
```

```
(7867.4595, 9603.081725000002)
```

With 95% Confidence Interval, the mean spending of female customers lie in the range (7867.459, 9603.082).

3. For Female customer with 99% Confidence Level



```
# 99% Confidence Level
y1 = np.percentile(bootstrap_3_female, 0.5)
y2 = np.percentile(bootstrap_3_female, 99.5)
```

```
y1,y2
```

```
(7867.4595, 9603.081725000002)
```

With 99% Confidence Interval, the mean spending of female customers lie in the range (7867.4595, 9603.917).

	90% CI	**95% CI**	**99% CI**
MALE	(8848.144, 10024.076)	(8741.456, 10133.311)	(8513.545, 10380.782)
FEMALE	(8174.314, 9292.295)	(7867.459, 9603.082)	(7867.4595, 9603.917)

Observations –

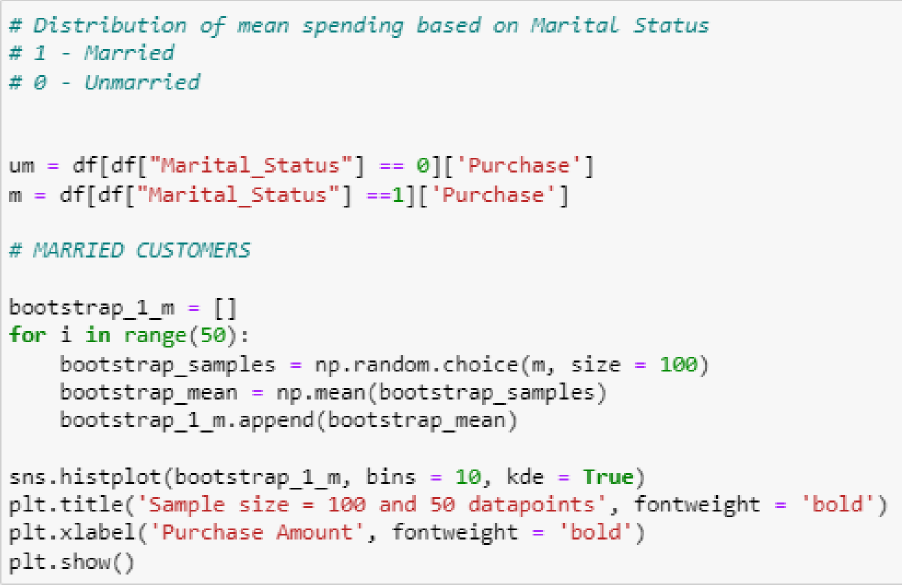
Confidence intervals are found to be overlapping. This concludes that there is no significant difference in the mean spending by male and female customers.

2. Marital Status

Distribution of mean spending based on Marital Status –

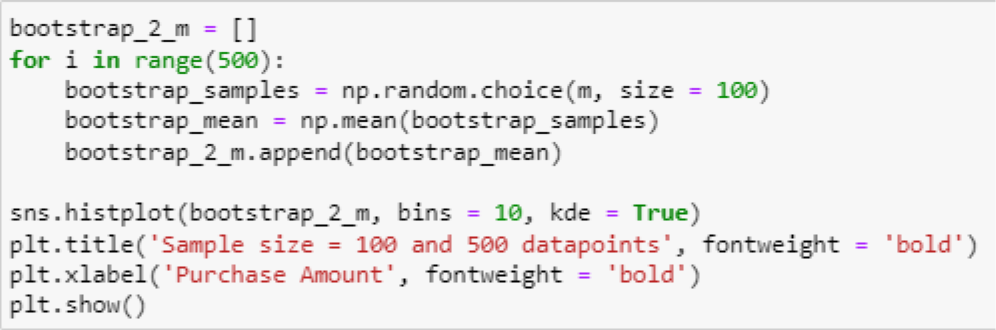
For Married customers

1. Taking sample size = 100 and checking for 50 datapoints.



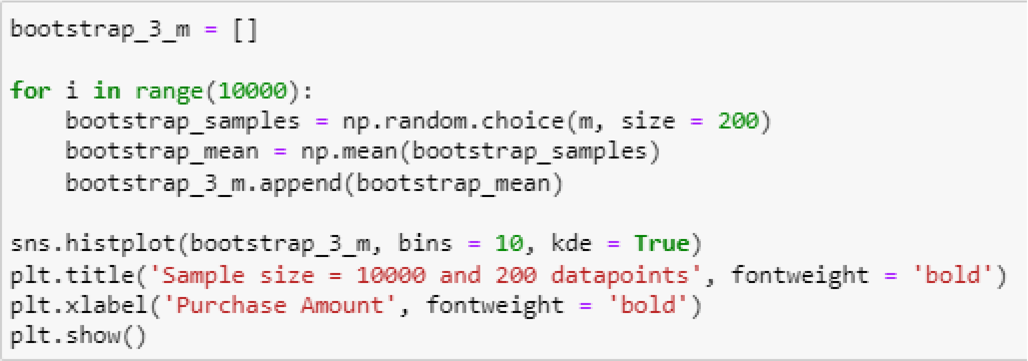
```
# Distribution of mean spending based on Marital Status
# 1 - Married
# 0 - Unmarried


um = df[df["Marital_Status"] == 0]['Purchase']
m = df[df["Marital_Status"] ==1]['Purchase']

# MARRIED CUSTOMERS

bootstrap_1_m = []
for i in range(50):
    bootstrap_samples = np.random.choice(m, size = 100)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_1_m.append(bootstrap_mean)

sns.histplot(bootstrap_1_m, bins = 10, kde = True)
plt.title('Sample size = 100 and 50 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

2. Taking sample size = 100 and checking for 500 datapoints.



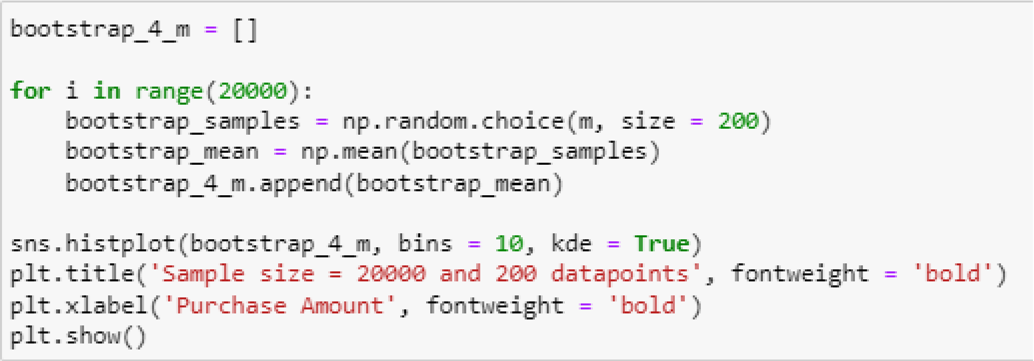
```
bootstrap_2_m = []
for i in range(500):
    bootstrap_samples = np.random.choice(m, size = 100)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_2_m.append(bootstrap_mean)

sns.histplot(bootstrap_2_m, bins = 10, kde = True)
plt.title('Sample size = 100 and 500 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

3. Taking sample size = 200 and checking for 10000 datapoints



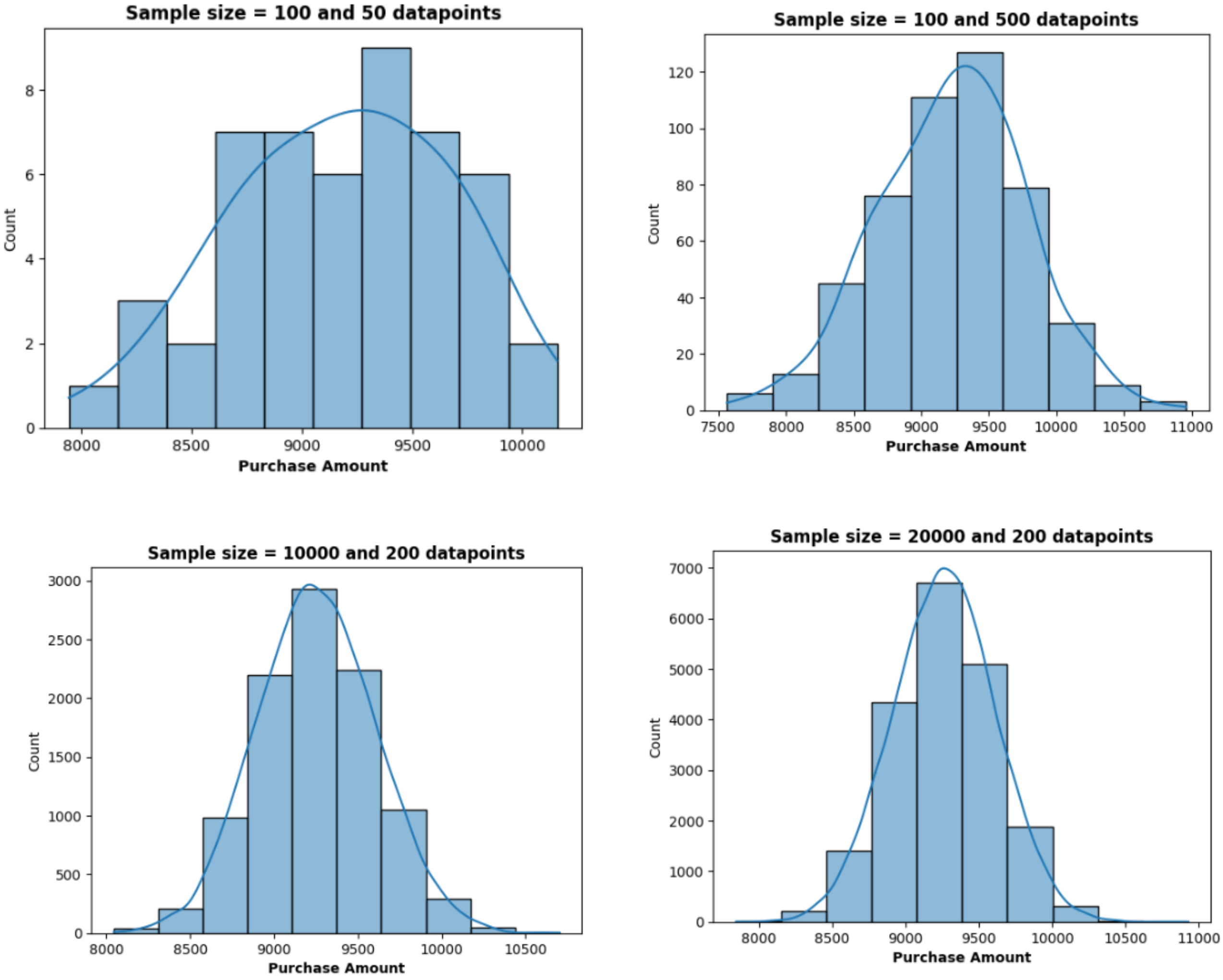
```
bootstrap_3_m = []

for i in range(10000):
    bootstrap_samples = np.random.choice(m, size = 200)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_3_m.append(bootstrap_mean)

sns.histplot(bootstrap_3_m, bins = 10, kde = True)
plt.title('Sample size = 10000 and 200 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

4. Taking sample size = 200 and checking for 10000 datapoints



```
bootstrap_4_m = []

for i in range(20000):
    bootstrap_samples = np.random.choice(m, size = 200)
    bootstrap_mean = np.mean(bootstrap_samples)
    bootstrap_4_m.append(bootstrap_mean)

sns.histplot(bootstrap_4_m, bins = 10, kde = True)
plt.title('Sample size = 20000 and 200 datapoints', fontweight = 'bold')
plt.xlabel('Purchase Amount', fontweight = 'bold')
plt.show()
```

Visualization of the Distribution of mean spending done by Married customers



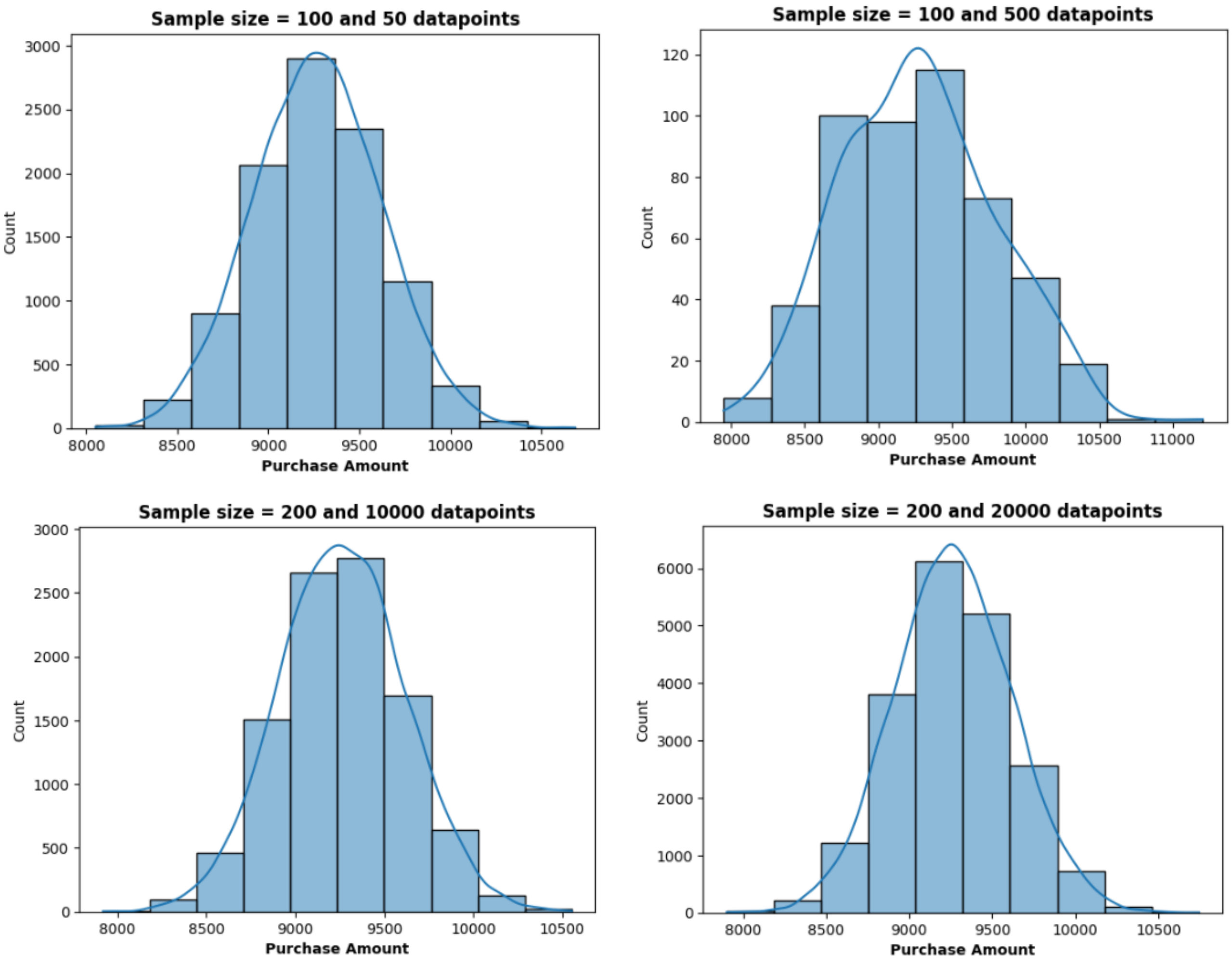
Sample size = 100 and 50 datapoints

Sample size = 100 and 500 datapoints

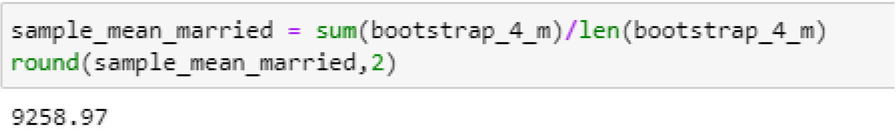
Sample size = 10000 and 200 datapoints

Sample size = 20000 and 200 datapoints

Visualization of the Distribution of mean spending done by Unmarried customers

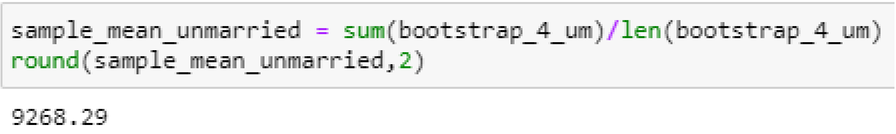


Validating the difference in the mean spending based on Marital Status –



```
sample_mean_married = sum(bootstrap_4_m)/len(bootstrap_4_m)
round(sample_mean_married,2)
```

```
9258.97
```



```
sample_mean_unmarried = sum(bootstrap_4_um)/len(bootstrap_4_um)
round(sample_mean_unmarried,2)
```

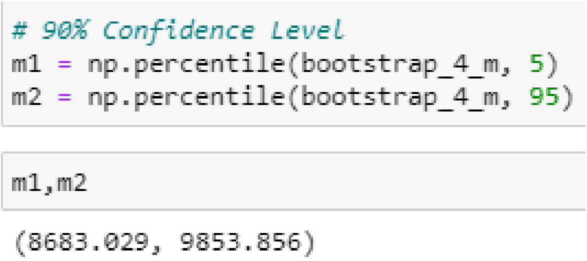
```
9268.29
```

Observations –

The average spending of the samples taken for both married as well as unmarried customers are not significantly different.

For Married customers

1. For Married customers with 90% Confidence Level



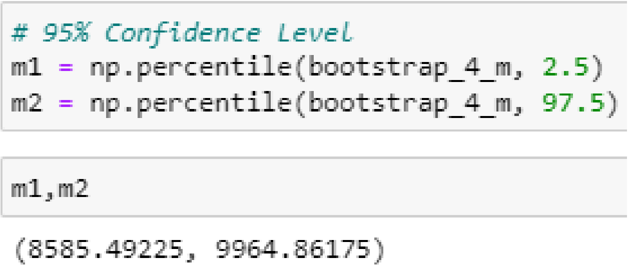
```
# 90% Confidence Level
m1 = np.percentile(bootstrap_4_m, 5)
m2 = np.percentile(bootstrap_4_m, 95)
```

```
m1,m2
```

```
(8683.029, 9853.856)
```

With 90% Confidence Interval, the mean spending of married customers lie in the range (8683.029, 9853.856).

2. For Married customers with 95% Confidence Level



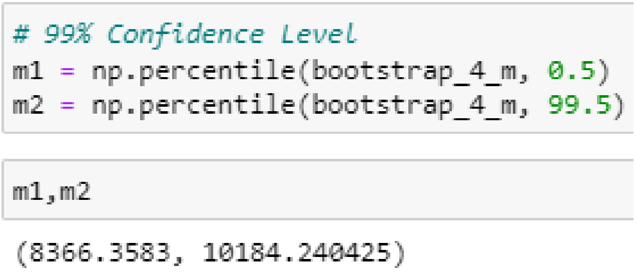
```
# 95% Confidence Level
m1 = np.percentile(bootstrap_4_m, 2.5)
m2 = np.percentile(bootstrap_4_m, 97.5)
```

```
m1,m2
```

```
(8585.49225, 9964.86175)
```

With 95% Confidence Interval, the mean spending of married customers lie in the range (8585.4922, 9964.8617).

3. For Married customers with 99% Confidence Level



```
# 99% Confidence Level
m1 = np.percentile(bootstrap_4_m, 0.5)
m2 = np.percentile(bootstrap_4_m, 99.5)
```

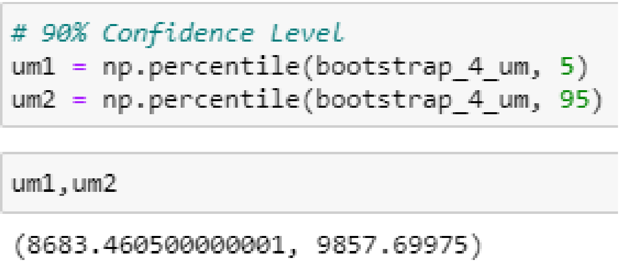
```
m1,m2
```

```
(8366.3583, 10184.240425)
```

With 99% Confidence Interval, the mean spending of married customers lie in the range (8366.3583, 10184.2404).

For Unmarried customers

1. For unmarried customers with 90% Confidence Level



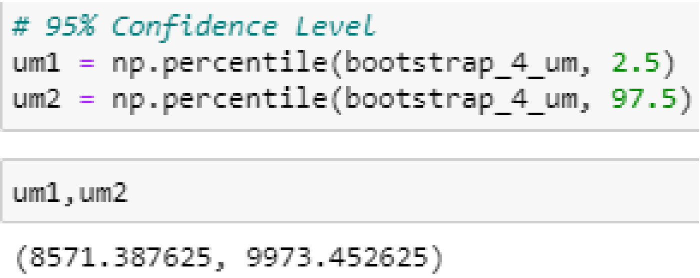
```
# 90% Confidence Level
um1 = np.percentile(bootstrap_4_um, 5)
um2 = np.percentile(bootstrap_4_um, 95)
```

```
um1,um2
```

```
(8683.460500000001, 9857.69975)
```

With 90% Confidence Interval, the mean spending of unmarried customers lie in the range (8683.4605, 9857.699).

2. For unmarried customers with 95% Confidence Level



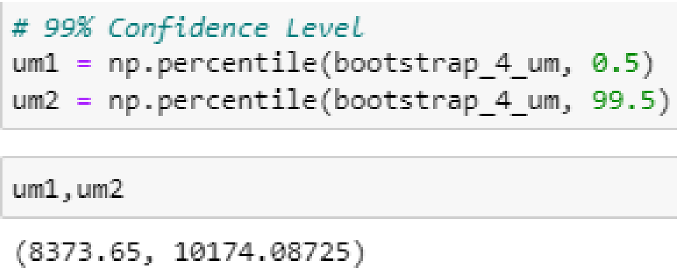
```
# 95% Confidence Level
um1 = np.percentile(bootstrap_4_um, 2.5)
um2 = np.percentile(bootstrap_4_um, 97.5)
```

```
um1,um2
```

```
(8571.387625, 9973.452625)
```

With 95% Confidence Interval, the mean spending of unmarried customers lie in the range (8571.3876, 9973.4526).

3. For unmarried customers with 99% Confidence Level



```
# 99% Confidence Level
um1 = np.percentile(bootstrap_4_um, 0.5)
um2 = np.percentile(bootstrap_4_um, 99.5)
```

```
um1,um2
```

```
(8373.65, 10174.08725)
```

With 99% Confidence Interval, the mean spending of unmarried customers lie in the range (8373.65, 10174.087).

	90% CI	**95% CI**	**99% CI**
MARRIED	(8683.029, 9853.856)	(8585.4922, 9964.8617)	(8366.3583, 10184.2404)
UNMARRIED	(8683.4605, 9857.699)	(8571.3876, 9973.4526)	(8373.65, 10174.087)

Observations –

Confidence intervals are found to be overlapping. This concludes that there is no significant difference in the mean spending by married and unmarried customers.

Recommendations

1. Men spent more money than women, so company should focus on retaining the male customers and getting more male customers.
2. Product_Category - 1, 5 & 8 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.
3. 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

4. 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.

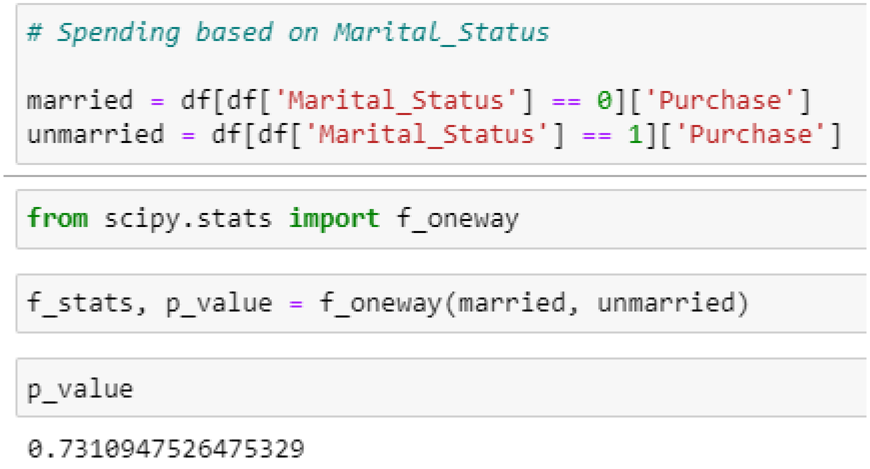
# Checking the variation in the mean/ average of customers using Hypothesis Testing.

Using ANOVA, we can find the variation in the mean spending for 5% significance.

1. **Spending based on Marital Status**

H0: All means are similar
Ha: Means are different.

Alpha = 0.05 (95% Confidence Level)



```
# Spending based on Marital_Status

married = df[df['Marital_Status'] == 0]['Purchase']
unmarried = df[df['Marital_Status'] == 1]['Purchase']
```

```
from scipy.stats import f_oneway
```

```
f_stats, p_value = f_oneway(married, unmarried)
```

```
p_value
```

```
0.7310947526475329
```

Since, p_value > alpha, We fail to reject the Null Hypothesis (H0).

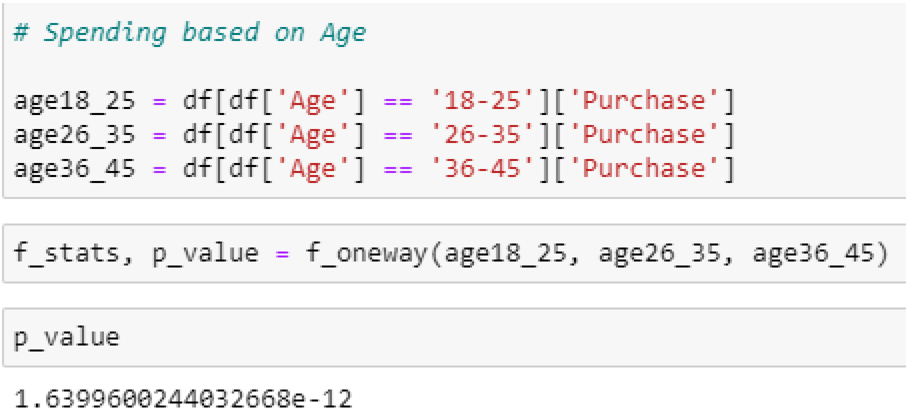
Hence, we can conclude that with a 95% Confidence Level, there is no difference in the average spending of customers based on their Marital Status.

2. **Spending based on Age**

H0: All means are similar

Ha: Means are different.

Alpha = 0.05 (95% Confidence Level)



```
# Spending based on Age

age18_25 = df[df['Age'] == '18-25']['Purchase']
age26_35 = df[df['Age'] == '26-35']['Purchase']
age36_45 = df[df['Age'] == '36-45']['Purchase']
```

```
f_stats, p_value = f_oneway(age18_25, age26_35, age36_45)
```

```
p_value
```

```
1.6399600244032668e-12
```

Since, p_value < alpha, We reject the Null Hypothesis (H0).

Hence, we can conclude that with a 95% Confidence Level, there is a significant difference in the average spending of customers based in the age-groups 18-25, 26-35, 36-45.

Recommendations –

Walmart must focus to retain the customers in these age-groups by providing discount offers and special contests to attract more customers belonging to these age-groups to increase their revenue.