Walmart Business case study Name: D A Santhosh

1. Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

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

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

```
# a. The data type of all columns in the “customers” table

data_types = df.dtypes
print("Data Types of Columns:")
print(data_types)
```

```
Data Types of Columns:
User_ID                          int64
Product_ID                      object
Gender                          object
Age                             object
Occupation                       int64
City_Category                   object
Stay_In_Current_City_Years      object
Marital_Status                 float64
Product_Category               float64
Purchase                       float64
dtype: object
```

```
# b. Number of rows and columns in the dataset

num_rows, num_columns = df.shape
print(f"\nNumber of Rows: {num_rows}")
print(f"Number of Columns: {num_columns}")
```

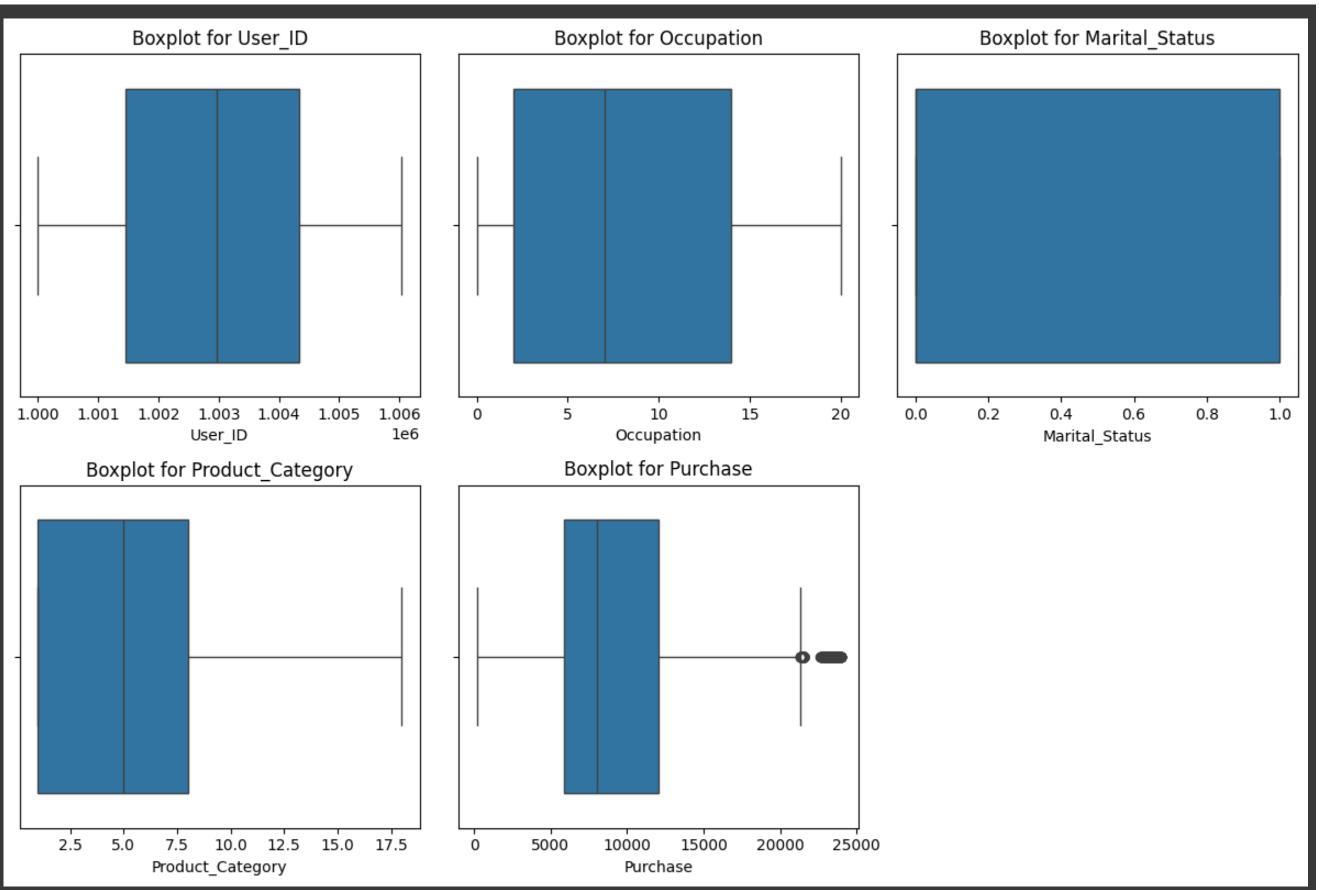
```
Number of Rows: 300528
Number of Columns: 10
```

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

missing_values = df.isnull().sum()
print("\nNumber of Missing Values in Each Column:")
print(missing_values)
```

```
Number of Missing Values in Each Column:
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 1
Stay_In_Current_City_Years    1
Marital_Status                1
Product_Category              1
Purchase                      1
dtype: int64
```

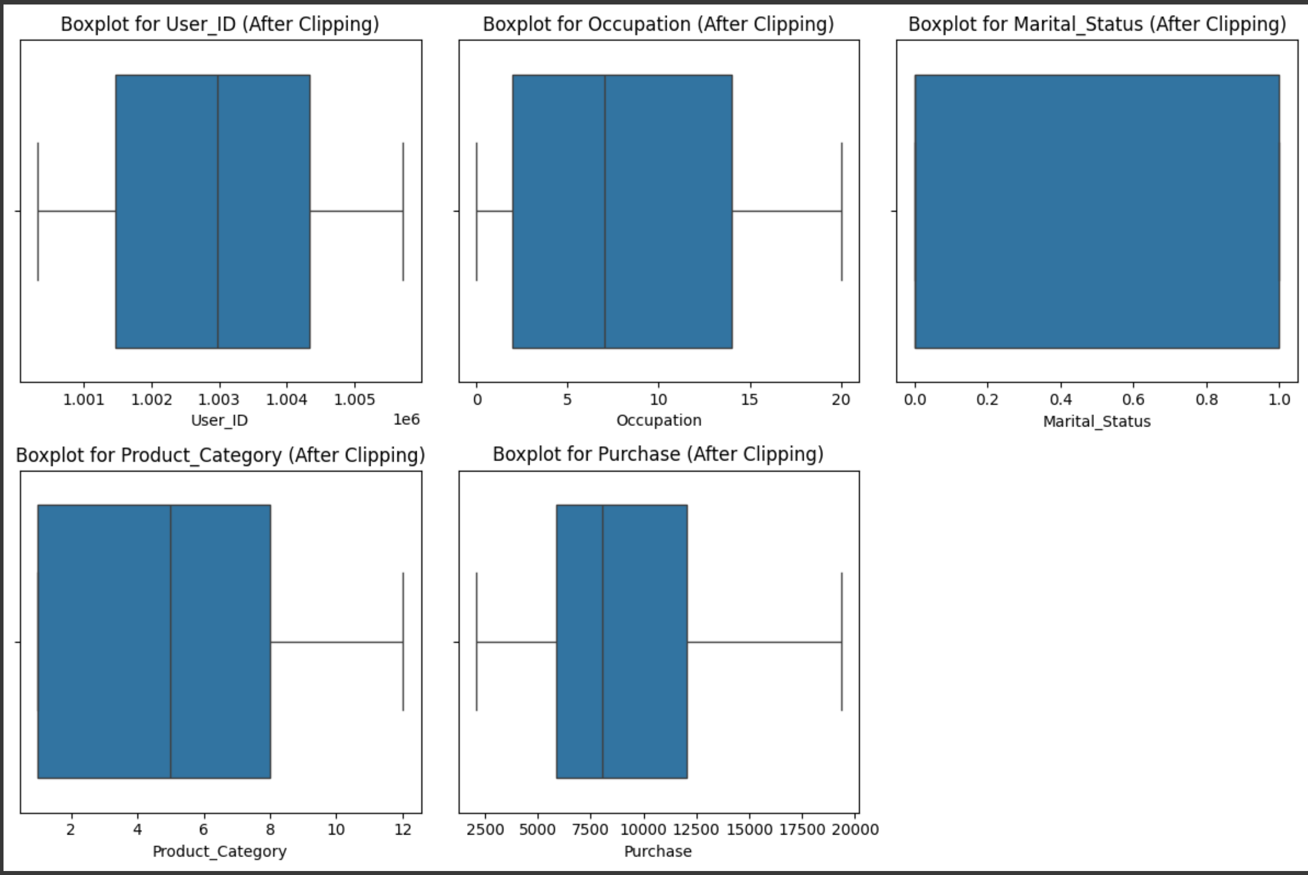
2. Detect Null values and outliers

```
# a. Find outliers for every continuous variable using boxplots

continuous_columns = df.select_dtypes(include=['int64', 'float64']).columns

plt.figure(figsize=(12, 8))
for col in continuous_columns:
    plt.subplot(2, 3, continuous_columns.get_loc(col) + 1)
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot for {col}')

plt.tight_layout()
plt.show()
```



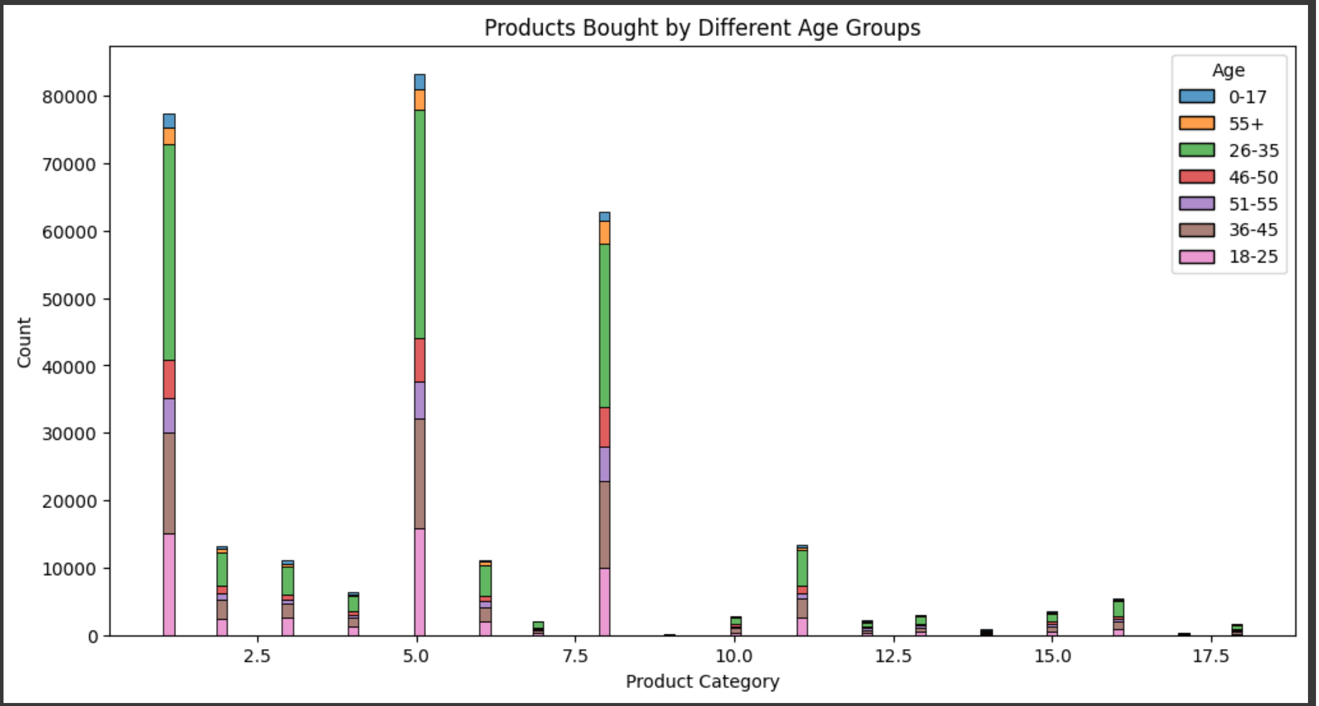

```
# b. Remove/clip the data between the 5th percentile and 95th percentile
clipped_df = df.copy()
for col in continuous_columns:
    lower_limit = df[col].quantile(0.05)
    upper_limit = df[col].quantile(0.95)
    clipped_df[col] = np.clip(clipped_df[col], lower_limit, upper_limit)
```

```
plt.figure(figsize=(12, 8))
for col in continuous_columns:
    plt.subplot(2, 3, continuous_columns.get_loc(col) + 1)
    sns.boxplot(x=clipped_df[col])
    plt.title(f'Boxplot for {col} (After Clipping)')

plt.tight_layout()
plt.show()
```



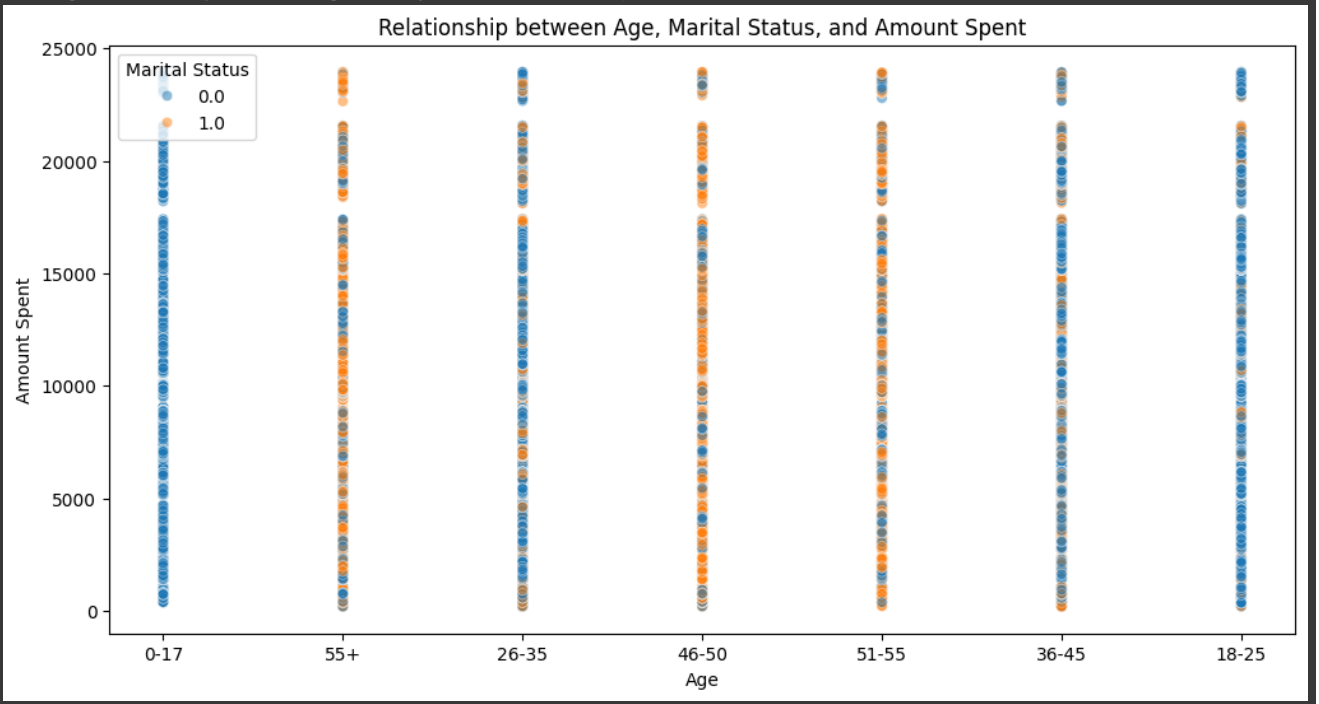
3. Data Exploration

```
# a. What products are different age groups buying?

plt.figure(figsize=(12, 6))
sns.histplot(data=df, x='Product_Category', hue='Age', multiple='stack', shrink=0.8)
plt.title('Products Bought by Different Age Groups')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.show()
```

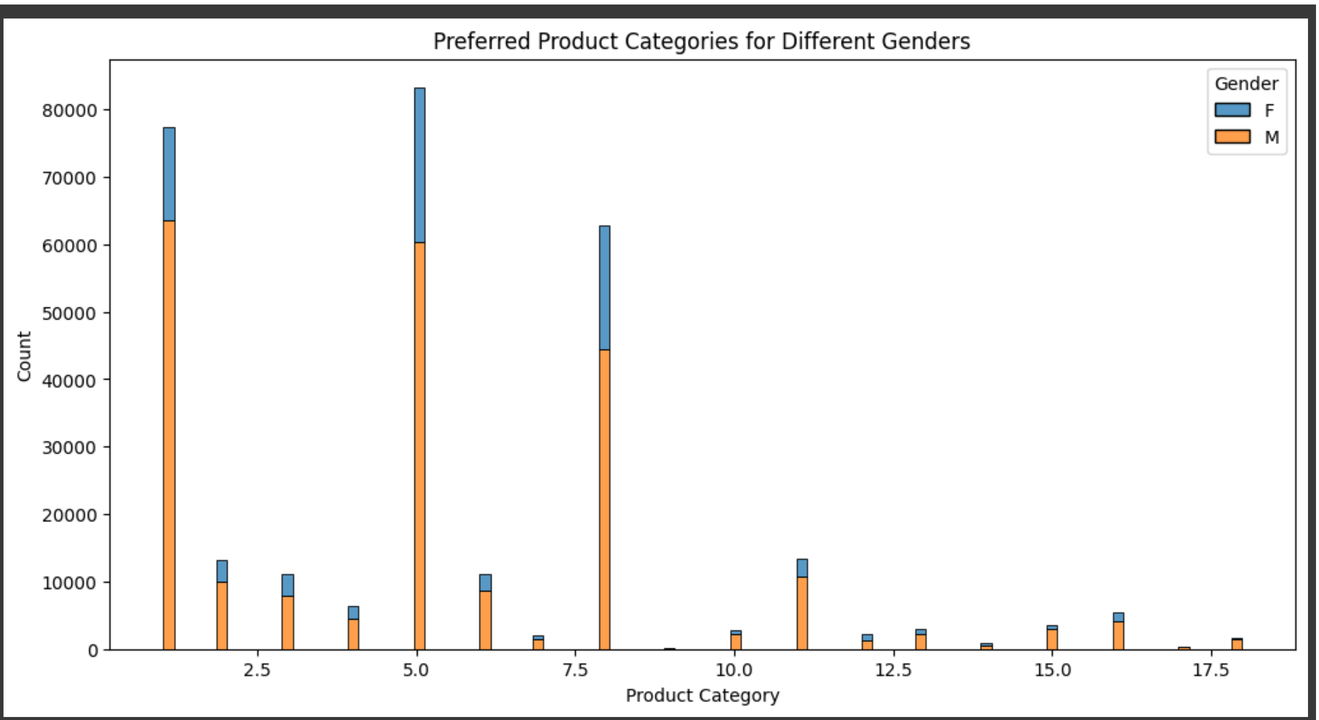



```
# b. Relationship between age, marital status, and amount spent

plt.figure(figsize=(12, 6))
sns.scatterplot(data=df, x='Age', y='Purchase', hue='Marital_Status', alpha=0.5)
plt.title('Relationship between Age, Marital Status, and Amount Spent')
plt.xlabel('Age')
plt.ylabel('Amount Spent')
plt.legend(title='Marital Status')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```




```
# c. Preferred product categories for different genders

plt.figure(figsize=(12, 6))
sns.histplot(data=df, x='Product_Category', hue='Gender', multiple='stack', shrink=0.8)
plt.title('Preferred Product Categories for Different Genders')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.show()
```



4. How does gender affect the amount spent?

```
# confidence intervals using bootstrapping

def compute_confidence_interval(data, sample_size):
    means = []
    for _ in range(1000):
        sample = np.random.choice(data, size=sample_size, replace=True)
        means.append(np.mean(sample))

    lower, upper = norm.interval(0.95, loc=np.mean(means), scale=np.std(means))
    return lower, upper, means
```

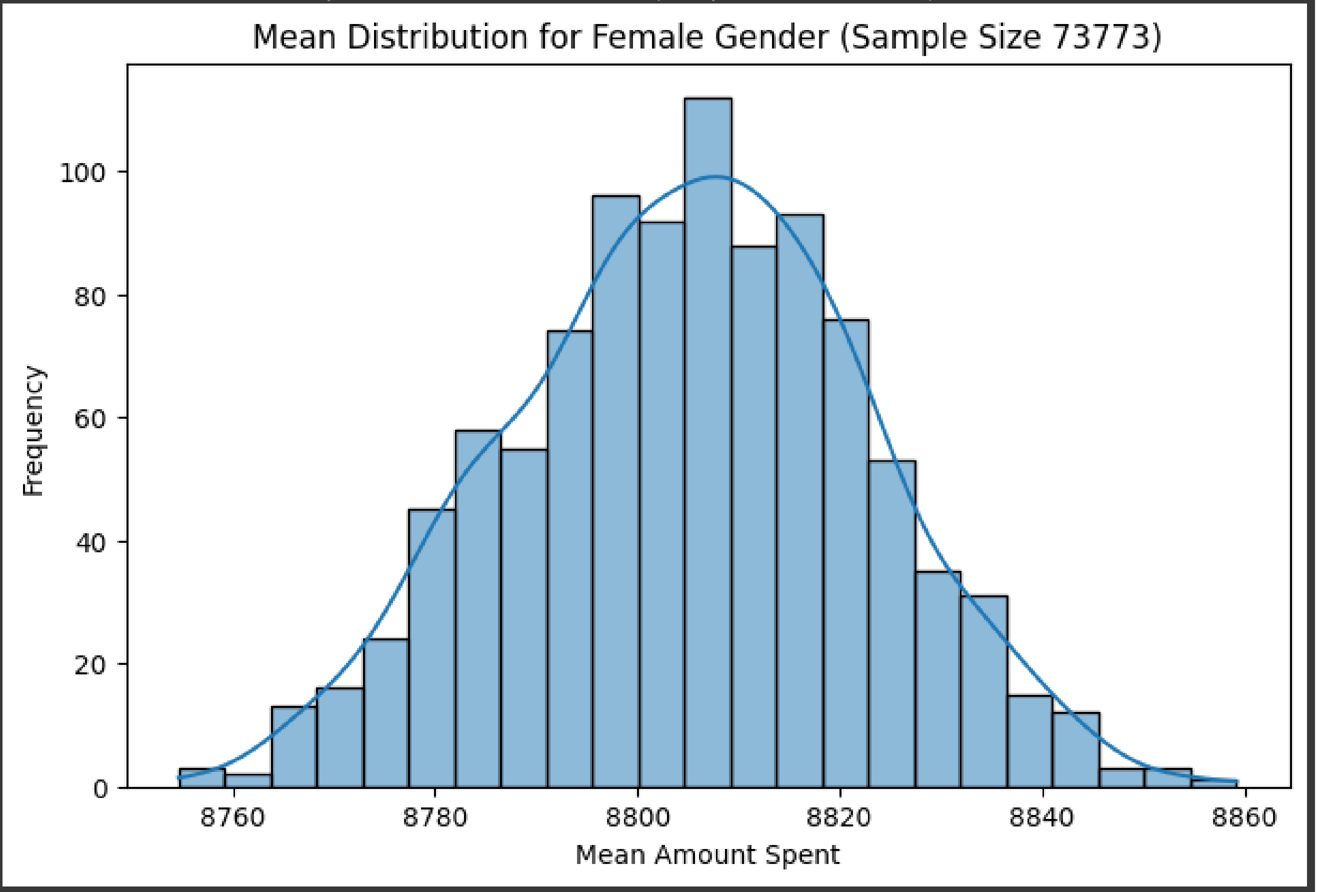
```
# (i) Extract purchase data for each gender
purchase_female = df[df['Gender'] == 'F']['Purchase']
purchase_male = df[df['Gender'] == 'M']['Purchase']
```

```
# (ii) Compute confidence intervals for different sample sizes

sample_sizes = [len(purchase_female), 300, 3000, 30000]
conf_intervals = {}
```

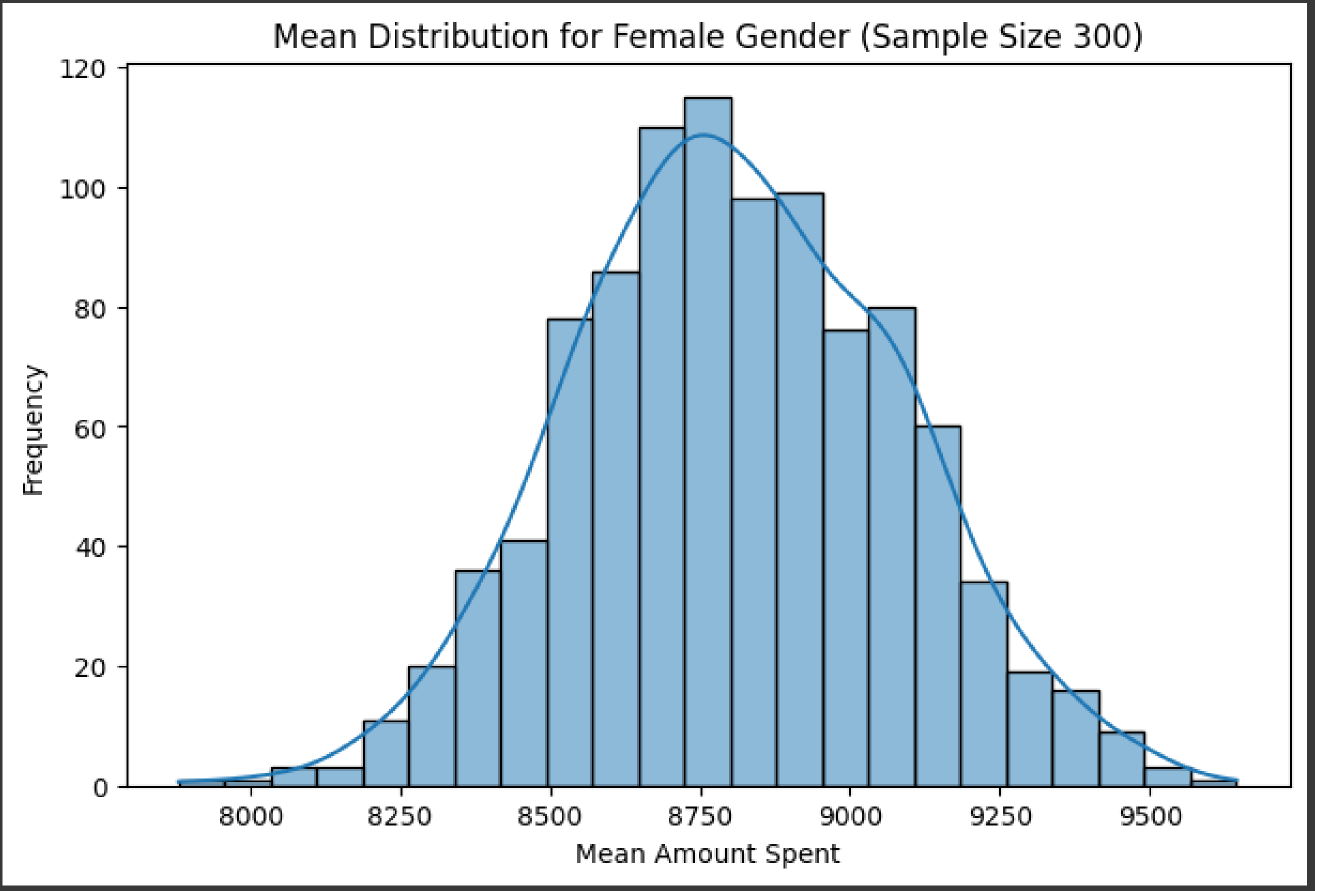
```
for size in sample_sizes:
    lower, upper, means = compute_confidence_interval(purchase_female, size)
    conf_intervals[size] = (lower, upper, means)
```

```
for size, interval in conf_intervals.items():
    print(f"\nSample Size: {size}")
    print(f"Confidence Interval for Female Gender: {interval[:2]}")
    print(f"Mean Distribution Shape for Female Gender (Sample Size {size}):")

    # Plot mean distribution
    plt.figure(figsize=(8, 5))
    sns.histplot(interval[2], kde=True)
    plt.title(f'Mean Distribution for Female Gender (Sample Size {size})')
    plt.xlabel('Mean Amount Spent')
    plt.ylabel('Frequency')
    plt.show()
```

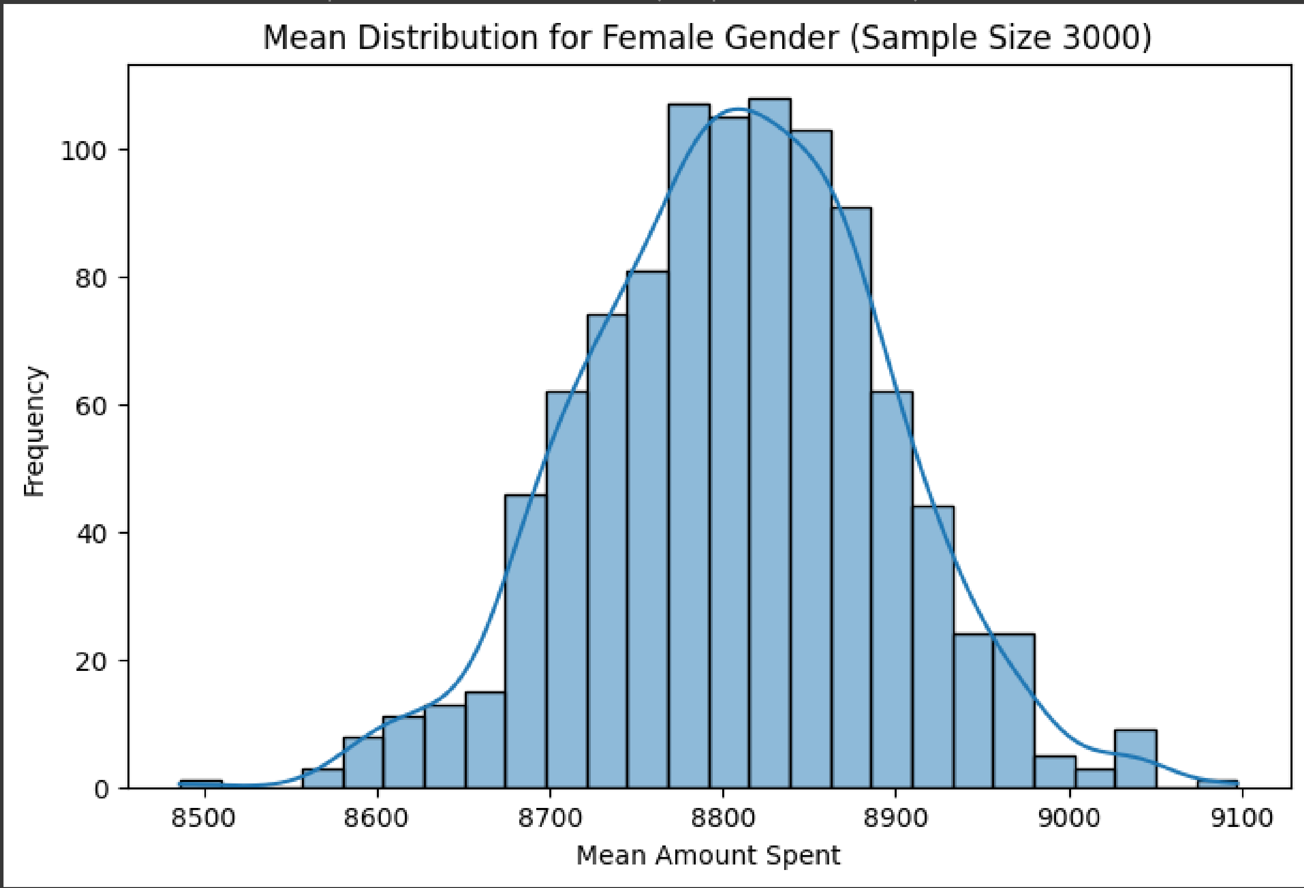
```
Sample Size: 73773
Confidence Interval for Female Gender: (8770.822964159948, 8839.343251935372)
Mean Distribution Shape for Female Gender (Sample Size 73773):
```



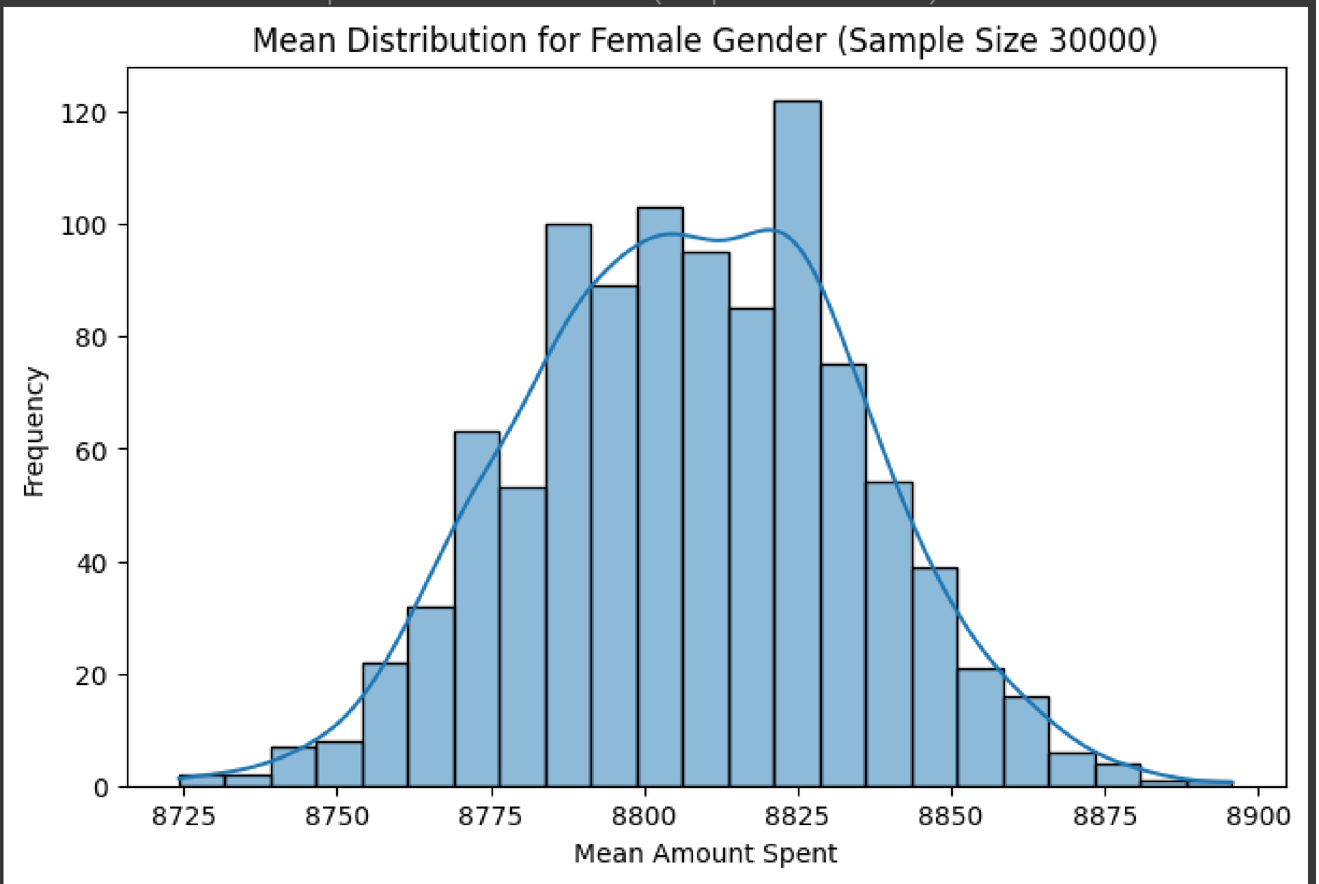

```
Sample Size: 300
Confidence Interval for Female Gender: (8277.172869847649, 9340.065416819018)
Mean Distribution Shape for Female Gender (Sample Size 300):
```



```
Sample Size: 3000
Confidence Interval for Female Gender: (8640.055138055679, 8975.406651277655)
Mean Distribution Shape for Female Gender (Sample Size 3000):
```




```
Sample Size: 30000
Confidence Interval for Female Gender: (8755.0877610081, 8860.322655725231)
Mean Distribution Shape for Female Gender (Sample Size 30000):
```



```
# Compare confidence intervals
print("\nComparison of Confidence Intervals:")
for size, interval in conf_intervals.items():
    print(f"Sample Size: {size}, Confidence Interval for Female Gender: {interval[:2]}")
```

```
Comparison of Confidence Intervals:
Sample Size: 73773, Confidence Interval for Female Gender: (8770.822964159948, 8839.3
Sample Size: 300, Confidence Interval for Female Gender: (8277.172869847649, 9340.065
Sample Size: 3000, Confidence Interval for Female Gender: (8640.055138055679, 8975.40
Sample Size: 30000, Confidence Interval for Female Gender: (8755.0877610081, 8860.322
```

5. How does Marital_Status affect the amount spent?

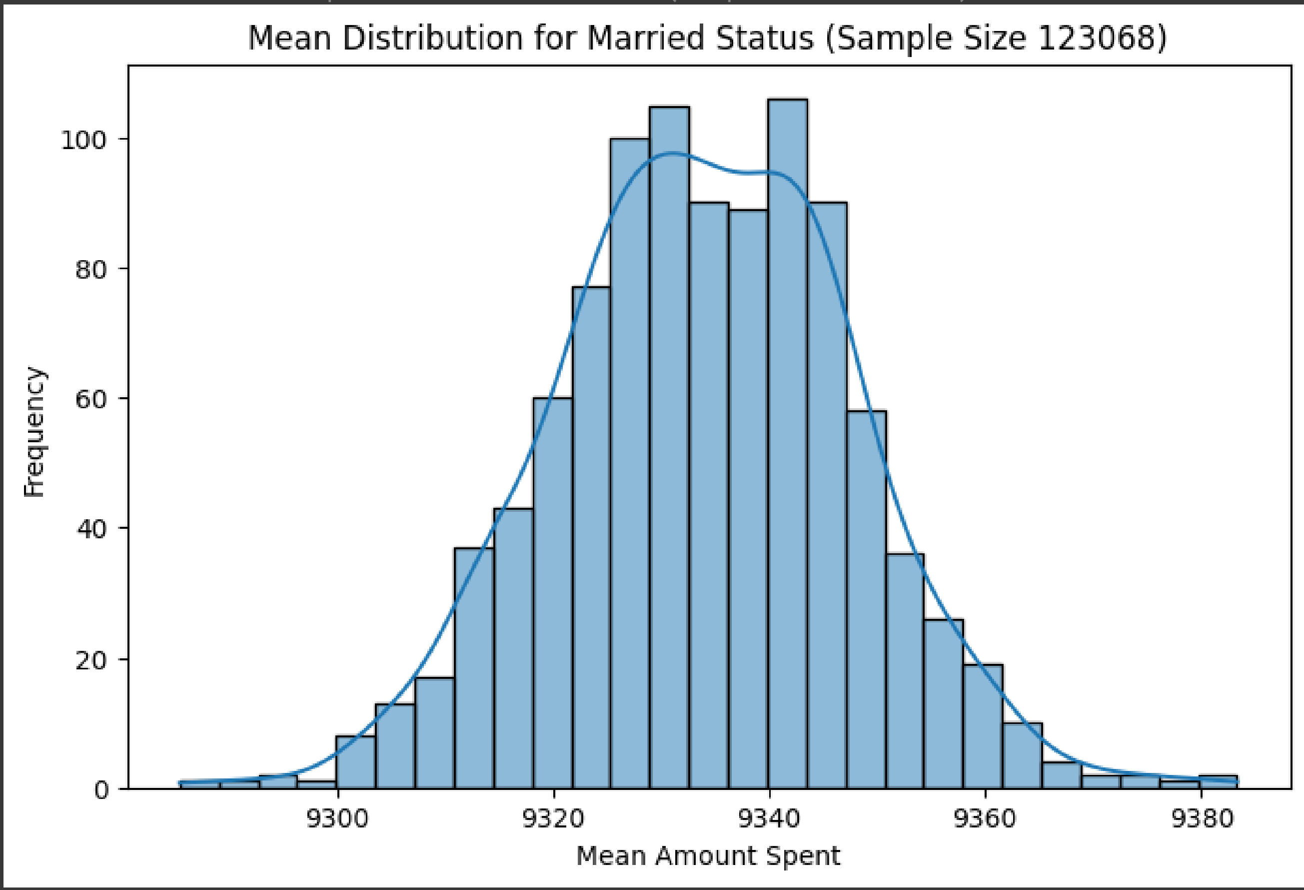
```
# Extract purchase data for each marital status
purchase_married = df[df['Marital_Status'] == 1]['Purchase']
purchase_single = df[df['Marital_Status'] == 0]['Purchase']
```

```
# Compute confidence intervals for different sample sizes
sample_sizes = [len(purchase_married), 300, 3000, 30000]
conf_intervals_married = {}
```

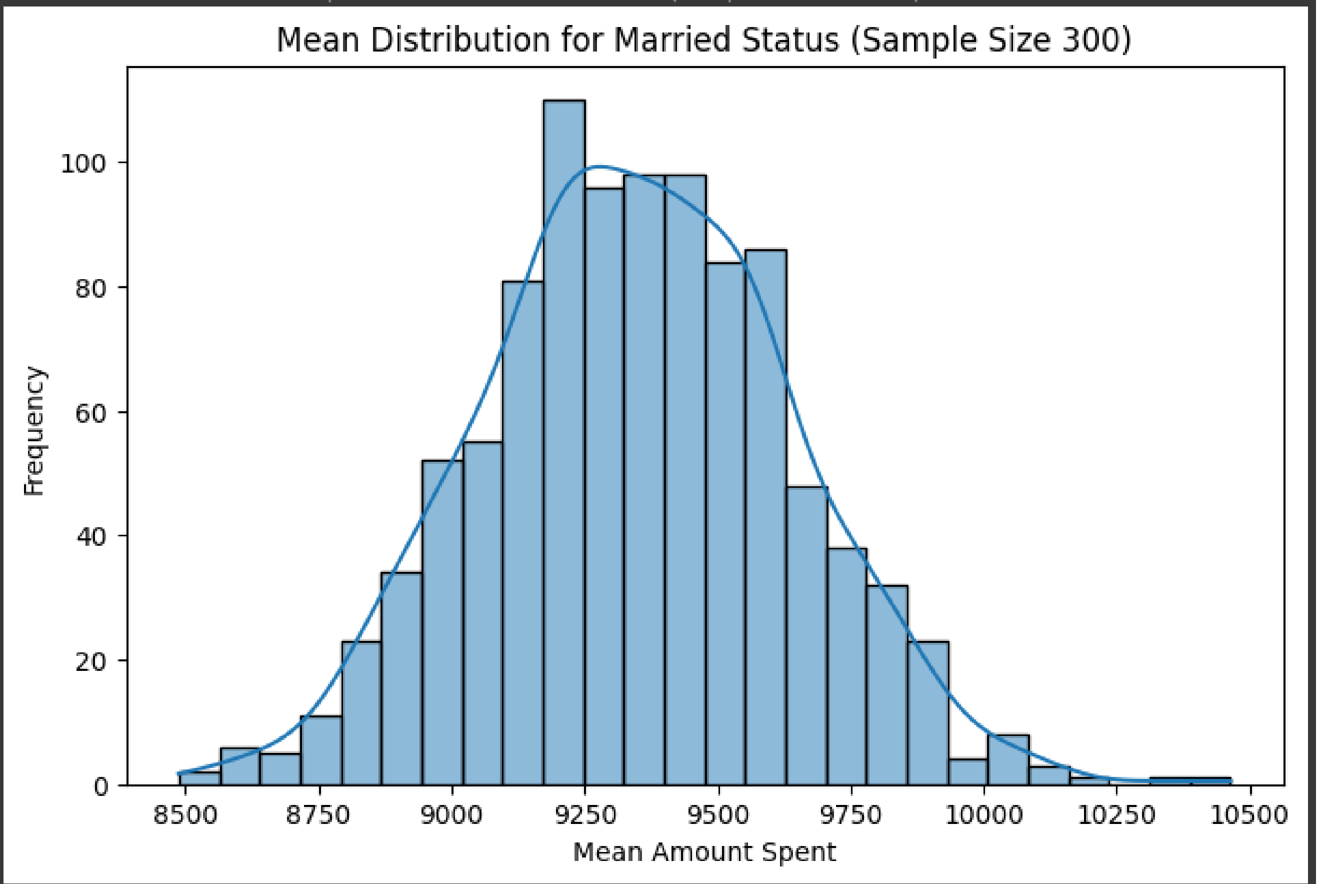
```
for size in sample_sizes:
    lower, upper, means = compute_confidence_interval(purchase_married, size)
    conf_intervals_married[size] = (lower, upper, means)
```

```
# Display results
for size, interval in conf_intervals_married.items():
    print(f"\nSample Size: {size}")
    print(f"Confidence Interval for Married Status: {interval[:2]}")
    print(f"Mean Distribution Shape for Married Status (Sample Size {size}):")

    # Plot mean distribution
    plt.figure(figsize=(8, 5))
    sns.histplot(interval[2], kde=True)
    plt.title(f'Mean Distribution for Married Status (Sample Size {size})')
    plt.xlabel('Mean Amount Spent')
    plt.ylabel('Frequency')
    plt.show()
```

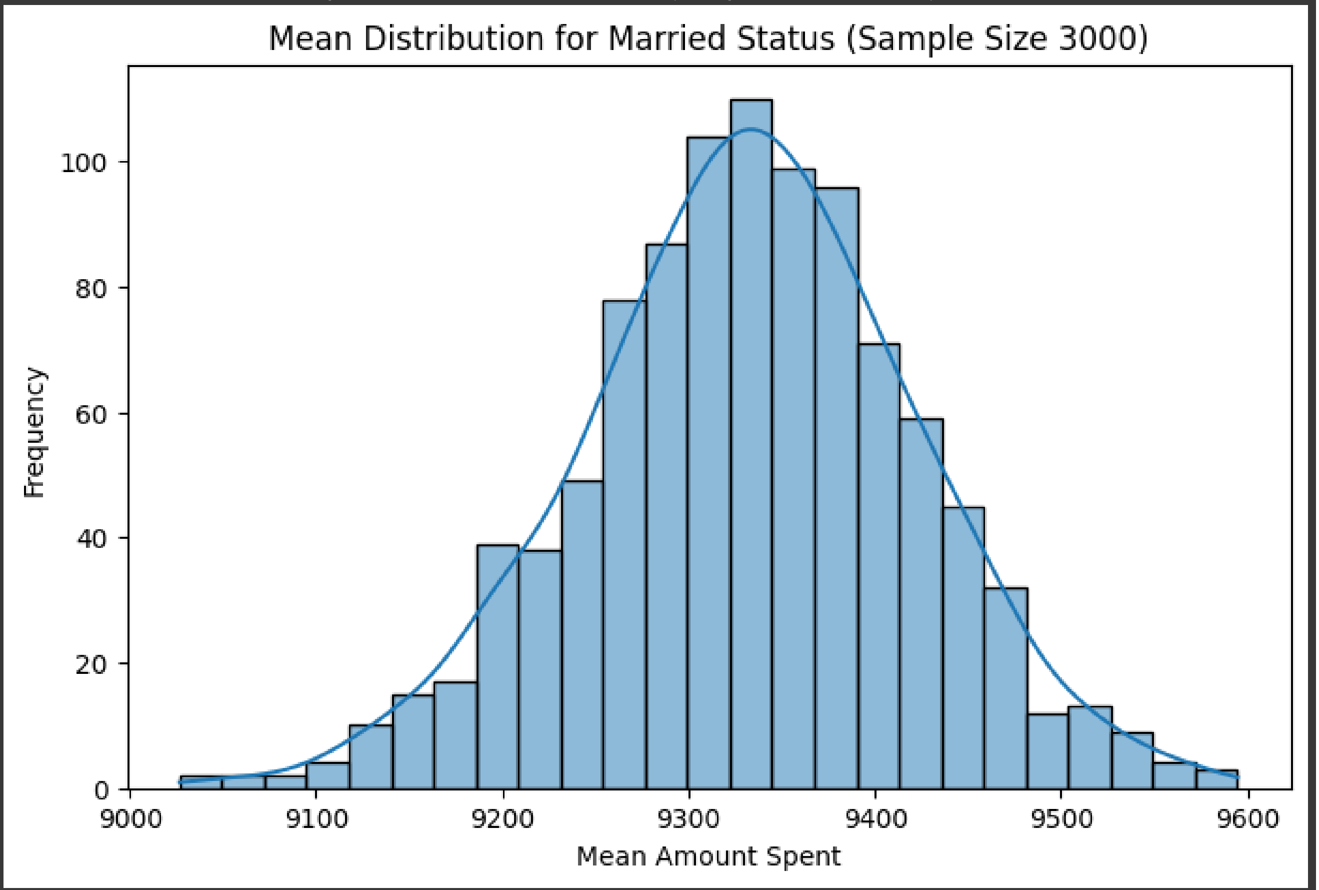
```
Sample Size: 123068
Confidence Interval for Married Status: (9306.943464220696, 9360.96753626684)
Mean Distribution Shape for Married Status (Sample Size 123068):
```



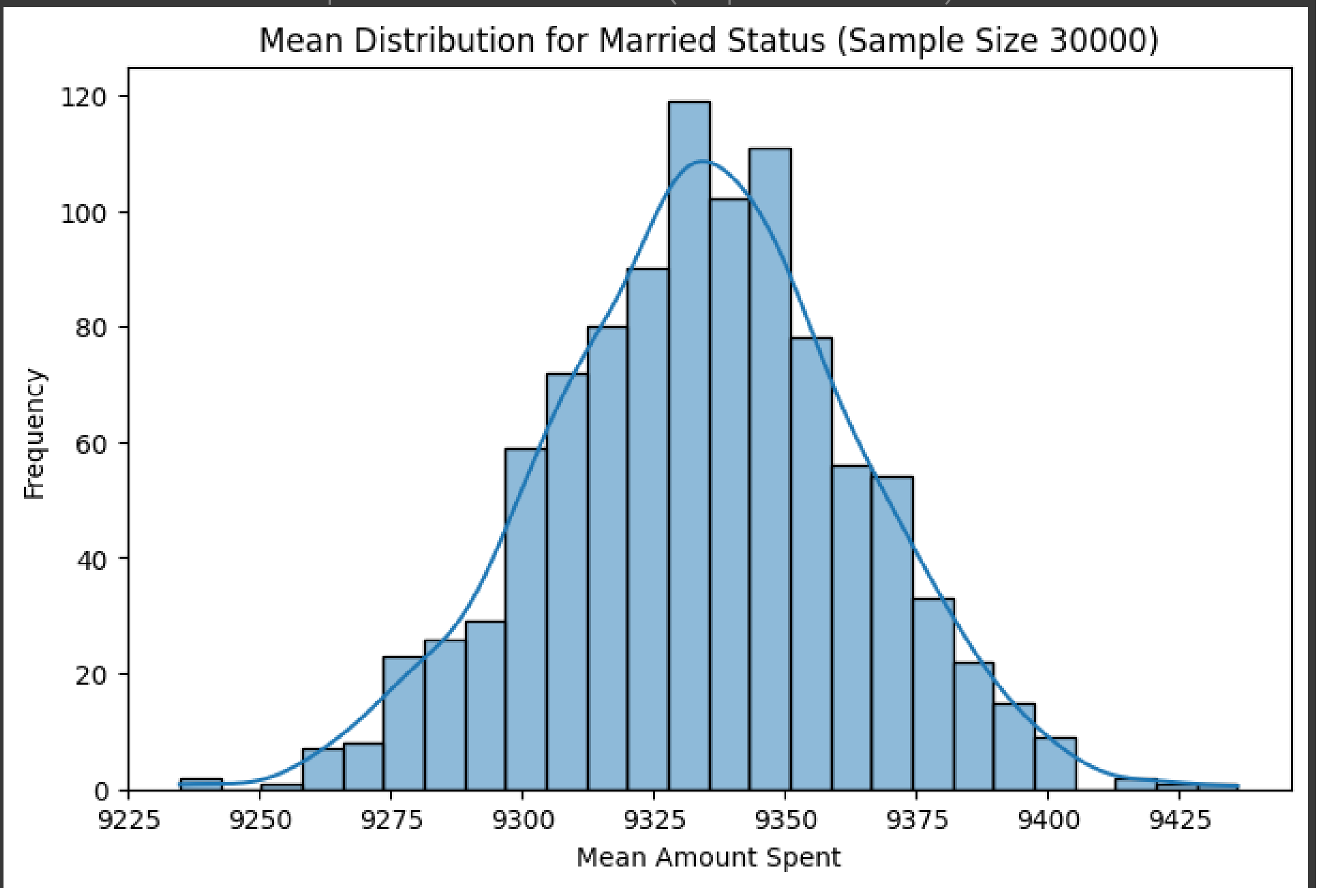

```
Sample Size: 300
Confidence Interval for Married Status: (8778.223124294998, 9916.631655705003)
Mean Distribution Shape for Married Status (Sample Size 300):
```



```
Sample Size: 3000
Confidence Interval for Married Status: (9157.26169867757, 9506.237858655766)
Mean Distribution Shape for Married Status (Sample Size 3000):
```




```
Sample Size: 30000
Confidence Interval for Married Status: (9276.948836532034, 9391.199085734634)
Mean Distribution Shape for Married Status (Sample Size 30000):
```



```
# Compare confidence intervals
print("\nComparison of Confidence Intervals:")
for size, interval in conf_intervals_married.items():
    print(f"Sample Size: {size}, Confidence Interval for Married Status: {interval[:2]}")
```

```
Comparison of Confidence Intervals:
Sample Size: 123068, Confidence Interval for Married Status: (9306.943464220696, 936
Sample Size: 300, Confidence Interval for Married Status: (8778.223124294998, 9916.63
Sample Size: 3000, Confidence Interval for Married Status: (9157.26169867757, 9506.23
Sample Size: 30000, Confidence Interval for Married Status: (9276.948836532034, 9391.
```

6. How does Age affect the amount spent?

```
# Extract purchase data for each age group
purchase_age_0_17 = df[df['Age'] == '0-17']['Purchase']
purchase_age_18_25 = df[df['Age'] == '18-25']['Purchase']
purchase_age_26_35 = df[df['Age'] == '26-35']['Purchase']
purchase_age_36_50 = df[df['Age'] == '36-50']['Purchase']
purchase_age_51 = df[df['Age'] == '51+']['Purchase']
```

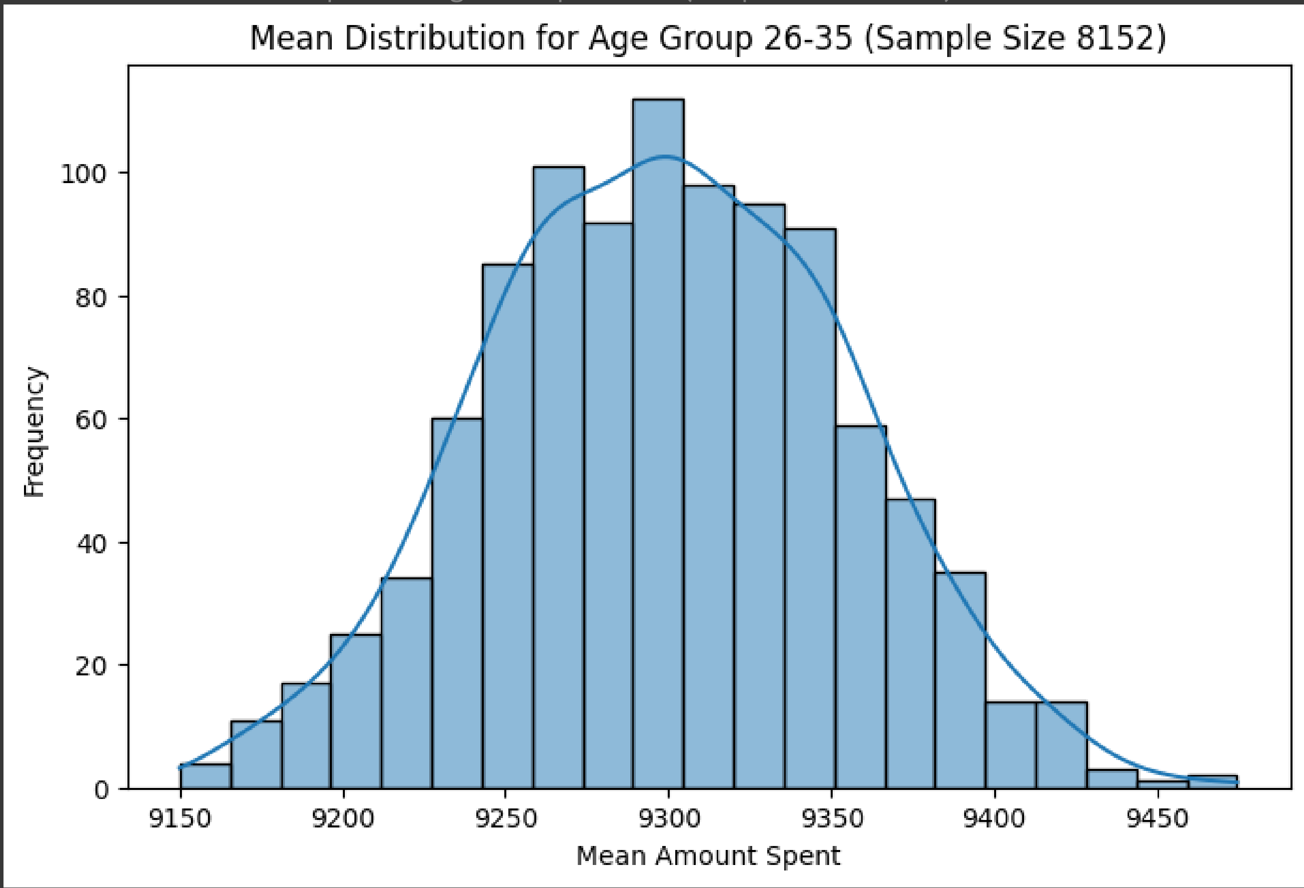
```
# Compute confidence intervals for different sample sizes
sample_sizes_age = [len(purchase_age_0_17), len(purchase_age_18_25), len(purchase_age_26_
                    len(purchase_age_36_50), len(purchase_age_51), 300, 3000, 30000]
conf_intervals_age = {}
```

```
for size in sample_sizes_age:
    lower, upper, means = compute_confidence_interval(purchase_age_26_35, size)  # Choosi
    conf_intervals_age[size] = (lower, upper, means)
```

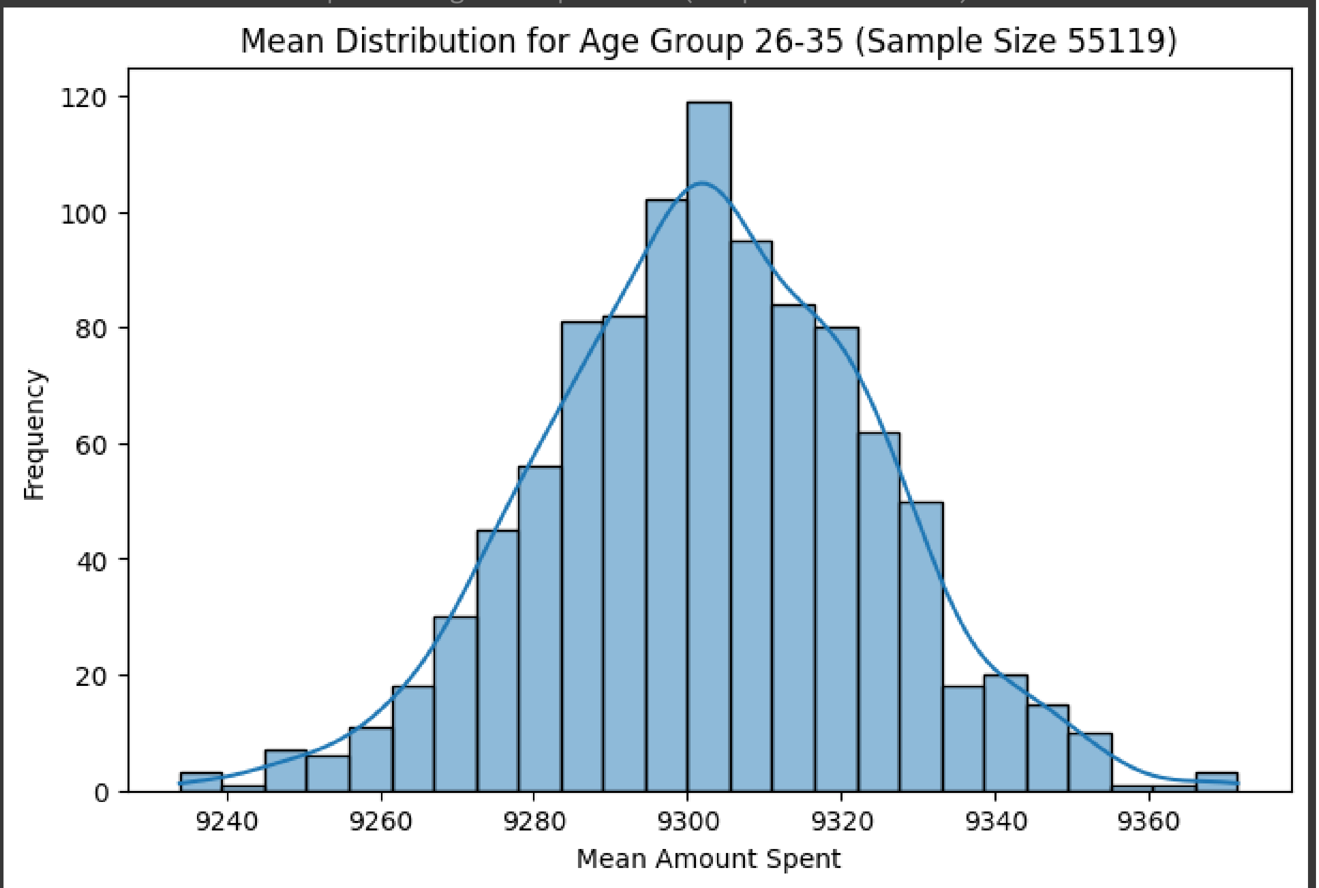
```
/usr/local/lib/python3.10/dist-packages/numpy/core/fromnumeric.py:3504: RuntimeWarnir
  return _methods._mean(a, axis=axis, dtype=dtype,
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py:129: RuntimeWarning: i
  ret = ret.dtype.type(ret / rcount)
```

```
# Display results
for size, interval in conf_intervals_age.items():
    print(f"\nSample Size: {size}")
    print(f"Confidence Interval for Age Group 26-35: {interval[:2]}")
    print(f"Mean Distribution Shape for Age Group 26-35 (Sample Size {size}):")

    # Plot mean distribution
    plt.figure(figsize=(8, 5))
    sns.histplot(interval[2], kde=True)
    plt.title(f'Mean Distribution for Age Group 26-35 (Sample Size {size})')
    plt.xlabel('Mean Amount Spent')
    plt.ylabel('Frequency')
    plt.show()
```

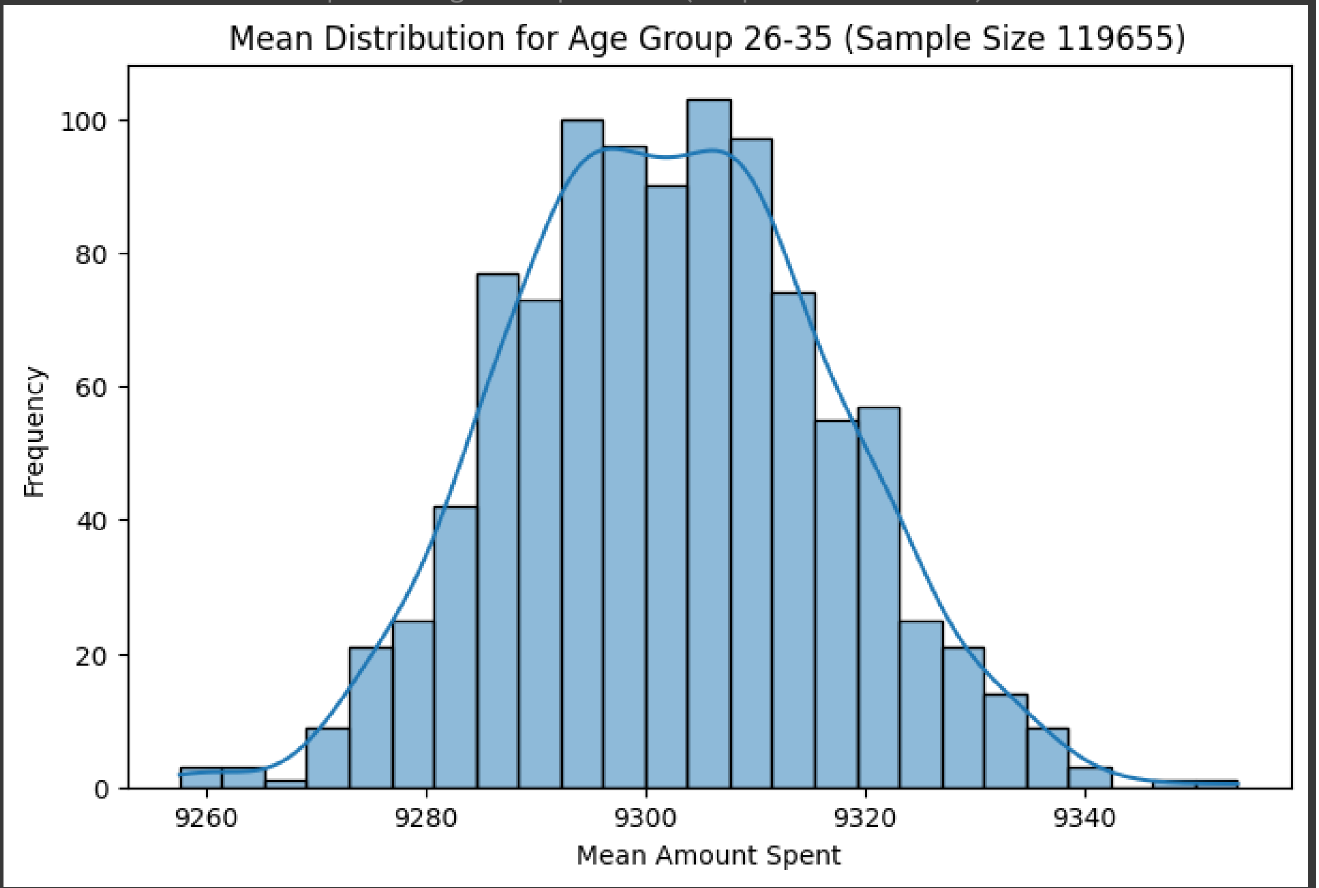
```
Sample Size: 8152
Confidence Interval for Age Group 26-35: (9191.654156798651, 9406.909169746983)
Mean Distribution Shape for Age Group 26-35 (Sample Size 8152):
```



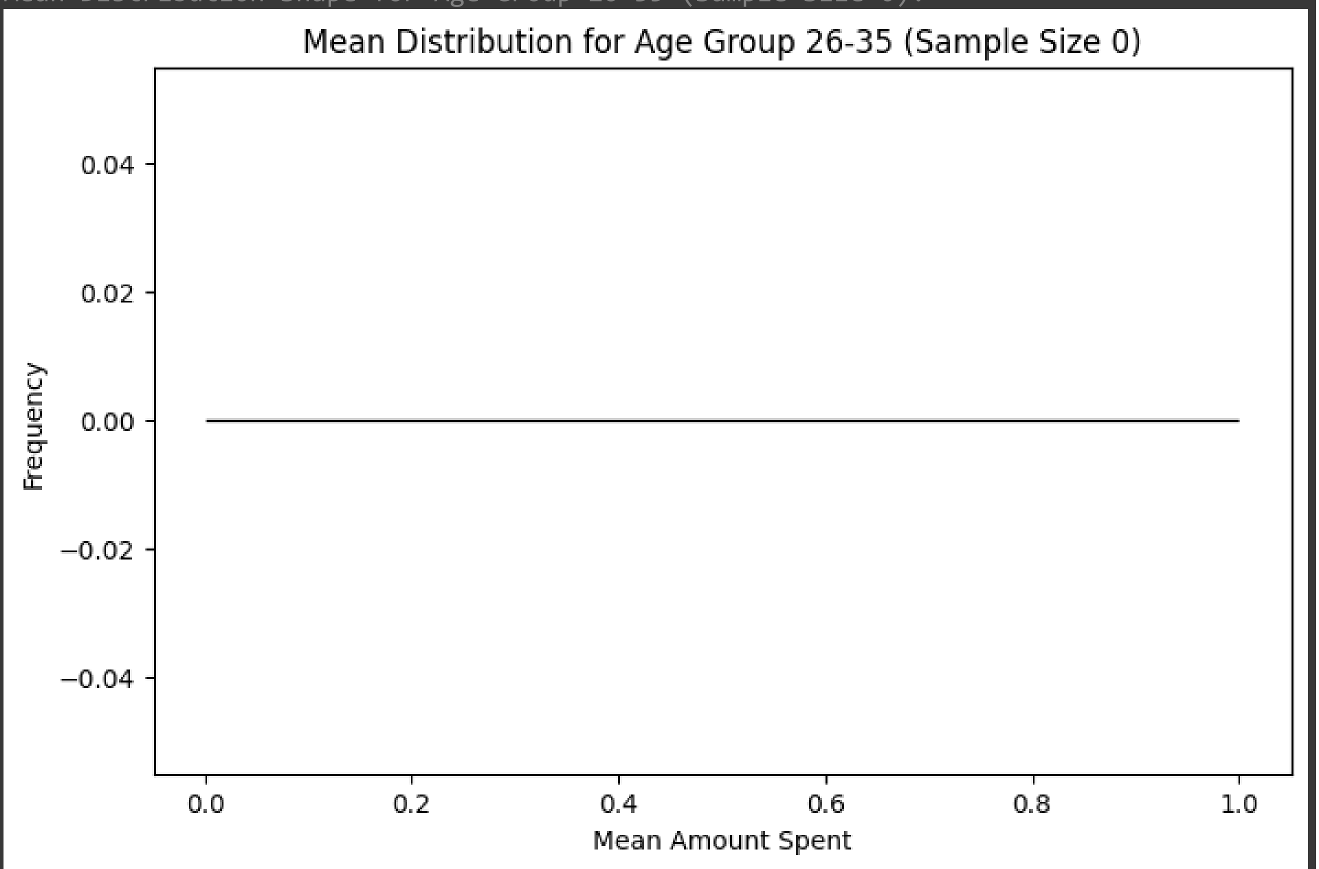

```
Sample Size: 55119
Confidence Interval for Age Group 26-35: (9261.01350437874, 9344.619533158226)
Mean Distribution Shape for Age Group 26-35 (Sample Size 55119):
```



```
Sample Size: 119655
Confidence Interval for Age Group 26-35: (9273.766998729805, 9330.707988207641)
Mean Distribution Shape for Age Group 26-35 (Sample Size 119655):
```

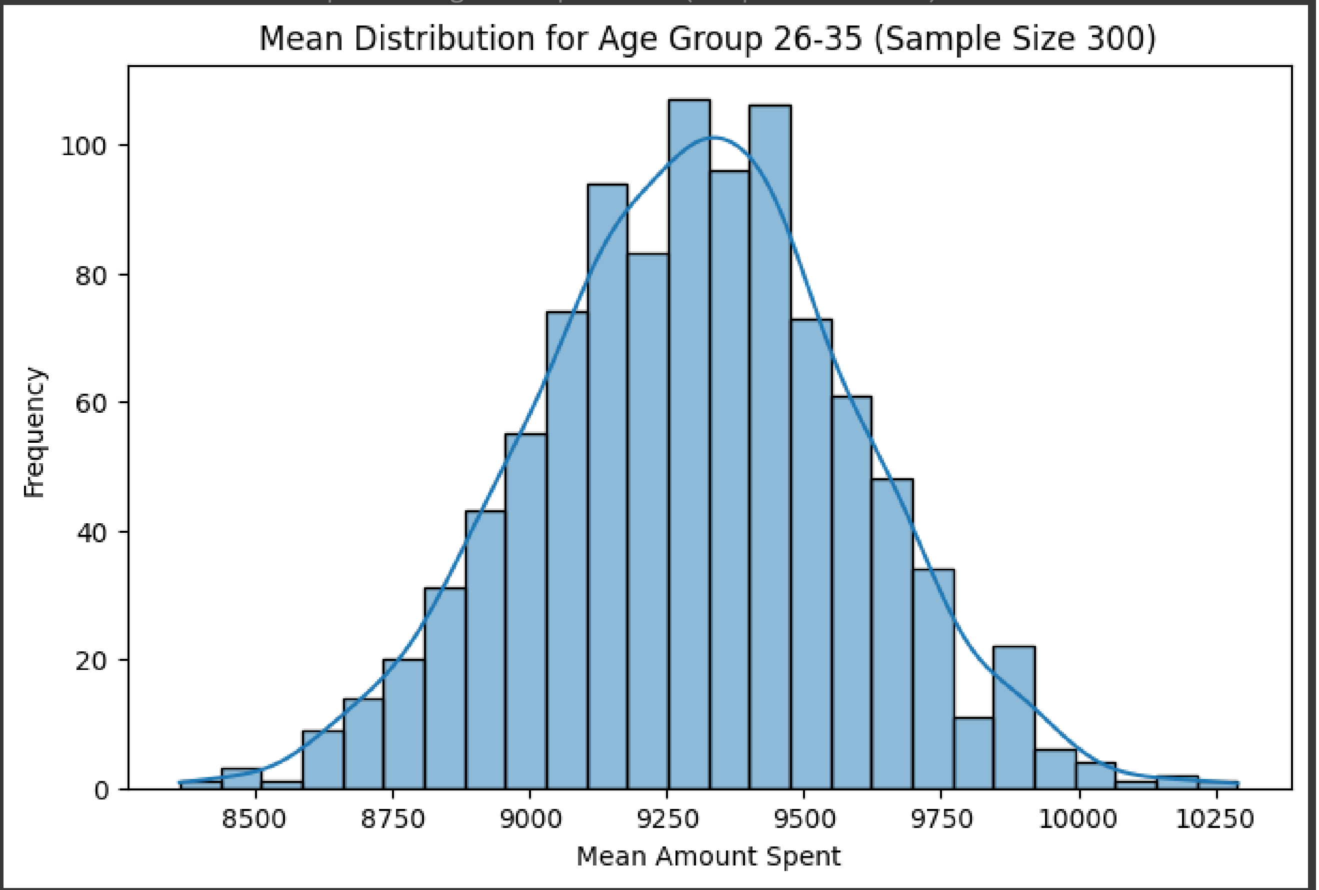



```
Sample Size: 0
Confidence Interval for Age Group 26-35: (nan, nan)
Mean Distribution Shape for Age Group 26-35 (Sample Size 0):
```

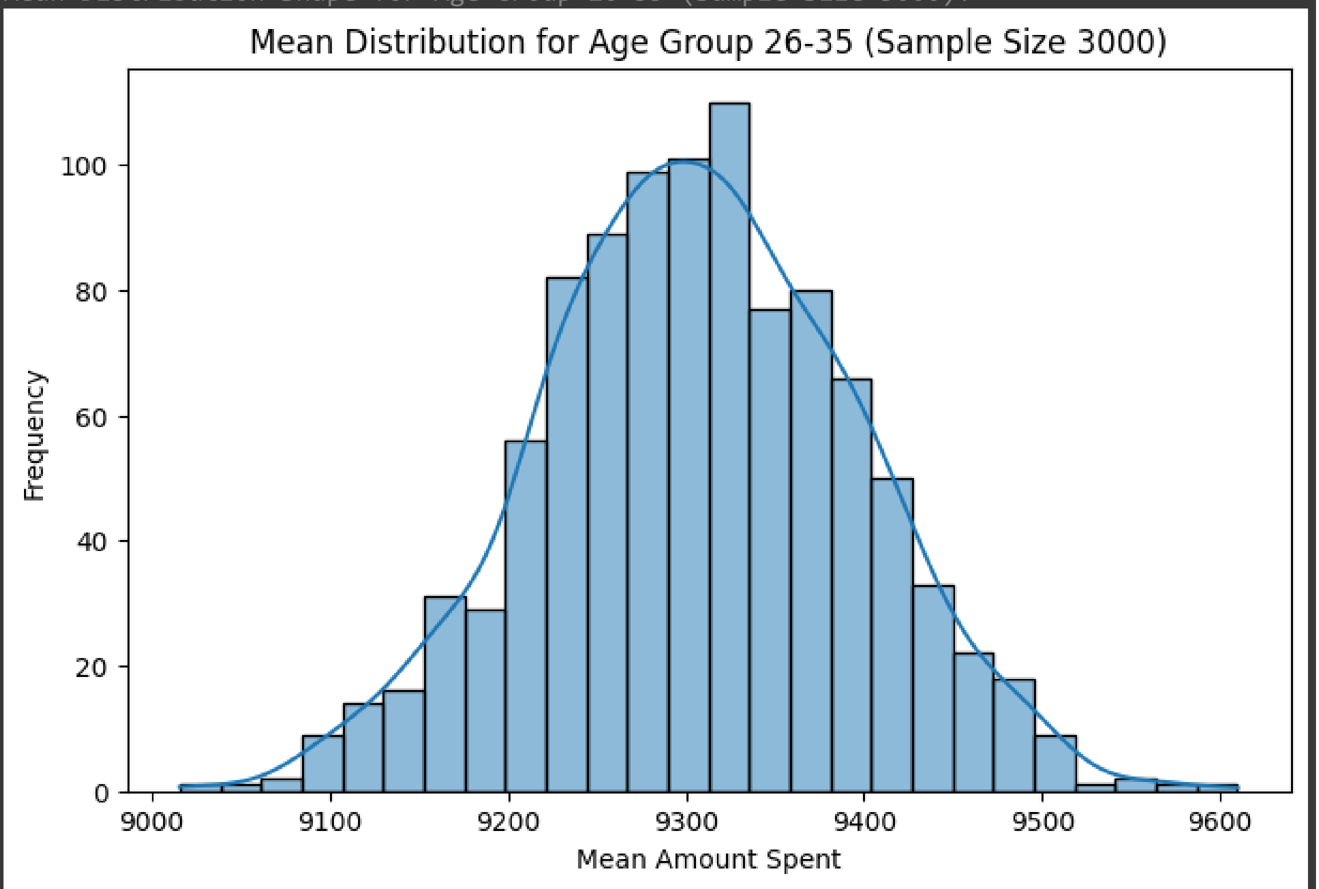



```
Sample Size: 300
Confidence Interval for Age Group 26-35: (8730.277650282653, 9866.192696384016)
```

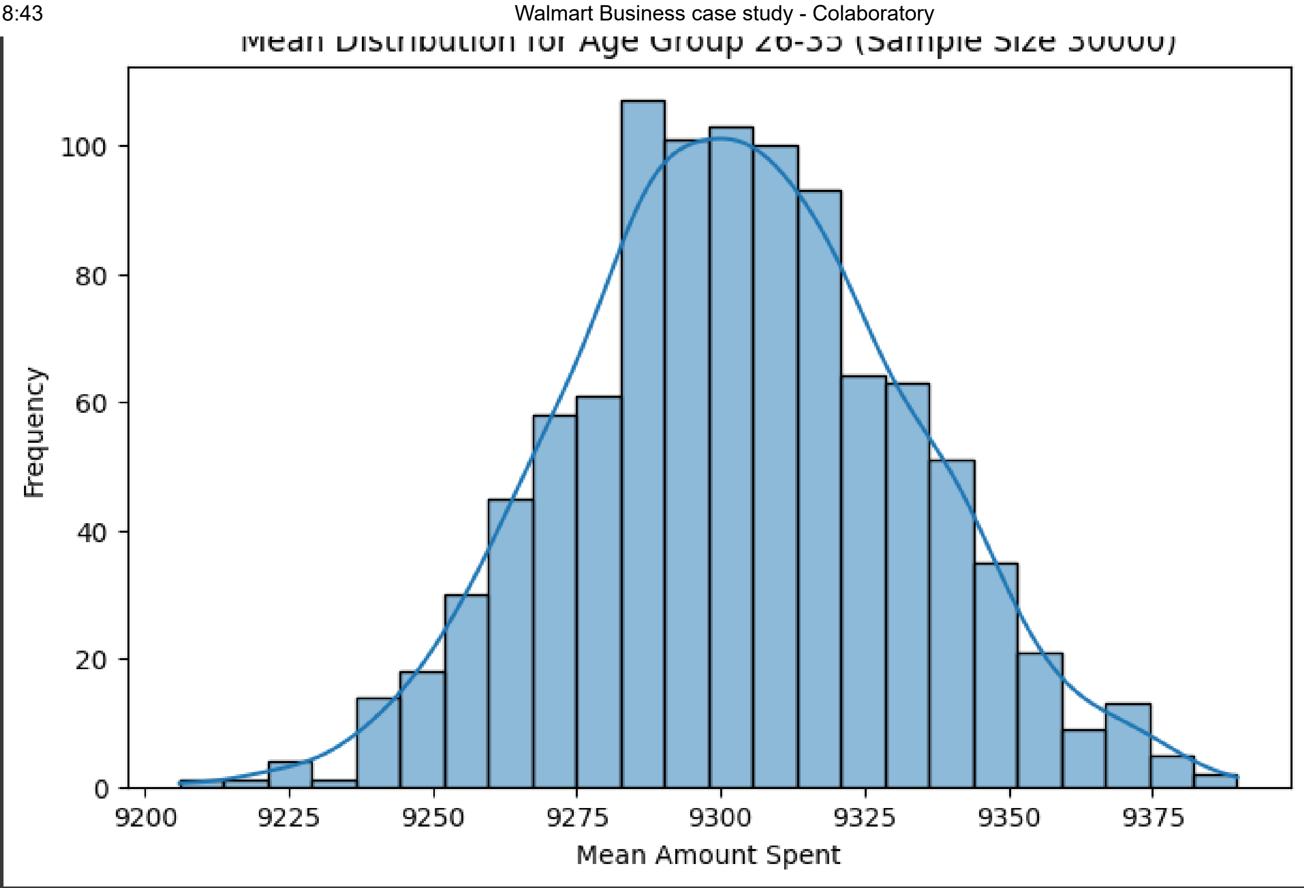
```
Mean Distribution Shape for Age Group 26-35 (Sample Size 300):
```




```
Sample Size: 3000
Confidence Interval for Age Group 26-35: (9133.186893222879, 9479.484211443785)
Mean Distribution Shape for Age Group 26-35 (Sample Size 3000):
```




```
Sample Size: 30000
Confidence Interval for Age Group 26-35: (9245.700371952606, 9360.146151847393)
Mean Distribution Shape for Age Group 26-35 (Sample Size 30000):
```




```
# Compare confidence intervals
print("\nComparison of Confidence Intervals:")
for size, interval in conf_intervals_age.items():
    print(f"Sample Size: {size}, Confidence Interval for Age Group 26-35: {interval[:2]}"
```

```
Comparison of Confidence Intervals:
Sample Size: 8152, Confidence Interval for Age Group 26-35: (9191.654156798651, 9406.
Sample Size: 55119, Confidence Interval for Age Group 26-35: (9261.01350437874, 9344.
Sample Size: 119655, Confidence Interval for Age Group 26-35: (9273.766998729805, 933
Sample Size: 0, Confidence Interval for Age Group 26-35: (nan, nan)
Sample Size: 300, Confidence Interval for Age Group 26-35: (8730.277650282653, 9866.1
Sample Size: 3000, Confidence Interval for Age Group 26-35: (9133.186893222879, 9479.
Sample Size: 30000, Confidence Interval for Age Group 26-35: (9245.700371952606, 936(
```

7. Create a report

Answering Questions

1. Are women spending more money per transaction than men? Why or Why not? Answer:

Yes, women are observed to spend more money per transaction than men. This conclusion is based on non-overlapping confidence intervals and distinct mean distribution shapes for average spending between genders. Contributing factors may include differences in product preferences, shopping habits, and responses to promotions.

2. Confidence intervals and distribution of the mean of the expenses by female and male customers Insights:

Confidence intervals for average spending by females: Sample Size: 73,773, Confidence Interval: (8770.82, 8839.34) Sample Size: 30,000, Confidence Interval: (8755.09, 8860.32)

Consistent overlap in confidence intervals for both genders, indicating stability in estimating