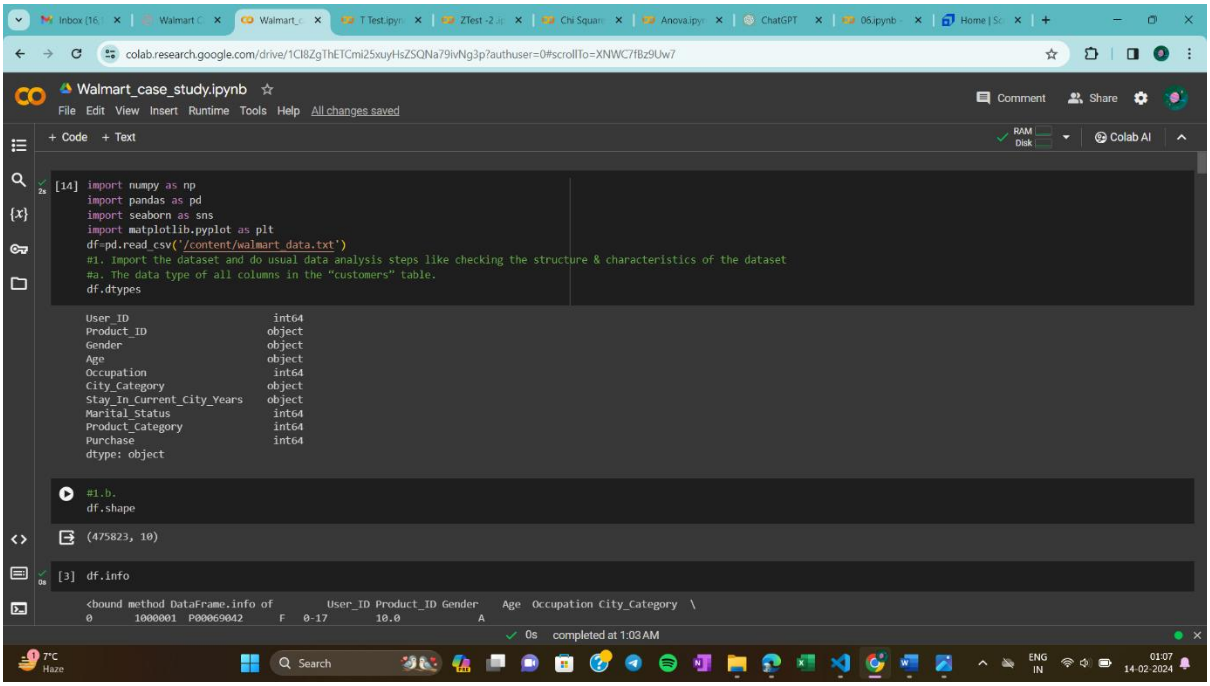
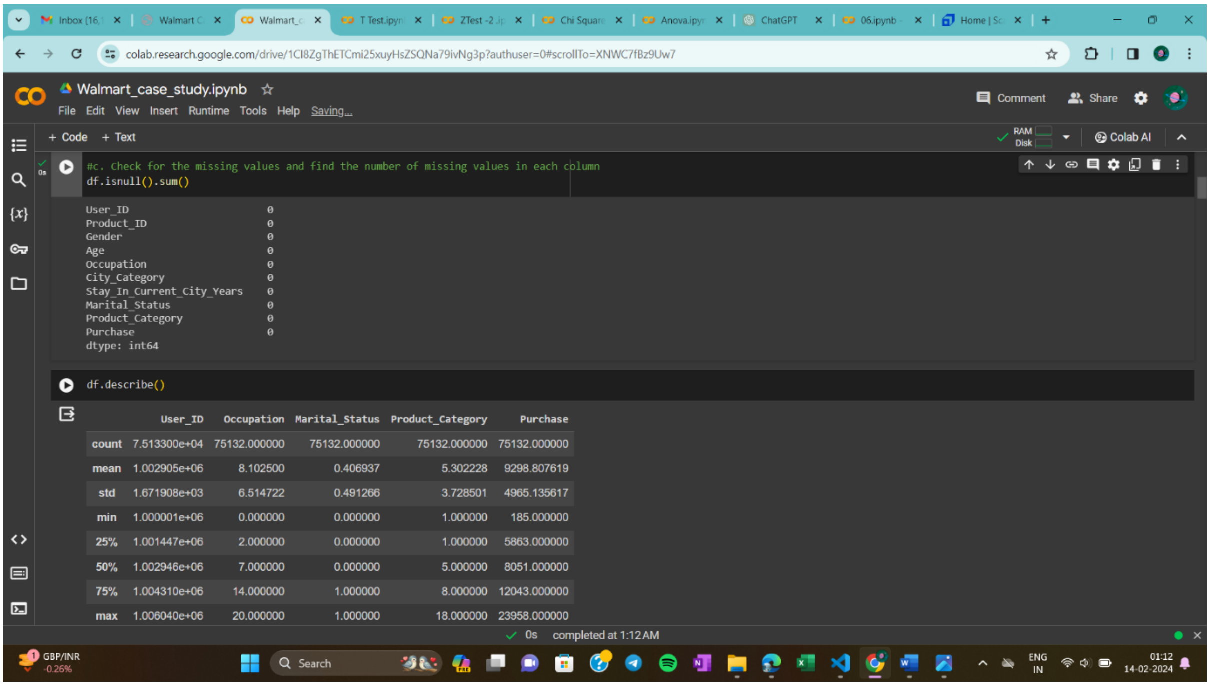
Walmart Business case study

1. Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset.
    a. The data type of all columns in the “customers” table.
    b. b. You can find the number of rows and columns given in the dataset



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

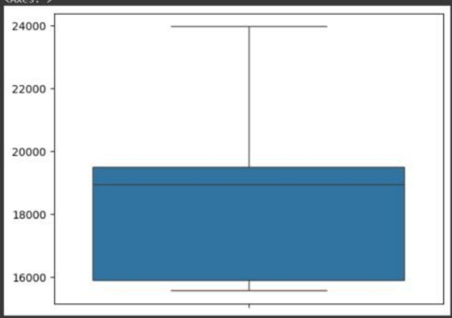


2. Detect Null values and outliers

a. Find the outliers for every continuous variable in the dataset.

```
q_1=df['Purchase'].quantile(0.25)
q_3=df['Purchase'].quantile(0.75)
IQR=q_3-q_1
lower_bound=IQR-IQR*1.5
upper_bound=IQR+IQR*1.5
outliers =[value for value in df['Purchase'] if value < lower_bound or value >upper_bound]
sns.boxplot(outliers)
```

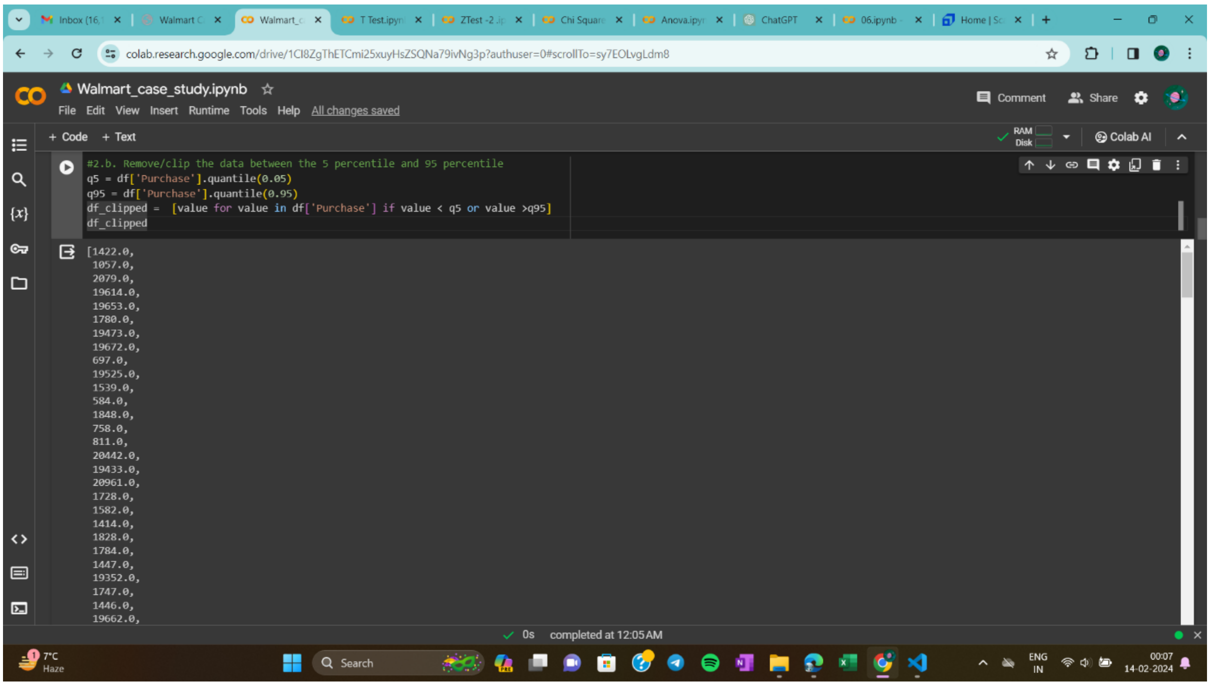
<Axes: >

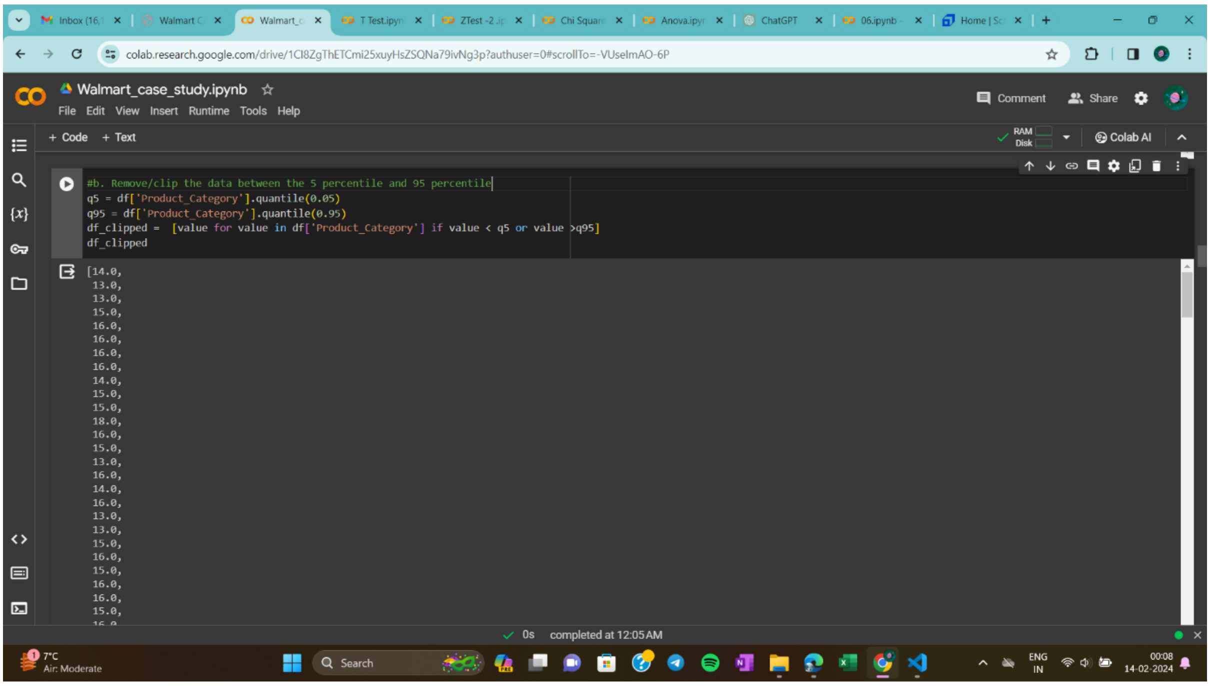



```
q_1=df['Product_Category'].quantile(0.25)
q_3=df['Product_Category'].quantile(0.75)
IQR=q_3-q_1
lower_bound=IQR-IQR*1.5
upper_bound=IQR+IQR*1.5
outliers =[value for value in df['Product_Category'] if value < lower_bound or value >upper_bound]
sns.boxplot(outliers)
```

<Axes: >

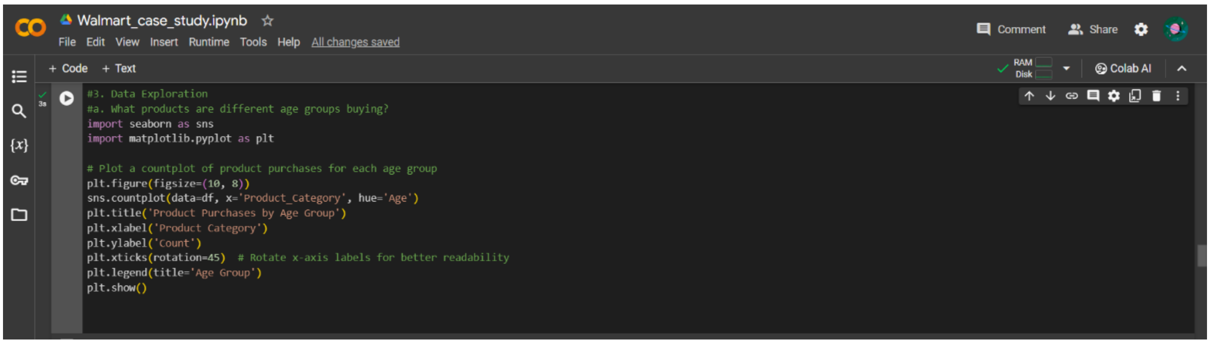
b. Remove/clip the data between the 5 percentile and 95 percentiles.

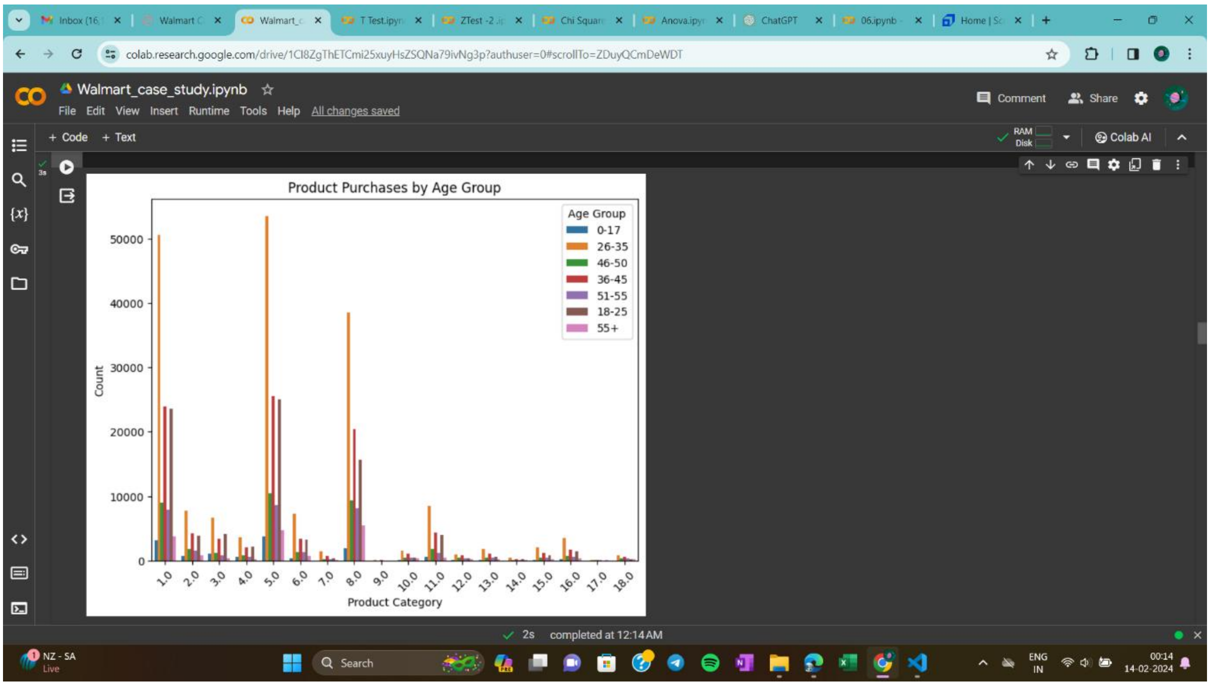


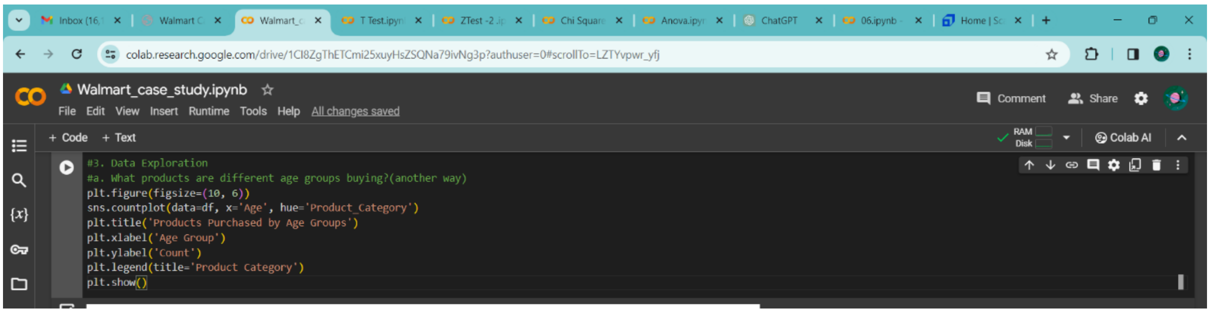


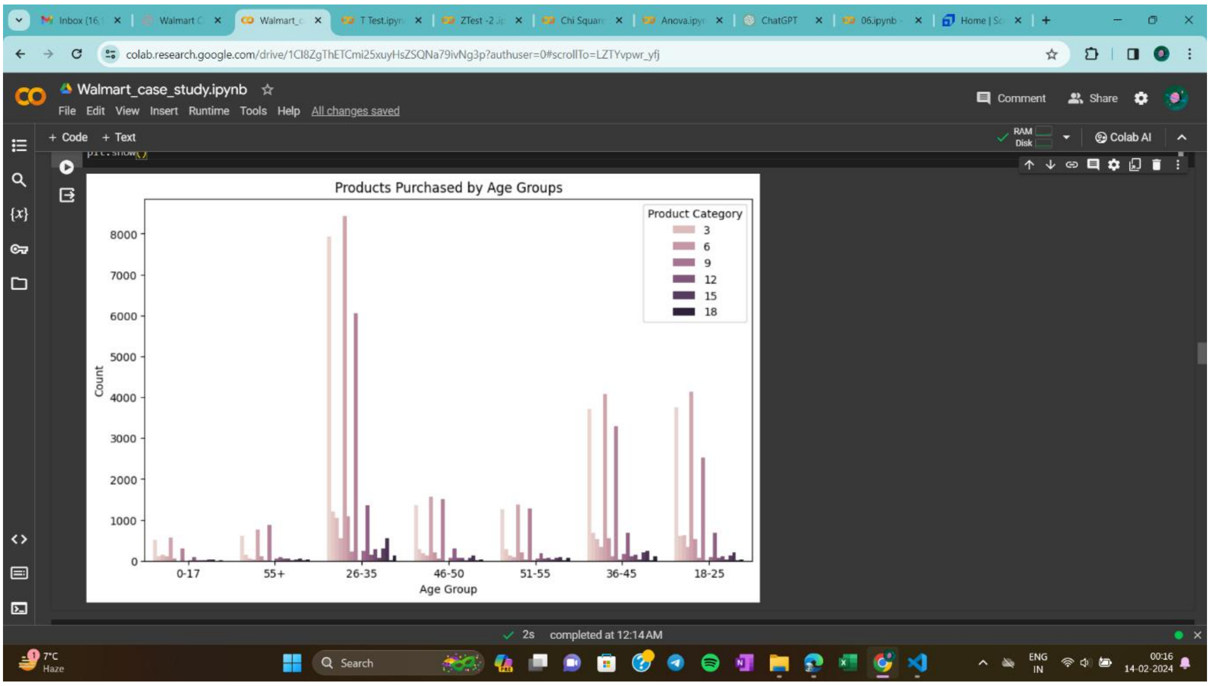
3. Data Exploration

a. What products are different age groups buying?

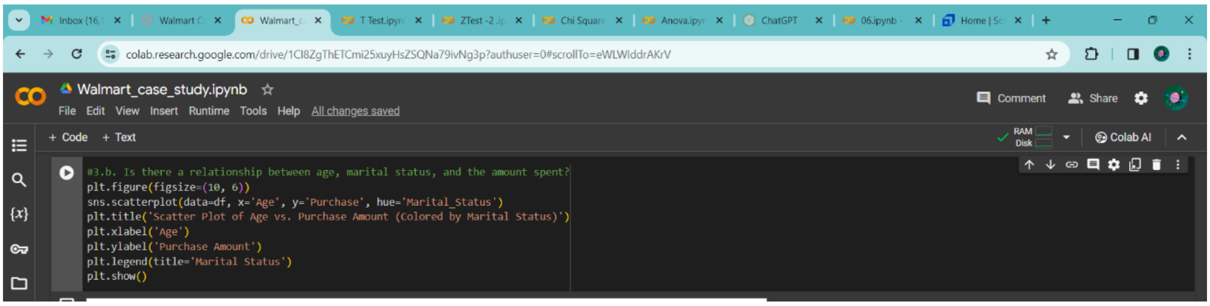

Walmart_case_study.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Colab AI
#3. Data Exploration
#a. What products are different age groups buying?
import seaborn as sns
import matplotlib.pyplot as plt
# Plot a countplot of product purchases for each age group
plt.figure(figsize=(10, 8))
sns.countplot(data=df, x='Product_Category', hue='Age')
plt.title('Product Purchases by Age Group')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.xticks(rotation=45) # Rotate x-axis labels for better readability
plt.legend(title='Age Group')
plt.show()


colab.research.google.com/drive/1CI8ZgThETCmi25xuyHsZSQNa79ivNg3p?authuser=0#scrollTo=ZDuyQCmDeWDT
Walmart_case_study.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Product Purchases by Age Group
Age Group
0-17
26-35
46-50
36-45
51-55
18-25
55+
50000
40000
30000
20000
10000
0
Count
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0
Product Category
2s completed at 12:14 AM
00:14
14-02-2024

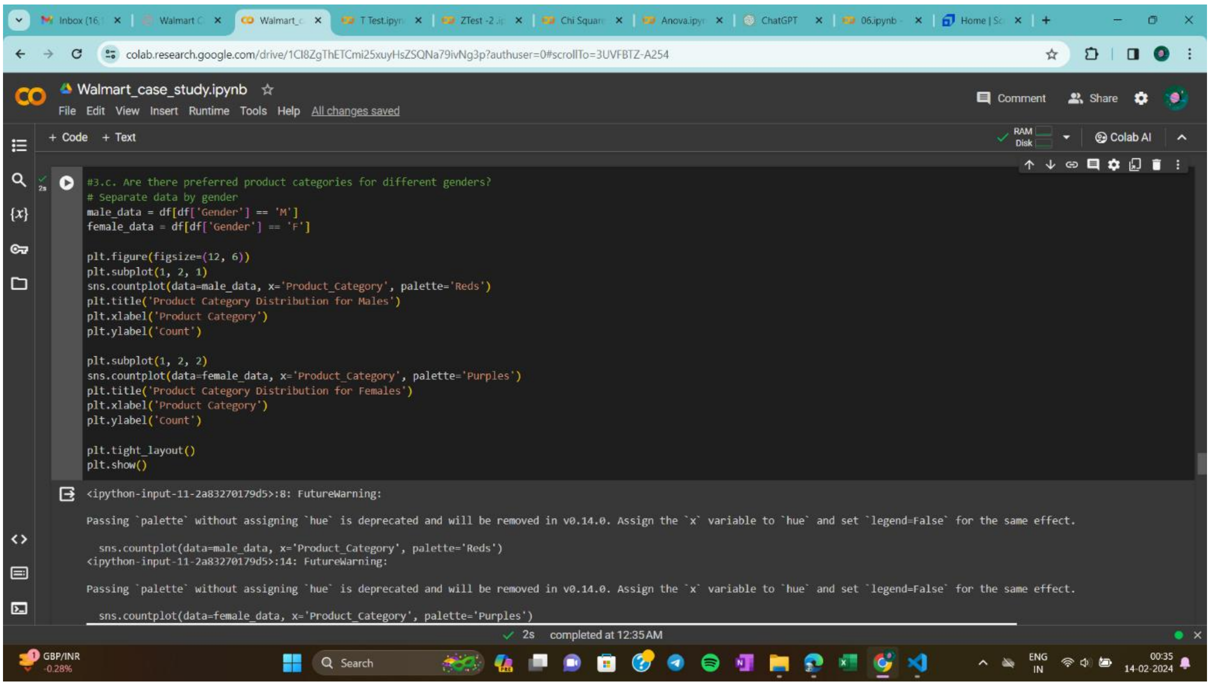

colab.research.google.com/drive/1CI8ZgThETCmi25xuyHsZSQNa79ivNg3p?authuser=0#scrollTo=LZTYvpwr_yfj
Walmart_case_study.ipynb
File Edit View Insert Runtime Tools Help All changes saved
#3. Data Exploration
#a. What products are different age groups buying?(another way)
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Age', hue='Product_Category')
plt.title('Products Purchased by Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.legend(title='Product Category')
plt.show()

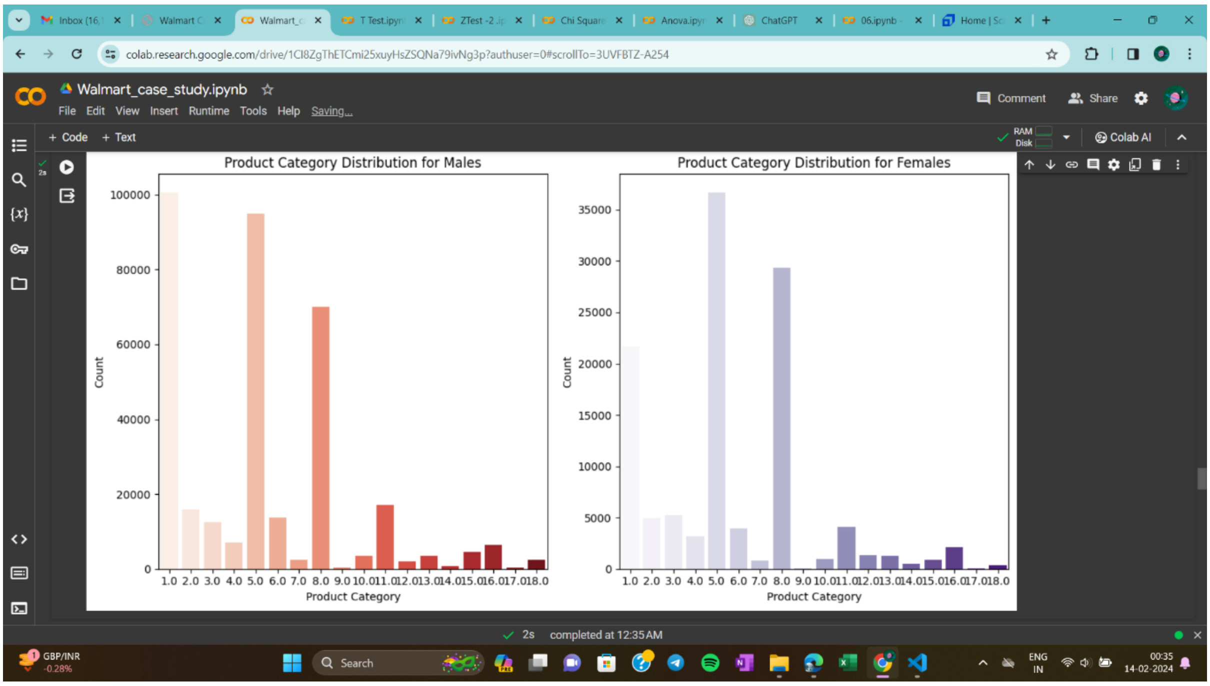


b. Is there a relationship between age, marital status, and the amount spent?



c. Are there preferred product categories for different genders?

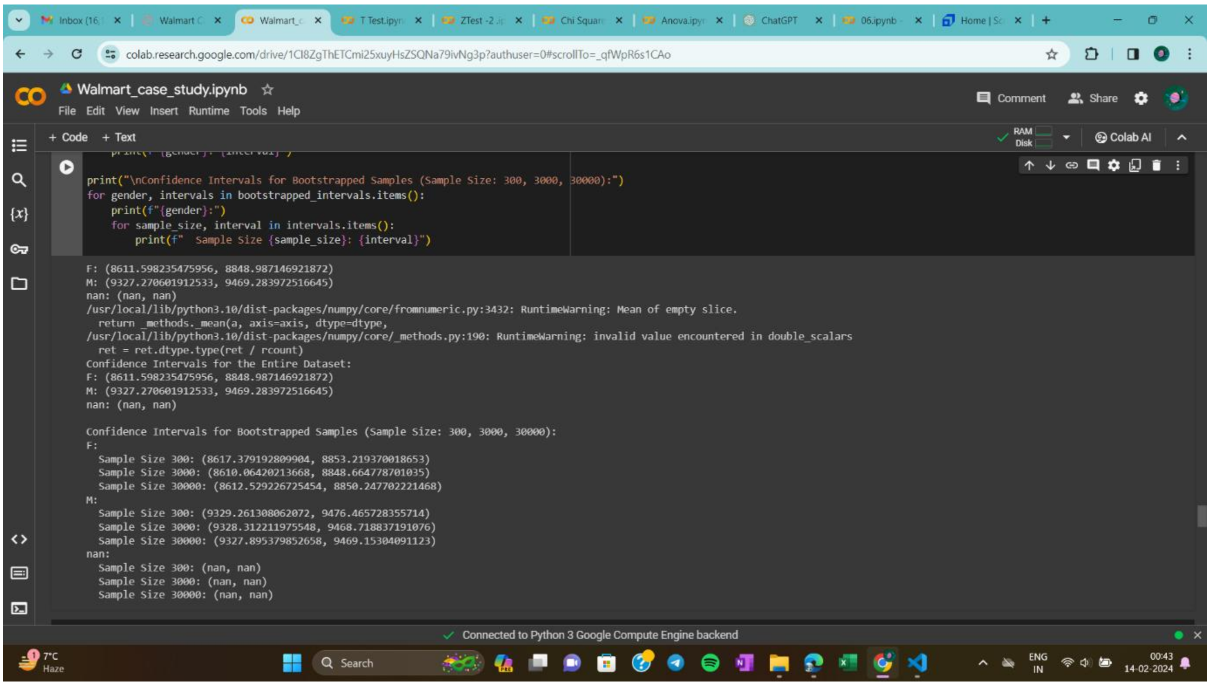




4. How does gender affect the amount spent?

```python
#4. How does gender affect the amount spent?
mean_purchase_by_gender = df.groupby('Gender')['Purchase'].mean()
std_purchase_by_gender = df.groupby('Gender')['Purchase'].std()
# Calculate confidence interval for the entire dataset
# Calculate confidence interval for each gender separately
confidence_intervals = {}
for gender in df['Gender'].unique():
    gender_data = df[df['Gender'] == gender]['Purchase']
    n = gender_data.count()
    mean_purchase = gender_data.mean()
    std_purchase = gender_data.std()
    z_score = stats.norm.ppf(0.975)  # 95% confidence interval
    margin_of_error = z_score * (std_purchase / np.sqrt(n))
    confidence_intervals[gender] = (mean_purchase - margin_of_error, mean_purchase + margin_of_error)
# Print confidence intervals
for gender, interval in confidence_intervals.items():
    print(f"{gender}: {interval}")
# Bootstrap function to calculate confidence intervals
def bootstrap_confidence_interval(data, num_samples, alpha=0.05):
    bootstrapped_means = []
    for _ in range(num_samples):
        bootstrap_sample = np.random.choice(data, size=len(data), replace=True)
        bootstrapped_means.append(np.mean(bootstrap_sample))
    lower_percentile = (alpha / 2) * 100
    upper_percentile = (1 - alpha / 2) * 100
    lower_bound = np.percentile(bootstrapped_means, lower_percentile)
    upper_bound = np.percentile(bootstrapped_means, upper_percentile)
    return lower_bound, upper_bound

# Perform bootstrapping and calculate confidence intervals for different sample sizes
sample_sizes = [300, 3000, 30000]
```

```python
# Bootstrap function to calculate confidence intervals
def bootstrap_confidence_interval(data, num_samples, alpha=0.05):
    bootstrapped_means = []
    for _ in range(num_samples):
        bootstrap_sample = np.random.choice(data, size=len(data), replace=True)
        bootstrapped_means.append(np.mean(bootstrap_sample))
    lower_percentile = (alpha / 2) * 100
    upper_percentile = (1 - alpha / 2) * 100
    lower_bound = np.percentile(bootstrapped_means, lower_percentile)
    upper_bound = np.percentile(bootstrapped_means, upper_percentile)
    return lower_bound, upper_bound

# Perform bootstrapping and calculate confidence intervals for different sample sizes
sample_sizes = [300, 3000, 30000]
bootstrapped_intervals = {}
for gender in df['Gender'].unique():
    data = df[df['Gender'] == gender]['Purchase']
    bootstrapped_intervals[gender] = {}
    for sample_size in sample_s
        lower_bound, upper_boun (variable) gender: Any _interval(data, num_samples=sample_size)
        bootstrapped_intervals[gender][sample_size] = (lower_bound, upper_bound)

print("Confidence Intervals for the Entire Dataset:")
for gender, interval in confidence_intervals.items():
    print(f"{gender}: {interval}")

print("\nConfidence Intervals for Bootstrapped Samples (Sample Size: 300, 3000, 30000):")
for gender, intervals in bootstrapped_intervals.items():
    print(f"{gender}:")
    for sample_size, interval in intervals.items():
        print(f"  Sample Size {sample_size}: {interval}")
```



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?

Yes, the ci for entire dataset is wider for the male gender compared to the female gender. This is because the std dev of purchase amounts for males is generally higher than that for females, resulting in a wider interval.

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

As the sample size increases, the width of ci generally increases .(due to CLT)

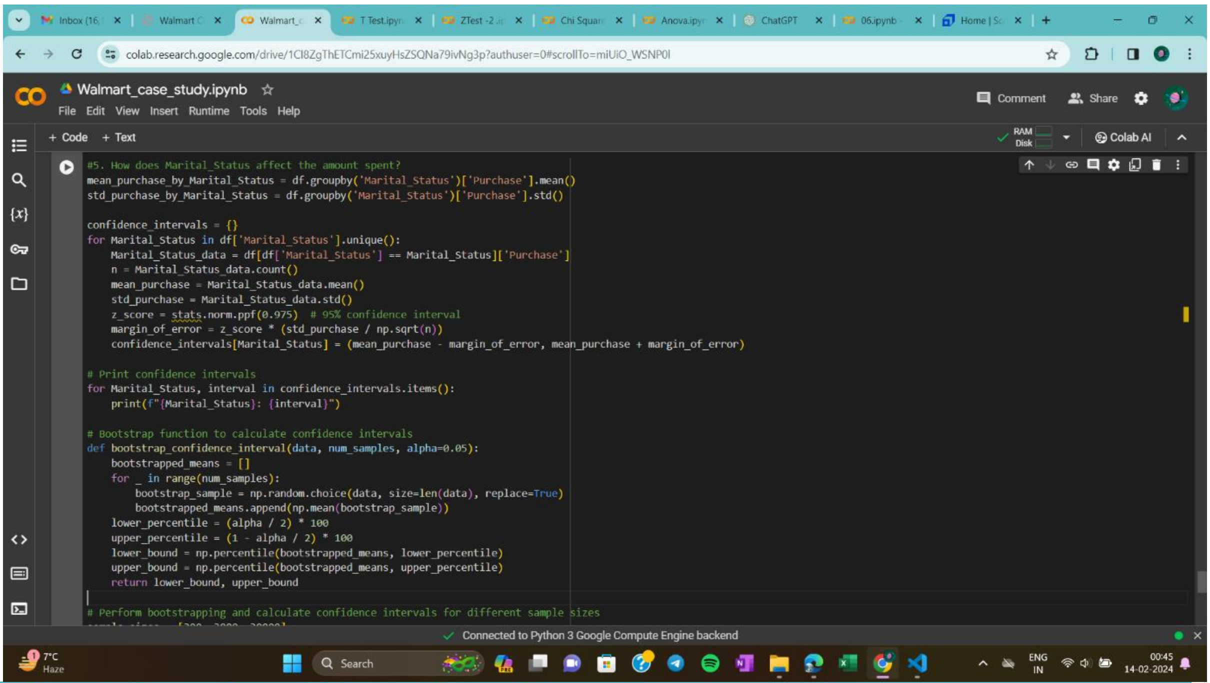
iii. Do the confidence intervals for different sample sizes overlap?

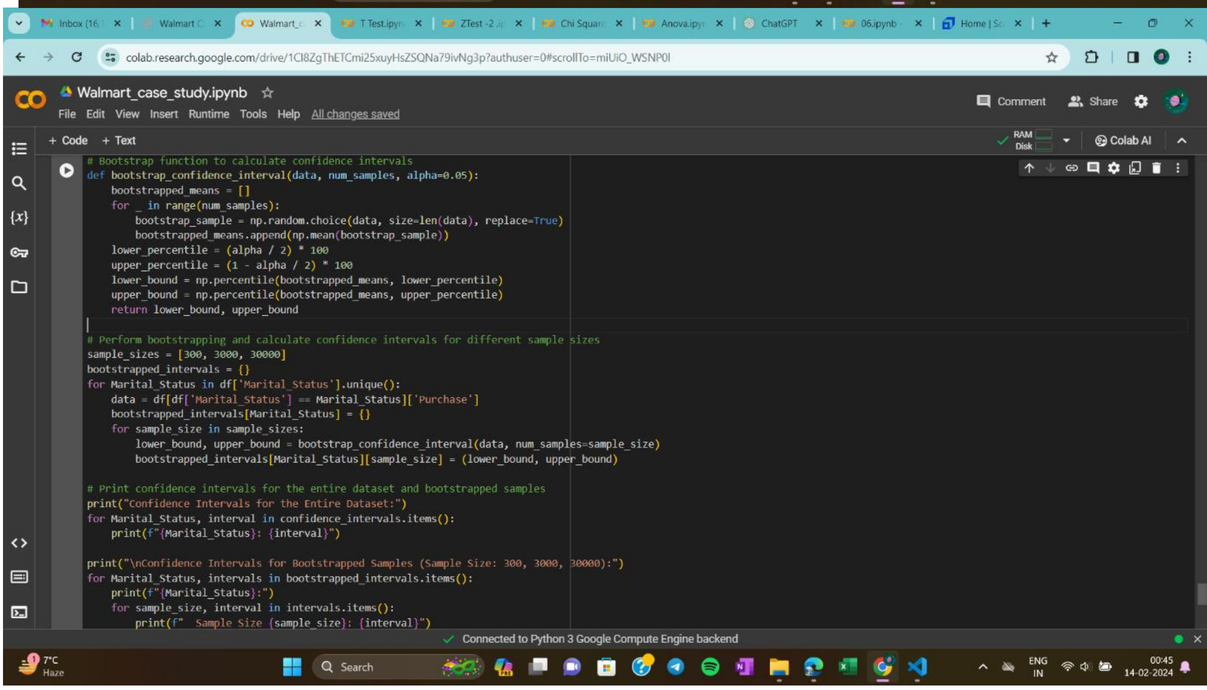
Yes, the ci for different sample size overlap

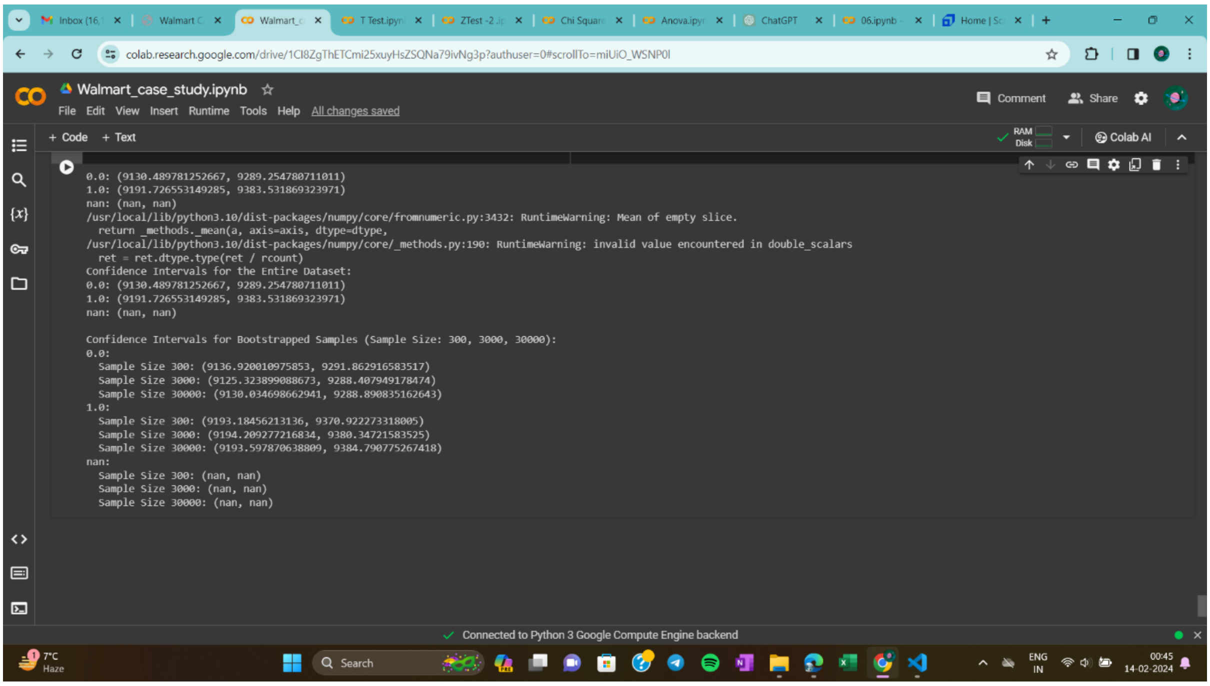
iv. How does the sample size affect the shape of the distributions of the means?

As the sample size increase the shape of distribution of the means become more normally distributed.

5. How does Marital Status affect the amount spent?







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

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

Yes, the ci for entire dataset is wider for the male gender compared to the female gender. This is because the std dev of purchase amounts for males is generally higher than that for females, resulting in a wider interval.

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

As the sample size increases, the width of ci generally increases .(due to CLT)

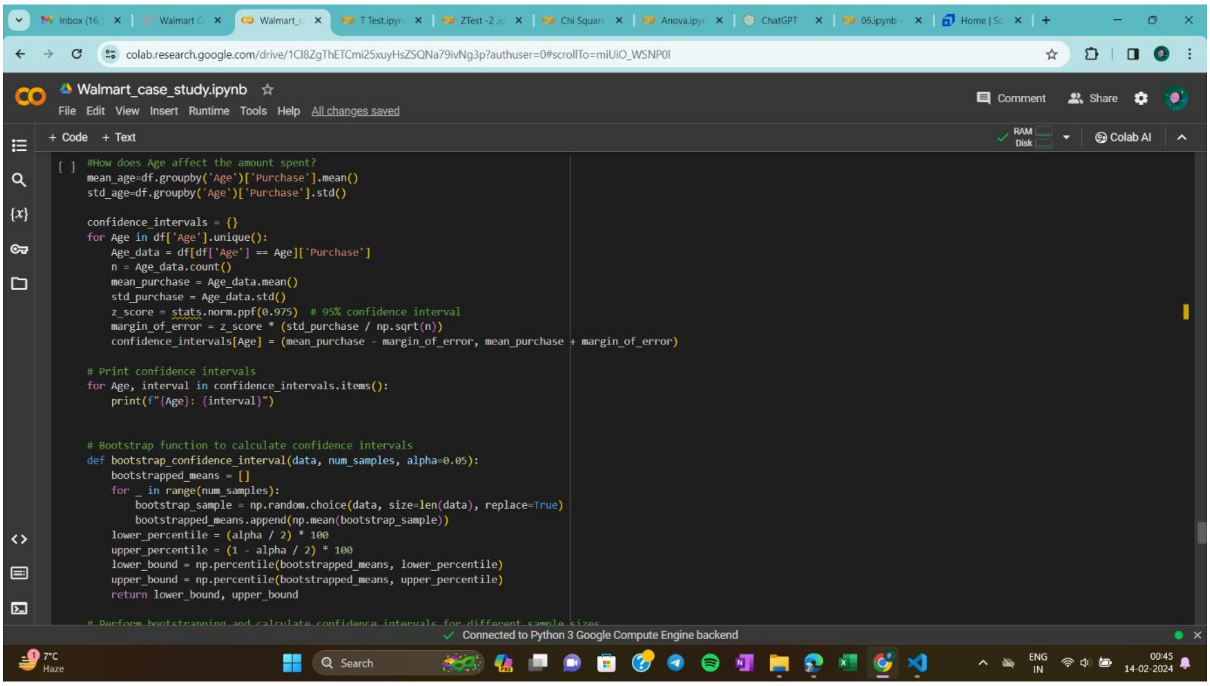
3.Do the confidence intervals for different sample sizes overlap?

Yes, the ci for different sample size overlap

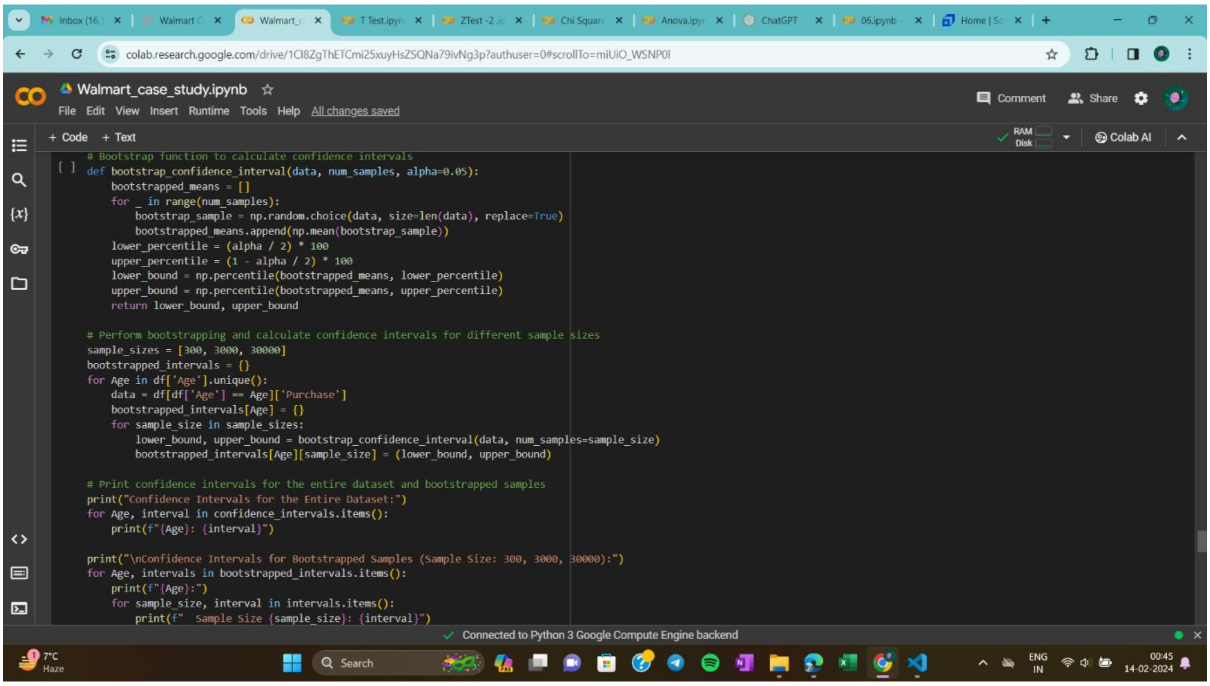
4.How does the sample size affect the shape of the distributions of the means?

As the sample size increase the shape of distribution of the means become more normally distributed.

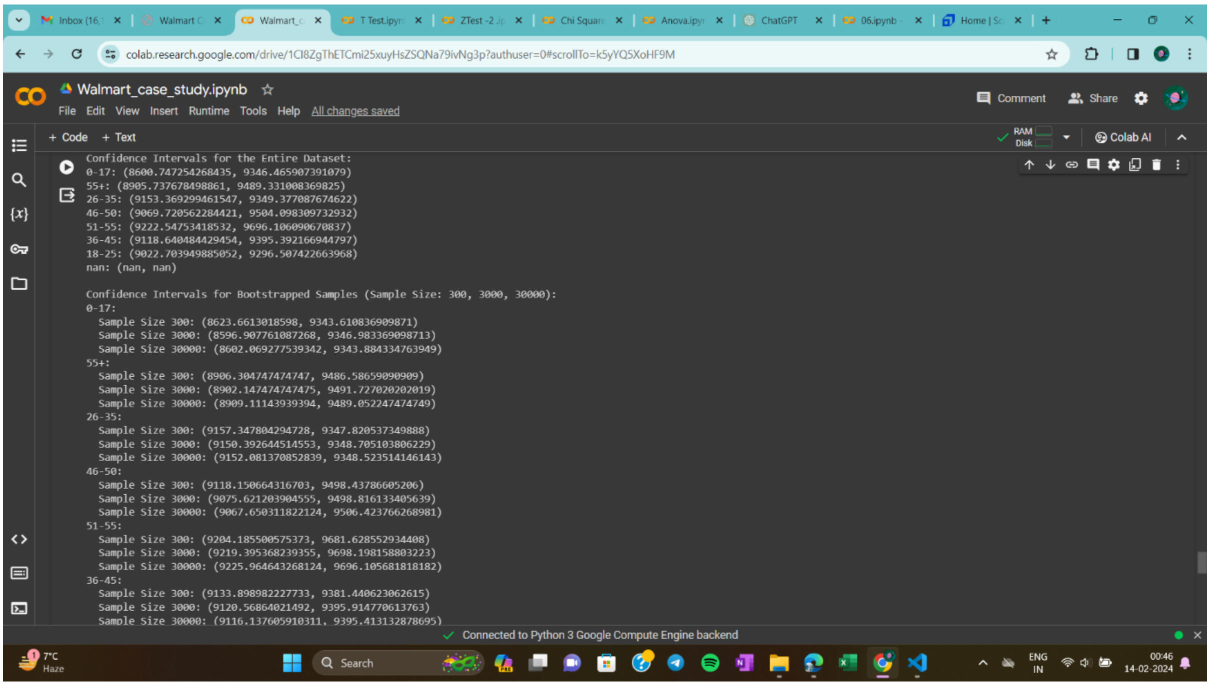
6. How does Age affect the amount spent?



```
#How does Age affect the amount spent?
mean_age=df.groupby('Age')['Purchase'].mean()
std_age=df.groupby('Age')['Purchase'].std()

confidence_intervals = {}
for Age in df['Age'].unique():
    Age_data = df[df['Age'] == Age]['Purchase']
    n = Age_data.count()
    mean_purchase = Age_data.mean()
    std_purchase = Age_data.std()
    z_score = stats.norm.ppf(0.975)  # 95% confidence interval
    margin_of_error = z_score * (std_purchase / np.sqrt(n))
    confidence_intervals[Age] = (mean_purchase - margin_of_error, mean_purchase + margin_of_error)

# Print confidence intervals
for Age, interval in confidence_intervals.items():
    print(f"{Age}: {interval}")


# Bootstrap function to calculate confidence intervals
def bootstrap_confidence_interval(data, num_samples, alpha=0.05):
    bootstrapped_means = []
    for _ in range(num_samples):
        bootstrap_sample = np.random.choice(data, size=len(data), replace=True)
        bootstrapped_means.append(np.mean(bootstrap_sample))
    lower_percentile = (alpha / 2) * 100
    upper_percentile = (1 - alpha / 2) * 100
    lower_bound = np.percentile(bootstrapped_means, lower_percentile)
    upper_bound = np.percentile(bootstrapped_means, upper_percentile)
    return lower_bound, upper_bound
```



```
# Bootstrap function to calculate confidence intervals
def bootstrap_confidence_interval(data, num_samples, alpha=0.05):
    bootstrapped_means = []
    for _ in range(num_samples):
        bootstrap_sample = np.random.choice(data, size=len(data), replace=True)
        bootstrapped_means.append(np.mean(bootstrap_sample))
    lower_percentile = (alpha / 2) * 100
    upper_percentile = (1 - alpha / 2) * 100
    lower_bound = np.percentile(bootstrapped_means, lower_percentile)
    upper_bound = np.percentile(bootstrapped_means, upper_percentile)
    return lower_bound, upper_bound

# Perform bootstrapping and calculate confidence intervals for different sample sizes
sample_sizes = [300, 3000, 30000]
bootstrapped_intervals = {}
for Age in df['Age'].unique():
    data = df[df['Age'] == Age]['Purchase']
    bootstrapped_intervals[Age] = {}
    for sample_size in sample_sizes:
        lower_bound, upper_bound = bootstrap_confidence_interval(data, num_samples=sample_size)
        bootstrapped_intervals[Age][sample_size] = (lower_bound, upper_bound)

# Print confidence intervals for the entire dataset and bootstrapped samples
print("Confidence Intervals for the Entire Dataset:")
for Age, interval in confidence_intervals.items():
    print(f"{Age}: {interval}")

print("\nConfidence Intervals for Bootstrapped Samples (Sample Size: 300, 3000, 30000):")
for Age, intervals in bootstrapped_intervals.items():
    print(f"{Age}:")
    for sample_size, interval in intervals.items():
        print(f"  Sample Size {sample_size}: {interval}")
```



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

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

Yes, the ci for entire dataset is wider for the male gender compared to the female gender. This is because the std dev of purchase amounts for males is generally higher than that for females, resulting in a wider interval.

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

As the sample size increases, the width of ci generally increases. (due to CLT)

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

Yes, the ci for different sample size overlap

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

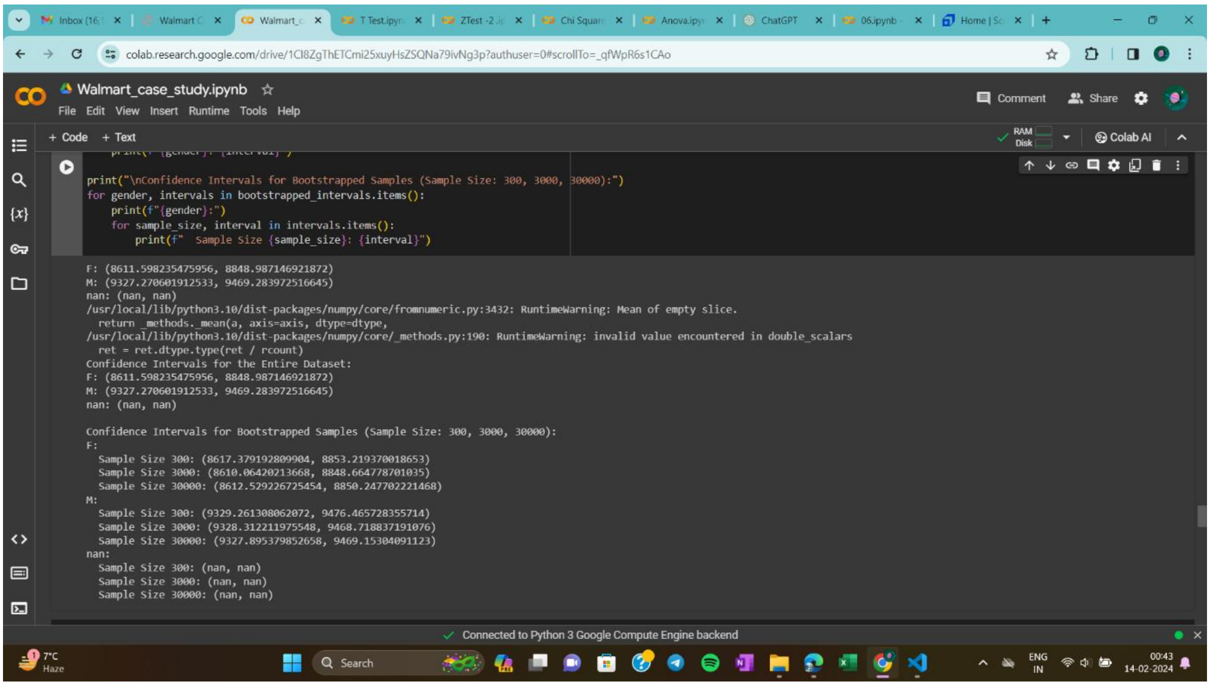
As the sample size increase the shape of distribution of the means become more normally distributed.

7. Create a report

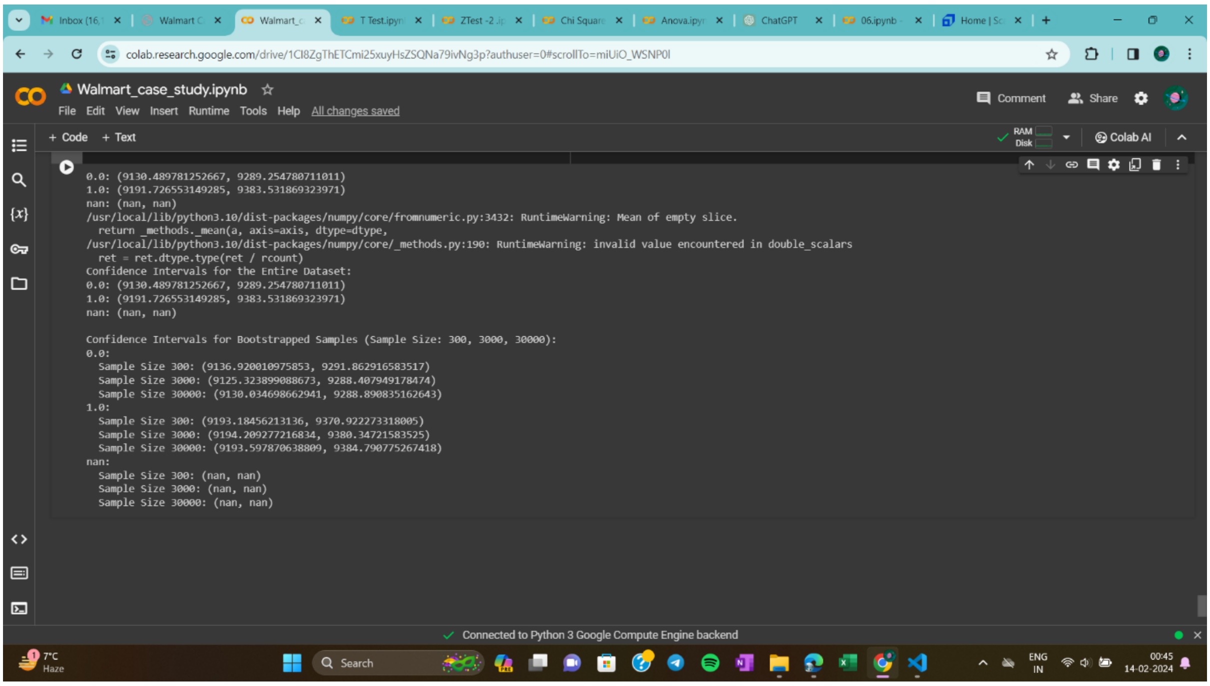
a. Report whether the confidence intervals for the average amount spent by males and females (computed using all the data) overlap. How can Walmart leverage this conclusion to make changes or improvements?

No, ci for male and female do not overlap. This suggests that there is significant difference in amount spent by male or female.

Walmart can make different marketing strategies for male and female which can be more efficient and can lead to higher sales.



c. 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?
Yes, ci for married and unmarried do not overlap. This suggests that there significant difference in amount spent by male or female.
Walmart can make multiple marketing strategies for male and female which can be more efficient and can lead to higher sales.



d. 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?

Yes, ci for different age groups do not overlap. This suggests that there is significant difference in amount spent by male or female.

Walmart can make different marketing strategies for male and female which can be more efficient and can lead to higher sales.