Importing recommended Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Importing dataset and usual data analysis

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv?1641285094
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv?1641285094
To: /content/walmart_data.csv?1641285094
100% 23.0M/23.0M [00:00<00:00, 164MB/s]
```

```
df = pd.read_csv('/content/walmart_data.csv?1641285094')
```

```
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

Shape of Dataset

```
print(f'The walmart dataset has: \n{df.shape[0]}-rows and {df.shape[1]}-columns')
```

```
The walmart dataset has:
550068-rows and 10-columns
```

Data Type of all the 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
```

Since the given information for the columns of the dataset shows that even the categorical columns like occupation, Marital_status and product_category are of numerical data type. Thus it needs to be converted to object data type.

Conversion of data types of Categorical columns Occupation, Marital_Status and Product_Category

```
cols = ['Occupation', 'Marital_Status', 'Product_Category']
df[cols] = df[cols].astype('object')
```

```
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  object
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  object
 8   Product_Category            550068 non-null  object
 9   Purchase                    550068 non-null  int64
dtypes: int64(2), object(8)
memory usage: 42.0+ MB
```

```
df.describe(include='all')
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category
count	5.500680e+05	550068	550068	550068	550068.0	550068	550068	550068.0	550068.0
unique	NaN	3631	2	7	21.0	3	5	2.0	20.0
top	NaN	P00265242	M	26-35	4.0	B	1	0.0	5.0
freq	NaN	1880	414259	219587	72308.0	231173	193821	324731.0	150933.0
mean	1.003029e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	1.727592e+03	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	1.000001e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	1.001516e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	1.003077e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	1.004478e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	1.006040e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1. There are 7 unique age groups present in the dataset and age group between 26-35 has made the highest purchase.
2. There are total 3631 unique product present in the dataset where product id P00265242 tops this list.
3. The minimum purchase is 1984 and the maximum is 19336 where the mean of purchase is 9256.71. 75% of purchase is made at 12054
4. There are 3 unique city category.
5. There are 5 unique product category.
6. Males are topping the list as they are purchasing more.

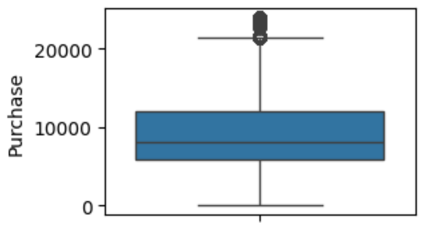
2. Detecting Null Values and Outliers

```
df.describe()
```

	User_ID	Purchase
count	5.500680e+05	550068.000000
mean	1.003029e+06	9263.968713
std	1.727592e+03	5023.065394
min	1.000001e+06	12.000000
25%	1.001516e+06	5823.000000
50%	1.003077e+06	8047.000000
75%	1.004478e+06	12054.000000
max	1.006040e+06	23961.000000

Detecting Outliers

```
plt.figure(figsize=(3,2))
sns.boxplot(df['Purchase'])
plt.show()
```



```
df['Purchase'] =  df['Purchase'].clip(lower = df['Purchase'].quantile(0.05), upper = df['Purchase'].quantile(0.95))
```

```
df.describe()
```

	User_ID	Purchase
count	5.500680e+05	550068.000000
mean	1.003029e+06	9256.710489
std	1.727592e+03	4855.947166
min	1.000001e+06	1984.000000
25%	1.001516e+06	5823.000000
50%	1.003077e+06	8047.000000
75%	1.004478e+06	12054.000000
max	1.006040e+06	19336.000000

Checking for Null Values

```
df.isnull().sum()
```

```
                                0
                  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
```

Observations and Actions

1. The dataset has 550068 rows and 10 columns
2. Purchase amount has outliers as the max purchased amount was 23961 while it's mean wass 9263.968.
3. This outliers have been clipped between 5th percentile and 95th percentile
4. There are no null values present in the dataset

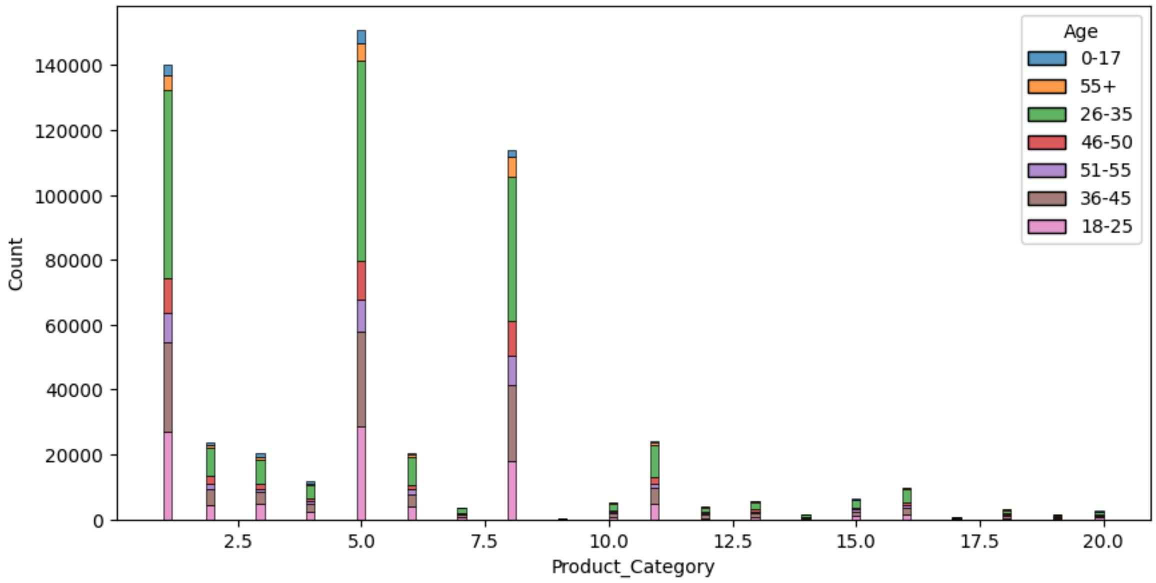
3. Data Exploration

a. What products are different age groups buying

```
age_groups = df['Age'].unique()
print(f'The unique age groups are: {age_groups}')
print('Number of age groups: ', len(age_groups))
```

```
The unique age groups are: ['0-17' '55+' '26-35' '46-50' '51-55' '36-45' '18-25']
Number of age groups:  7
```

```
plt.figure(figsize=(10,5))
sns.histplot(data = df, x = 'Product_Category', hue = 'Age', multiple = 'stack')
plt.show()
```

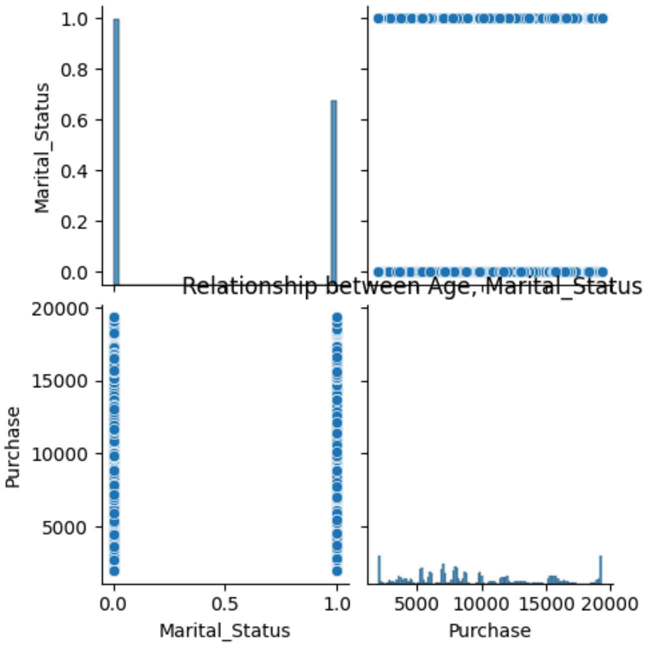


Observations:

1. The most common highest selling product that each age group is buying belongs to product category 5 followed by product category 1 and 8 respectively.
2. Age group 26-35 is the most effective bbuyers for each product category.

b. Is there any relationship between Age, Marital Status and the amount spent

```
sns.pairplot(df[['Age', 'Marital_Status', 'Purchase']])
plt.title('Relationship between Age, Marital_Status and Purchase')
plt.show()
```



Relationship between Age, Marital_Status and Purchase

Observations:

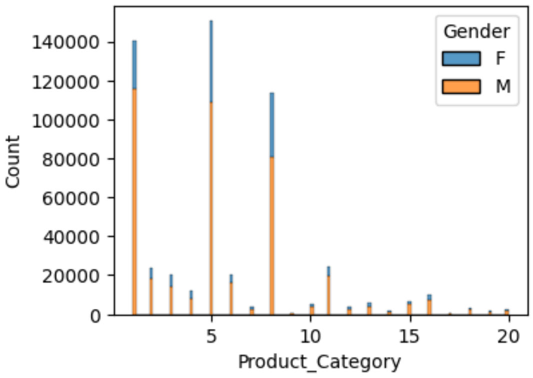
Yes there's a relationship between Age, Marital_Status and Purchase. Single customers having age between 26-35 use to have higher purchase ratio as compared to married customers.

```
grouped_data = df.groupby(['Age', 'Marital_Status'])['Purchase'].mean()
print(grouped_data)
```

```
Age    Marital_Status
0-17   0                 8940.649053
18-25  0                 9215.764183
       1                 8995.105702
26-35  0                 9242.877723
       1                 9245.174074
36-45  0                 9393.742155
       1                 9215.193533
46-50  0                 8951.386840
       1                 9301.401654
51-55  0                 9545.115509
       1                 9503.009291
55+    0                 9532.120512
       1                 9209.546362
Name: Purchase, dtype: float64
```

c. Are there preferred Product Categories for different Genders?

```
plt.figure(figsize=(4,3))
sns.histplot(data = df, x = 'Product_Category', hue = 'Gender', multiple = 'stack')
plt.show()
```



Observations:

Yes there sees to be preferred category according to genders as male customers tops the list as it is being seen that Walmart has more options for males as compare to females.

4. How does gender affect the Amount Spent

```
print("Unique values in Gender:", df['Gender'].unique())
```

```
Unique values in Gender: ['F' 'M']
```

```
print("Missing values in Gender:", df['Gender'].isnull().sum())
```

```
Missing values in Gender: 0
```

```
print("Missing values in Purchase:", df['Purchase'].isnull().sum())
```

```
Missing values in Purchase: 0
```

Dropping rows with missing purchase values

```
purchase_male = df.loc[(df['Gender'] == 'M') & (~df['Purchase'].isnull()), 'Purchase']
purchase_female = df.loc[(df['Gender'] == 'F') & (~df['Purchase'].isnull()), 'Purchase']
```

Checking the sizes of the purchases Gender wise

```
print("Size of purchase male:", len(purchase_male))
print("Size of purchase female:", len(purchase_female))
```

```
Size of purchase male: 414259
Size of purchase female: 135809
```

```
import numpy as np
import scipy.stats as stats

def bootstrap_ci(df, sample_size, num_bootstraps=1000):
    if len(df) == 0:
        raise ValueError("Input data for bootstrap sampling is empty")
    means = []
    for _ in range(num_bootstraps):
        sample = np.random.choice(df, size=sample_size, replace=True)
        means.append(np.mean(sample))

    ci = np.percentile(means, [2.5, 97.5])
    return ci
```

Purchase data Gender wise

```
purchase_male = df[df['Gender'] == 'M']['Purchase']
purchase_female = df[df['Gender'] == 'F']['Purchase']
```

Calculating ci for different sample sizes

```
ci_male_300 = bootstrap_ci(purchase_male, 300)
ci_female_300 = bootstrap_ci(purchase_female, 300)
ci_male_3000 = bootstrap_ci(purchase_male, 3000)
ci_female_3000 = bootstrap_ci(purchase_female, 3000)
ci_male_30000 = bootstrap_ci(purchase_male, 30000)
ci_female_30000 = bootstrap_ci(purchase_female, 30000)
```

```
print(f"CLT for males having sample size 300: Lower Bound - {ci_male_300[0]} and Upper Bound - {ci_male_300[1]}")
print(f"CLT for females having sample size 300: Lower Bound - {ci_female_300[0]} and Upper Bound - {ci_female_300[1]}")
print(f"CLT for males having sample size 300: Lower Bound - {ci_male_3000[0]} and Upper Bound - {ci_male_3000[1]}")
print(f"CLT for females having sample size 300: Lower Bound - {ci_female_3000[0]} and Upper Bound - {ci_female_3000[1]}")
print(f"CLT for males having sample size 300: Lower Bound - {ci_male_30000[0]} and Upper Bound - {ci_male_30000[1]}")
print(f"CLT for females having sample size 300: Lower Bound - {ci_female_30000[0]} and Upper Bound - {ci_female_30000[1]}")
```

```
CLT for males having sample size 300: Lower Bound - 8898.009083333334 and Upper Bound - 10023.37725
CLT for females having sample size 300: Lower Bound - 8224.526083333334 and Upper Bound - 9250.686083333334
CLT for males having sample size 300: Lower Bound - 9248.651600000001 and Upper Bound - 9616.184383333333
CLT for females having sample size 300: Lower Bound - 8574.293291666667 and Upper Bound - 8899.02515
CLT for males having sample size 300: Lower Bound - 9368.370273333334 and Upper Bound - 9482.6523925
CLT for females having sample size 300: Lower Bound - 8688.305295 and Upper Bound - 8786.9428275
```

1. Is the confidence interval computed using the entire dataset wider for one of the genders? Why is this case?

Ans 1. Yes the confidence interval computed using the entire data set is wider for Male Gender as compared with the females. This is because if the data for one gender has more variance in spending it will lead to wider connfidence interval.

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

Ans 2. As the sample size increases, the confidence interval generally becomes narrower due to better estimation of the population mean.

3. Do the confidence interval for the different sample sizes overlap?

Ans 3. The confidence interval for smaller sample sizes overlap significantly, but the overlap decreases as the sample size increases, indicating more precise estimates.

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

Ans 4. As the sample size increases, the distribution of the means becomes narrower and more symmetric, approximating a normal distribution due to the central limit theorem.

Insights:

I: Male customers consistently spend more on average compared to female customers

II: Marketing campaigns and product promotions could focus on products preferred by male customers during peak shopping season

III: For female customers, identifying specific product category where spending is higher can help create targeted campaigns to boost sales.

5. How does the Marital Status affect the amount spent?

```
purchase_married = df[df['Marital_Status'] == 1]['Purchase']
purchase_unmarried = df[df['Marital_Status'] == 0]['Purchase']
```

```
print("Size of purchase married:", len(purchase_married))
print("Size of purchase unmarried:", len(purchase_unmarried))
```

```
Size of purchase married: 225337
Size of purchase unmarried: 324731
```

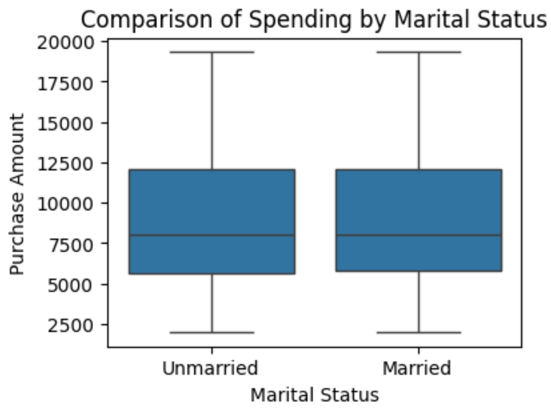
```
mean_married = purchase_married.mean()
std_married = purchase_married.std()
count_married = len(purchase_married)
```

```
mean_unmarried = purchase_unmarried.mean()
std_unmarried = purchase_unmarried.std()
count_unmarried = len(purchase_unmarried)
```

```
print(f"Married: Mean={mean_married}, Std={std_married}, Count={count_married}")
print(f"Unmarried: Mean={mean_unmarried}, Std={std_unmarried}, Count={count_unmarried}")
```

```
Married: Mean=9253.669823420034, Std=4843.48644692002, Count=225337
Unmarried: Mean=9258.820463706883, Std=4864.581471475336, Count=324731
```

```
plt.figure(figsize=(4,3))
sns.boxplot(x='Marital_Status', y='Purchase', data=df)
plt.title('Comparison of Spending by Marital Status')
plt.xticks([0, 1], ['Unmarried', 'Married'])
plt.xlabel('Marital Status')
plt.ylabel('Purchase Amount')
plt.show()
```



```
from scipy.stats import ttest_ind

t_stat, p_value = ttest_ind(purchase_married, purchase_unmarried, equal_var=False)
print(f"T-Statistic: {t_stat}, P-Value: {p_value}")
```

```
T-Statistic: -0.3871666222599602, P-Value: 0.6986330280619595
```

Here the P-Value is not significant as it is lying below 0.05

```
import numpy as np

def bootstrap_ci(df, num_bootstraps=1000, ci=95):
    means = []

    for _ in range(num_bootstraps):
        sample = np.random.choice(df, size=len(df), replace=True)
        means.append(np.mean(sample))

    lower_bound = np.percentile(means, (100 - ci) / 2)
    upper_bound = np.percentile(means, 100 - (100 - ci) / 2)

    return lower_bound, upper_bound
```

```
ci_married = bootstrap_ci(purchase_married)
ci_unmarried = bootstrap_ci(purchase_unmarried)
```

```
print(f"Married CI: {ci_married}")
print(f"Umarried CI: {ci_unmarried}")
```

```
Married CI: (9233.586013947997, 9275.222056630735)
Umarried CI: (9241.273890466262, 9276.6313196923)
```

1. There is not much difference between the mean and std purchase of both married and unmarried customers.
2. There is not much stastical difference in spending between married and unmarried individuals.
3. Unmarried individuals usually spend more than Married customers.
4. Unmarried customers average spend is 9258.82 and Married customers average spend is 9253.66
5. Walmart can leverage these insights to create targeted promotions for married customers example: family-oriented bundles and, for unmarried customers example: lifestyle products.

6. How does age affect the amount spent?

```
age_groups = df['Age'].unique()
```

```
age_groups
```

```
array(['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25'],
      dtype=object)
```

```
ci_age_groups = {}
for age in age_groups:
    purchase_by_age = df[df['Age'] == age]['Purchase']
    ci_age_groups[age] = bootstrap_ci(purchase_by_age, 300)
```

```
ci_age_groups
```

```
{'0-17': (8862.31071877897, 9019.36578267779),
 '55+': (9265.260611979167, 9396.15518391927),
 '26-35': (9225.012267916589, 9264.887617094819),
 '46-50': (9157.250600643312, 9246.559123432748),
 '51-55': (9461.895144281967, 9561.808558219267),
 '36-45': (9294.065896075917, 9353.28357898612),
 '18-25': (9136.448604505318, 9196.002312612884)}
```

Insights:

1. Age group 26-35 could be targeted with premium products and larger family-oriented bundles
2. Age group 18-25 could benefit from discounts or deals on budget friendly products.
3. Older customers may be drawn to high quality or necessity driven purchases.

A. Checking whether the average spending of males and females overlap or not using the CLT method?

1. There's a significant difference between the spending between genders as amount spent by male customers is larger than the amount spent by the female customers.
2. Walmart can market products uniformly to both genders but explore potential trends in product preferences.

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

Recommendations

- Gender Based Spending:
  - Walmart can target both genders equally but explore gender-specific product preferences especially for the male customers as the tend to spend more on walmart products.
- Spending by Martial Status:

```
*  The interval does not overlap as it is been observed that single customer spend more as and when compared with the married customers. Walmart c
```

- Age Group Spending:
  - The age group 26-35 spends significantly more as compared to any other group. Walmart can prioritize marketing efforts toward this demographic.
  - Provide budget friendly deals for 18-25 and practical products for 36-45.