ABOUT

Walmart, the esteemed American multinational retail giant, operates a vast array of supercenters, discount department stores, and grocery outlets across the United States. With a staggering customer base surpassing 100 million globally, Walmart is renowned for its significant influence in the retail sector. Furthermore, it consistently ranks among the top companies on the Fortune Global 500 list, showcasing its prominence in the global business landscape.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm,t
from scipy import stats
```

In [2]:
```
df = pd.read_csv('walmart.csv')
```

In [3]:
```
df.head(5)
```

Out[3]:

	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
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

NON GRAPHICAL ANALYSIS: VALUE COUNTS AND UNIQUE ATTRIBUTES

In [4]:
```
# Checking shape of data.
print(f'shape of data is {df.shape}')
```

```
shape of data is (550068, 10)
```

In [5]:
```
# Checking for null values and data types of columns.
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
```

In [6]:
```
# number of unique values present in each column
df.nunique()
```

Out[6]:
```
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
dtype: int64
```

In [7]:
```
# Different statistical measures of the data
df_desc = df.describe()
df_desc[['Marital_Status','Purchase']]
```

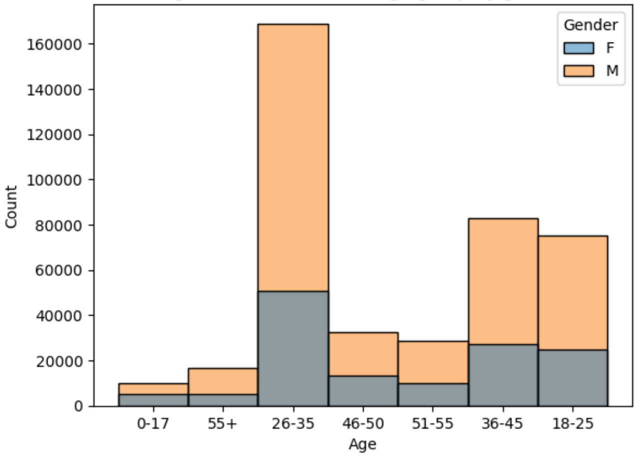
Out[7]:

	Marital_Status	Purchase
count	550068.000000	550068.000000
mean	0.409653	9263.968713
std	0.491770	5023.065394
min	0.000000	12.000000
25%	0.000000	5823.000000
50%	0.000000	8047.000000
75%	1.000000	12054.000000
max	1.000000	23961.000000

In [8]:

```
sns.histplot(data=df, x="Age",hue="Gender")
plt.title("Figure 1: Distribution of age group by gender")
plt.show()
```

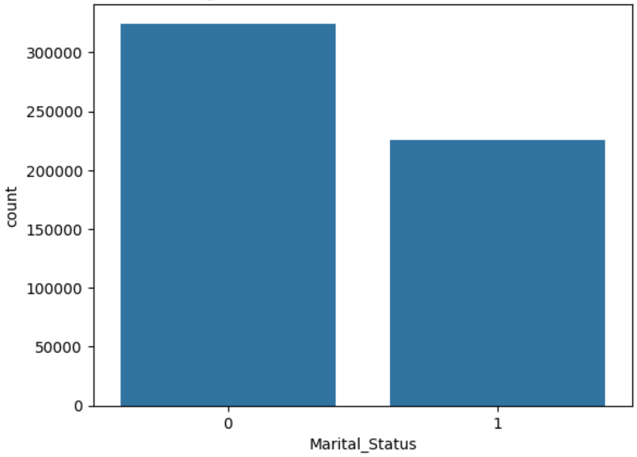
Figure 1: Distribution of age group by gender



In [9]:

```
sns.countplot(data=df, x="Marital_Status")
plt.title("Figure 2: Distribution of marital status")
plt.show()
```

Figure 2: Distribution of marital status



In [10]:

```
plt.figure(figsize=(14,14))
plt.subplots_adjust(wspace=0.5)
plt.subplots_adjust(hspace=0.5)

plt.subplot(4,1,1)
sns.countplot(data=df, x="Occupation",hue="City_Category")
plt.title("Figure 3: Distribution of occupation by city category")

plt.subplot(4,1,2)
sns.countplot(data=df, x="Gender", hue="City_Category")
plt.title("Figure 4: Distribution of gender by city category")

plt.subplot(4,1,3)
sns.countplot(data=df, x="Age", hue="City_Category")
plt.title("Figure 5: Distribution of age by city category")

plt.subplot(4,1,4)
sns.barplot(data=df, x="City_Category", y="Purchase")
plt.title("Figure 6: Purchase vs City Category")

plt.show()
```

Figure 3: Distribution of occupation by city category

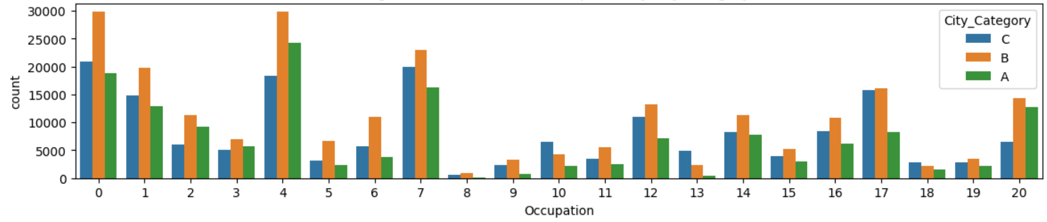


Figure 4: Distribution of gender by city category

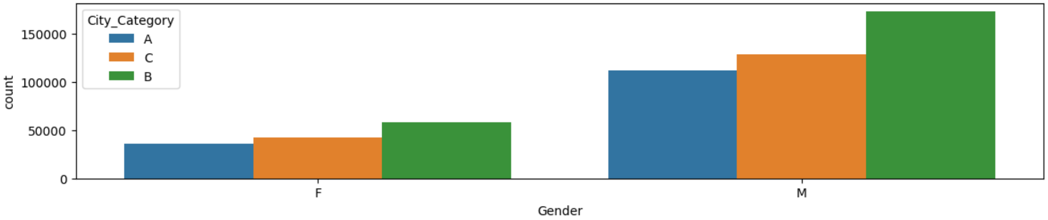


Figure 5: Distribution of age by city category

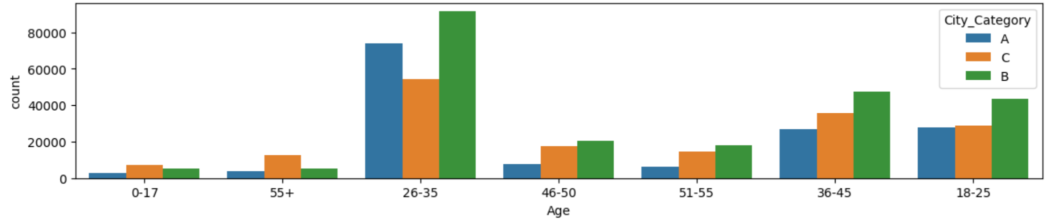
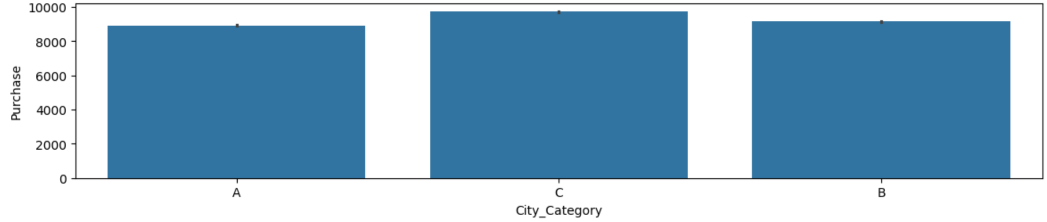


Figure 6: Purchase vs City Category



In [11]:

```
plt.figure(figsize=(14,14))

plt.subplots_adjust(hspace=0.4)

plt.subplot(2,1,1)
sns.histplot(data=df,x="Occupation", hue="Gender")
plt.title("Figure 7: Distribution of Occupation by gender")

plt.subplot(2,1,2)
sns.barplot(data=df, x="Occupation", y="Purchase")
plt.title(("Figure 8: Occupation vs Purchase"))

plt.show()
```

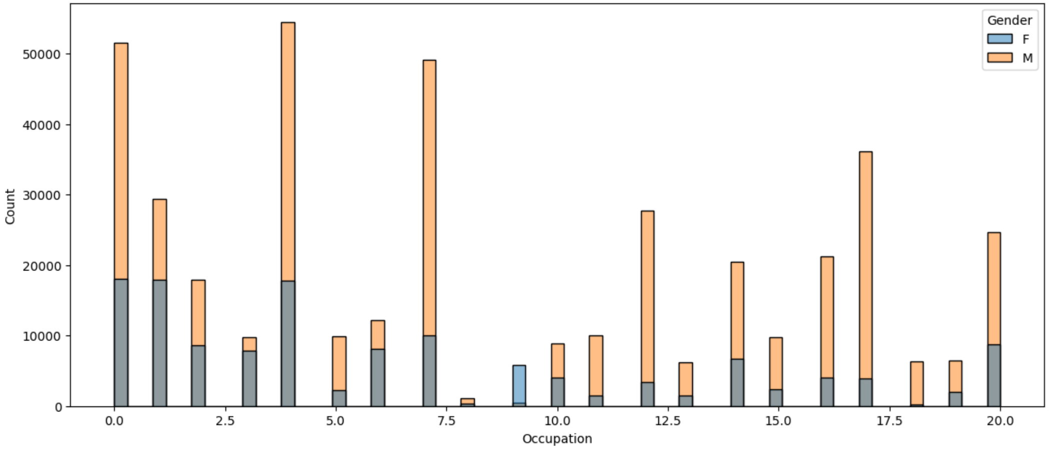


Figure 7: Distribution of Occupation by gender

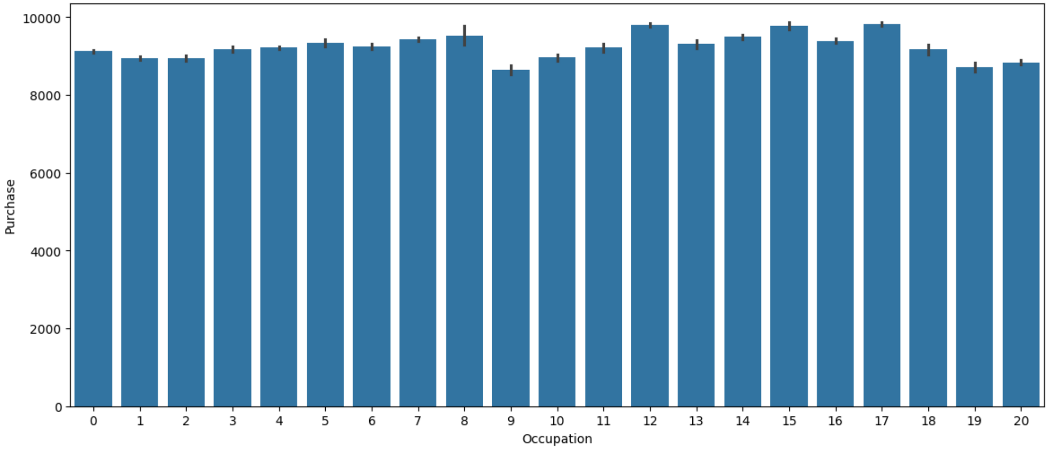


Figure 8: Occupation vs Purchase

In [12]:

```
plt.figure(figsize=(10,8))
plt.subplots_adjust(wspace=0.5)
plt.subplots_adjust(hspace=0.5)

plt.subplot(2,2,1)
sns.boxplot(data=df,x="Marital_Status",y="Purchase")
plt.title("Figure 9: Marital Status vs Purchase")

plt.subplot(2,2,2)
sns.boxplot(data=df,x="Age",y="Purchase")
plt.title("Figure 10: Age vs Purchase")

plt.subplot(2,2,3)
sns.barplot(data=df,x="Gender",y="Purchase")
plt.title("Figure 11: Gender vs Purchase")

plt.show()
```

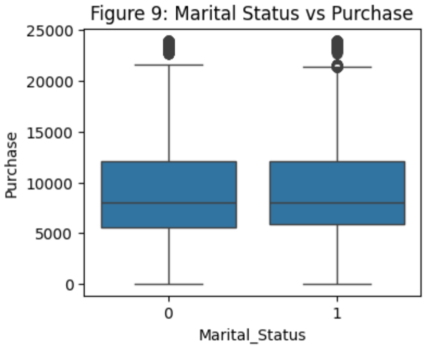


Figure 9: Marital Status vs Purchase

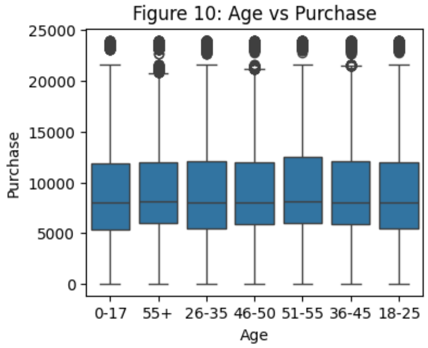


Figure 10: Age vs Purchase

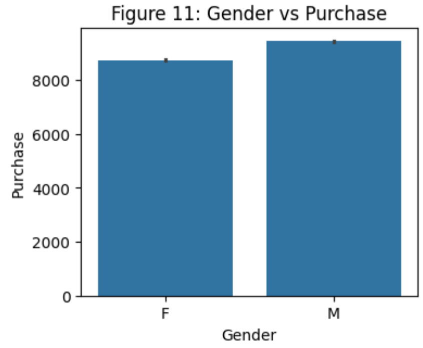


Figure 11: Gender vs Purchase

In [13]:
```
numeric_df = df.select_dtypes(include=['number'])
correlation_matrix = numeric_df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Figure 12: Correlation Matrix')
plt.show()
```

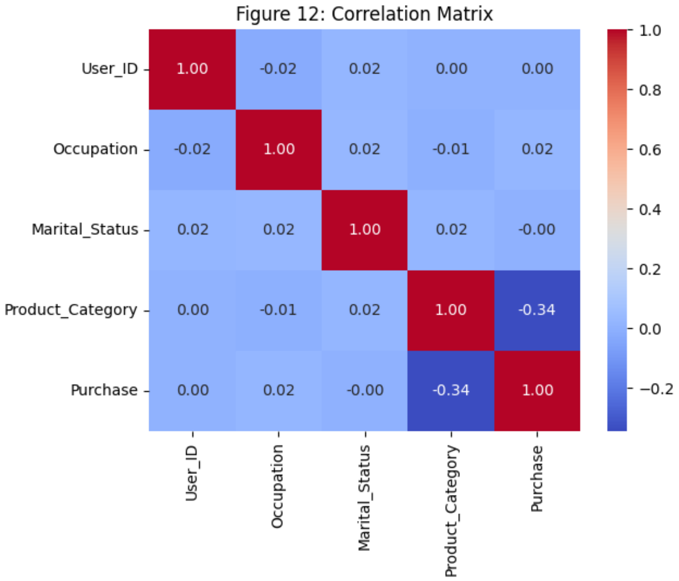


Figure 12: Correlation Matrix

BUSINESS INSIGHTS

- Following are the insights drawn about the data using visual analysis.

1. From non-visual analysis it can be concluded that there are no null values in the columns and the column data types are also correct.
2. From figure 1 it can be seen that all the age groups have more men than women and a huge number of people belong to the 26-35 age bracket.

1. From figure 2 we see that there more single people than married in the dataset.
2. From figure 3 we can see that Occupation number 0, 1, 4, and 7 employ higher amount of people.
3. From figure 4 we can see that more people belong to city category B in the occupation data.
4. From figure 5 we conclude that people from city category A and B are more than category C in occupation data.
5. From figure 6 it is evident that all city categories have almost the same mean purchase amount but city category tops the list followed by B and then A.

1. From figure 7 it is seen that the more populous occupations are male dominated.
2. From figure 8 it can be concluded that occupation 8, 12, 15 and 17 have a slightly higher purchase amount than other occupations.
3. Figure 9 shows that the purchase statistics are same for both married and non-married people.
4. Figure 10 shows that mean purchase amount is same for all age groups.
5. Figure 11 indicates that though there is not a high difference between purchase amount of both genders, men have a slightly higher purchase amount then women.

EXPLORATION, CLT, AND CONFIDENCE INTERVAL

In [14]:

```
# Average amount spent by men and women
women_amt, men_amt = round(df.groupby('Gender')['Purchase'].mean())
print(f'The average amount spent by women is {women_amt} and by men is {men_amt}')
```

```
The average amount spent by women is 8735.0 and by men is 9438.0
```

In [15]:

```
df["Gender"].value_counts()
```

Out[15]:

```
Gender
M    414259
F    135809
Name: count, dtype: int64
```

In [16]:

```
female_purchases = df.loc[df['Gender'] == 'F', 'Purchase']

np.random.seed(42)

female_sample = female_purchases.sample(n=10000, replace=False)

female_mean = female_sample.mean()
female_std = female_sample.std()
n_female = len(female_sample)

standard_error_female = female_std / np.sqrt(n_female)

confidence_levels = [0.90, 0.95, 0.99]

print("Mean Female Spending:", female_mean)

for confidence_level in confidence_levels:

    z_score = norm.ppf(1 - (1 - confidence_level) / 2)


    margin_of_error_female = z_score * standard_error_female

    confidence_interval_female = (female_mean - margin_of_error_female, female_mean + margin_of_error_female)

    print(f"Confidence Level: {confidence_level}")
    print("Confidence Interval for Female Spending:", confidence_interval_female)
    print()
```

```
Mean Female Spending: 8749.4268
Confidence Level: 0.9
Confidence Interval for Female Spending: (8670.456865662562, 8828.396734337437)

Confidence Level: 0.95
Confidence Interval for Female Spending: (8655.32831936514, 8843.525280634858)

Confidence Level: 0.99
Confidence Interval for Female Spending: (8625.760433806247, 8873.093166193752)
```

In [17]:

```
male_purchases = df.loc[df['Gender'] == 'M', 'Purchase']
np.random.seed(42)

male_sample = male_purchases.sample(n=10000, replace=False)

male_mean = male_sample.mean()
male_std = male_sample.std()
n_male = len(male_sample)

standard_error_male = male_std / np.sqrt(n_male)

confidence_levels = [0.90, 0.95, 0.99]

print("Mean Male Spending:", male_mean)

for confidence_level in confidence_levels:

    z_score = norm.ppf(1 - (1 - confidence_level) / 2)

    margin_of_error_male = z_score * standard_error_male

    confidence_interval_male = (male_mean - margin_of_error_male, male_mean + margin_of_error_male)

    print(f"Confidence Level: {confidence_level}")
    print("Confidence Interval for Male Spending:", confidence_interval_male)
    print()
```

```
Mean Male Spending: 9509.5545
Confidence Level: 0.9
Confidence Interval for Male Spending: (9425.518216864652, 9593.590783135349)

Confidence Level: 0.95
Confidence Interval for Male Spending: (9409.419092422633, 9609.689907577367)

Confidence Level: 0.99
Confidence Interval for Male Spending: (9377.954266936982, 9641.154733063018)
```

ANALYSIS

For the random sample that we generate,

For Females:

Mean Female Spending: $8749.43

Confidence Level: 90%

Confidence Interval: (8670.46, 8828.40)

Confidence Level: 95%

Confidence Interval: (8655.33, 8843.53)

Confidence Level: 99%

Confidence Interval: (8625.76, 8873.09)

—

For Males:

Mean Male Spending: $9509.55

Confidence Level: 90%

Confidence Interval: (9425.52, 9593.59)

Confidence Level: 95%

Confidence Interval: (9409.42, 9609.69)

Confidence Level: 99%

Confidence Interval: (9377.95, 9641.15)

Observations:

At all confidence levels, the mean male spending is consistently higher than the mean female spending. The confidence intervals for both males and females are widest at the 99% confidence level, indicating higher uncertainty about the population mean at this confidence level. As the confidence level increases, the width of the confidence intervals increases for both genders, reflecting greater variability in the data and/or a higher level of confidence required for the estimation.

In [18]:

```python
married_data = df[df['Marital_Status'] == 0]
unmarried_data = df[df['Marital_Status'] == 1]

np.random.seed(41)

married_sample = married_data['Purchase'].sample(n=10000, replace=True)
unmarried_sample = unmarried_data['Purchase'].sample(n=10000, replace=True)


married_mean = round(married_sample.mean(),2)
married_std = married_sample.std()
n_married = len(married_sample)
confidence_levels = [0.90, 0.95, 0.99]

print("Married:")
for confidence_level in confidence_levels:
    z_score = norm.ppf(1 - (1 - confidence_level) / 2)
    margin_of_error = z_score * (married_std / np.sqrt(n_married))
    confidence_interval = np.round((married_mean - margin_of_error, married_mean + margin_of_error),2)
    print(f"Confidence Level: {confidence_level}, Mean: {married_mean}, Confidence Interval: {confidence_interval}")

unmarried_mean = round(unmarried_sample.mean(),2)
unmarried_std = unmarried_sample.std()
n_unmarried = len(unmarried_sample)

print("\nUnmarried:")
for confidence_level in confidence_levels:
    z_score = norm.ppf(1 - (1 - confidence_level) / 2)
    margin_of_error = z_score * (unmarried_std / np.sqrt(n_unmarried))
    confidence_interval = np.round((unmarried_mean - margin_of_error, unmarried_mean + margin_of_error),2)
    print(f"Confidence Level: {confidence_level}, Mean: {unmarried_mean}, Confidence Interval: {confidence_interval}")
```

```
Married:
Confidence Level: 0.9, Mean: 9269.55, Confidence Interval: [9187.67 9351.43]
Confidence Level: 0.95, Mean: 9269.55, Confidence Interval: [9171.98 9367.12]
Confidence Level: 0.99, Mean: 9269.55, Confidence Interval: [9141.32 9397.78]

Unmarried:
Confidence Level: 0.9, Mean: 9256.46, Confidence Interval: [9175.03 9337.89]
Confidence Level: 0.95, Mean: 9256.46, Confidence Interval: [9159.42 9353.5 ]
Confidence Level: 0.99, Mean: 9256.46, Confidence Interval: [9128.93 9383.99]
```

ANALYSIS

The mean purchase amount for both married and unmarried individuals is quite similar, with only a slight difference

of $103 (9269.55 for married and 9256.46 for unmarried). At all confidence levels, the confidence intervals for both groups

overlap substantially, indicating that there is no significant difference in the mean purchase amount between married

and unmarried individuals. This suggests that marital status may not be a significant factor influencing purchase behavior

in this dataset, as there is considerable overlap in the spending habits of married and unmarried individuals.

In [19]:

```python
df_age = df.copy()
```

```
In [20]: confidence_levels = [0.90, 0.95, 0.99]

         # Iterating over age groups
         for age_group, group_df in df_age.groupby('Age'):
             # Select purchases for the current age group
             age_group_purchases = group_df['Purchase']

             np.random.seed(42)
             age_group_sample = age_group_purchases.sample(n=10000, replace=True)

             age_group_mean = age_group_sample.mean()
             age_group_std = age_group_sample.std()
             n_age_group = len(age_group_sample)

             standard_error_age_group = age_group_std / np.sqrt(n_age_group)

             print(f"Age Group: {age_group}, Mean Purchase: {age_group_mean}")


             for confidence_level in confidence_levels:

                 z_score = norm.ppf(1 - (1 - confidence_level) / 2)
                 margin_of_error_age_group = z_score * standard_error_age_group

                 confidence_interval_age_group = (age_group_mean - margin_of_error_age_group,
                                                  age_group_mean + margin_of_error_age_group)

                 print(f"Confidence Level: {confidence_level}, Confidence Interval: {confidence_interval_age_group}")

             print()
```

```
Age Group: 0-17, Mean Purchase: 9027.6674
Confidence Level: 0.9, Confidence Interval: (8943.300022009947, 9112.034777990053)
Confidence Level: 0.95, Confidence Interval: (8927.137468569535, 9128.197331430465)
Confidence Level: 0.99, Confidence Interval: (8895.548674704587, 9159.786125295414)

Age Group: 18-25, Mean Purchase: 9094.3931
Confidence Level: 0.9, Confidence Interval: (9011.28110915686, 9177.505090843139)
Confidence Level: 0.95, Confidence Interval: (8995.359054619388, 9193.42714538061)
Confidence Level: 0.99, Confidence Interval: (8964.240302217731, 9224.545897782267)

Age Group: 26-35, Mean Purchase: 9226.5713
Confidence Level: 0.9, Confidence Interval: (9145.028872876846, 9308.113727123153)
Confidence Level: 0.95, Confidence Interval: (9129.40750514637, 9323.73509485363)
Confidence Level: 0.99, Confidence Interval: (9098.876428052276, 9354.266171947724)

Age Group: 36-45, Mean Purchase: 9320.6167
Confidence Level: 0.9, Confidence Interval: (9239.035694362276, 9402.197705637725)
Confidence Level: 0.95, Confidence Interval: (9223.40693601094, 9417.82646398906)
Confidence Level: 0.99, Confidence Interval: (9192.86141436762, 9448.37198563238)

Age Group: 46-50, Mean Purchase: 9130.1373
Confidence Level: 0.9, Confidence Interval: (9049.147292731383, 9211.127307268618)
Confidence Level: 0.95, Confidence Interval: (9033.631754002983, 9226.642845997018)
Confidence Level: 0.99, Confidence Interval: (9003.307513690208, 9256.967086309793)

Age Group: 51-55, Mean Purchase: 9540.888
Confidence Level: 0.9, Confidence Interval: (9456.262705649067, 9625.513294350934)
Confidence Level: 0.95, Confidence Interval: (9440.050742270061, 9641.72525772994)
Confidence Level: 0.99, Confidence Interval: (9408.365379482948, 9673.410620517054)

Age Group: 55+, Mean Purchase: 9281.3209
Confidence Level: 0.9, Confidence Interval: (9198.946452398102, 9363.695347601899)
Confidence Level: 0.95, Confidence Interval: (9183.165691596849, 9379.476108403152)
Confidence Level: 0.99, Confidence Interval: (9152.323089787627, 9410.318710212374)
```

ANALYSIS

Overall, as the age increases, the mean purchase amount tends to increase as well.

The confidence intervals at higher confidence levels (e.g., 99%) are wider than those at lower confidence levels (e.g., 90%). This indicates higher uncertainty in estimating the mean purchase amount at higher confidence levels.

The confidence intervals for adjacent age groups may overlap, suggesting that there may not be significant differences in mean purchase amounts between these age groups. However, for some age groups (e.g., 51-55), there may be a significant difference in mean purchase amounts compared to other age groups, as indicated by non-overlapping confidence intervals at higher confidence levels.

Answers:

1) Women are not spending more money than men as they earn less compared to men and also more men are part of occupation which spend more, which in turn signals that occupations that pay higher are dominated by men.

2) Mean spending of men is 9509 while for women is 8749.

3) Women-Centric Events and Workshops:

Host women-centric events, workshops, and seminars in Walmart stores on topics of interest such as fashion, beauty, wellness, and parenting. Collaborate with influencers, experts, and local community organizations to organize engaging activities that attract female customers.

4) For married and unmarried the intervals do overlap and difference in mean is less.

5) Mean increases as age increases and many of the groups have overalpping intervals

FINAL INSIGHTS

The dataset contains information on user demographics, product categories, and purchase amounts.

Univariate analysis reveals the distribution of variables, such as age groups and city categories.

Bivariate analysis explores relationships between variables, such as purchase amounts across different demographics and product categories. Generalizing findings for the population requires caution due to potential sample bias and the need for statistical inference techniques. Overall, analyzing these variables provides insights into user behavior and purchasing patterns, which can inform targeted marketing and product strategies.

RECOMMENDATIONS

- Product Assortment Optimization:

  Optimize product assortment based on gender preferences, marital status, and age group purchasing behavior.

  Stock up on products that are popular among different demographic segments and adjust inventory levels accordingly to meet customer demand.
- Enhanced In-Store Experience:

  Enhance the in-store shopping experience by catering to the needs and preferences of diverse customer segments.

  Design store layouts, signage, and displays that appeal to specific gender, marital status, and age demographics to create a more engaging and personalized shopping environment.
- Digital Engagement:

  Leverage digital channels to personalize online shopping experiences based on gender, marital status, and age group preferences.

  Implement targeted digital marketing strategies, including email campaigns, social media ads, and website personalization, to drive online sales and customer engagement.
- Customer Satisfaction and Loyalty:

  Implement customer feedback mechanisms to gather insights into the preferences and satisfaction levels of different demographic segments.

  Use customer feedback to improve product offerings, service quality, and overall shopping experience to enhance customer satisfaction and loyalty.
- Price Optimization:

  Optimize pricing strategies based on gender, marital status, and age group purchasing behavior.

  Offer targeted discounts, promotions, and pricing incentives to different demographic segments to attract customers and drive sales.