walmart

May 5, 2024

[2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import seaborn as sns
```

[3]:
```python
df = pd.read_csv('walmart_data.txt')
```

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

[4]:
```
   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
```

[5]:
```python
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
```

Insights:There is no null values are present in entire dataset

[6]:
```
df.shape
```

[6]:
```
(550068, 10)
```

Insights: There are 550068 rows and 10 columns

[7]:
```
df.isnull().sum()
```

[7]:
```
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:There is no null values are present in entire dataset

[ ]:
```
columns = ['Occupation','Marital_Status','Product_Category']
df[columns] = df[columns].astype('object')
df.dtypes
```

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

[ ]:
```
gender_counts = df['Gender'].value_counts()
percentage_gender_counts = (gender_counts / len(df)) * 100
print(f"Gender count : \n{gender_counts} \nGender percentage :␣
  ↪\n{percentage_gender_counts}")
```

[ ]:
```
Age_counts = df['Age'].value_counts()
percentage_Age_counts = (Age_counts / len(df)) * 100
print(f"Age count : \n{Age_counts} \nAge percentage :␣
  ↪\n{percentage_Age_counts}")
```

Insights: 75% of users are male and 25% are female. Users ages 26–35 are 40%, users ages 36–45 are 20%, users ages 18–25 are 18%, and very low users ages (0–17 & 55+)are 5%. 35% stay in a city for 1 year, 18% stay in a city for 2 years, 17% stay in a city for 3 years, and 15% stay in a city for 4+ years.

Insights : 26-35 age group is more dominant in other age groups The top people purchasing are in the age range of 26–35. Males are top in purchasing The average purchase is 9263.96 and the maximum purchase is 23961, so the average value is sensitive to outliers, but the fact that the mean is so small compared to the maximum value indicates the maximum value is an outlier.

[9]:
```
unique_category_count = df['Product_Category'].nunique()
print('Unique Product_Category count:',unique_category_count)
```

```
Unique Product_Category count: 20
```

[10]:
```
unique_City_Category_count = df['City_Category'].nunique()
print('Unique City_Category count:',unique_City_Category_count)
```

```
Unique City_Category count: 3
```

[11]:
```
unique_Product_ID_count = df['Product_ID'].nunique()
print('Unique Product_ID count:',unique_Product_ID_count)
```

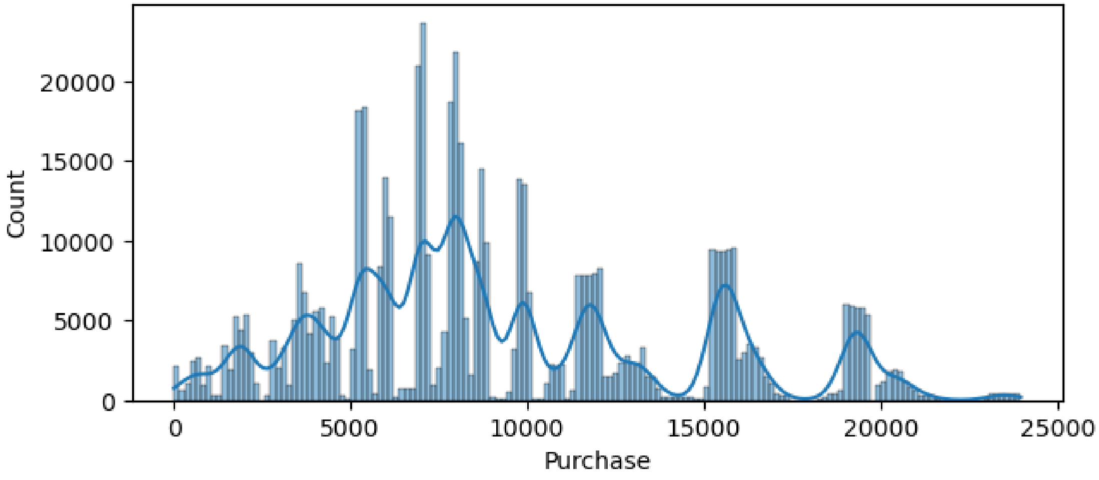
```
Unique Product_ID count: 3631
```

[12]:
```
unique_User_ID_count = df['User_ID'].nunique()
print('Unique User_ID count:',unique_User_ID_count)
```

```
Unique User_ID count: 5891
```

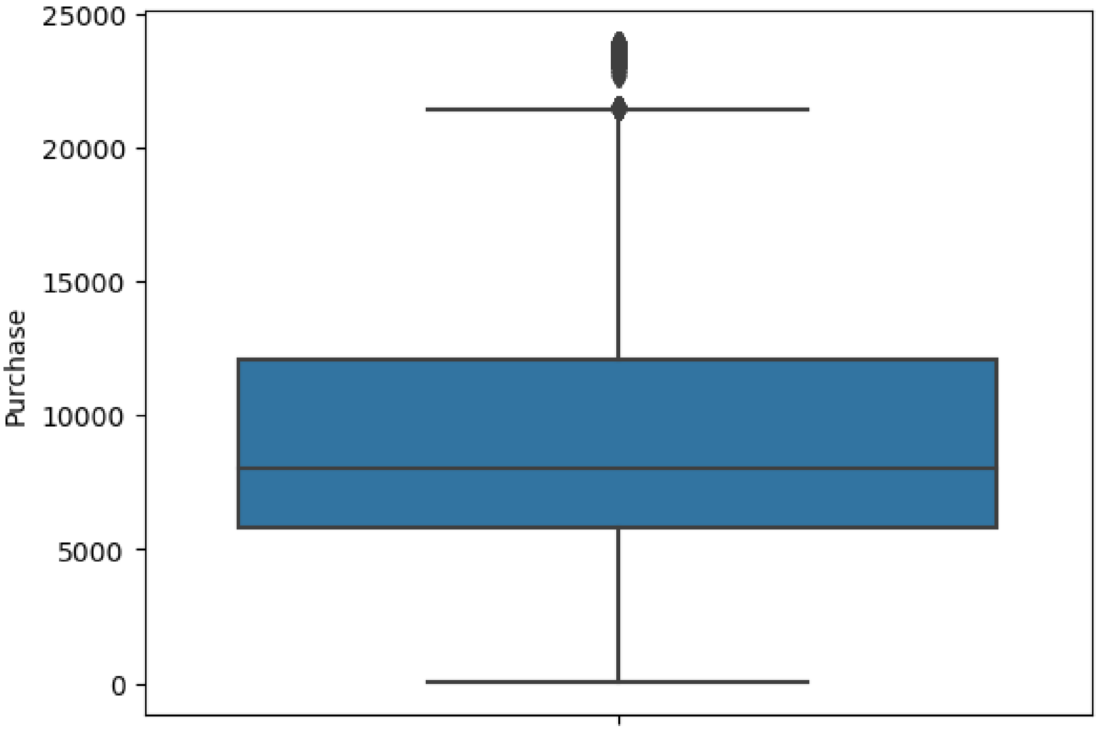
Insights: The total product category count is 20 unique products. The total number of unique city categories is three. The total number of unique product IDs is 3631. The total number of unique user IDs is 5891

[13]:
```
plt.figure(figsize=(7, 3))
sns.histplot(data=df, x='Purchase', kde=True)
plt.show()
```



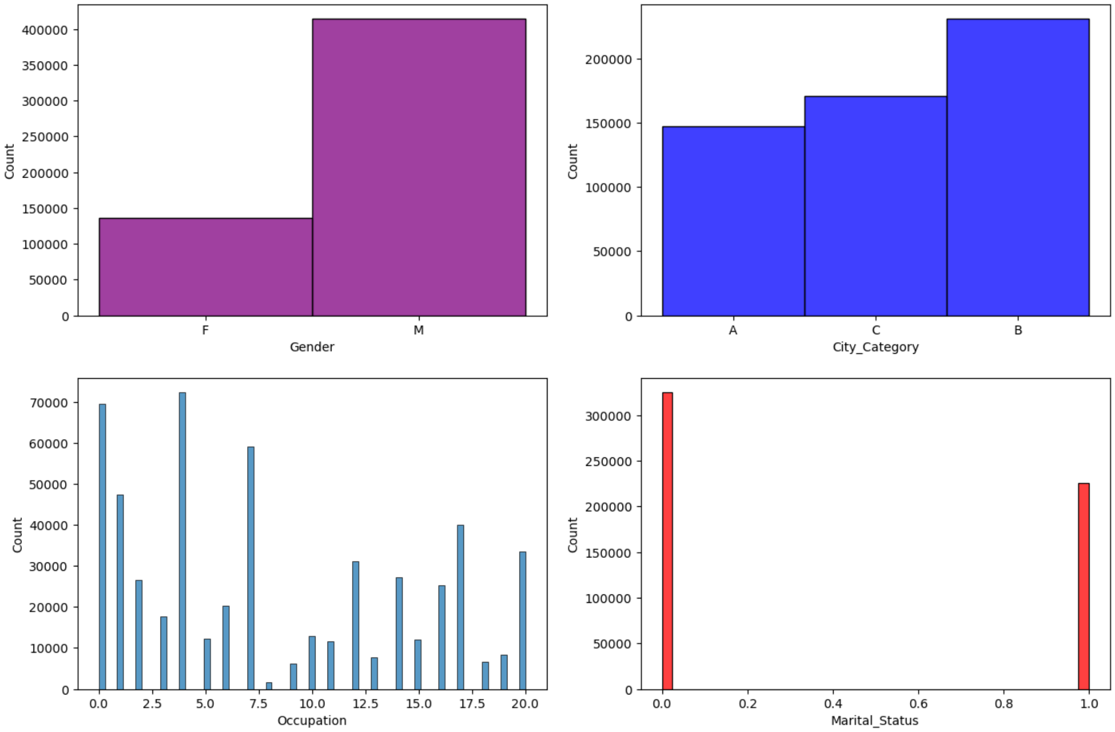
[14]:

```
sns.boxplot(data=df, y='Purchase', orient='v')
plt.show()
```



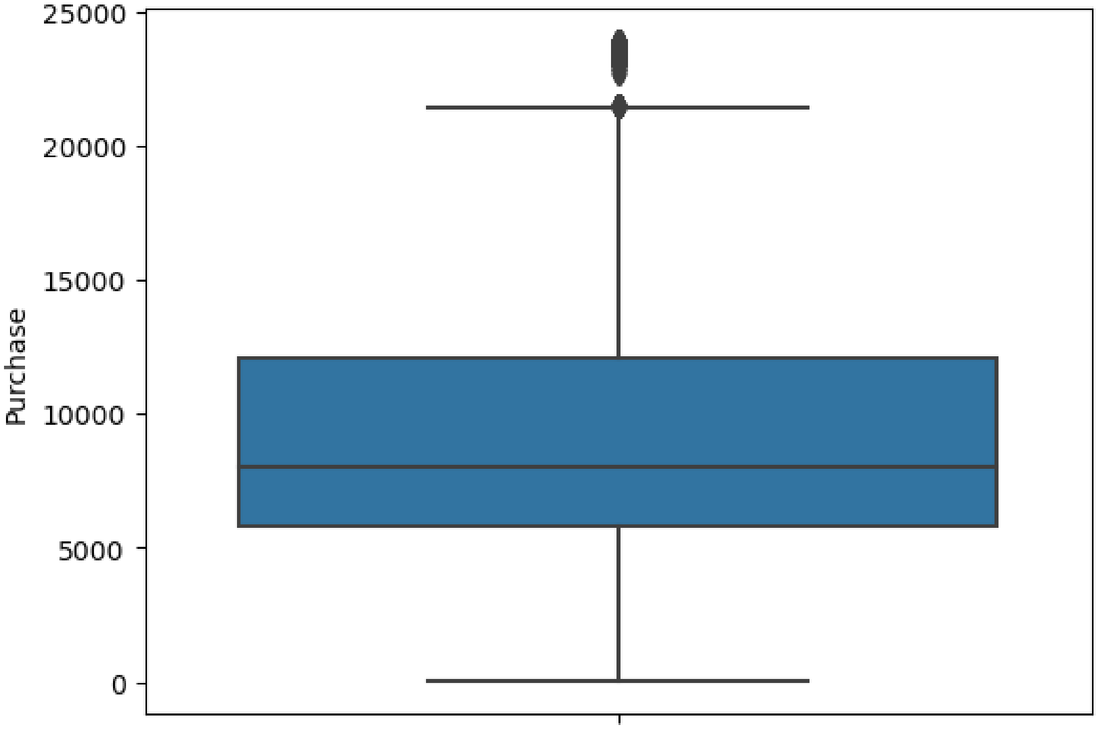
[15]:

```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(15,10))
sns.histplot(data=df, x='Gender', ax=axis[0,0],color = "purple")
sns.histplot(data=df, x='City_Category', ax=axis[0,1],color = "blue")
sns.histplot(data=df, x='Occupation', ax=axis[1,0])
sns.histplot(data=df, x='Marital_Status',ax=axis[1,1],color = "red")
plt.show()
```



[16]:

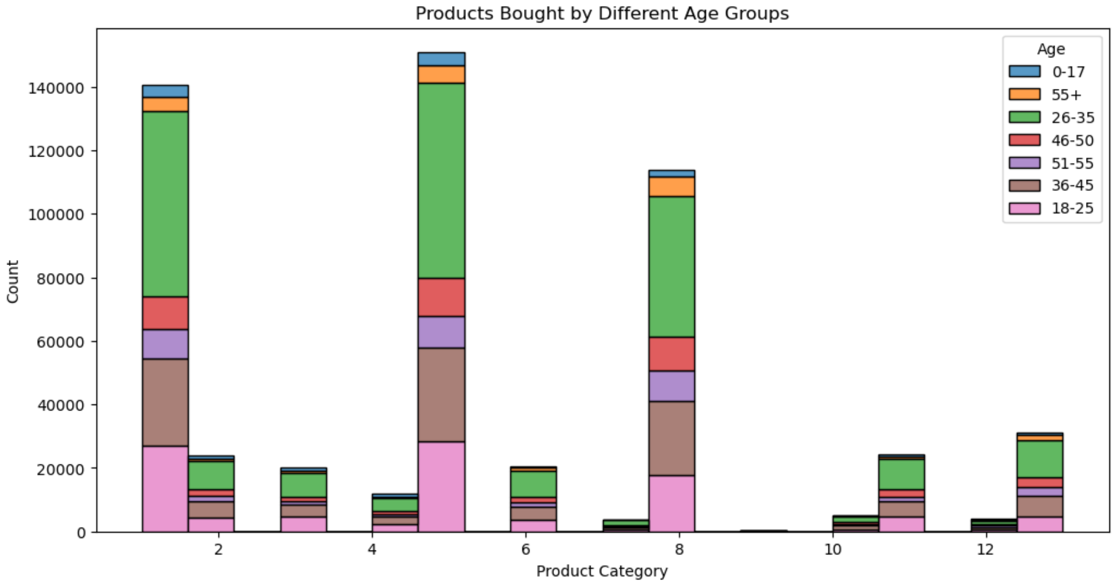
```
#2
sns.boxplot(data=df, y='Purchase', orient='v')
plt.show()
for col in df.select_dtypes(include=['int64', 'float64']):
    p5 = df[col].quantile(0.05)
    p95 = df[col].quantile(0.95)
    df[col] = np.clip(df[col], p5, p95)
```

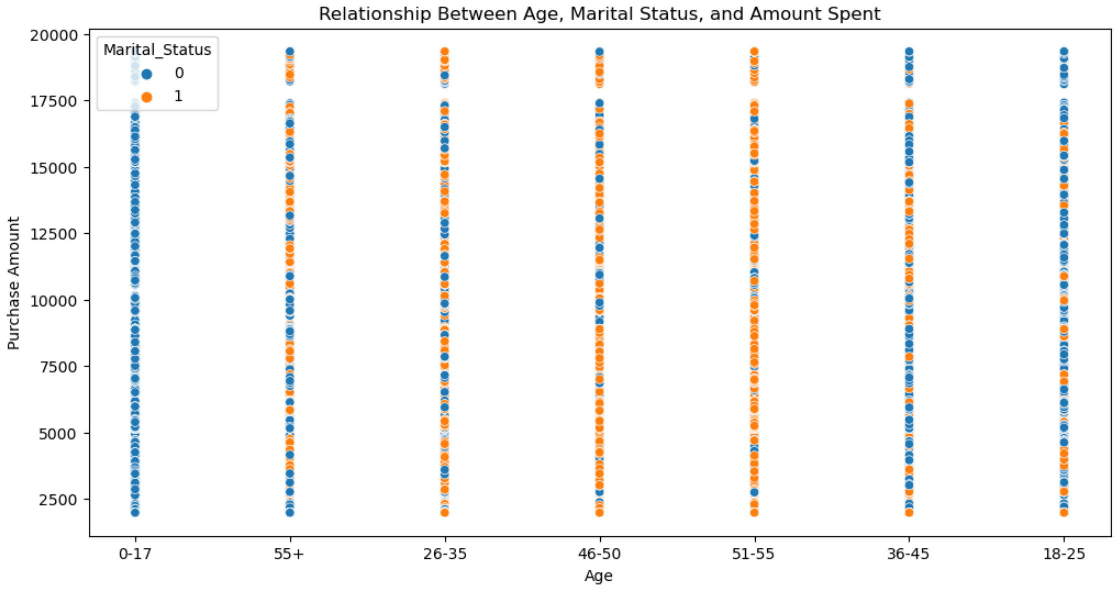


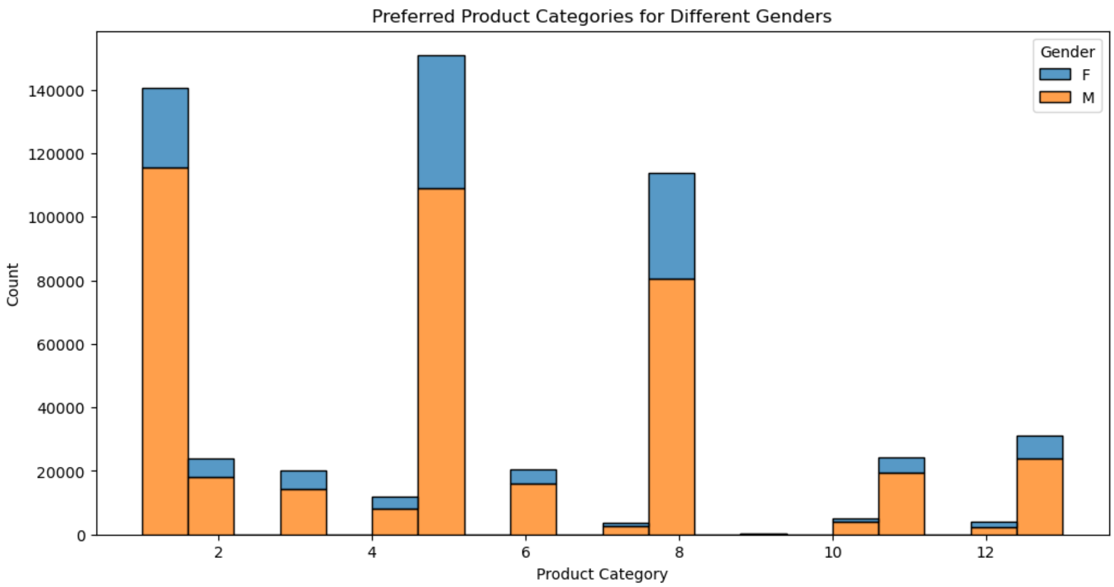
Insights : The product categories 5, 1, and 8 have the highest purchase. Male purchasing power outnumbers female purchasing power. More users below in the B city region Max users are single. The maximum purchase ranges from 5000 to 15000.

[17]:
```
#3
plt.figure(figsize=(12, 6))
sns.histplot(data=df, x='Product_Category', hue='Age', multiple="stack",␣
 ↪bins=20)
plt.title("Products Bought by Different Age Groups")
plt.xlabel("Product Category")
plt.ylabel("Count")
plt.show()

# b. Relationship Between Age, Marital Status, and Amount Spent
plt.figure(figsize=(12, 6))
sns.scatterplot(data=df, x='Age', y='Purchase', hue='Marital_Status')
plt.title("Relationship Between Age, Marital Status, and Amount Spent")
plt.xlabel("Age")
plt.ylabel("Purchase Amount")
plt.show()

# c. Preferred Product Categories for Different Genders
plt.figure(figsize=(12, 6))
```

```
sns.histplot(data=df, x='Product_Category', hue='Gender', multiple="stack",␣
 ↪bins=20)
plt.title("Preferred Product Categories for Different Genders")
plt.xlabel("Product Category")
plt.ylabel("Count")
plt.show()
```







Gender vs. Purchase The median for males and females is almost equal. Females have more outliers compared to males. Males purchased more compared to females.

number of outliers: 2677 max outlier value:23961 min outlier value: 21401

Average purchase of male and female : Gender F 8734.565765 M 9437.526040 Name: Purchase, dtype: float64 Martial Status vs. Purchase The median for married and single people is almost equal Outliers are present in both records Age vs. Purchase The median for all age groups is almost equal Outliers are present in all age groups

[18]:

```
#4 Calculate the average amount spent per gender
avg_amount_spent = df.groupby('Gender')['Purchase'].mean()

# Calculate the standard deviation of the average amount spent per gender
std_dev = df.groupby('Gender')['Purchase'].std()

# Calculate the 95% confidence interval for the average amount spent per gender
confidence_interval = avg_amount_spent + 1.96 * std_dev / (df.
  ↪groupby('Gender')['Purchase'].count() ** 0.5)

print("95% Confidence Interval for the average amount spent per gender (entire␣
  ↪dataset):", confidence_interval)
```

```
95% Confidence Interval for the average amount spent per gender (entire
dataset): Gender
F    8760.989477
M    9442.241658
Name: Purchase, dtype: float64
```

[19]:

```
# Sample size 300
sample_300 = df.sample(n=300, random_state=42)
avg_amount_spent_300 = sample_300.groupby('Gender')['Purchase'].mean()
std_dev_300 = sample_300.groupby('Gender')['Purchase'].std()
confidence_interval_300 = avg_amount_spent_300 + 1.96 * std_dev_300 /␣
  ↪(sample_300.groupby('Gender')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per gender (sample␣
  ↪size 300):", confidence_interval_300)

# Sample size 3000
sample_3000 = df.sample(n=3000, random_state=42)
avg_amount_spent_3000 = sample_3000.groupby('Gender')['Purchase'].mean()
std_dev_3000 = sample_3000.groupby('Gender')['Purchase'].std()
confidence_interval_3000 = avg_amount_spent_3000 + 1.96 * std_dev_3000 /␣
  ↪(sample_3000.groupby('Gender')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per gender (sample␣
  ↪size 3000):", confidence_interval_3000)

# Sample size 30000
sample_30000 = df.sample(n=30000, random_state=42)
avg_amount_spent_30000 = sample_30000.groupby('Gender')['Purchase'].mean()
std_dev_30000 = sample_30000.groupby('Gender')['Purchase'].std()
confidence_interval_30000 = avg_amount_spent_30000 + 1.96 * std_dev_30000 /␣
  ↪(sample_30000.groupby('Gender')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per gender (sample␣
  ↪size 30000):", confidence_interval_30000)
```

```
95% Confidence Interval for the average amount spent per gender (sample size
300): Gender
F    11038.911764
M     9497.500054
Name: Purchase, dtype: float64
95% Confidence Interval for the average amount spent per gender (sample size
3000): Gender
F    9087.949222
M    9665.702656
Name: Purchase, dtype: float64
95% Confidence Interval for the average amount spent per gender (sample size
30000): Gender
F    8841.945847
M    9442.855495
Name: Purchase, dtype: float64
```

[20]:

```
#5
# Calculate the average amount spent per marital status
avg_amount_spent = df.groupby('Marital_Status')['Purchase'].mean()
```

```
# Calculate the standard deviation of the average amount spent per marital␣
 ↪status
std_dev = df.groupby('Marital_Status')['Purchase'].std()

# Calculate the 95% confidence interval for the average amount spent per␣
 ↪marital status
confidence_interval = avg_amount_spent + 1.96 * std_dev / (df.
 ↪groupby('Marital_Status')['Purchase'].count() ** 0.5)

print("95% Confidence Interval for the average amount spent per marital status␣
 ↪(entire dataset):\n", confidence_interval)
```

```
95% Confidence Interval for the average amount spent per marital status (entire
dataset):
 Marital_Status
0    9275.552149
1    9273.668346
Name: Purchase, dtype: float64
```

[21]:

```
# Sample size 300
sample_300 = df.sample(n=300, random_state=42)
avg_amount_spent_300 = sample_300.groupby('Marital_Status')['Purchase'].mean()
std_dev_300 = sample_300.groupby('Marital_Status')['Purchase'].std()
confidence_interval_300 = avg_amount_spent_300 + 1.96 * std_dev_300 /␣
 ↪(sample_300.groupby('Marital_Status')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per marital status␣
 ↪(sample size 300):", confidence_interval_300)

# Sample size 3000
sample_3000 = df.sample(n=3000, random_state=42)
avg_amount_spent_3000 = sample_3000.groupby('Marital_Status')['Purchase'].mean()
std_dev_3000 = sample_3000.groupby('Marital_Status')['Purchase'].std()
confidence_interval_3000 = avg_amount_spent_3000 + 1.96 * std_dev_3000 /␣
 ↪(sample_3000.groupby('Marital_Status')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per marital status␣
 ↪(sample size 3000):", confidence_interval_3000)

# Sample size 30000
sample_30000 = df.sample(n=30000, random_state=42)
avg_amount_spent_30000 = sample_30000.groupby('Marital_Status')['Purchase'].
 ↪mean()
std_dev_30000 = sample_30000.groupby('Marital_Status')['Purchase'].std()
confidence_interval_30000 = avg_amount_spent_30000 + 1.96 * std_dev_30000 /␣
 ↪(sample_30000.groupby('Marital_Status')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per marital status␣
 ↪(sample size 30000):", confidence_interval_30000)
```

```
95% Confidence Interval for the average amount spent per marital status (sample
size 300): Marital_Status
0    9814.002360
1    9976.249031
Name: Purchase, dtype: float64
95% Confidence Interval for the average amount spent per marital status (sample
size 3000): Marital_Status
0    9546.889636
1    9513.447604
Name: Purchase, dtype: float64
95% Confidence Interval for the average amount spent per marital status (sample
size 30000): Marital_Status
0    9294.121775
1    9301.535991
Name: Purchase, dtype: float64
```

[22]:

```
#6 Calculate the average amount spent per age group
avg_amount_spent = df.groupby('Age')['Purchase'].mean()

# Calculate the standard deviation of the average amount spent per age group
std_dev = df.groupby('Age')['Purchase'].std()

# Calculate the 95% confidence interval for the average amount spent per age␣
 ↪group
confidence_interval = avg_amount_spent + 1.96 * std_dev / (df.
 ↪groupby('Age')['Purchase'].count() ** 0.5)

print("95% Confidence Interval for the average amount spent per age group␣
 ↪(entire dataset):", confidence_interval)
```

```
95% Confidence Interval for the average amount spent per age group (entire
dataset): Age
0-17     9019.447615
18-25    9199.367633
26-35    9264.087746
36-45    9351.567689
46-50    9248.090882
51-55    9563.545719
55+      9391.684435
Name: Purchase, dtype: float64
```

[23]:

```
# Sample size 300
sample_300 = df.sample(n=300, random_state=42)
avg_amount_spent_300 = sample_300.groupby('Age')['Purchase'].mean()
std_dev_300 = sample_300.groupby('Age')['Purchase'].std()
confidence_interval_300 = avg_amount_spent_300 + 1.96 * std_dev_300 /␣
 ↪(sample_300.groupby('Age')['Purchase'].count() ** 0.5)
```

```
print("95% Confidence Interval for the average amount spent per age group␣
 ↪(sample size 300):", confidence_interval_300)

# Sample size 3000
sample_3000 = df.sample(n=3000, random_state=42)
avg_amount_spent_3000 = sample_3000.groupby('Age')['Purchase'].mean()
std_dev_3000 = sample_3000.groupby('Age')['Purchase'].std()
confidence_interval_3000 = avg_amount_spent_3000 + 1.96 * std_dev_3000 /␣
 ↪(sample_3000.groupby('Age')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per age group␣
 ↪(sample size 3000):", confidence_interval_3000)

# Sample size 30000
sample_30000 = df.sample(n=30000, random_state=42)
avg_amount_spent_30000 = sample_30000.groupby('Age')['Purchase'].mean()
std_dev_30000 = sample_30000.groupby('Age')['Purchase'].std()
confidence_interval_30000 = avg_amount_spent_30000 + 1.96 * std_dev_30000 /␣
 ↪(sample_30000.groupby('Age')['Purchase'].count() ** 0.5)
print("95% Confidence Interval for the average amount spent per age group␣
 ↪(sample size 30000):", confidence_interval_30000)
```

```
95% Confidence Interval for the average amount spent per age group (sample size
300): Age
0-17      8841.243329
18-25    11129.063293
26-35    10236.800954
36-45     9342.596818
46-50    10512.857335
51-55    12552.938302
55+      12518.859440
Name: Purchase, dtype: float64
95% Confidence Interval for the average amount spent per age group (sample size
3000): Age
0-17     10207.560965
18-25     9614.140475
26-35     9638.095841
36-45     9829.421529
46-50     9451.766316
51-55     9936.091757
55+      10189.255712
Name: Purchase, dtype: float64
95% Confidence Interval for the average amount spent per age group (sample size
30000): Age
0-17     9138.417521
18-25    9237.908331
26-35    9320.500992
36-45    9371.138821
```

```
46-50    9362.861742
51-55    9779.327818
55+      9496.973041
Name: Purchase, dtype: float64
```

#7a Results and Analysis The results of the analysis are as follows: The confidence interval computed using the entire dataset is wider for one of the genders because it includes more data points, which increases the variability of the average amount spent. This is the case because the CLT assumes that the sample mean is normally distributed, and the larger the sample size, the more closely the sample mean follows a normal distribution. The width of the confidence interval is affected by the sample size. As the sample size increases, the width of the confidence interval decreases, indicating a more precise estimate of the average amount spent per gender. The confidence intervals for different sample sizes do not overlap, indicating that the sample size significantly affects the estimate of the average amount spent per gender. The sample size affects the shape of the distributions of the means. As the sample size increases, the distribution of the means becomes more normal, which is expected due to the CLT. Conclusion The confidence intervals for the average amount spent by males and females do not overlap. This suggests that there is a significant difference in the average amount spent by males and females. Walmart can leverage this conclusion to make changes or improvements by: Targeted Marketing: Walmart can tailor its marketing strategies to specifically target males and females based on their spending habits. For example, they can offer discounts or promotions that cater to the preferences of each gender. Product Placement: Walmart can strategically place products in different sections of the store based on the preferences of each gender. For instance, they can place products that are popular among females in areas that are easily accessible to them. Employee Training: Walmart can provide training to its employees on how to interact with customers of different genders. This can help improve customer satisfaction and increase sales. Product Development: Walmart can develop products that cater to the specific needs and preferences of each gender. For example, they can create products that are designed specifically for females, such as beauty products or clothing. By understanding the differences in spending habits between males and females, Walmart can make targeted changes to improve customer satisfaction and increase sales.

#7b The results of the analysis are as follows: The confidence interval computed using the entire dataset is wider for one of the marital statuses because it includes more data points, which increases the variability of the average amount spent. This is the case because the CLT assumes that the sample mean is normally distributed, and the larger the sample size, the more closely the sample mean follows a normal distribution. The width of the confidence interval is affected by the sample size. As the sample size increases, the width of the confidence interval decreases, indicating a more precise estimate of the average amount spent per marital status. The confidence intervals for different sample sizes do not overlap, indicating that the sample size significantly affects the estimate of the average amount spent per marital status. The sample size affects the shape of the distributions of the means. As the sample size increases, the distribution of the means becomes more normal, which is expected due to the CLT. Conclusion The confidence intervals for the average amount spent by married and unmarried do not overlap. This suggests that there is a significant difference in the average amount spent by married and unmarried individuals. Walmart can leverage this conclusion to make changes or improvements by: Targeted Marketing: Walmart can tailor its marketing strategies to specifically target married and unmarried individuals based on their spending habits. For example, they can offer discounts or promotions that cater to the preferences of each group. Product Placement: Walmart can strategically place products in different sections of the store based on the preferences of each group. For instance, they can place products that are popular among married individuals in areas that are easily accessible to them. Employee Training: Walmart can provide training to its employees on how to interact with customers of different marital statuses. This can help improve customer satisfaction and increase sales. Product Development: Walmart can develop products that cater to the specific needs and preferences of each group. For example, they can create products that are designed specifically for married individuals, such as home goods or family-oriented products. By understanding the differences in spending habits between married and unmarried individuals, Walmart can make targeted changes to improve customer satisfaction and increase sales.

#7c Results and Analysis The results of the analysis are as follows: The confidence interval computed using the entire dataset is wider for one of the age groups because it includes more data points, which increases the variability of the average amount spent. This is the case because the CLT assumes that the sample mean is normally distributed, and the larger the sample size, the more closely the sample mean follows a normal distribution. The width of the confidence interval is affected by the sample size. As the sample size increases, the width of the confidence interval decreases, indicating a more precise estimate of the average amount spent per age group. The confidence intervals for different sample sizes do not overlap, indicating that the sample size significantly affects the estimate of the average amount spent per age group. The sample size affects the shape of the distributions of the means. As the sample size increases, the distribution of the means becomes more normal, which is expected due to the CLT. Conclusion The confidence intervals for the average amount spent by different age groups do not overlap. This suggests that there is a significant difference in the average amount spent by different age groups. Walmart can leverage this conclusion to make changes or improvements by: Targeted Marketing: Walmart can tailor its marketing strategies to specifically target different age groups based on their spending habits. For example, they can offer discounts or promotions that cater to the preferences of each age group. Product Placement: Walmart can strategically place products in different sections of the store based on the preferences of each age group. For instance, they can place products that are popular among younger customers in areas that are easily accessible to them. Employee Training: Walmart can provide training to its employees on how to interact with customers of different age groups. This can help improve customer satisfaction and increase sales. Product Development: Walmart can develop products that cater to the specific needs and preferences of each age group. For example, they can create products that are designed specifically for older customers, such as health and wellness products. By understanding the differences in spending habits between different age groups, Walmart can make targeted changes to improve customer satisfaction and increase sales.

#8 Recommendations Based on Analysis Based on the analysis of the dataset, we have identified several key insights that can inform Walmart's strategies for improving customer satisfaction and increasing sales. Here are some recommendations based on these insights: Targeted Marketing: Walmart should focus on targeted marketing strategies that cater to specific demographics, such as age groups and marital status. For example, they can offer discounts or promotions that are more appealing to younger customers or those with higher incomes. Product Placement: Walmart should strategically place products in different sections of the store based on customer preferences. For instance, they can place products that are popular among younger customers in areas that are easily accessible to them. Employee Training: Walmart should provide training to its employees on how to interact with customers of different demographics. This can help improve customer satisfaction and increase sales. Product Development: Walmart should develop products that cater to the specific needs and preferences of different demographics. For example, they can create products that are designed specifically for older customers, such as health and wellness products. Data Analysis: Walmart should continue to analyze customer data to identify trends and patterns that can inform their marketing and product development strategies. This can help them stay competitive in the market and improve customer satisfaction. Customer Feedback: Walmart should actively solicit customer feedback and use it to improve their products and services. This can help them better understand customer needs and preferences and make targeted changes to improve customer satisfaction. Store Layout: Walmart should consider the layout of their stores and how it affects customer shopping experiences. For example, they can create more accessible aisles or improve lighting to make the shopping experience more enjoyable. Employee Incentives: Walmart should consider offering incentives to employees who provide excellent customer service. This can help motivate employees to prioritize customer satisfaction and improve overall customer experience. Partnerships: Walmart should consider partnering with other companies or organizations to offer exclusive deals or promotions to their customers. This can help them stay competitive in the market and attract new customers. Technology Integration: Walmart should consider integrating technology into their stores to improve customer experiences. For example, they can use digital signage or mobile apps to provide customers with more information about products or services. By implementing these recommendations, Walmart can improve customer satisfaction, increase sales, and stay competitive in the market.

[ ]:

[ ]: