

Walmart - Confidence Interval and CLT

Problem Statement

The Management team at Walmart Inc. wants to understand the customer spending patterns across variuos demographics such as gender, age, location, occupation, and marital status to help the business make better decisions.

Objective

Analyze customer purchase behavior using the walmart dataset to identify customers spending patterns and provide actionable insights for better decisions. The objectives are as follows:


- Identify whether women spend more on Black Friday than men.
- Determine which product categories are preferred by different genders.
- Examine how age, location, occupation, and marital status impact spending habits.
- Identify whether there is a relationship between age, marital status, and the amount spent.
- Perform statistical analysis, including confidence intervals, to determine the significance of spending differences between genders.

Importing libraries

```
import pandas as pd
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the data

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



	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	1422


Basic Analysis of Data

```
df.shape
```



```
(550068, 10)
```

```
# data type of each column
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

```

```

# Checking null values
print(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

```

```
df[df.duplicated()]
```

```

↗ User_ID Product_ID Gender Age Occupation City_Category Stay_In_Current_City_Years Marital_Status Product_Category Purchase

```

There is no duplicate records in the dataset.

```

# Statistical Summary
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

Observation

- Both mean and median of the product category are 5, it suggests that Category 5 is widely purchased across all demographics.
- Median purchase amount is ₹8,047, suggesting most customers spend around this range.
- The spread of data points varies from the mean value, with the minimum Purchase amount is being USD 12, while maximum being USD 23,961, indicating the possibility of outliers.

```
df.describe(include="O")
```

```

↗

```

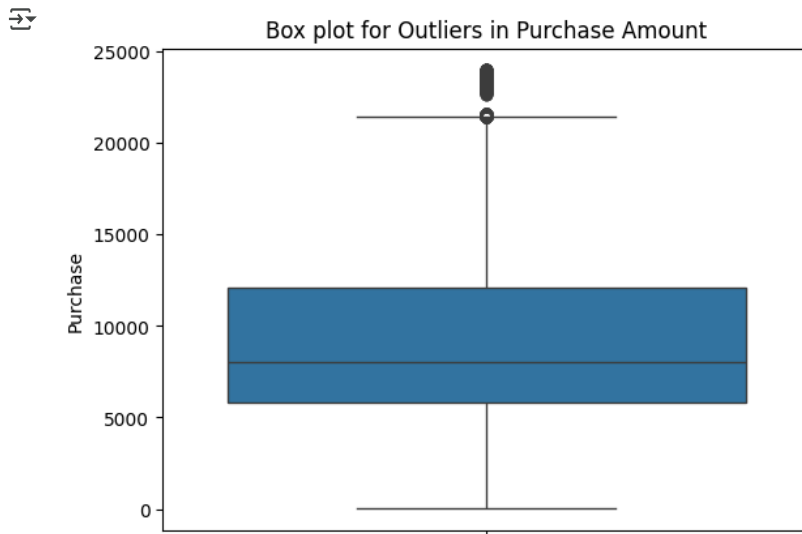
	Product_ID	Gender	Age	City_Category	Stay_In_Current_City_Years
count	550068	550068	550068	550068	550068
unique	3631	2	7	3	5
top	P00265242	M	26-35	B	1
freq	1880	414259	219587	231173	193821

Observation

- P00265242 is the most purchased product.
- Male made the highest number of purchases.
- Customers age group 26-35 made the highest purchase.
- City_Category B has made the highest number of purchases.

✓ Detect outliers

```
plt.title("Box plot for Outliers in Purchase Amount")
sns.boxplot(df['Purchase'])
plt.show()
```



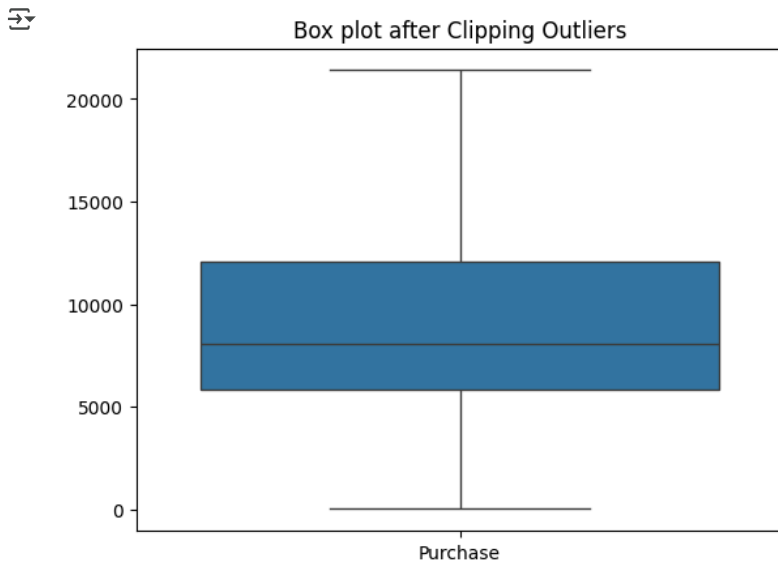
```
Q1 = df['Purchase'].quantile(0.25)
Q3 = df['Purchase'].quantile(0.75)
df_median=df['Purchase'].median()
IQR = Q3 - Q1
lower_bound = Q1 - (1.5 * IQR)
upper_bound = Q3 + (1.5 * IQR)
print("upper_bound :", upper_bound)
print("lower_bound :", lower_bound)
print(df_median)
outliers=df[(df['Purchase'] < lower_bound) | (df['Purchase'] > upper_bound)]
print(outliers.shape[0])
```

```
upper_bound : 21400.5
lower_bound : -3523.5
8047.0
2677
```

Observation

- Any value above the upper bound is considered an outlier.
- There are 2677 outliers in the purchase column.
- Since purchase amount plays a key role in understanding customer spending patterns, instead of removing these outliers, we can clip them to limit their impact.

```
plt.title("Box plot after Clipping Outliers")
clipped_data = np.clip(df['Purchase'], lower_bound, upper_bound)
sns.boxplot([clipped_data])
plt.show()
```



✓ Exploratory Data Analysis

Unique Value Count

```
df.columns
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
```

Observation

From above observation we will keep only the purchase column as numeric and convert the others into categories for easier analysis.

```
# Data type conversion
walmart_df=df.copy()
columns_to_convert = walmart_df.columns[:-1]
walmart_df[columns_to_convert] = walmart_df[columns_to_convert].apply(lambda x: x.astype('category'))
walmart_df['Marital_Status'] = walmart_df['Marital_Status'].apply(lambda x: 'Married' if x == 1 else 'Single')
```

```
walmart_df["Gender"].value_counts(normalize=True).round(2)
```

```
proportion
Gender
M      0.75
F      0.25
```

dtype: float64

```
walmart_df.groupby("Gender",observed=True)["User_ID"].nunique()
```

```
User_ID
Gender
F      1666
M      4225
```

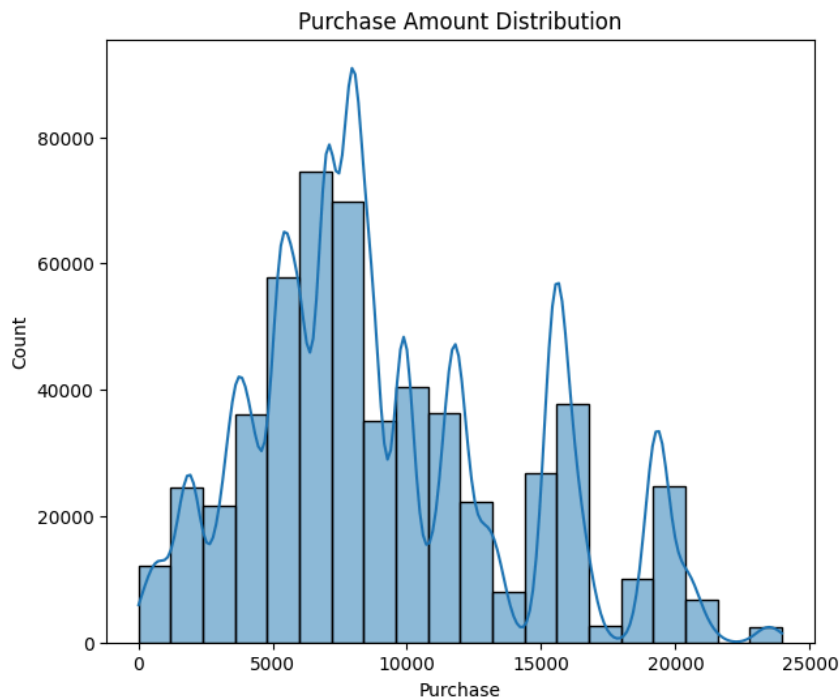
dtype: int64

Observation

The dataset clearly shows that male appear significantly more than female.

✓ Univariate Analysis

```
plt.figure(figsize=(7,6))
plt.title("Purchase Amount Distribution")
sns.histplot(df['Purchase'],bins=20,kde=True)
plt.show()
```

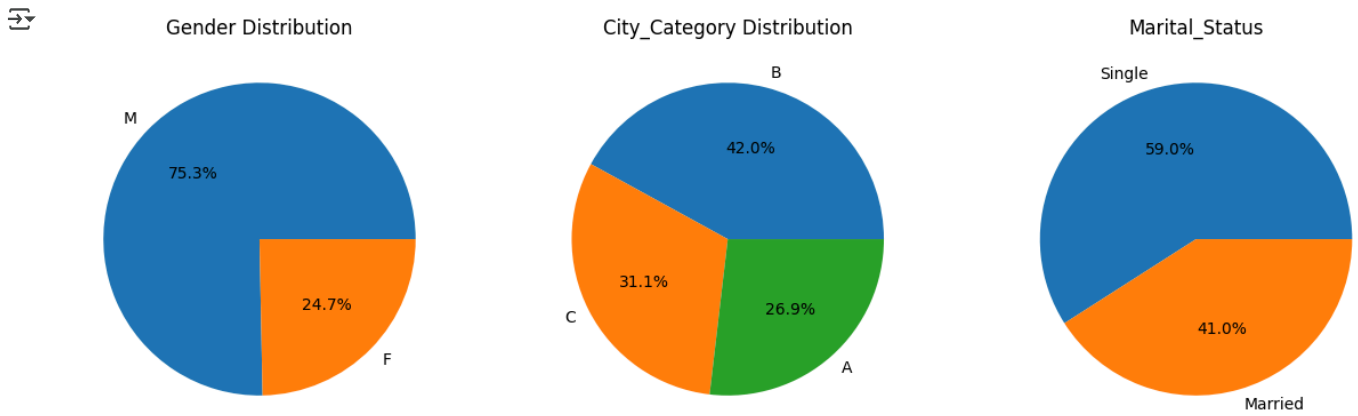


Insights

- We can clearly see that purchase amount is not normally distributed, it has multiple peaks, meaning different groups of customers have distinct buying habits.
- The highest peak is around **\$ 8,000**, suggesting that majority of purchases fall within the **5,000 to 10,000** range.
- The second highest notable peak is around **\$ 15,000**.
- A smaller group of customers have purchases exceeding **\$ 21,000**.

✓ Categorical Variables

```
fig, axes = plt.subplots(1, 3, figsize=(15,10))
columns=['Gender','City_Category','Marital_Status']
titles = ['Gender Distribution', 'City_Category Distribution', 'Marital_Status']
for i, col in enumerate(columns):
    walmart_df[col].value_counts(normalize=True).plot(kind='pie', autopct='%1.1f%%', ax=axes[i])
    axes[i].set_title(titles[i])
    axes[i].set_ylabel('')
plt.show()
```

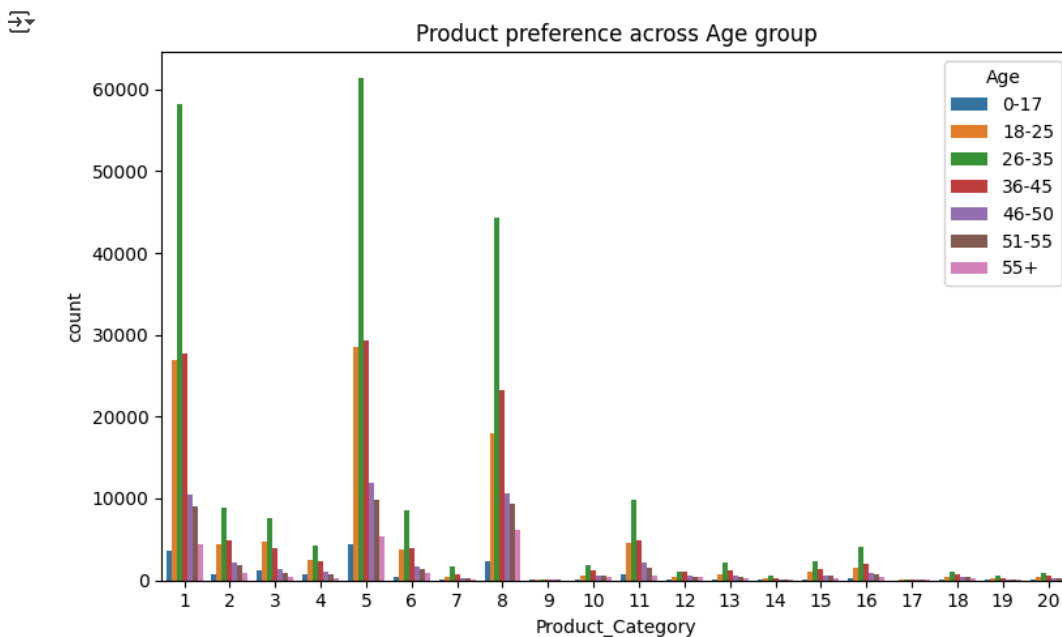


Insights

- The percentage of male is higher compared to female, which clearly indicates that male customers made significantly higher purchases than female during black friday sales.
- Among all the cities, people in Category B made the highest purchases, followed by those in Category C.
- Single customers made higher purchases than married customers.

Product_Category vs Age

```
plt.figure(figsize=(8,5))
plt.title("Product preference across Age group")
sns.countplot(x="Product_Category",hue="Age",data=walmart_df)
plt.tight_layout()
```



Insights

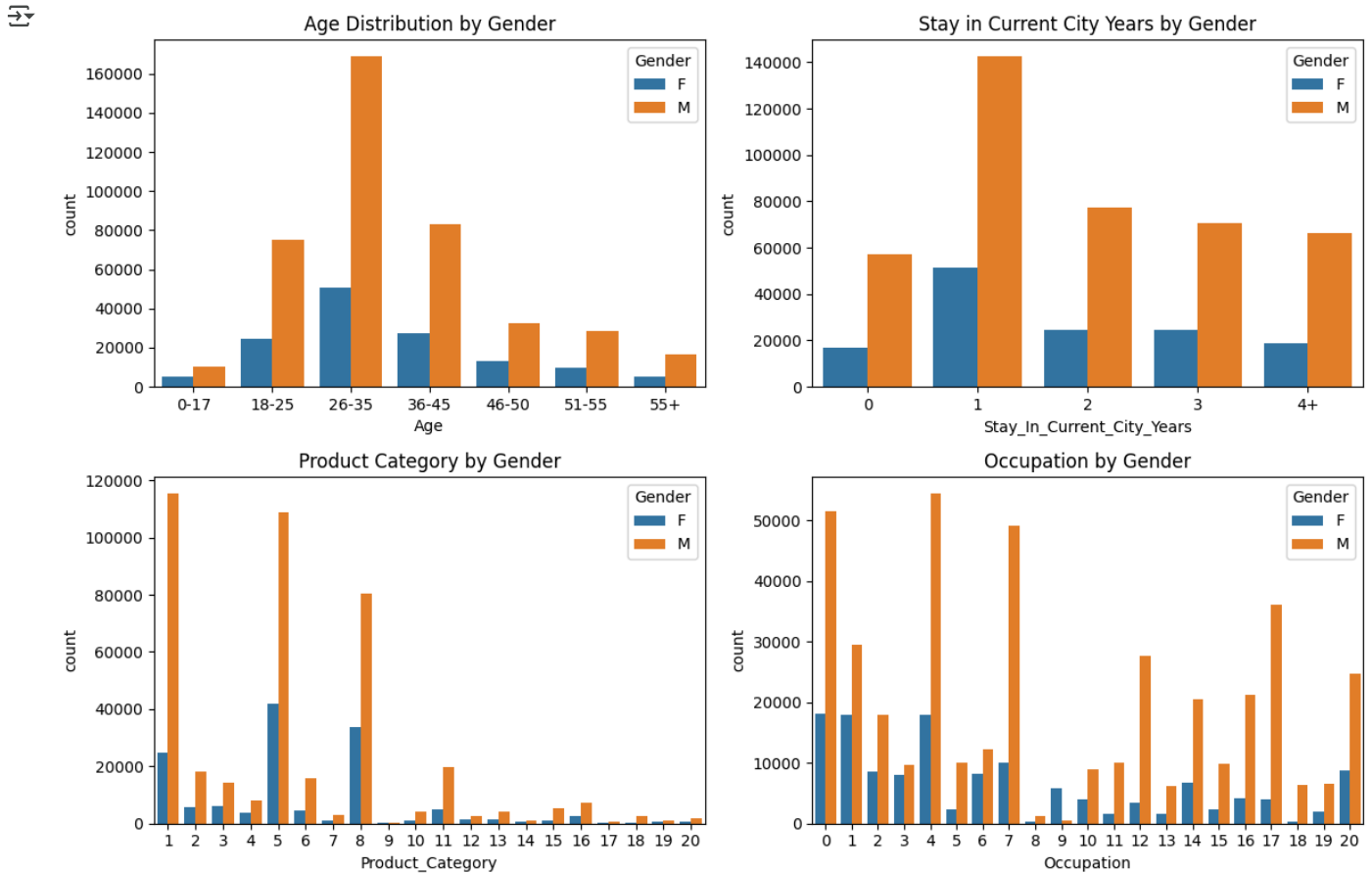
- **Product Categories 1, 5, and 8** are the top most purchased across all age groups.
- The age group 26-35 is the dominant group across all product categories. Product Categories 1 and 5 have more than 50,000 customers in the 26-35 age group.
- The second highest purchase frequency comes from the **18-25 and 36-45** age groups.

Distribution of Gender

```

columns = ['Age', 'Stay_In_Current_City_Years', 'Product_Category', 'Occupation']
titles = ['Age Distribution by Gender', 'Stay in Current City Years by Gender', 'Product Category by Gender', 'Occupation by Gender']
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
axes = axes.flatten()
for i, column in enumerate(columns):
    sns.countplot(x=column, hue="Gender", data=walmart_df, ax=axes[i])
    axes[i].set_title(titles[i])
plt.tight_layout()
plt.show()

```



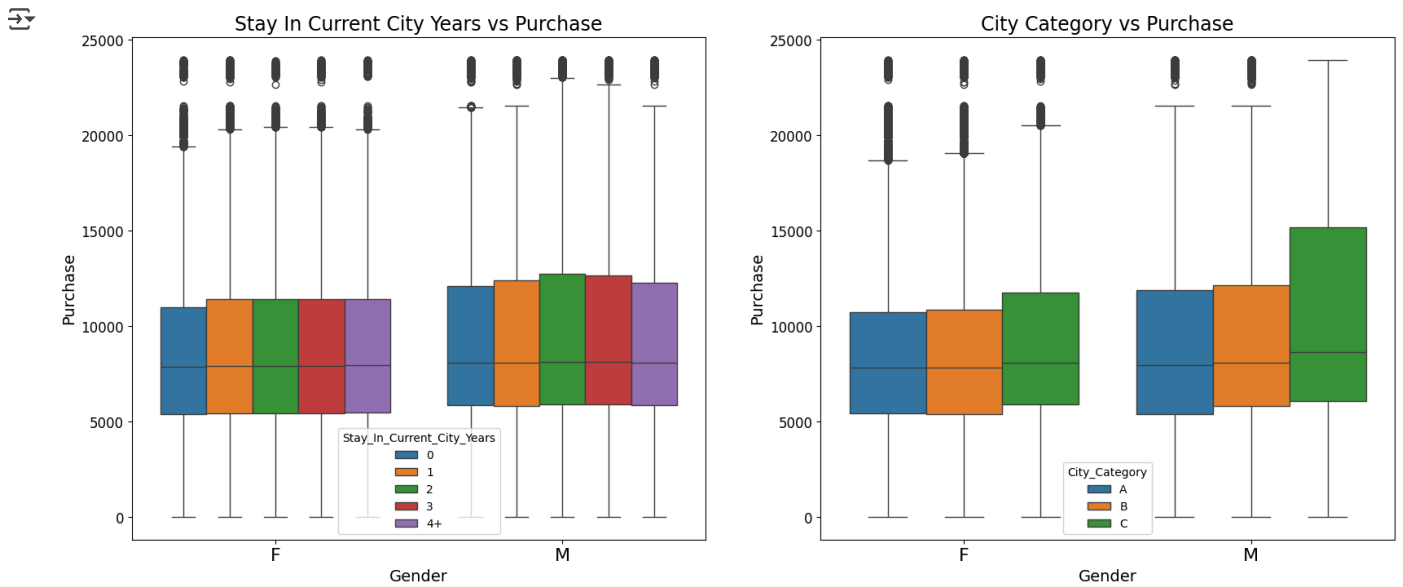
- Across all age groups, male made higher purchases than female. The most notable group is 26-35, where the difference is significantly high.
- Customers who have lived in their current city for one year made the highest purchases than who have been living there for over 4 years especially male.
- The top product categories are have highest customers in both male and female.
- Customers in Occupation categories 0, 4, 7 made most frequent purchase especially male. While female customers contributed across all the occupation categories, males had significantly higher presense.
- So we can infer that males contibuted significantly across all the variables.

✓ Spending Patterns Across Locations

```

features = [("Gender", "Stay_In_Current_City_Years", "Stay In Current City Years vs Purchase"), ("Gender", "City_Category", "City Category \
    ]
fig, axes = plt.subplots(nrows=1, ncols=len(features), figsize=(20,8))
for i, (x_col, hue_col, title) in enumerate(features):
    sns.boxplot(x=x_col, y="Purchase", hue=hue_col, data=walmart_df, ax=axes[i])
    axes[i].set_title(title, fontsize=17)
    axes[i].set_xlabel(x_col, fontsize=14)
    axes[i].set_ylabel("Purchase", fontsize=14)
    axes[i].tick_params(axis='x', labels=16)
    axes[i].tick_params(axis='y', labels=12)
plt.show()

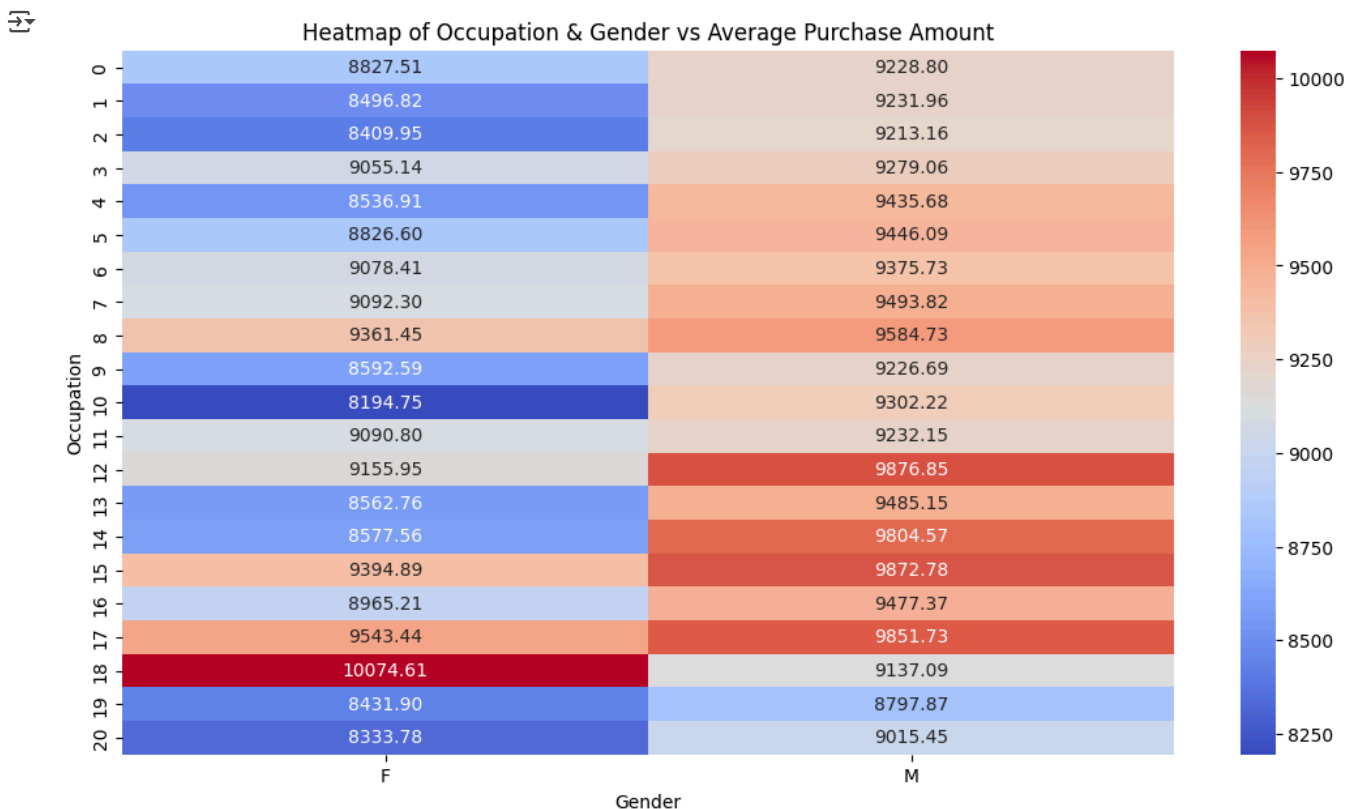
```



- Interestingly, male who live in City Category C did not show any outliers, indicating that they tend to spend more uniformly during Black Friday, rather than making extreme purchases like in other city categories.
- Customers who have stayed in the city for different years spend about the same per transaction regardless of how long they've been there.

✓ Occupation & Spending Patterns

```
pivot_table = walmart_df.pivot_table(values='Purchase', index='Occupation', columns='Gender', aggfunc='mean', observed=True)
plt.figure(figsize=(13, 7))
sns.heatmap(pivot_table, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Heatmap of Occupation & Gender vs Average Purchase Amount")
plt.xlabel("Gender")
plt.ylabel("Occupation")
plt.show()
```



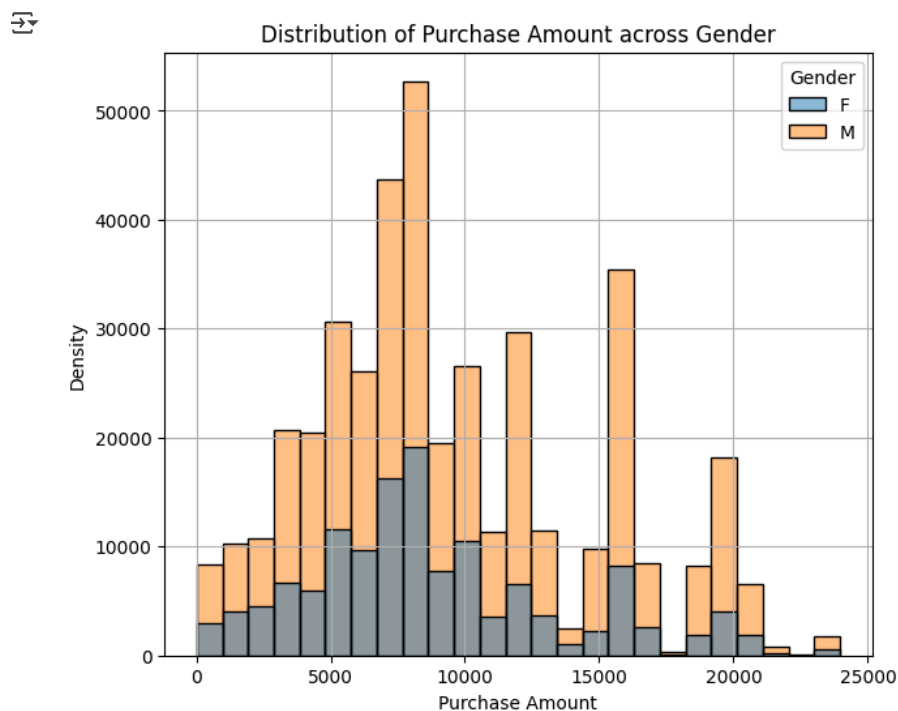
- We can clearly see that female spending is inconsistent. For some purchases they spent less, sometimes spent high amount.
- Male on the other hand spending consistently throughout black friday with average spending higher than female.
- We can infer that inconsistency in spending could be one of the reasons why women spend less than men.

✓ Average Purchase Amount by Gender

```
walmart_df.groupby('Gender',observed=False)['Purchase'].describe()
```

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

```
plt.figure(figsize=(7,6))
sns.histplot(x="Purchase",hue="Gender",bins=25,data=walmart_df, edgecolor="black")
plt.xlabel("Purchase Amount")
plt.ylabel("Density")
plt.title("Distribution of Purchase Amount across Gender")
plt.grid(True)
plt.show()
```



Insights

- The KDE plot shows that both males and females have different buying habits, with multiple peaks in the amount they spend.
- we can infer that men usually spend significantly high than female.
- The data is not normally distributed, showing a wider spread right side, meaning people tend to spend more in larger amounts.
- The average purchase amount is between 8000 to 9000. However based on this data we cannot conclude that the population parameter will fall with in this range as since the data represents sample. It may or may not fall with in this range.

✓ Central Limit Theorem

To estimate the population parameter, we can take samples and apply the Central Limit Theorem. If we take large samples and repeat the process multiple times the sample mean will be close to population mean. Also if we plot these means their distribution will be approximately normal, regardless of population distribution.

```
#For entire dataset
def data_ci(data, variable, category,n_iterations=1000, confidence_level=0.95):
```

```

category_data = data[data[variable] == category]['Purchase']
bootstrap_means = []

for _ in range(n_iterations):
    bootstrap_sample = category_data.sample(n=len(category_data), replace=True, random_state=None)
    bootstrap_mean = bootstrap_sample.mean()
    bootstrap_means.append(bootstrap_mean)

bootstrap_means = np.array(bootstrap_means)
lower_bound = np.percentile(bootstrap_means, (1 - confidence_level) / 2 * 100)
upper_bound = np.percentile(bootstrap_means, (1 + confidence_level) / 2 * 100)
bootstrap_mean = bootstrap_means.mean()
ci_width = upper_bound - lower_bound
print(f'{category} Mean: {bootstrap_mean:0.2f}, Confidence Interval: ({lower_bound:0.2f}, {upper_bound:0.2f}), Width: {ci_width:0.2f}')

#bootstrapping for samples
def sample_ci(data, variable, category, sample_sizes=[300, 3000, 30000], n_iterations=1000, confidence_level=0.95):

    for sample_size in sample_sizes:
        category_data = data[data[variable] == category]['Purchase']
        bootstrap_means = []

        for _ in range(n_iterations):
            bootstrap_sample = category_data.sample(n=sample_size, replace=True, random_state=None)
            bootstrap_mean = bootstrap_sample.mean()
            bootstrap_means.append(bootstrap_mean)

        bootstrap_means = np.array(bootstrap_means)
        lower_bound = np.percentile(bootstrap_means, (1 - confidence_level) / 2 * 100)
        upper_bound = np.percentile(bootstrap_means, (1 + confidence_level) / 2 * 100)
        bootstrap_mean = bootstrap_means.mean()
        ci_width = upper_bound - lower_bound
        print(f'Sample Size: {sample_size}')
        print(f'{category} Mean: {bootstrap_mean:0.2f}, Confidence Interval: ({lower_bound:0.2f}, {upper_bound:0.2f}), Width: {ci_width:0.2f}')

#Visualize
def bootstrap_sample_means(data, sample_size, iterations=1000):
    sample_means = [
        np.mean(np.random.choice(data, size=sample_size, replace=True))
        for _ in range(iterations)
    ]
    return sample_means

def visualize_sample_means_distribution(data, variable, categories, sample_sizes=[300, 3000, 30000], iterations=1000, colors=None):
    fig, axes = plt.subplots(1, len(sample_sizes), figsize=(25,8))
    if colors is None:
        colors = ['green', 'red']
    if len(categories) > len(colors):
        colors = colors * (len(categories) // len(colors)) + colors[:len(categories) % len(colors)]

    for idx, size in enumerate(sample_sizes):
        ax = axes[idx]

        for cat_idx, category in enumerate(categories):
            category_data = data[data[variable] == category]['Purchase']
            sample_means = bootstrap_sample_means(category_data, size, iterations)
            sns.histplot(sample_means, kde=True, bins=30, label=f'{category}', color=colors[cat_idx], stat='density', ax=ax)
        ax.set_title(f'Distribution of Bootstrap Sample Means (Size = {size})')
        ax.set_xlabel('Sample Means')
        ax.set_ylabel('Density')
        ax.legend(title="Category")
        ax.grid(True)
    plt.tight_layout()
    plt.show()

```


✓ Confidence intervals for the Average amount spent per gender

Entire Dataset

```

categories=['M','F']
for category in categories:
    data_ci(walmart_df, 'Gender', category)

```

 M Mean: 9437.89, Confidence Interval: (9422.68, 9452.96), Width: 30.28
 F Mean: 8734.84, Confidence Interval: (8709.86, 8760.90), Width: 51.05

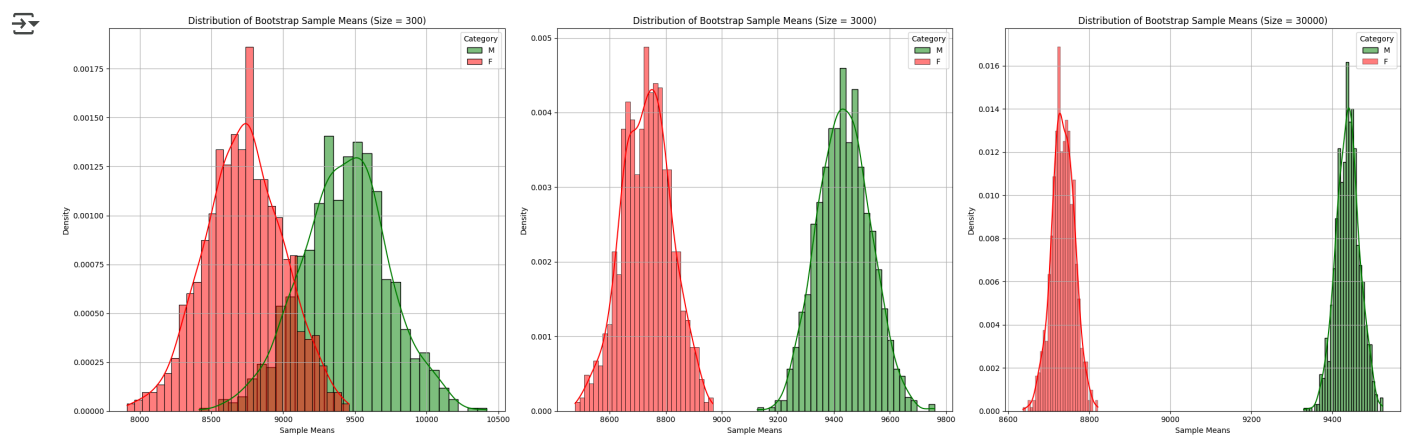
For Sample

```
sample_ci(walmart_df, 'Gender', 'M', sample_sizes=[300, 3000, 30000], n_iterations=1000, confidence_level=0.95)
sample_ci(walmart_df, 'Gender', 'F', sample_sizes=[300, 3000, 30000], n_iterations=1000, confidence_level=0.95)
```

```
↩ Sample Size: 300
M Mean: 9432.78, Confidence Interval: (8832.56, 10084.11), Width: 1251.55
Sample Size: 3000
M Mean: 9439.16, Confidence Interval: (9256.17, 9631.52), Width: 375.35
Sample Size: 30000
M Mean: 9437.71, Confidence Interval: (9378.31, 9495.55), Width: 117.23
Sample Size: 300
F Mean: 8730.75, Confidence Interval: (8176.77, 9260.10), Width: 1083.33
Sample Size: 3000
F Mean: 8738.47, Confidence Interval: (8569.29, 8902.99), Width: 333.70
Sample Size: 30000
F Mean: 8734.55, Confidence Interval: (8681.55, 8786.80), Width: 105.25
```

✓ Distribution of Sample Mean Purchase Amount Across Genders

```
categories = ['M', 'F']
visualize_sample_means_distribution(walmart_df, variable='Gender', categories=categories, sample_sizes=[300, 3000, 30000], iterations=1000)
```



- As the sample size increases the distribution becomes more normal, and the curve gets narrower with less spread.
- The confidence interval for the entire dataset is **wider for female**, due to variations in the average purchase amounts. Some female might have spent significantly more, while others spent much less.
- As sample size increases the width of confidence interval decreases.
- The calculated confidence intervals for sample size 300 overlap. But as size increases they become more separated.
- We can say with 95% confidently that the population average will lie between the confidence intervals for **male (9378.31, 9495.55)** and female (**8681.55, 8786.80**). we can infer that the average purchase amount for male is higher than for female.
- As the sample size increases the distribution of the mean becomes narrower and tighter eventually forming a sharper higher peak.

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

Women spend less per transaction because they tend to shop more often for everyday things like groceries and household items, which are usually cheaper. Women may also be more careful with their spending, looking for discounts and comparing prices, which leads to smaller purchases at a time.

How can Walmart leverage this conclusion to make changes or improvements?

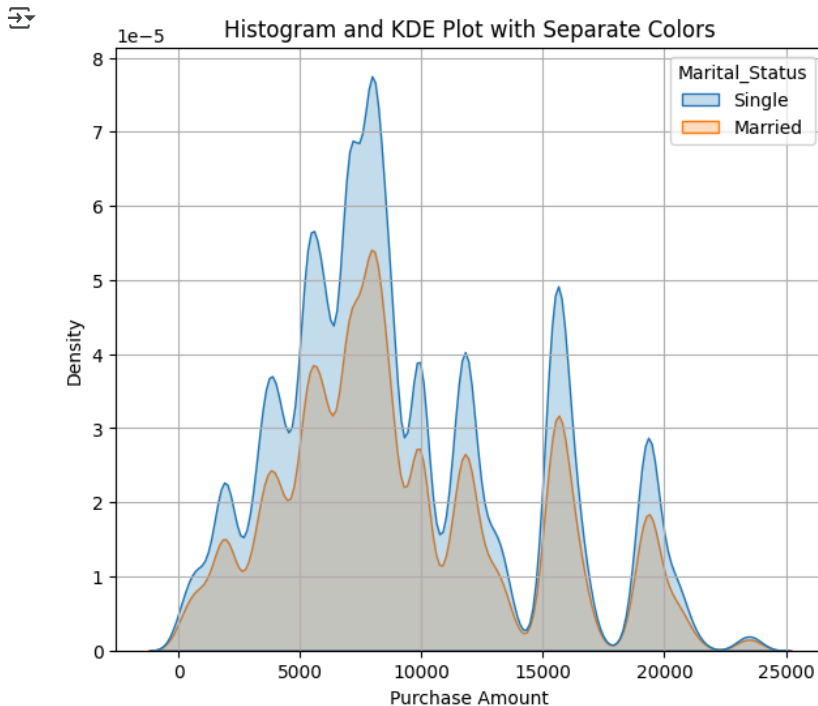
Men and women spending pattern is different. Walmart can use this to create special offers, stock more of the products each gender likes, and improve their ads to match these shopping habits.

✓ Average Purchase Amount by Marital_Status

```
walmart_df.groupby('Marital_Status', observed=False)['Purchase'].describe()
```

	count	mean	std	min	25%	50%	75%	max
Marital_Status								
Single	324731.0	9265.907619	5027.347859	12.0	5605.0	8044.0	12061.0	23961.0
Married	225337.0	9261.174574	5016.897378	12.0	5843.0	8051.0	12042.0	23961.0

```
plt.figure(figsize=(7,6))
sns.kdeplot(x="Purchase",hue="Marital_Status",fill=True,data=walmart_df)
plt.xlabel("Purchase Amount")
plt.ylabel("Density")
plt.title("Histogram and KDE Plot with Separate Colors")
plt.grid(True)
plt.show()
```



Insights

- Both single and married individuals made purchases ranging from low to extremely high amounts.
- Single purchases higher than married. The average amount spent per purchase is almost same for both single and married. Let's verify this using confidence interval

✓ Confidence intervals for the average amount spent per Marital_Status.

Entire Dataset

```
categories=['Single','Married']
for category in categories:
    data_ci(walmart_df,"Marital_Status",category)
```

Single Mean: 9266.25, Confidence Interval: (9248.61, 9283.62), Width: 35.00
 Married Mean: 9261.49, Confidence Interval: (9240.30, 9282.88), Width: 42.58

For sample

```
sample_ci(walmart_df,'Marital_Status','Single',sample_sizes=[300, 3000, 30000], n_iterations=1000, confidence_level=0.95)
```

```
Sample Size: 300
Single Mean: 9274.40, Confidence Interval: (8746.83, 9824.96), Width: 1078.13
Sample Size: 3000
Single Mean: 9261.09, Confidence Interval: (9084.46, 9430.89), Width: 346.43
Sample Size: 30000
Single Mean: 9265.92, Confidence Interval: (9206.36, 9321.07), Width: 114.71
```

```
sample_ci(walmart_df,'Marital_Status','Married',sample_sizes=[300, 3000, 30000], n_iterations=1000, confidence_level=0.95)
```

```

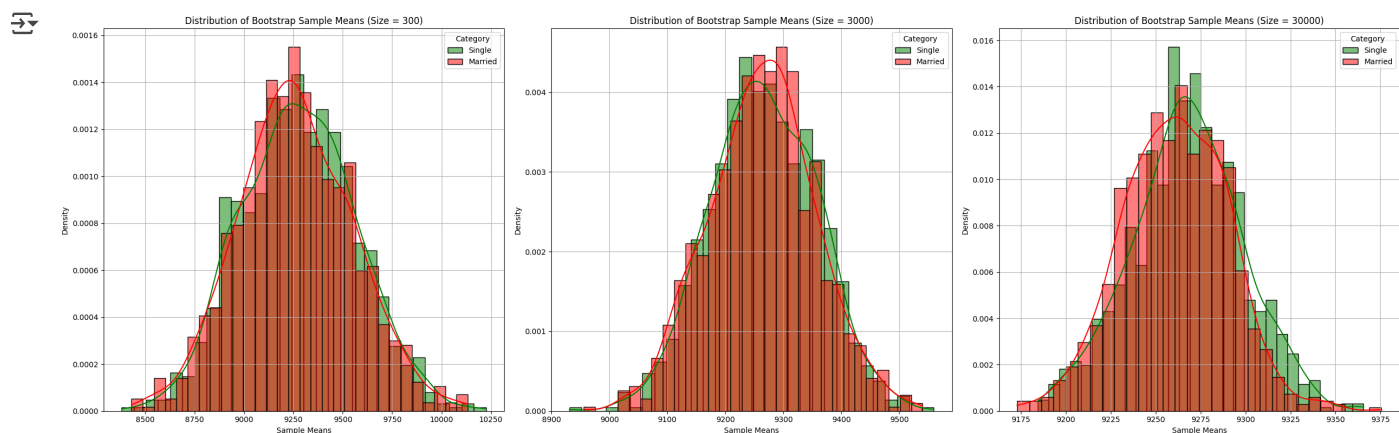
Sample Size: 300
Married Mean: 9251.56, Confidence Interval: (8646.23, 9851.15), Width: 1204.92
Sample Size: 3000
Married Mean: 9263.08, Confidence Interval: (9095.04, 9438.44), Width: 343.40
Sample Size: 30000
Married Mean: 9260.88, Confidence Interval: (9204.97, 9316.02), Width: 111.06

```

```

categories = ['Single', 'Married']
visualize_sample_means_distribution(walmart_df, variable='Marital_Status', categories=categories, sample_sizes=[300, 3000, 30000], iteratic

```



Insights

- The confidence interval computed using the entire dataset not wider for either group.
- For each of the three sample sizes the confidence intervals for both the groups are overlap. This suggests that the mean purchase amounts for either group are statistically close to each other across different sample sizes.
- We can say with 95% confidence that the population mean for married individuals spending falls between for **married (9204.97, 9316.02)** and for **single ((9206.36, 9321.07))**.

How can Walmart leverage this conclusion to make changes or improvements?

Since there is no big difference in spending between married and single customer, Walmart can keep things simple by offering the same promotions and products to everyone. This saves time and effort while still reaching all customers effectively.

✓ Average purchase amount for each age group

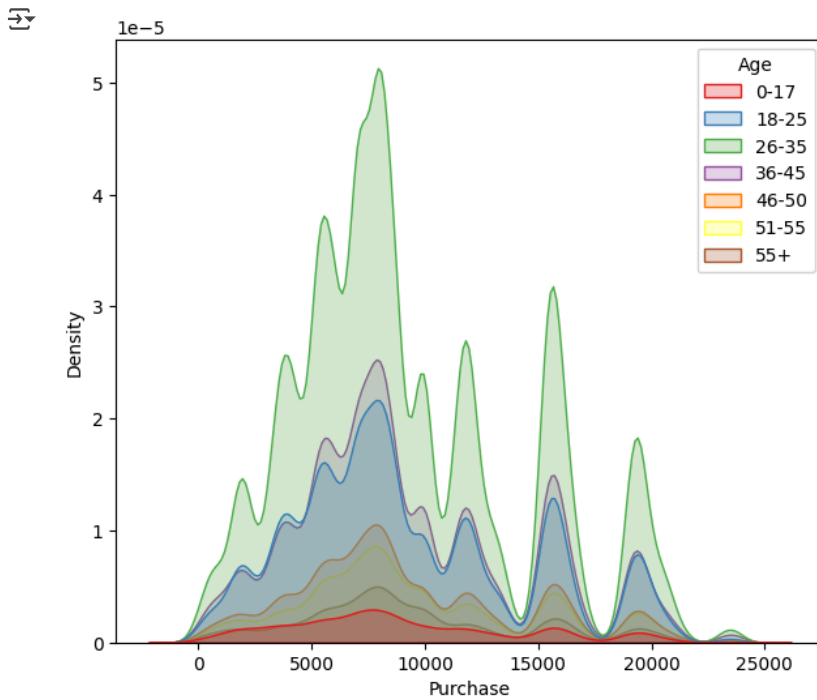
```
walmart_df.groupby('Age',observed=True)['Purchase'].describe()
```

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

```

plt.figure(figsize=(7,6))
sns.kdeplot(x='Purchase',hue='Age',fill=True,data=walmart_df,palette='Set1')
plt.show()

```



Insights

- Age group 26-35 has made significantly higher purchases, while the 0-17 age group has made very low.
- Overall middle aged group purchased significantly more than both older and younger groups.

✓ Confidence intervals for the average amount spent by Age Group

Entire Dataset

```
categories = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
for category in categories:
    data_ci(walmart_df, 'Age', category, confidence_level=0.95)
```

0-17 Mean: 8935.38, Confidence Interval: (8849.11, 9015.66), Width: 166.55
 18-25 Mean: 9169.23, Confidence Interval: (9137.15, 9201.80), Width: 64.65
 26-35 Mean: 9252.43, Confidence Interval: (9231.54, 9273.35), Width: 41.81
 36-45 Mean: 9331.78, Confidence Interval: (9301.83, 9360.88), Width: 59.05
 46-50 Mean: 9208.11, Confidence Interval: (9161.46, 9256.31), Width: 94.85
 51-55 Mean: 9534.29, Confidence Interval: (9484.39, 9583.10), Width: 98.71
 55+ Mean: 9337.09, Confidence Interval: (9270.80, 9399.34), Width: 128.54

For Samples

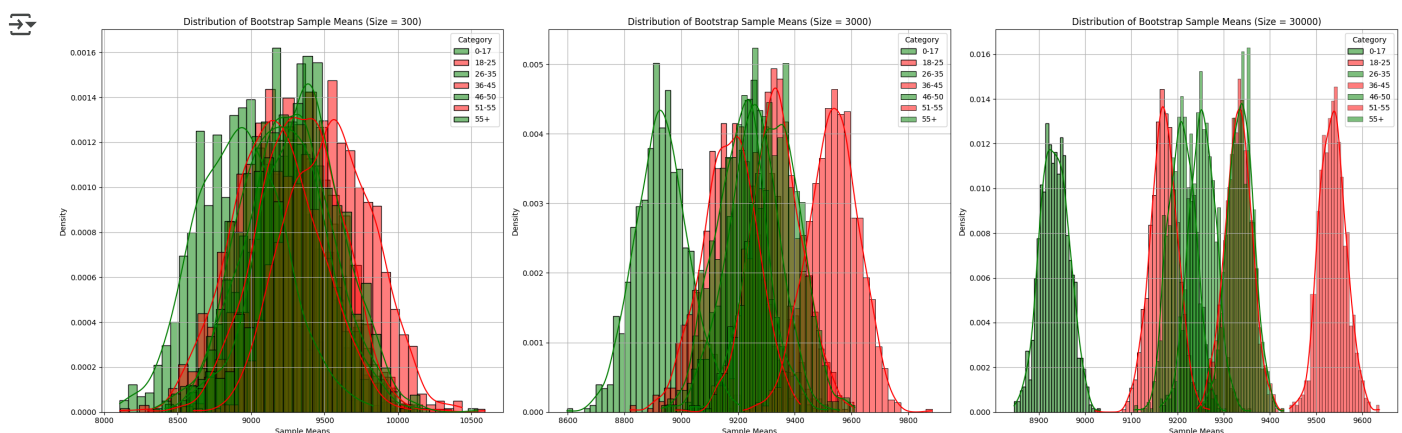
```
categories = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
for category in categories:
    sample_ci(walmart_df, 'Age', category, sample_sizes=[300, 3000, 30000], n_iterations=1000, confidence_level=0.95)
```

Sample Size: 300
 0-17 Mean: 8927.64, Confidence Interval: (8366.87, 9473.13), Width: 1106.26
 Sample Size: 3000
 0-17 Mean: 8931.36, Confidence Interval: (8742.13, 9116.28), Width: 374.16
 Sample Size: 30000
 0-17 Mean: 8934.38, Confidence Interval: (8880.84, 8991.87), Width: 111.03
 Sample Size: 300
 18-25 Mean: 9159.62, Confidence Interval: (8599.75, 9702.01), Width: 1102.26
 Sample Size: 3000
 18-25 Mean: 9168.82, Confidence Interval: (8988.81, 9365.46), Width: 376.65
 Sample Size: 30000
 18-25 Mean: 9170.05, Confidence Interval: (9114.96, 9225.61), Width: 110.65
 Sample Size: 300
 26-35 Mean: 9264.55, Confidence Interval: (8695.46, 9867.60), Width: 1172.14
 Sample Size: 3000
 26-35 Mean: 9251.22, Confidence Interval: (9076.22, 9419.29), Width: 343.08
 Sample Size: 30000
 26-35 Mean: 9253.18, Confidence Interval: (9199.58, 9313.18), Width: 113.60
 Sample Size: 300
 36-45 Mean: 9327.42, Confidence Interval: (8805.41, 9881.85), Width: 1076.44
 Sample Size: 3000
 36-45 Mean: 9331.81, Confidence Interval: (9141.16, 9515.52), Width: 374.36

Sample Size: 30000
 36-45 Mean: 9331.19, Confidence Interval: (9276.61, 9387.49), Width: 110.88
 Sample Size: 300
 46-50 Mean: 9210.90, Confidence Interval: (8667.86, 9738.65), Width: 1070.79
 Sample Size: 3000
 46-50 Mean: 9209.31, Confidence Interval: (9024.83, 9388.40), Width: 363.57
 Sample Size: 30000
 46-50 Mean: 9208.42, Confidence Interval: (9152.62, 9266.21), Width: 113.59
 Sample Size: 300
 51-55 Mean: 9530.22, Confidence Interval: (8999.35, 10109.27), Width: 1109.92
 Sample Size: 3000
 51-55 Mean: 9538.39, Confidence Interval: (9365.24, 9711.51), Width: 346.26
 Sample Size: 30000
 51-55 Mean: 9533.81, Confidence Interval: (9477.60, 9595.06), Width: 117.46
 Sample Size: 300
 55+ Mean: 9350.03, Confidence Interval: (8793.43, 9905.82), Width: 1112.39
 Sample Size: 3000
 55+ Mean: 9336.83, Confidence Interval: (9163.24, 9513.65), Width: 350.41
 Sample Size: 30000
 55+ Mean: 9336.29, Confidence Interval: (9283.08, 9392.42), Width: 109.33

✓ Distribution of sample mean across age group

```
categories = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
visualize_sample_means_distribution(walmart_df, variable='Age', categories=categories, sample_sizes=[300, 3000, 30000], iterations=1000)
```



Insights

- As the sample size increases the confidence interval become narrower. So the estimated population parameters are more reliable.
- The average spending amount for the 0-17 age group is lower compared to other age groups.
- **18-25, 26-35, 46-50** customers in these age groups are overlaps, indicating similarities in their spending patterns.
- The age groups **36-45 and 55+** have similarities in spending as their samples overlaps.
- Although the age group 51-55 makes fewest purchases, their average amount per purchase is higher than other age groups.
- The population mean will fall in the below range: 0 - 17 (8880.84, 8991.87) -18 - 25 (9114.96, 9225.61) -26 - 35 (9199.58, 9313.18) -36 - 45 (9276.61, 9387.49) -46 - 50 (9152.62, 9266.21) -51 - 55 (9477.60, 9595.06)
- 55+ (9283.08, 9392.42)

How can Walmart leverage this conclusion to make changes or improvements?

For 18-25, 26-35, and 46-50

Target Shared Interests: Walmart can focus on products like electronics, fashion, and health items that appeal across these age groups.

Bundle Offers: Combine fitness gear (dumbbells, resistance bands, yoga mats) with supplements (protein powders, vitamins) for a holistic health package.

student discounts: Walmart can offer student discounts on tech, books, school supplies, and fashion to attract the 18-25 group. For 26-35 and 46-50, discounts on family essentials can appeal to parents shopping for students.

For 36-45, 55+

Easy Shopping Solutions: Personal shopping assistance or easy online ordering, to make shopping more convenient for these groups.

Sustainability Focus: Promote eco-friendly or sustainable home products that might appeal to both groups, given their focus on home improvement and wellness.

▼ Recommendations

- Walmart should retain male customers. Focus on investing in targeted ads and promotions for women, improving the shopping experience with more relevant products, and offering more discounts on beauty and fashion.
- Walmart should prioritize keeping product categories 5, 8, and 1 well-stocked, as they are highly purchased by customers.
- Walmart could introduce a food court and a kids' play area to increase its appeal, which would encourage more families, especially married couples with kids, to visit the store.
- Customers in city category C whose purchasing pattern is vary from the rest. They may belong to affluent or executive-level occupations. Therefore walmart should focus on bringing in more targeted marketing strategies to retain customers.
- Walmart can increase sales in the 0-17 age group by creating promotions that appeal to both independent young consumers and parents purchasing for them.
- Walmart can offer loyalty cards with special privileges to attract customers who have been living in the current city for more than four years.
- Walmart should focus on retaining customers in occupations 0, 4, and 7 by sending personalized email offers.