```
from google.colab import files
files.upload()
```

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

```
df= pd.read_csv('walmart_data.csv')
df.head()
```

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

1.) checking the structure & characteristics of the dataset

a) . The data type of all columns in the "customers" table.

```
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
```

b) . You can find the number of rows and columns given in the dataset

```
df.shape
```

```
(550068, 10)
```

Insights: ▶ There are 550068 records and 10 columns in the dataset.

c) . Check for the missing values and find the number of missing values in each column

```
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
```

Insights : ▶ By looking at the data, we can say that there are no null values present in any of the columns.

==

---

2. Detect Null values and outliers

a.) Find the outliers for every continuous variable in 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

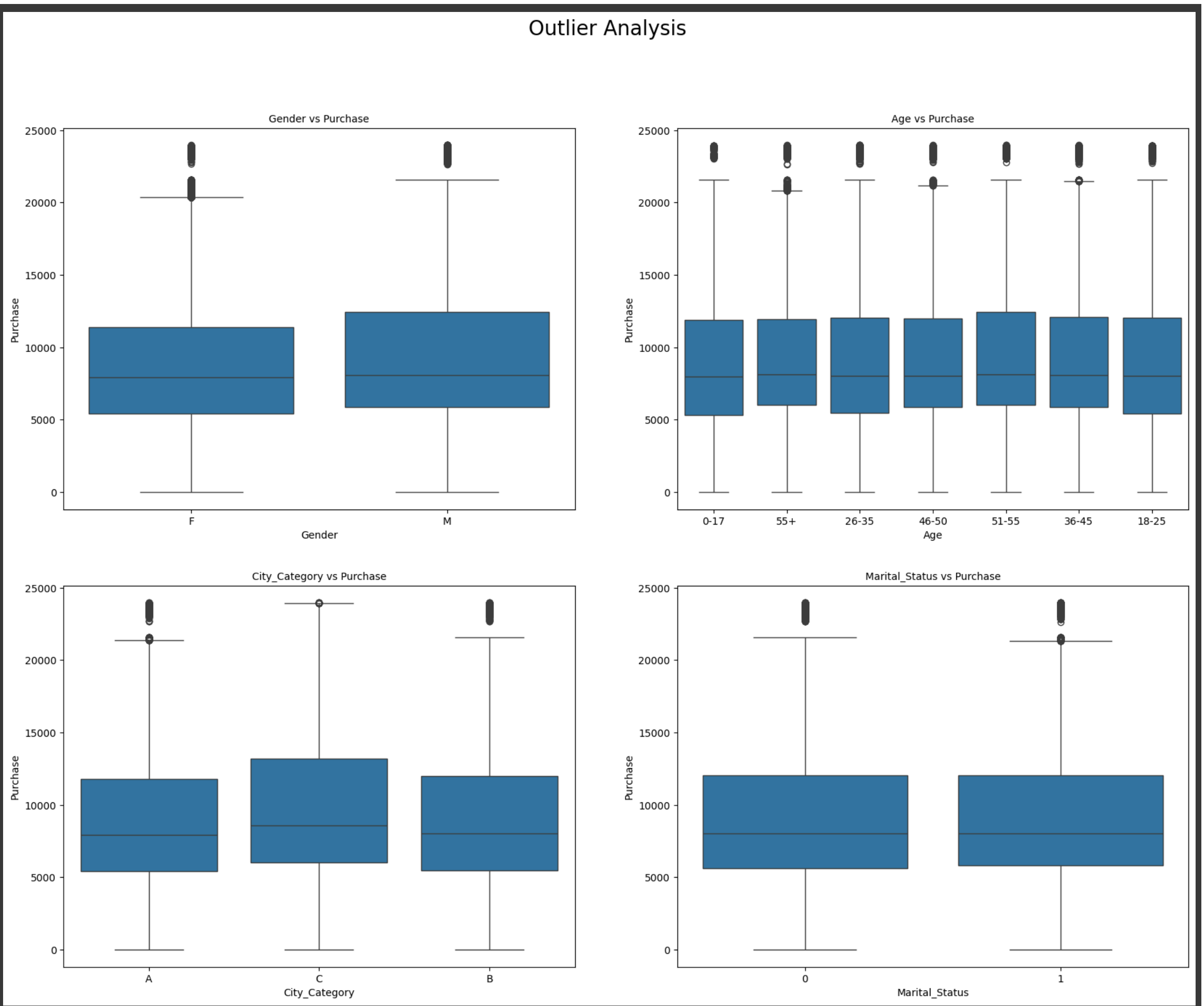
```
#Purchase amount is the countinous variable in the data set
plt.figure(figsize=(20,15))
plt.suptitle("Outlier Analysis",fontsize=20)

plt.subplot(2,2,1)
sns.boxplot(data=df,x="Gender",y="Purchase")
plt.xlabel("Gender")
plt.ylabel("Purchase")
plt.title("Gender vs Purchase",fontsize=10)

plt.subplot(2,2,2)
sns.boxplot(data=df,x="Age",y="Purchase")
plt.xlabel("Age")
plt.ylabel("Purchase")
plt.title("Age vs Purchase",fontsize=10)

plt.subplot(2,2,3)
sns.boxplot(data=df,x="City_Category",y="Purchase")
plt.xlabel("City_Category")
plt.ylabel("Purchase")
plt.title("City_Category vs Purchase",fontsize=10)

plt.subplot(2,2,4)
sns.boxplot(data=df,x="Marital_Status",y="Purchase")
plt.xlabel("Marital_Status")
plt.ylabel("Purchase")
plt.title("Marital_Status vs Purchase",fontsize=10)
plt.show()
```



b) . Remove/clip the data between the 5 percentile and 95 percentile

```
# unique values in each column
for i in df.columns:
  print(i,' : ',df[i].nunique())
```

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

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

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

```
# we can see that Gender ,Age and City_Category column are in string(object) data type so we need to convert this datatype into integer
df_copy = df.copy()


df_copy['Gender'].replace(['M', 'F'], [1, 0], inplace=True)

df_copy['City_Category'].replace(['A', 'B','C'], [0, 1,2], inplace=True)

df_copy['Age'].replace(['0-17', '18-25', '26-35','36-45','46-50','51-55','55+'], [0, 1, 2,3,4,5,6], inplace=True)

# cliping the data np.clip() between the 5 percentile and 95 percentile
cols=['Gender','Age','City_Category','Marital_Status']

for col in cols:
  percentile = df_copy[col].quantile([0.05,0.95]).values
  df_copy[col] = np.clip(df_copy[col], percentile[0], percentile[1])
```

3. Data Exploration

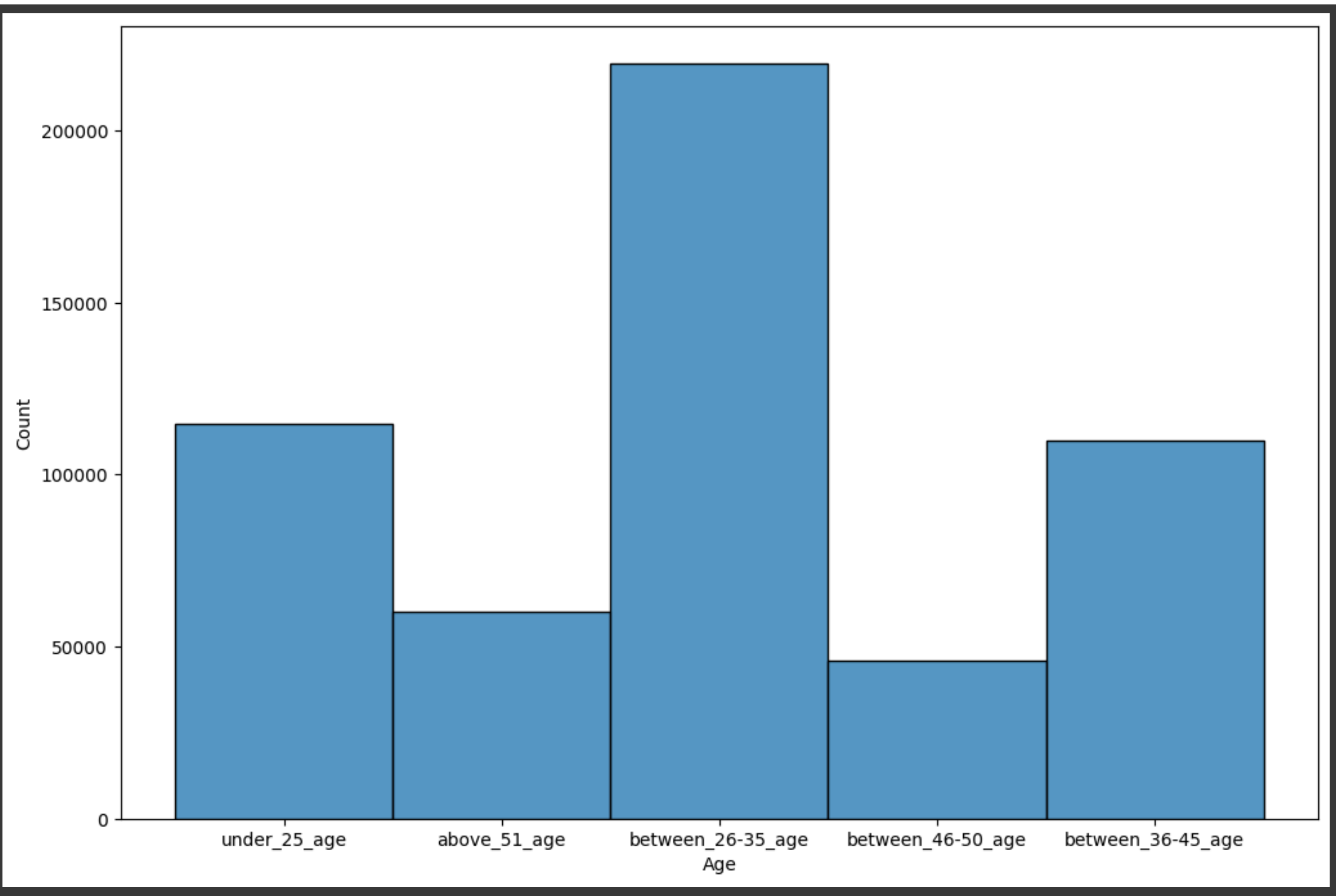
a. What products are different age groups buying?

```
df_copy['Age'].unique()
# since we alrady replace age group with numeric value and applied np.clip , we are getting here total 5 group
# let's rename this age group with better meaningful word
df_copy['Age'].replace([1,2,3,4,5],['under_25_age','between_26-35_age','between_36-45_age','between_46-50_age','above_51_age'],inplace=
# Grouping by AgeGroup and Product, and counting occurrences
grouped_data =  df_copy.groupby(['Age']).size().reset_index(name='Count')
grouped_data
```

	Age	Count
0	above_51_age	60005
1	between_26-35_age	219587
2	between_36-45_age	110013
3	between_46-50_age	45701
4	under_25_age	114762

Next steps: Generate code with grouped_data View recommended plots

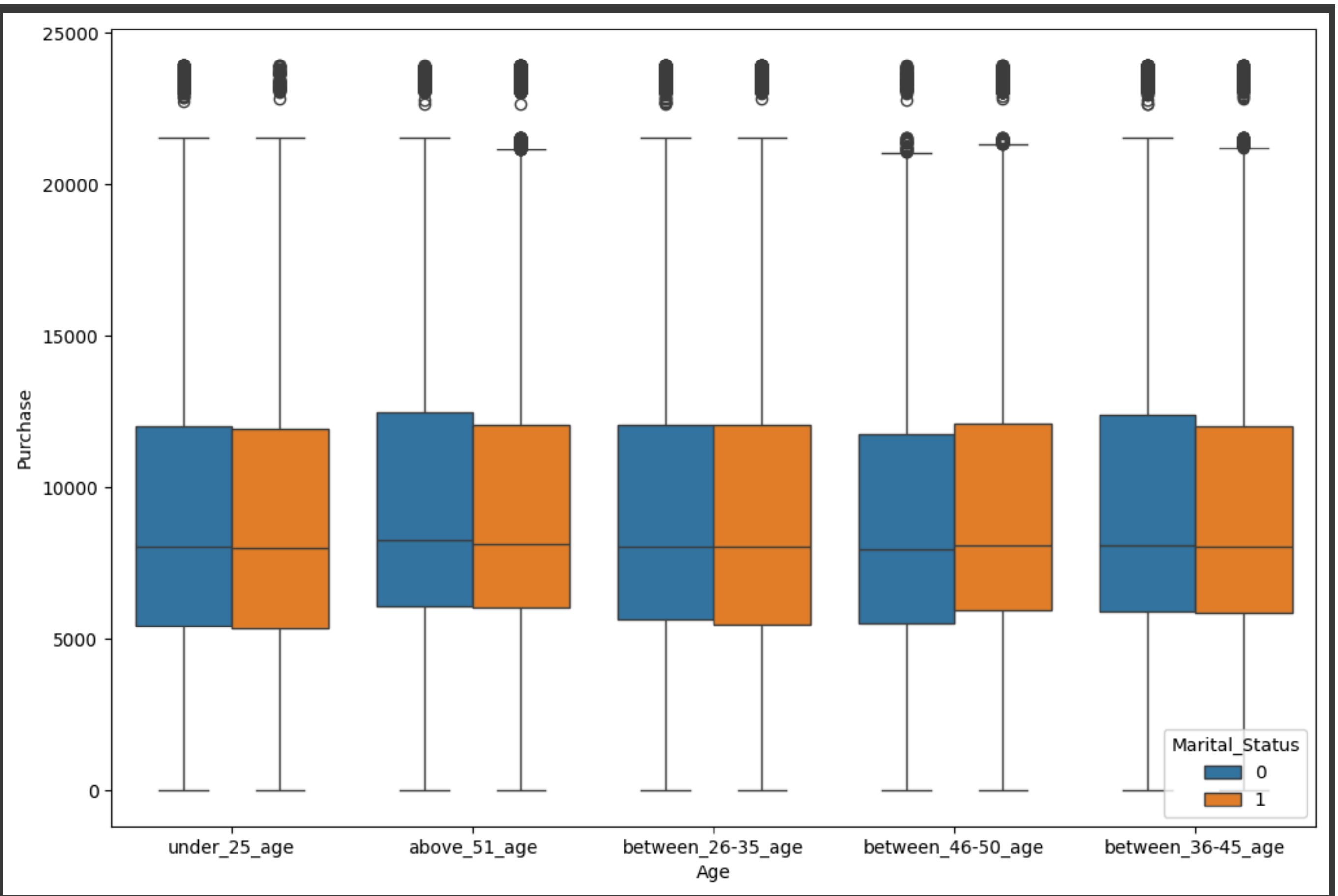
```
plt.figure(figsize=(12,8))
sns.histplot(data=df_copy,x='Age',binwidth=2)
plt.show()
```



Insights ▶ we can observe that age group between 26 and 35 are more in buying products while age group between 46 and 50 are least buyer among these.

b. Is there a relationship between age, marital status, and the amount spent

```
# You can do multivariate analysis to find the relationship between age,marital status, and the amount spent
plt.figure(figsize=(12,8))
sns.boxplot(data=df_copy, x='Age',y='Purchase', hue='Marital_Status')
plt.show()
```

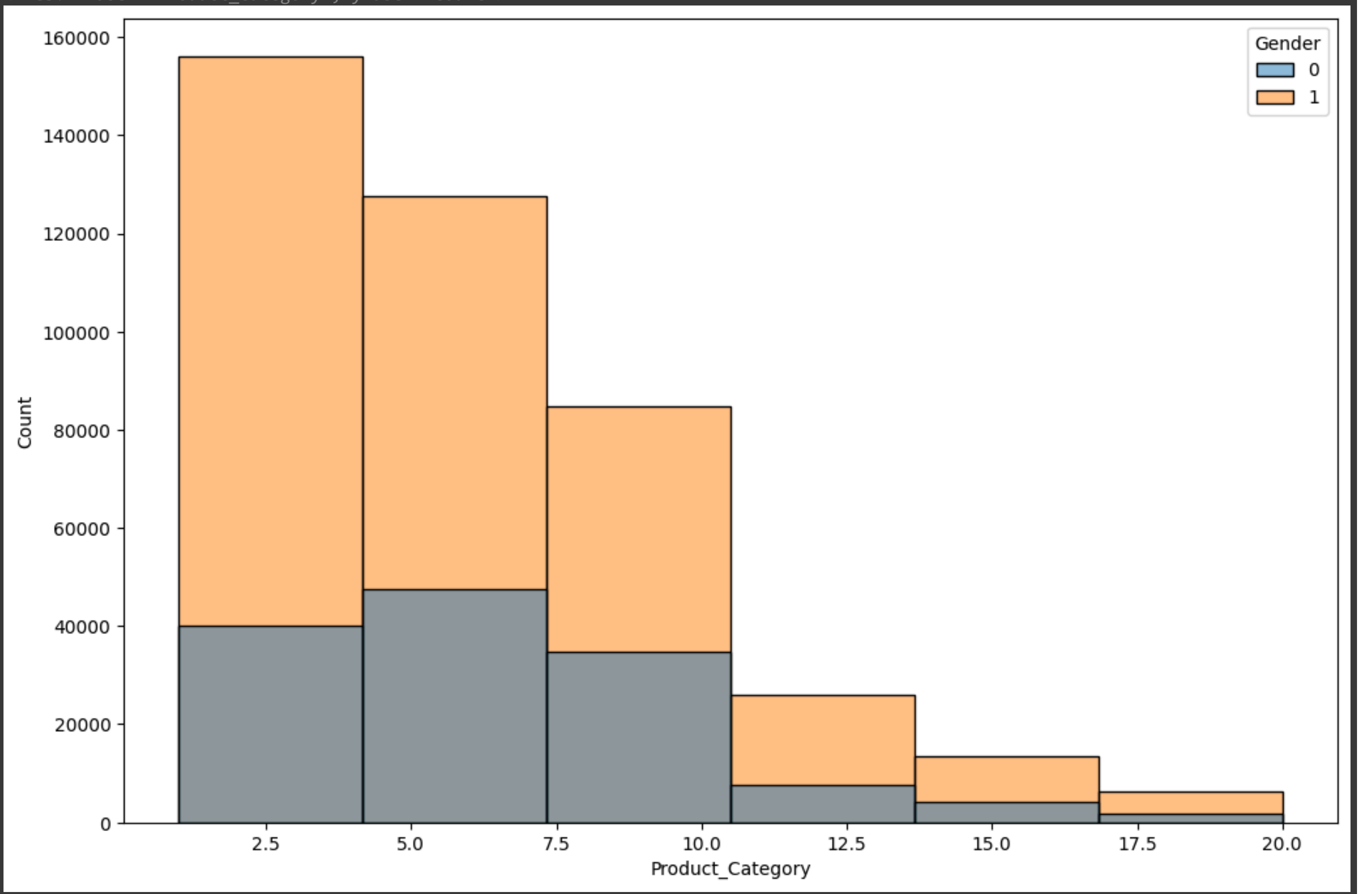


insights ▶ product purchase in all age group againts married and unmarried are almost same, there is no significant difference among them.

c. Are there preferred product categories for different genders?

```
# here Gender -->  0 : Female, 1: Male
plt.figure(figsize=(12,8))
sns.histplot(data=df_copy,x='Product_Category',hue='Gender',binwidth=3)
```

```
<Axes: xlabel='Product_Category', ylabel='Count'>
```



Insights ▶ here product category(2.5) has more user male & female where category (20) has lowest user for male as well female.

---

4. How does gender affect the amount spent? ⏳

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

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

```
male_sample_means=[df[df["Gender"]=="M"].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(male_sample_means)
```

```
9434.919346666666
```

```
Female_sample_means=[df[df["Gender"]=="F"].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(Female_sample_means)
```

```
8726.744033333332
```

From the above,unable to conclude since there is overlap in spending behaviour in both male and female. Hence will increase the sample size to 3000

```
male_sample_means1=[df[df["Gender"]=="M"].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(male_sample_means1)
```

```
9439.547392666665
```

```
Female_sample_means1=[df[df["Gender"]=="F"].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(Female_sample_means1)
```

```
8733.495035333335
```

```
# at 95% sample size for 3000 sample size
np.percentile(male_sample_means1,[2.5,97.5]),np.percentile(Female_sample_means1,[2.5,97.5])
```

```
(array([9261.56901667, 9627.99446667]), array([8569.10198333, 8908.27730833]))
```

a. From the above calculated CLT answer the following questions.

i. Is the confidence interval computed using the entire dataset wider for one of the genders? Why is this the case?

Answer: Confidence interval computed using the entire dataset,we were not able to conclude because of the outliers which pulls the mean value of both male and female to be in the same levels casuing overlap.hence unable to make decision

ii. How is the width of the confidence interval affected by the sample size?

Answer : Performed CI with sample size of 300 for 1000 iterations at 95% and 90% confidence interval,the results were same - there was overlap of levels in both male and female - hence unable to conclude

iii. Do the confidence intervals for different sample sizes overlap?

Answer : Increased sample size to 3000 for 1000 iterations at 95% CI, there was no overlap. at 95% confidence Interval (3000 sample size) we were able to conclude that mean spending of male is more than the female.

iv. How does the sample size affect the shape of the distributions of the means?

Answer : increasing the sample size,decreases the standard error.

---

6. How does Age affect the amount spent? 📽

```
df.groupby("Age")[["Purchase"]].describe().reset_index()
```

	Age	Purchase							
		count	mean	std	min	25%	50%	75%	max
0	0-17	15102.0	8933.464640	5111.114046	12.0	5328.0	7986.0	11874.0	23955.0
1	18-25	99660.0	9169.663606	5034.321997	12.0	5415.0	8027.0	12028.0	23958.0
2	26-35	219587.0	9252.690633	5010.527303	12.0	5475.0	8030.0	12047.0	23961.0
3	36-45	110013.0	9331.350695	5022.923879	12.0	5876.0	8061.0	12107.0	23960.0
4	46-50	45701.0	9208.625697	4967.216367	12.0	5888.0	8036.0	11997.0	23960.0
5	51-55	38501.0	9534.808031	5087.368080	12.0	6017.0	8130.0	12462.0	23960.0
6	55+	21504.0	9336.280459	5011.493996	12.0	6018.0	8105.5	11932.0	23960.0

```
Age=['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']
Age_sample_means1=[df[df["Age"]=='0-17'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
Age_sample_means2=[df[df["Age"]=='18-25'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
Age_sample_means3=[df[df["Age"]=='26-35'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
Age_sample_means4=[df[df["Age"]=='36-45'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
Age_sample_means5=[df[df["Age"]=='46-50'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
Age_sample_means6=[df[df["Age"]=='51-55'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
Age_sample_means7=[df[df["Age"]=='55+'].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
```

```
np.mean(Age_sample_means1),np.mean(Age_sample_means2),np.mean(Age_sample_means3),np.mean(Age_sample_means4),np.mean(Age_sample_means5),ı
```

```
(8928.660600000001,
 9170.23251,
 9249.387606666665,
 9339.857473333333,
 9217.058433333334,
```

```
 9544.250853333333,
 9330.799053333332)
```

```
#95 % CI
np.percentile(Age_sample_means1,[2.5,97.5]),np.percentile(Age_sample_means2,[2.5,97.5]),np.percentile(Age_sample_means3,[2.5,97.5]),np.p
```

```
(array([8381.11125   , 9447.99991667]),
 array([8619.80958333, 9742.40691667]),
 array([8723.15175   , 9789.99641667]),
 array([8762.12891667, 9900.0555    ]),
 array([8618.73241667, 9807.83216667]),
 array([ 8971.34691667, 10080.5555    ]),
 array([8791.31791667, 9888.41033333]))
```

```
#90 % CI
np.percentile(Age_sample_means1,[5,95]),np.percentile(Age_sample_means2,[5,95]),np.percentile(Age_sample_means3,[5,95]),np.percentile(Ag
```

```
(array([8468.85483333, 9374.67083333]),
 array([8689.6475    , 9643.77716667]),
 array([8798.5615    , 9704.65416667]),
 array([8851.5975    , 9814.10133333]),
 array([8736.543, 9676.555]),
 array([ 9059.5315, 10027.9485]),
 array([8859.31466667, 9808.69733333]))
```

From the above,unable to conclude since there is overlap in spending behaviour in all age groups. Hence will increase the sample size to 3000

```
A_sample_means1=[df[df["Age"]=='0-17'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
A_sample_means2=[df[df["Age"]=='18-25'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
A_sample_means3=[df[df["Age"]=='26-35'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
A_sample_means4=[df[df["Age"]=='36-45'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
A_sample_means5=[df[df["Age"]=='46-50'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
A_sample_means6=[df[df["Age"]=='51-55'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
A_sample_means7=[df[df["Age"]=='55+'].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
```

```
np.mean(A_sample_means1),np.mean(A_sample_means2),np.mean(A_sample_means3),np.mean(A_sample_means4),np.mean(A_sample_means5),np.mean(A_s
```

```
(8933.779318333334,
 9167.832130333334,
 9251.37171333333,
 9330.512127333332,
 9212.469621,
 9533.595188000001,
 9336.134118999998)
```

```
#95 % CI
np.percentile(A_sample_means1,[2.5,97.5]),np.percentile(A_sample_means2,[2.5,97.5]),np.percentile(A_sample_means3,[2.5,97.5]),np.percent
```

```
(array([8736.89369167, 9113.38751667]),
 array([8984.20313333, 9346.65355   ]),
 array([9076.78334167, 9432.568375  ]),
 array([9141.04623333, 9507.365625  ]),
 array([9041.59281667, 9390.91395833]),
 array([9357.07371667, 9722.09941667]),
 array([9152.45346667, 9509.61113333]))
```

```
#90 % CI
np.percentile(A_sample_means1,[5,95]),np.percentile(A_sample_means2,[5,95]),np.percentile(A_sample_means3,[5,95]),np.percentile(A_sample
```

```
(array([8774.09695    , 9083.98896667]),
 array([9017.49946667, 9325.86713333]),
 array([9096.6733     , 9398.13213333]),
 array([9173.99688333, 9481.8575    ]),
 array([9062.26813333, 9366.0822    ]),
 array([9376.70981667, 9691.95811667]),
 array([9180.90153333, 9482.94378333]))
```

From the above,unable to conclude since there is overlap in spending behaviour in all age groups. Hence will increase the sample size to 30000

```
AA_sample_means1=[df[df["Age"]=='0-17'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
AA_sample_means2=[df[df["Age"]=='18-25'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
AA_sample_means3=[df[df["Age"]=='26-35'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
AA_sample_means4=[df[df["Age"]=='36-45'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
AA_sample_means5=[df[df["Age"]=='46-50'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
AA_sample_means6=[df[df["Age"]=='51-55'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
AA_sample_means7=[df[df["Age"]=='55+'].sample(30000,replace=True)["Purchase"].mean() for i in range(1000)]
```

```
np.mean(AA_sample_means1),np.mean(AA_sample_means2),np.mean(AA_sample_means3),np.mean(AA_sample_means4),np.mean(AA_sample_means5),np.mea
```

```
(8932.6787742,
 9168.819344633333,
 9252.355721000002,
 9330.802069233332,
 9208.575406366666,
 9535.306462533332,
 9335.751921799998)
```

```
#95 % CI
np.percentile(AA_sample_means1,[2.5,97.5]),np.percentile(AA_sample_means2,[2.5,97.5]),np.percentile(AA_sample_means3,[2.5,97.5]),np.perc
```

```
(array([8877.43594917, 8988.55435833]),
 array([9113.23343   , 9225.90841417]),
 array([9196.03460667, 9306.19355583]),
 array([9275.97633583, 9387.67510833]),
 array([9152.94200917, 9269.64457583]),
 array([9480.28308   , 9593.50839333]),
 array([9281.1688025 , 9391.11868083]))
```

```
#90 % CI
np.percentile(AA_sample_means1,[5,95]),np.percentile(AA_sample_means2,[5,95]),np.percentile(AA_sample_means3,[5,95]),np.percentile(AA_sa
```

```
(array([8884.96937667, 8979.989435  ]),
 array([9121.00625   , 9216.90182167]),
 array([9206.70086667, 9296.394045  ]),
 array([9282.534965  , 9378.88702333]),
 array([9160.00717167, 9256.69243333]),
 array([9488.14569   , 9580.46127833]),
 array([9291.33468   , 9383.33637833]))
```

```
#85 % CI
np.percentile(AA_sample_means1,[7.5,92.5]),np.percentile(AA_sample_means2,[7.5,92.5]),np.percentile(AA_sample_means3,[7.5,92.5]),np.perc
```

```
(array([8890.37929583, 8974.47508833]),
 array([9125.79644167, 9207.40635083]),
 array([9211.959955, 9292.71687 ]),
 array([9288.5599675, 9371.474745 ]),
 array([9164.70924667, 9250.23643583]),
 array([9494.61040917, 9576.40409   ]),
 array([9295.28756333, 9377.93691917]))
```

```
#80 % CI
np.percentile(AA_sample_means1,[10,90]),np.percentile(AA_sample_means2,[10,90]),np.percentile(AA_sample_means3,[10,90]),np.percentile(A/
```

```
(array([8894.88918667, 8969.65257   ]),
 array([9130.91647   , 9203.14649333]),
 array([9217.85872667, 9288.91247667]),
 array([9293.97318, 9366.85249]),
 array([9169.84675667, 9245.85481333]),
 array([9499.49069   , 9571.21373333]),
 array([9299.16034, 9373.97229]))
```

a. From the above calculated CLT answer the following questions.

i. Is the confidence interval computed using the entire dataset wider for one of the genders? Why is this the case?

Answer : Performed CI with sample size of 300 for 1000 iterations at 95%,90% confidence interval,the results were same - there was overlap of spending in all age groups - hence unable to conclude.

ii. How is the width of the confidence interval affected by the sample size?

Answer : Increased sample size to 3000 for 1000 iterations at 95%,90% CI - there was overlap of spending in all age groups - hence unable to conclude.

iii. Do the confidence intervals for different sample sizes overlap?

Answer : Increased sample size to 30000 for 1000 iterations at 95%,90%,85%,80% CI - not all age groups have overlap.hence the population is the same as sample.

iv. How does the sample size affect the shape of the distributions of the means?

Answer : Hence 95% confidence level,the age group 51-55 spends the most and (0-17) the least.netween 18-50 the spending habit is almost the same.

---

7. Create a report 📄

a. Report whether the confidence intervals for the average amount spent by males and females (computed using all the data) overlap. How can Walmart leverage this conclusion to make changes or improvements?

Answer : On 95% Confidence Interval, we can conclude that men average spending is more than the females.hence walmart must focus on women products to improve the avg spending either through offers,discounts,advertising to increase footfalls.

b. Report whether the confidence intervals for the average amount spent by married and unmarried (computed using all the data) overlap. How can Walmart leverage this conclusion to make changes or improvements?

```
unmarried_sample_means=[df[df["Marital_Status"]==0].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(unmarried_sample_means)
```

```
9273.522620000002
```

```
married_sample_means=[df[df["Marital_Status"]==1].sample(300,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(married_sample_means)
```

```
9268.917256666666
```

From the above,unable to conclude since there is overlap in spending behaviour in both male and female. Hence will increase the sample size to 3000

```
unmarried_sample_means1=[df[df["Marital_Status"]==0].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(unmarried_sample_means1)
married_sample_means1=[df[df["Marital_Status"]==1].sample(3000,replace=True)["Purchase"].mean() for i in range(1000)]
np.mean(married_sample_means1)
```

```
9268.917256666666
```

```
# at 95% sample size for 3000 sample size
np.percentile(unmarried_sample_means1,[2.5,97.5]),np.percentile(married_sample_means1,[2.5,97.5])
```

```
(array([9097.13585 , 9447.763975]), array([9077.81754167, 9447.04378333]))
```

Answer : Interms of Marital status, there is no significant change in terms of spending.hence walmart can target the audiences based on these aspect to improve their spending, hence increase in sales for the business.

c. Report whether the confidence intervals for the average amount spent by different age groups (computed using all the data) overlap. How can Walmart leverage this conclusion to make changes or improvements?

Answer: Performed CI with sample size of 300 for 1000 iterations at 95%,90% confidence interval,the results were same - there was overlap of spending in all age groups - hence unable to conclude.

- Interms of Age group, there is slight significant change interms of spending at 80% Confidence interval between the age groups. age group(50-55) spends the most and agegroup(0-17) is the least.overall it remains same between age(18-40).

---

8. Recommendations ✅

1. Walmart should provide some good offers to Women as they usually like to shop more. When they see some nice offers or discounts, they will definetly buy more.
2. Walmart can also launch some Age wise discounts or offer goodies/freebies to customers to attract people from all age groups.
3. Range of products which interest the people should also be diversified.

4. There is good opporuntiy for walmart to club the marital status and age groups to create a criteria levels and do target producting

Start coding or generate with AI.

+ Code + Text