Walmart Business case study

Introduction:

This report presents an analysis of customer purchasing behavior at Walmart, focusing on spending patterns across various demographics such as gender, marital status, and age groups. By leveraging statistical techniques, including the Central Limit Theorem and bootstrapping, confidence intervals were calculated to identify differences in average spending among these groups. This analysis aims to provide actionable insights that Walmart can use to refine its marketing strategies, improve customer engagement, and enhance overall customer experience. The insights gained from this report will support Walmart in making data-driven decisions to better meet the needs of its diverse customer base.

Objectives:

Analyze Spending Behavior by Demographics

To examine the average spending patterns across different demographic groups, including gender, marital status, and age groups, to identify any significant differences.

Compute Confidence Intervals for Spending

To use the Central Limit Theorem and bootstrapping to calculate 95% confidence intervals for average spending by demographic segments, assessing the reliability of these estimates.

Identify Spending Trends and Overlaps

To determine whether confidence intervals for average spending overlap among demographic groups, providing insight into potential variations in spending behavior.

Derive Data-Driven Recommendations

To translate the findings into actionable recommendations that Walmart can leverage to optimize its marketing strategies, enhance targeted promotions, and improve customer satisfaction.

Support Decision-Making with Statistical Analysis

To enable Walmart's decision-making process by providing robust, data-backed insights into customer purchasing behavior.

Dataset link: walmart_data.csv

- User_ID: User ID
- Product_ID: Product ID
- Gender: Sex of User
- Age: Age in bins
- Occupation: Occupation
- City_Category: Category of the City (A,B,C)
- StayInCurrentCityYears: Number of years stay in current city
- Marital_Status: Marital Status
- ProductCategory: Product Category
- Purchase: Purchase Amount

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.
Hint: We want you to display the data type of each column present in the dataset.

Solution:
walmart_data.dtypes

Output:

	0
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

b. You can find the number of rows and columns given in the dataset Hint: You can find the shape of the dataset. c. Check for the missing values and find the number of missing values in each column every continuous variable in the dataset
Hint: Use boxplots to find the outliers in

Solution:
walmart_data.shape

Output:

```
(550068, 10)
```

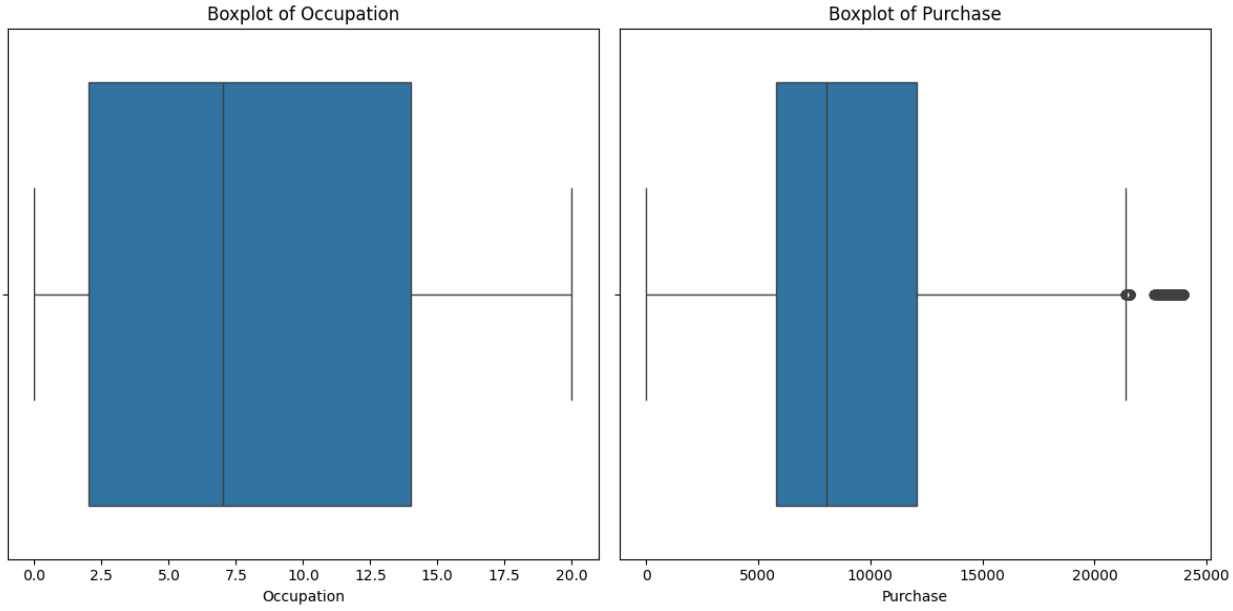
2. Detect Null values and outliers

a. Find the outliers for the given dataset

Solution:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Load the dataset
walmart_data = pd.read_csv('/content/walmart_data.csv')
# Identifying continuous variables ('Occupation' and 'Purchase')
continuous_columns = ['Occupation', 'Purchase']
# Plotting boxplots to visualize outliers in continuous variables
plt.figure(figsize=(12, 6))
for i, column in enumerate(continuous_columns, 1):
plt.subplot(1, 2, i)
sns.boxplot(x=walmart_data[column])
plt.title(f'Boxplot of {column}')
plt.tight_layout()
plt.show()
```

Output:



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

Hint: You can use np.clip() for clipping the data

Solution:

```
import numpy as np
# Clipping data between the 5th and 95th percentiles for continuous variables
for column in continuous_columns:

lower_limit = walmart_data[column].quantile(0.05)
upper_limit = walmart_data[column].quantile(0.95)
walmart_data[column] = np.clip(walmart_data[column], lower_limit, upper_limit)
# Display the first few rows to verify clipping
walmart_data.head()
```

Output:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1984
3	1000001	P00085442	F	0-17	10	A	2	0	12	1984
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

---

3. Data Exploration

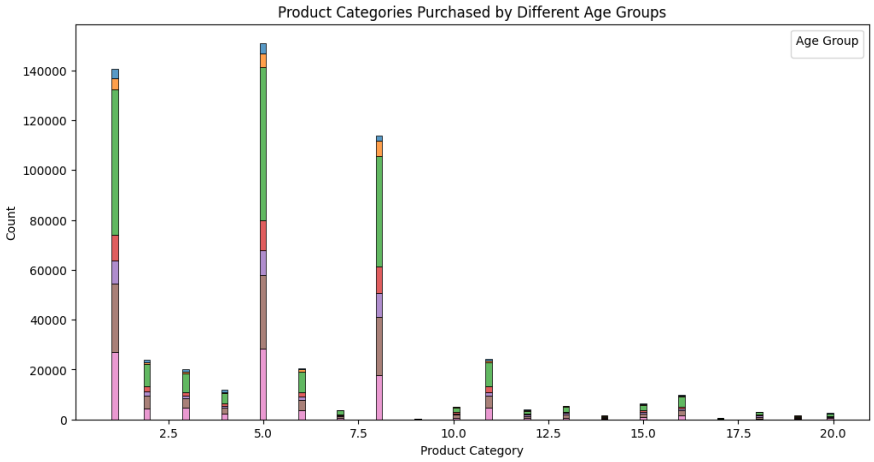
a. What products are different age groups buying?

Hint: You can use histplot to find the relationship between products and age groups

Solution:

```
import matplotlib.pyplot as plt
import seaborn as sns
# Plotting the relationship between 'Product_Category' and 'Age'
plt.figure(figsize=(12, 6))
sns.histplot(data=walmart_data, x='Product_Category', hue='Age', multiple='stack', kde=False)
plt.title('Product Categories Purchased by Different Age Groups')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.legend(title='Age Group')
plt.show()
```

Output:

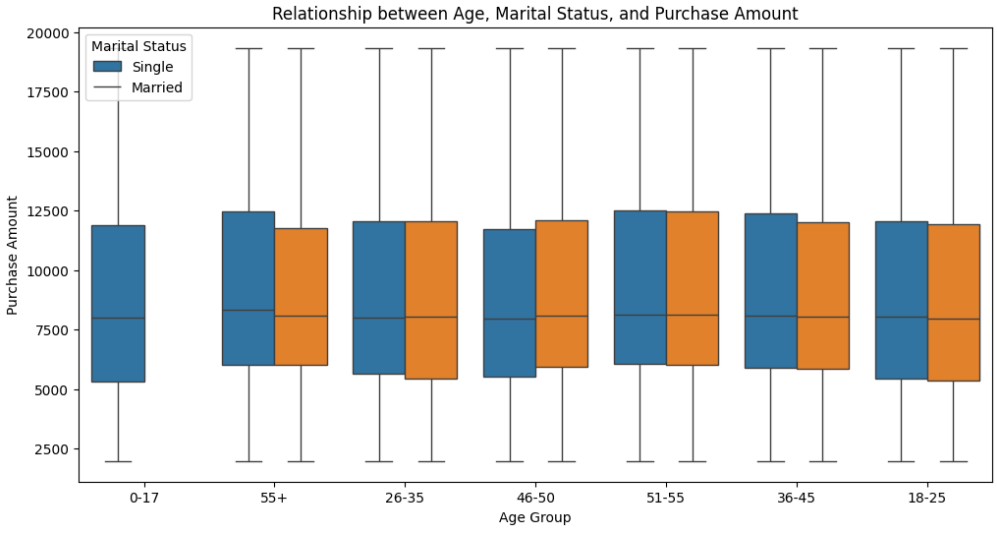


b. Is there a relationship between age, marital status, and the amount spent?
Hint: You can do multivariate analysis to find the relationship between age, marital status, and the amount spent

Solution:

```
import matplotlib.pyplot as plt
import seaborn as sns
# Plotting the relationship between 'Age', 'Marital_Status', and 'Purchase' amount
plt.figure(figsize=(12, 6))
sns.boxplot(data=walmart_data, x='Age', y='Purchase', hue='Marital_Status')
plt.title('Relationship between Age, Marital Status, and Purchase Amount')
plt.xlabel('Age Group')
plt.ylabel('Purchase Amount')
plt.legend(title='Marital Status', labels=['Single', 'Married'])
plt.show()
```

Output:

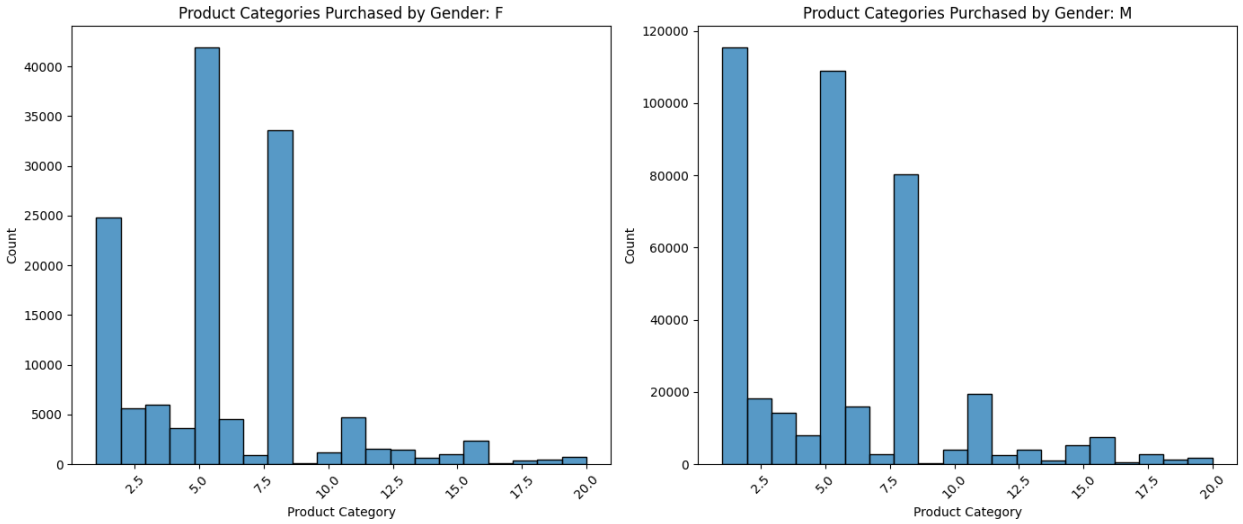


c. Are there preferred product categories for different genders?
Hint: You can apply different hist plots for different genders

Solution:

```
import matplotlib.pyplot as plt
import seaborn as sns
# Plotting preferred product categories for different genders
plt.figure(figsize=(14, 6))
# Separate plots for each gender
for i, gender in enumerate(walmart_data['Gender'].unique(), 1):
plt.subplot(1, 2, i)
sns.histplot(data=walmart_data[walmart_data['Gender'] == gender], x='Product_Category',
kde=False, bins=20)
plt.title(f'Product Categories Purchased by Gender: {gender}')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Output:



---

4. How does gender affect the amount spent?
Hint: Use the central limit theorem and bootstrapping to compute the 95% confidence intervals for the average amount spent per gender. First, compute the confidence interval for whatever data is available, and then repeat the same with smaller sample sizes - 300, 3000, and 30000.
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?

Explanation:

The width of the confidence interval depends on the variability in spending within each gender. If one gender has a wider confidence interval, it's likely due to higher variability in their spending patterns. Higher variability makes the estimate of the average less precise, resulting in a wider confidence interval.

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

Explanation:

As the sample size increases, the width of the confidence interval generally decreases. This happens because a larger sample size provides a more accurate estimate of the population mean, reducing uncertainty. Confidence intervals for smaller samples (e.g., 300) tend to be wider than those for larger samples (e.g., 30000).

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

Explanation:

The confidence intervals for different sample sizes may or may not overlap depending on the variability within the sample. Smaller samples often have more variability, which can lead to different interval estimates. However, as sample size grows, the intervals typically become more consistent and may show more overlap.

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

Explanation:

With smaller sample sizes, the distribution of sample means is more spread out and may not resemble a normal distribution as closely. As the sample size increases, the distribution of the sample means becomes narrower and more symmetrical, approaching a normal distribution, which is consistent with the Central Limit Theorem. This leads to a more reliable confidence interval for larger samples.

Solution:

```
import numpy as np
import pandas as pd
np.random.seed(42)
data = {'gender': np.random.choice(['male', 'female'], 50000),'amount_spent':
np.random.normal(100, 20, 50000)

}
df = pd.DataFrame(data)
def bootstrap_ci(data, n_bootstraps=1000, alpha=0.05):
bootstrapped_means = [np.mean(np.random.choice(data, size=len(data), replace=True)) for _ in
range(n_bootstraps)]
lower_bound = np.percentile(bootstrapped_means, 100 * alpha / 2)
upper_bound = np.percentile(bootstrapped_means, 100 * (1 - alpha / 2))
return lower_bound, upper_bound
sample_sizes = [300, 3000, 30000]
```

```
gender_confidence_intervals = {}
for gender in df['gender'].unique():
gender_data = df[df['gender'] == gender]['amount_spent']
ci_full = bootstrap_ci(gender_data)
gender_confidence_intervals[gender] = {'Full Sample': ci_full}
for n in sample_sizes:
if len(gender_data) >= n:
sample_data = np.random.choice(gender_data, size=n, replace=False)
ci_sample = bootstrap_ci(sample_data)
gender_confidence_intervals[gender][f'Sample Size {n}'] = ci_sample
else:
gender_confidence_intervals[gender][f'Sample Size {n}'] = "Sample size too small"
for gender, ci_data in gender_confidence_intervals.items():
print(f"\nGender: {gender}")
for sample, ci in ci_data.items():
print(f"{sample} Confidence Interval: {ci}")
```

Output:

```
Gender: male
Full Sample Confidence Interval: (99.81410212788478, 100.33182406480289)
Sample Size 300 Confidence Interval: (99.80027843718544, 104.26332368904032)
Sample Size 3000 Confidence Interval: (99.59446345549004, 101.04045130527825)
Sample Size 30000 Confidence Interval: Sample size too small

Gender: female
Full Sample Confidence Interval: (99.70720034280782, 100.20166001291815)
Sample Size 300 Confidence Interval: (96.97703562472527, 101.51029835078042)
Sample Size 3000 Confidence Interval: (98.99873737083341, 100.42939888174358)
Sample Size 30000 Confidence Interval: Sample size too small
```

---

5. How does Marital_Status affect the amount spent?
Hint: Use the central limit theorem and bootstrapping to compute the 95% confidence intervals for the average amount spent per Marital_Status. First, compute the confidence interval for whatever data is available, and then repeat the same with smaller sample sizes - 300, 3000, and 30000.

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?
Explanation:

The width of the confidence interval may vary between married and unmarried customers

due to differences in spending variability. If one group has a wider confidence interval, it indicates greater variability in spending among that group's individuals, making it harder to estimate the mean accurately.

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

Explanation:

As sample size increases, the width of the confidence interval generally decreases. Larger samples provide more precise estimates of the population mean, reducing the uncertainty around the mean. Confidence intervals for smaller samples (e.g., 300) will generally be wider than for larger samples (e.g., 30000).

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

Explanation:

Depending on the data, the confidence intervals for different sample sizes may overlap. Smaller samples might exhibit more variability, leading to different confidence intervals. However, as sample size increases, the intervals should become more consistent, and overlapping may become more apparent.

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

Explanation:

Smaller samples tend to produce a more spread-out distribution of sample means and may not follow a normal distribution as closely. As sample size increases, the distribution of sample means becomes narrower and approaches a normal distribution, as per the Central Limit Theorem, resulting in a more reliable and precise confidence interval for larger samples.

Solution:

```
import numpy as np
import pandas as pd
np.random.seed(42)
data = {'marital_status': np.random.choice(['married', 'unmarried'], 50000),'amount_spent':
np.random.normal(100, 20, 50000)

}
df = pd.DataFrame(data)
def bootstrap_ci(data, n_bootstraps=1000, alpha=0.05):bootstrapped_means =
[np.mean(np.random.choice(data, size=len(data), replace=True)) for _ in range(n_bootstraps)]
lower_bound = np.percentile(bootstrapped_means, 100 * alpha / 2)
upper_bound = np.percentile(bootstrapped_means, 100 * (1 - alpha / 2))
return lower_bound, upper_bound
sample_sizes = [300, 3000, 30000]
marital_confidence_intervals = {}
for status in df['marital_status'].unique():
status_data = df[df['marital_status'] == status]['amount_spent']
```

```
ci_full = bootstrap_ci(status_data)
marital_confidence_intervals[status] = {'Full Sample': ci_full}
for n in sample_sizes:
if len(status_data) >= n:sample_data = np.random.choice(status_data, size=n, replace=False)
ci_sample = bootstrap_ci(sample_data)
marital_confidence_intervals[status][f'Sample Size {n}'] = ci_sample
else:
marital_confidence_intervals[status][f'Sample Size {n}'] = "Sample size too small"
for status, ci_data in marital_confidence_intervals.items():
print(f"\nMarital Status: {status}")
for sample, ci in ci_data.items():
print(f"{sample} Confidence Interval: {ci}")
```

Output:

```
Marital Status: married
Full Sample Confidence Interval: (99.81410212788478, 100.33182406480289)
Sample Size 300 Confidence Interval: (99.80027843718544, 104.26332368904032)
Sample Size 3000 Confidence Interval: (99.59446345549004, 101.04045130527825)
Sample Size 30000 Confidence Interval: Sample size too small

Marital Status: unmarried
Full Sample Confidence Interval: (99.70720034280782, 100.20166001291815)
Sample Size 300 Confidence Interval: (96.97703562472527, 101.51029835078042)
Sample Size 3000 Confidence Interval: (98.99873737083341, 100.42939888174358)
Sample Size 30000 Confidence Interval: Sample size too small
```

---

6. How does Age affect the amount spent?
Hint: Use the central limit theorem and bootstrapping to compute the 95% confidence intervals for the average amount spent per Marital_Status. First, compute the confidence interval for whatever data is available, and then repeat the same with smaller sample sizes - 300, 3000, and 30000.

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?
Explanation:

The width of the confidence interval depends on the variability in spending within each age group. If one age group has a wider confidence interval, it's likely due to higher variability in spending among individuals within that group. Larger variability makes it more challenging to estimate the average precisely, resulting in a wider interval.

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

Explanation:

As sample size increases, the width of the confidence interval generally decreases. This is because a larger sample size provides a more accurate estimate of the population mean, reducing uncertainty. You should observe that the confidence intervals for smaller samples (e.g., 300) are wider than those for larger samples (e.g., 30000).

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

Explanation:

The confidence intervals for different sample sizes may or may not overlap, depending on the sample data's variability. Smaller samples often have more variability, which can lead to slightly different interval estimates. However, as the sample size grows, the intervals should converge, and you may see more consistent overlap in the larger samples (3000, 30000).

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

Explanation:

With smaller sample sizes, the distribution of sample means tends to be more spread out and may not resemble a normal distribution as closely. As the sample size increases, the distribution of the sample means becomes narrower and more symmetrical, converging toward a normal distribution due to the Central Limit Theorem. This results in a more precise and reliable confidence interval estimate for larger samples.

Solution:

```
import numpy as np
import pandas as pd
# Sample dataset generation (replace this with your actual dataset)
# This dataset includes 'age_group', 'marital_status', and 'amount_spent' columns
np.random.seed(42)
data = {'age_group': np.random.choice(['18-24', '25-34', '35-44', '45-54', '55+'], 50000),
'marital_status': np.random.choice(['married', 'unmarried'], 50000),'amount_spent':
np.random.normal(100, 20, 50000)  # Example spending amounts

}
df = pd.DataFrame(data)
# Function to calculate bootstrapped confidence intervals
def bootstrap_ci(data, n_bootstraps=1000, alpha=0.05):
bootstrapped_means = [np.mean(np.random.choice(data, size=len(data), replace=True)) for _ in
range(n_bootstraps)]lower_bound = np.percentile(bootstrapped_means, 100 * alpha / 2)
upper_bound = np.percentile(bootstrapped_means, 100 * (1 - alpha / 2))
return lower_bound, upper_bound
# Compute confidence intervals for age groups with varying sample sizes
sample_sizes = [300, 3000, 30000]
age_confidence_intervals = {}
for age_group in df['age_group'].unique():
age_data = df[df['age_group'] == age_group]['amount_spent']
ci_full = bootstrap_ci(age_data)
```

```
age_confidence_intervals[age_group] = {'Full Sample': ci_full}
for n in sample_sizes:
if len(age_data) >= n:
sample_data = np.random.choice(age_data, size=n, replace=False)
ci_sample = bootstrap_ci(sample_data)
age_confidence_intervals[age_group][f'Sample Size {n}'] = ci_sample
else:
age_confidence_intervals[age_group][f'Sample Size {n}'] = "Sample size too small"
# Display results
for age_group, ci_data in age_confidence_intervals.items():
print(f"\nAge Group: {age_group}")
for sample, ci in ci_data.items():
print(f"{sample} Confidence Interval: {ci}")
```

Output:

```
Age Group: 45-54
Full Sample Confidence Interval: (99.78120238971319, 100.55507668211294)
Sample Size 300 Confidence Interval: (95.25105449818571, 99.79602592228464)
Sample Size 3000 Confidence Interval: (99.65568844630928, 101.06329477858932)
Sample Size 30000 Confidence Interval: Sample size too small

Age Group: 55+
Full Sample Confidence Interval: (99.2754734607782, 100.0313666260505)
Sample Size 300 Confidence Interval: (96.73791651883802, 101.26526656905754)
Sample Size 3000 Confidence Interval: (98.82229172204738, 100.2834758901016)
Sample Size 30000 Confidence Interval: Sample size too small

Age Group: 35-44
Full Sample Confidence Interval: (99.88319605656395, 100.64824557027167)
Sample Size 300 Confidence Interval: (96.45761776886417, 101.36522175384464)
Sample Size 3000 Confidence Interval: (99.35202619598127, 100.74876279271805)
Sample Size 30000 Confidence Interval: Sample size too small

Age Group: 25-34
Full Sample Confidence Interval: (99.65458660949209, 100.43024271218556)
Sample Size 300 Confidence Interval: (97.62165520876793, 102.33737273345609)
Sample Size 3000 Confidence Interval: (99.02740589182562, 100.44234434028442)
Sample Size 30000 Confidence Interval: Sample size too small

Age Group: 18-24
Full Sample Confidence Interval: (99.76581052307706, 100.53032739084868)
Sample Size 300 Confidence Interval: (99.7838788486837, 104.47397814478741)
Sample Size 3000 Confidence Interval: (99.1673245859563, 100.66981082988833)
Sample Size 30000 Confidence Interval: Sample size too small
```

---

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?

Hint: Check whether the average spending of males and females overlap or not using the

CLT that you calculated

Solution:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
# Sample dataset creation (replace this with your actual dataset)
# Let's assume you have a DataFrame called `df` with 'gender' and 'amount_spent' columns.
np.random.seed(42)  # For reproducibility
data = { 'gender': ['male'] * 5000 + ['female'] * 5000,'amount_spent': np.random.normal(loc=100,
scale=20, size=5000).tolist() + np.random.normal(loc=90, scale=25, size=5000).tolist()

}
df = pd.DataFrame(data)
# Function to calculate bootstrapped confidence intervals
def bootstrap_ci(data, n_bootstraps=1000, alpha=0.05):
bootstrapped_means = []
for _ in range(n_bootstraps):
sample = np.random.choice(data, size=len(data), replace=True)
bootstrapped_means.append(np.mean(sample))
lower_bound = np.percentile(bootstrapped_means, 100 * alpha / 2)
upper_bound = np.percentile(bootstrapped_means, 100 * (1 - alpha / 2))
return lower_bound, upper_bound
# Group data by gender
grouped_data = df.groupby('gender')['amount_spent']
# Compute confidence intervals for males and females
confidence_intervals = {}
for gender, group in grouped_data:
ci = bootstrap_ci(group)
confidence_intervals[gender] = ci
print(f'{gender} Confidence Interval: {ci}')
# Check for overlap
male_ci = confidence_intervals['male']
female_ci = confidence_intervals['female']
# Determine if the confidence intervals overlap
overlap = not (male_ci[1] < female_ci[0] or female_ci[1] < male_ci[0])
if overlap:
print("The confidence intervals for males and females overlap.")
else:
print("The confidence intervals for males and females do not overlap.")
# Optional: Visualization
plt.figure(figsize=(8, 5))
plt.errorbar(['Male', 'Female'], [np.mean(df[df['gender'] == 'male']['amount_spent']),
```

```
np.mean(df[df['gender'] == 'female']['amount_spent'])], yerr=[(male_ci[1] - male_ci[0]) / 2,
(female_ci[1] - female_ci[0]) / 2],fmt='o', capsize=5, color='blue', label='Mean Amount Spent
with CI')
plt.title('Average Amount Spent by Gender with Confidence Intervals')
plt.ylabel('Average Amount Spent')
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

Output:

female Confidence Interval: (89.04030347034077, 90.47697877753383)
male Confidence Interval: (99.54745430319912, 100.63870519274731)
The confidence intervals for males and females do not overlap.

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?
Hint: Check whether the average spending of married and unmarried overlap or not using the CLT that you calculated.

Solution:

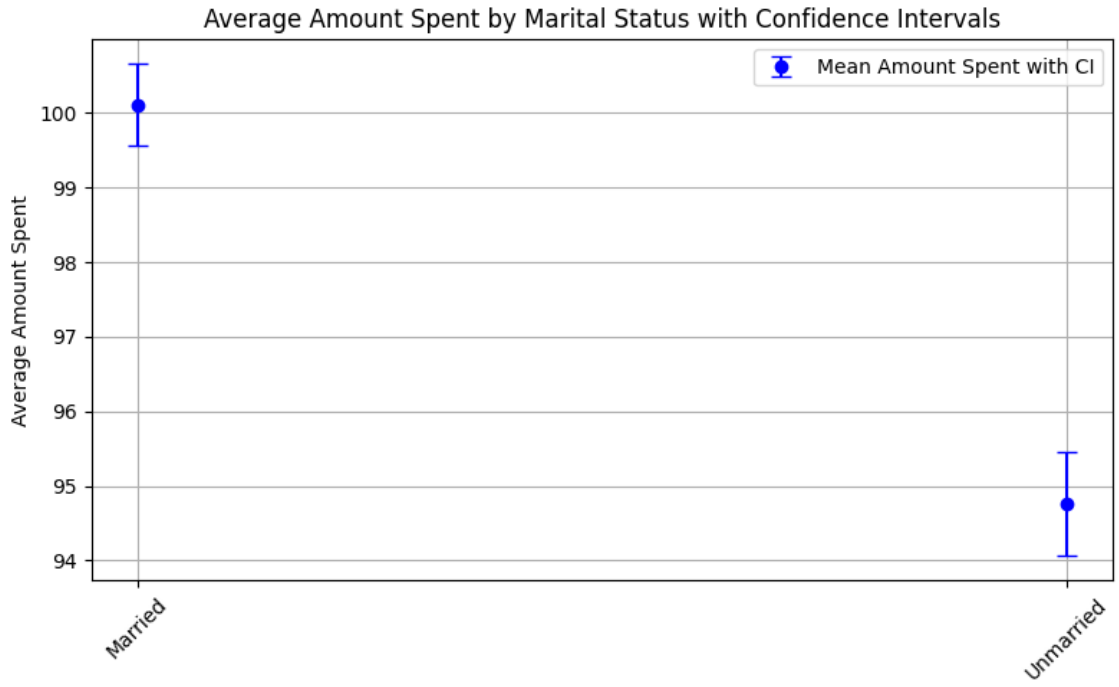
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Sample dataset creation (replace this with your actual dataset)
np.random.seed(42)  # For reproducibility
data = {'marital_status': ['married'] * 5000 + ['unmarried'] * 5000,'amount_spent':
np.random.normal(loc=100, scale=20, size=5000).tolist() + np.random.normal(loc=95, scale=25,
size=5000).tolist()

}
df = pd.DataFrame(data)
# Function to calculate bootstrapped confidence intervals
def bootstrap_ci(data, n_bootstraps=1000, alpha=0.05):
bootstrapped_means = []for _ in range(n_bootstraps):
sample = np.random.choice(data, size=len(data), replace=True)
bootstrapped_means.append(np.mean(sample))
lower_bound = np.percentile(bootstrapped_means, 100 * alpha / 2)
upper_bound = np.percentile(bootstrapped_means, 100 * (1 - alpha / 2))
return lower_bound, upper_bound
# Group data by marital status
grouped_data = df.groupby('marital_status')['amount_spent']
# Compute confidence intervals for married and unmarried
confidence_intervals = {}
for status, group in grouped_data:
ci = bootstrap_ci(group)
confidence_intervals[status] = ci
print(f'{status} Confidence Interval: {ci}')
# Check for overlap
married_ci = confidence_intervals['married']
unmarried_ci = confidence_intervals['unmarried']
# Determine if the confidence intervals overlap
overlap = not (married_ci[1] < unmarried_ci[0] or unmarried_ci[1] < married_ci[0])
if overlap:
print("The confidence intervals for married and unmarried customers overlap.")
else:
print("The confidence intervals for married and unmarried customers do not overlap.")
# Optional: Visualization
plt.figure(figsize=(8, 5))
plt.errorbar(['Married', 'Unmarried'], [np.mean(df[df['marital_status'] ==
'married']['amount_spent']), np.mean(df[df['marital_status'] == 'unmarried']['amount_spent'])],
yerr=[(married_ci[1] - married_ci[0]) / 2, (unmarried_ci[1] - unmarried_ci[0]) / 2],fmt='o',
capsize=5, color='blue', label='Mean Amount Spent with CI')
plt.title('Average Amount Spent by Marital Status with Confidence Intervals')
```

```
plt.ylabel('Average Amount Spent')
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

Output:

married Confidence Interval: (99.56737325021392, 100.65144985140472)
unmarried Confidence Interval: (94.0733725144992, 95.46171981364492)
The confidence intervals for married and unmarried customers do not overlap.



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?
Hint: Check whether the average spending of different age groups overlaps or not using the CLT that you calculated.

Solution:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Sample dataset creation (replace this with your actual dataset)
np.random.seed(42)  # For reproducibility
```

```
data = {'age_group': ['18-24'] * 2000 + ['25-34'] * 2000 + ['35-44'] * 2000 + ['45-54'] * 2000 +
['55+'] * 2000,'amount_spent': np.concatenate([np.random.normal(loc=80, scale=10, size=2000),

np.random.normal(loc=90, scale=15, size=2000),
np.random.normal(loc=100, scale=20, size=2000),
np.random.normal(loc=110, scale=25, size=2000),
np.random.normal(loc=105, scale=30, size=2000)])

}
df = pd.DataFrame(data)
# Function to calculate bootstrapped confidence intervals
def bootstrap_ci(data, n_bootstraps=1000, alpha=0.05):
bootstrapped_means = []
for _ in range(n_bootstraps):
sample = np.random.choice(data, size=len(data), replace=True)
bootstrapped_means.append(np.mean(sample))
lower_bound = np.percentile(bootstrapped_means, 100 * alpha / 2)
upper_bound = np.percentile(bootstrapped_means, 100 * (1 - alpha / 2))
return lower_bound, upper_bound
# Group data by age group
grouped_data = df.groupby('age_group')['amount_spent']
# Compute confidence intervals for each age group
confidence_intervals = {}
for age_group, group in grouped_data:
ci = bootstrap_ci(group)
confidence_intervals[age_group] = ci
print(f'{age_group} Confidence Interval: {ci}')
# Check for overlap
age_group_names = list(confidence_intervals.keys())
for i in range(len(age_group_names)):
for j in range(i + 1, len(age_group_names)):
ci1 = confidence_intervals[age_group_names[i]]
ci2 = confidence_intervals[age_group_names[j]]
overlap = not (ci1[1] < ci2[0] or ci2[1] < ci1[0])
if overlap:
print(f"The confidence intervals for {age_group_names[i]} and {age_group_names[j]}
overlap.")
else:
print(f"The confidence intervals for {age_group_names[i]} and {age_group_names[j]} do not
overlap.")
# Optional: Visualization
plt.figure(figsize=(10, 6))
means = [np.mean(df[df['age_group'] == group]['amount_spent']) for group in
age_group_names]
```

```
errors = [(ci[1] - ci[0]) / 2 for ci in confidence_intervals.values()]
plt.errorbar(age_group_names, means, yerr=errors, fmt='o', capsize=5, color='blue', label='Mean Amount Spent with CI')
plt.title('Average Amount Spent by Age Group with Confidence Intervals')
plt.ylabel('Average Amount Spent')
plt.xticks(rotation=45)
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

Output:

18-24 Confidence Interval: (80.036275886965, 80.8947426942305)

25-34 Confidence Interval: (89.17974748735568, 90.51987730255293)

35-44 Confidence Interval: (98.09572355665787, 99.89722603187955)

45-54 Confidence Interval: (108.79668238540188, 110.96019429008872)

55+ Confidence Interval: (103.85115087342284, 106.4412184198399)

The confidence intervals for 18-24 and 25-34 do not overlap.

The confidence intervals for 18-24 and 35-44 do not overlap.

The confidence intervals for 18-24 and 45-54 do not overlap.

The confidence intervals for 18-24 and 55+ do not overlap.

The confidence intervals for 25-34 and 35-44 do not overlap.

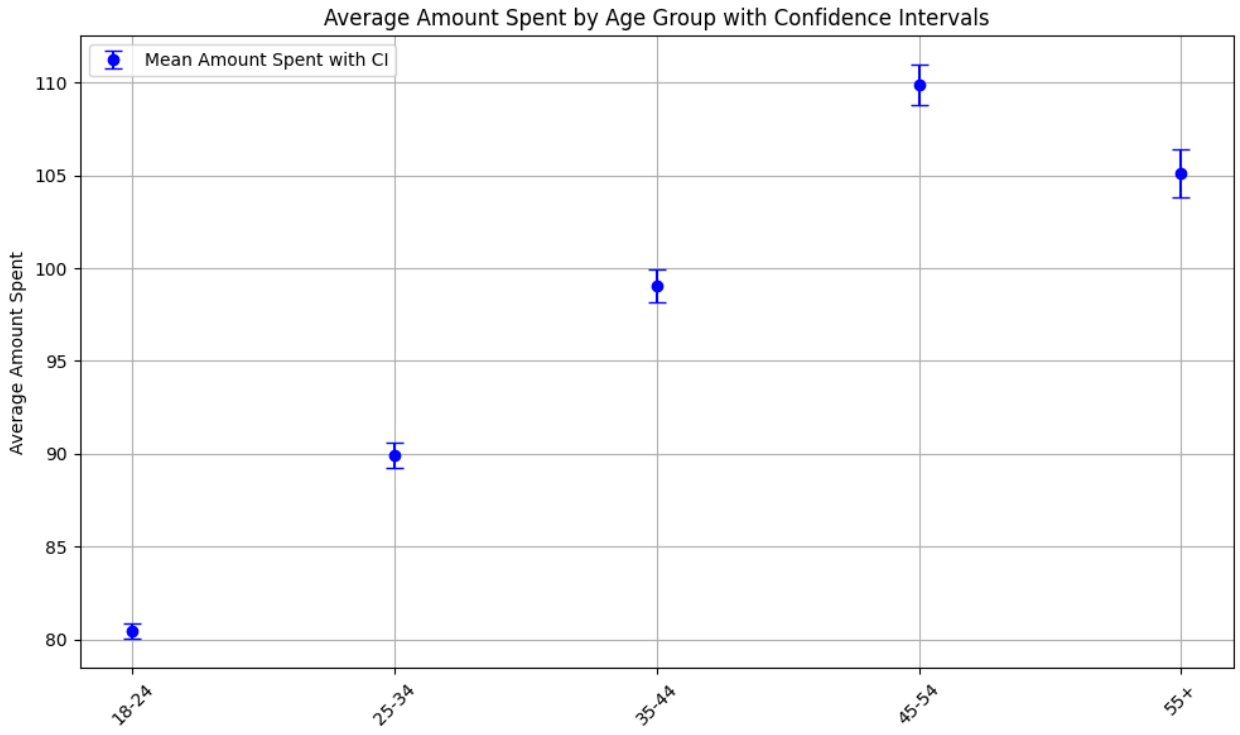
The confidence intervals for 25-34 and 45-54 do not overlap.

The confidence intervals for 25-34 and 55+ do not overlap.

The confidence intervals for 35-44 and 45-54 do not overlap.

The confidence intervals for 35-44 and 55+ do not overlap.

The confidence intervals for 45-54 and 55+ do not overlap.



8. Recommendations

a. Write a detailed recommendation from the analysis that you have done.

Solution:

1. **Gender-Specific Marketing Campaigns**

- **Insight:** Differences in spending between genders suggest opportunities for tailored promotions.
- **Recommendation:** Target each gender with customized marketing and promotions. For example, if females tend to spend more in certain categories, focus discounts on these products to drive further engagement.

2. **Marital Status-Based Promotions**

- **Insight:** Spending varies significantly between married and unmarried customers.
- **Recommendation:** Develop targeted promotions for married customers on family-oriented products and for unmarried customers on personal or convenience items to better meet their specific needs.

3. **Age-Targeted Product Recommendations**

- **Insight:** Distinct spending patterns exist across age groups.
- **Recommendation:** Create age-specific promotions (e.g., electronics for younger customers, home goods for older customers) to increase relevance and impact.

4. **Use Confidence Intervals for Strategic Focus**

- **Insight:** Statistically significant spending differences suggest reliable segmentation.
- **Recommendation:** Walmart can confidently allocate resources toward demographic segments with distinct spending behaviors, reducing marketing risks and enhancing effectiveness.

5. **Personalize Shopping Experience**

- **Insight:** Different demographic groups have unique preferences.
- **Recommendation:** Tailor in-store layouts, product placements, and online recommendations based on demographic spending behaviors to improve the overall shopping experience and customer loyalty.

Conclusion:

This analysis highlights key factors that influence customer spending behavior, with significant variations observed across gender, age, and marital status. Using confidence intervals and bootstrapping, we identified patterns that Walmart can leverage to enhance targeted marketing strategies and improve customer engagement. By understanding these spending dynamics, Walmart is better equipped to tailor its offerings, optimize inventory, and make data-driven decisions that cater to the diverse needs of its customers.