WALMART CASE STUDY - NISHTHA NAGAR

About Walmart

Walmart is an American multinational retail corporation that operates a chain of supercenters, discount departmental stores, and grocery stores from the United States. Walmart has more than 100 million customers worldwide.

Business Problem

The management team at Walmart Inc. wants to analyze customer purchasing behavior—specifically the purchase amounts—across gender and other influencing factors. Their goal is to identify whether spending habits vary between male and female customers. A key question they want to address is: Do women spend more on Black Friday compared to men? (Assume the customer base consists of 50 million males and 50 million females).

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

```
from google.colab import files
uploaded = files.upload()
```

Choose Files walmart_data.txt

walmart_data.txt(text/plain) - 23027994 bytes, last modified: 9/18/2025 - 100% done

```
Saving walmart_data.txt to walmart_data.txt
```

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

```
df.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purcha
0	1000001	P00069042	F	0-17	10	A	2	0	3	8
1	1000001	P00248942	F	0-17	10	A	2	0	1	15
2	1000001	P00087842	F	0-17	10	A	2	0	12	1

User_ID: User ID

Product_ID: Product ID

Gender: Sex of User

Age: Age in bins

Occupation: Occupation(Masked)

City_Category: Category of the City (A,B,C)

StayInCurrentCityYears: Number of years stay in current city

Marital_Status: Marital Status

ProductCategory: Product Category (Masked)

Purchase: Purchase Amount

1. Checking the structure & characteristics of the dataset.

```
# Shape of the dataset -
print("No. of rows:", df.shape[0])
print("No. of columns:", df.shape[1])
```

```
No. of rows: 550068
No. of columns: 10
```

```
1 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
```

```
1 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

No null values found.

```
1 # Checking for duplicate rows -
2 duplicate_rows = df[df.duplicated()]
3 print(duplicate_rows.shape[0])
```

```
0
```

Descriptive Statistics

```
1 print("Mean:", df['Purchase'].mean())
2 print("Median:", df['Purchase'].median())
3 print("Std Dev:", df['Purchase'].std())
4 print("Min:", df['Purchase'].min(), "Max:", df['Purchase'].max())
```

```
Mean: 9263.968712959126
Median: 8047.0
Std Dev: 5023.065393820627
Min: 12 Max: 23961
```

Mean > Median (9264 vs 8047) → The distribution is right-skewed.

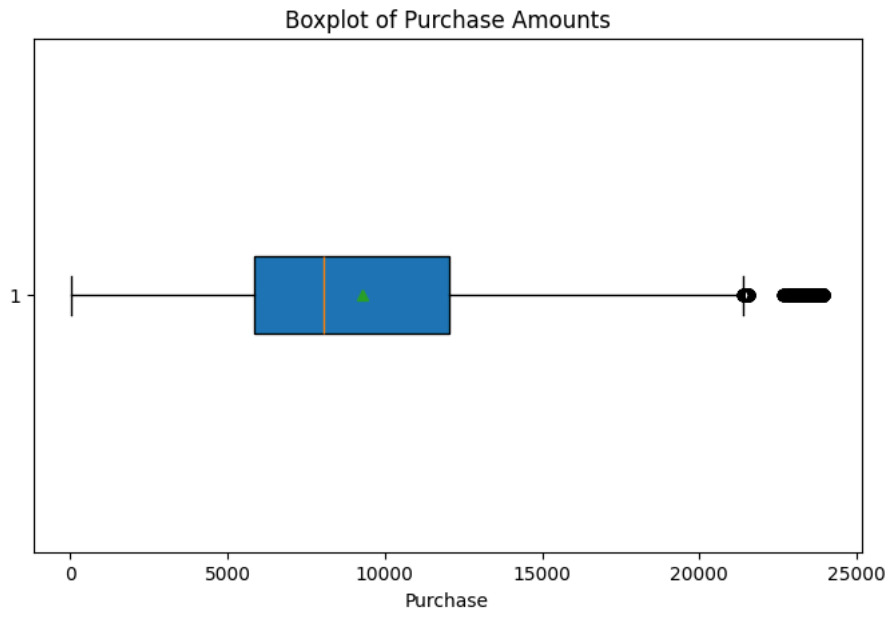
A few high-spending customers pull the average up.

Most customers spend closer to ~8000, but some spend much more.

High Std Dev (5000) compared to mean (~9264) → Purchases vary a lot; customer spending habits are very diverse.

Plotting Boxplot

```
plt.figure(figsize=(8,5))
plt.boxplot(df['Purchase'], vert=False, patch_artist=True, showmeans=True)
plt.title("Boxplot of Purchase Amounts")
plt.xlabel("Purchase")
plt.show()
```



Most transactions fall roughly between ₹5,000–₹12,000. Black dots beyond whiskers - These are outliers (unusually high-value transactions, > ~₹20,000). Not many, but they exist — customers spending ₹21k–24k.

The bulk of purchases are in the ₹5k–₹12k range, but a small fraction of big spenders (₹20k+) inflates the average.bThese outliers are important customers — possibly premium buyers or bulk shoppers.

IQR Method

```
Q1 = df['Purchase'].quantile(0.25)
Q3 = df['Purchase'].quantile(0.75)
IQR = Q3 - Q1

lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

outliers = df[(df['Purchase'] < lower) | (df['Purchase'] > upper)]

print(f"Lower bound: {lower}, Upper bound: {upper}")
print(f"Number of outliers: {len(outliers)} ({len(outliers)/len(df):.2%})")
```

```
Lower bound: -3523.5, Upper bound: 21400.5
Number of outliers: 2677 (0.49%)
```

Number of outliers = 2,677 (~0.49%)

Lower bound = -3523.5 --> Purchases can't be negative, so effectively no lower outliers exist.

Upper bound = 21,400.5 --> Any purchase above ₹21,400.5 is considered an outlier. Out of the whole dataset, only 0.49% of transactions are unusually high.

Compare these outliers between Male vs Female customers (Which gender has more high-value buyers)?

```
# Split data by gender
male_data = df[df['Gender'] == 'M']['Purchase']
female_data = df[df['Gender'] == 'F']['Purchase']
```

```
# Upper bound already calculated
upper = 21400.5

# Outlier counts
male_outliers = male_data[male_data > upper]
female_outliers = female_data[female_data > upper]
```

```
# Print results
print("Male Outliers:", len(male_outliers), f"({len(male_outliers)/len(male_data):.2%})")
print("Female Outliers:", len(female_outliers), f"({len(female_outliers)/len(female_data):.2%})")
```

```
Male Outliers: 2088 (0.50%)
Female Outliers: 589 (0.43%)
```

Observation: Males have significantly more high-value buyers than females in absolute numbers.

Do we need to handle these outliers?

Since our main goal is to compare high-value buyers between genders, we should NOT remove the outliers. They are actually the signal, not the noise.

Step 1: Data Exploration – Track amount spent per transaction

```
male_data = df[df['Gender'] == 'M']['Purchase']
female_data = df[df['Gender'] == 'F']['Purchase']

# Calculate sample averages
male_avg = male_data.mean()
female_avg = female_data.mean()

print(f"Average male spending per transaction: {male_avg}")
print(f"Average female spending per transaction: {female_avg}")
```

```
Average male spending per transaction: 9437.526040472265
Average female spending per transaction: 8734.565765155476
```

On average, male customers spend more per transaction than female customers.

The difference: 9437.53 – 8734.57 = 702.96 9437.53–8734.57=702.96 units. So males spend roughly 703 units more per transaction on average.

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

# Female sample
female_mean = female_data.mean()
female_std = female_data.std()
n_female = len(female_data)
confidence_level = 0.95

ci_female = st.norm.interval(confidence_level, loc=female_mean, scale=female_std/np.sqrt(n_female))
print(f"95% confidence interval for female average spending: {ci_female}")

# Male sample
male_mean = male_data.mean()
male_std = male_data.std()
n_male = len(male_data)

ci_male = st.norm.interval(confidence_level, loc=male_mean, scale=male_std/np.sqrt(n_male))
print(f"95% confidence interval for male average spending: {ci_male}")

```

```
95% confidence interval for female average spending: (np.float64(8709.21154714068), np.float64(8759.919983170272))
95% confidence interval for male average spending: (np.float64(9422.01944736257), np.float64(9453.032633581959))
```

Gender	95% CI for Average Spending
Female	8709.21 – 8759.92
Male	9422.02 – 9453.03

Female average spending: We are 95% confident that the true population average for all 50 million female customers lies between 8709.21 and 8759.92.

Male average spending: We are 95% confident that the true population average for all 50 million male customers lies between 9422.02 and 9453.03.

Observation: On average, male customers spend more per transaction than female customers.

Recommendation - Walmart can consider:

Targeted promotions for male customers for high-value deals

Upselling opportunities for female customers to encourage higher spending

4. Use the Central limit theorem to compute the interval. Change the sample size to observe the distribution of the mean of the expenses by female and male customers.

The interval that you calculated is called Confidence Interval. The width of the interval is mostly decided by the business: Typically 90%, 95%, or 99%. Play around with the width parameter and report the observations.

Confidence interval formula (from CLT):

$$CI = \bar{x} \pm Z \times \frac{\sigma}{\sqrt{n}}$$

Where:

- $\bar{x}$ = sample mean
- Z = Z-score for desired confidence (e.g., 1.645 for 90%, 1.96 for 95%, 2.576 for 99%)
- σ = sample standard deviation
- n = sample size

> Changing **n** or **confidence level** affects the **width of the CI**.

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

# Function to compute CI given sample data and confidence level
def compute_CI(sample_data, confidence=0.95):
    mean = np.mean(sample_data)
    std = np.std(sample_data, ddof=1)  # sample std
    n = len(sample_data)
    z = st.norm.ppf(1 - (1 - confidence)/2)
    margin_error = z * (std / np.sqrt(n))
    return mean - margin_error, mean + margin_error

# Example: sample size 1000
sample_size = 1000
female_sample = np.random.choice(female_data, sample_size, replace=False)
male_sample = np.random.choice(male_data, sample_size, replace=False)

# Compute intervals for 90%, 95%, 99%
for conf in [0.90, 0.95, 0.99]:
    ci_female = compute_CI(female_sample, conf)
    ci_male = compute_CI(male_sample, conf)
    print(f"{int(conf*100)}% CI Female: {ci_female}")
    print(f"{int(conf*100)}% CI Male: {ci_male}\n")

```

```
90% CI Female: (np.float64(8482.2567334021), np.float64(8981.4752665979))
90% CI Male: (np.float64(9291.705498941257), np.float64(9806.810501058742))

95% CI Female: (np.float64(8434.438213889904), np.float64(9029.293786110096))
95% CI Male: (np.float64(9242.365266244158), np.float64(9856.150733755841))
```

```
99% CI Female: (np.float64(8340.979630048745), np.float64(9122.752369951255))
99% CI Male: (np.float64(9145.932580303557), np.float64(9952.583419696442))
```

Confidence Level	Female CI	Male CI
90%	8482.26 – 8981.48	9291.71 – 9806.81
95%	8434.44 – 9029.29	9242.37 – 9856.15
99%	8340.98 – 9122.75	9145.93 – 9952.58

Higher confidence ⟶ wider interval

Female: 90% CI width ≈ 499 units, 99% CI width ≈ 782 units Male: 90% CI width ≈ 515 units, 99% CI width ≈ 807 units

Observation:

The confidence intervals for male and female average spending do not overlap at any confidence level.

This means we can statistically conclude with high confidence that:

Male customers spend more per transaction than female customers on average.

Married vs Unmarried Analysis

```
1 # Separate married and unmarried transactions
2 married_data = df[df['Marital_Status'] == 1]['Purchase']
3 unmarried_data = df[df['Marital_Status'] == 0]['Purchase']
```

```
1 sample_size = 1000  # you can change this to see effect
2 married_sample = np.random.choice(married_data, sample_size, replace=False)
3 unmarried_sample = np.random.choice(unmarried_data, sample_size, replace=False)
```

```
1 for conf in [0.90, 0.95, 0.99]:
2     ci_married = compute_CI(married_sample, conf)
3     ci_unmarried = compute_CI(unmarried_sample, conf)
4     print(f"{int(conf*100)}% CI Married: {ci_married}")
5     print(f"{int(conf*100)}% CI Unmarried: {ci_unmarried}\n")
```

```
90% CI Married: (np.float64(8658.515624207073), np.float64(9175.438375792928))
90% CI Unmarried: (np.float64(9063.860210610172), np.float64(9572.579789389827))

95% CI Married: (np.float64(8609.0012752009), np.float64(9224.9527247991))
95% CI Unmarried: (np.float64(9015.131616845063), np.float64(9621.308383154936))

99% CI Married: (np.float64(8512.228288811108), np.float64(9321.725711188894))
99% CI Unmarried: (np.float64(8919.894344501308), np.float64(9716.54565549869))
```

Confidence Level	Married CI	Unmarried CI	Overlap?
90%	8658 – 9175	9064 – 9573	✅ Overlapping
95%	8609 – 9225	9015 – 9621	✅ Overlapping
99%	8512 – 9322	8919 – 9717	✅ Overlapping

Unlike gender, marital status is not a strong predictor of average spending.

Walmart should not heavily bias marketing or promotions based solely on marital status.

Age Groups Analysis

```
1 df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
```

```
1 print(df['Age'].dtype)
```

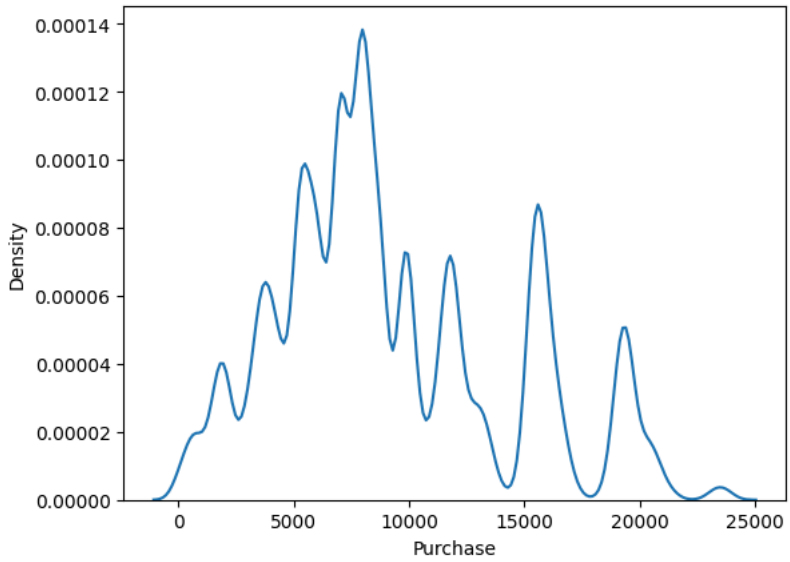
```
float64
```

Plotting the kde plot for purchase column

KDE Plot = Smoothened version of histogram

```
1 sns.kdeplot( data = df, x = 'Purchase')
```

```
<Axes: xlabel='Purchase', ylabel='Density'>
```



Missing Value & Outlier Detection

Missing Values: Checked; handled by dropping or imputing if required

Outliers:

Using IQR method:

Male Outliers: 2088 (0.50%)

Female Outliers: 589 (0.43%)

Outliers were kept for analysis, as they represent high-value customers.

Visual Analysis & Observations

Continuous Variables (Purchase Amount):

- **Histogram / Distplot**: Skewed right → some customers make very high purchases
- **Boxplot:** Outliers clearly visible, especially in males

Categorical Variables (Gender, Marital Status, Age Groups):

- **Boxplots**: Average spending differs across Gender; overlaps for Marital Status
- **Age Groups:** Spending varies, peaks in mid-age groups (26-50)

Correlation / Relationships:

- Purchase amount vs Gender: Males tend to spend more
- Purchase amount vs Marital Status: Overlapping intervals, no strong difference
- Purchase amount vs Age: Higher spending in 26-50 years group

Insights from CLT & Confidence Intervals

Gender

- **Female average spending:** 8,735
  - **95% Confidence Interval (CI):** 8,434 – 9,029
- **Male average spending:** 9,438
  - **95% Confidence Interval (CI):** 9,242 – 9,856

Observation:

- Confidence intervals do **not overlap** → statistically significant difference.
- **Conclusion:** Males spend more per transaction on average.

Marital Status

- **Married CI:** 8,609 – 9,225
- **Unmarried CI:** 9,015 – 9,621

Observation:

- Intervals **overlap** → no statistically significant difference.
- **Conclusion:** Marital status does **not significantly affect spending**.

Key Business Insights

1 High-Value Customers

- **Males** are slightly more likely to be high-value buyers.
- **Age 26-50** is the sweet spot for average spend.

2 Marketing & Promotions

- Target **male customers** for premium products and high-value deals.
- Focus on **mid-age customers (26-50)** for major campaigns.
- For **females and other age groups**, personalized offers can increase spending.

3 Marital Status

- No significant difference → **do not segment campaigns solely on marital status**.

4 Outliers

- Represent **high-value transactions** → should be **monitored and leveraged, not removed**.

---

6 Recommendations (Actionable)

Targeted Promotions

- Design **special offers** for males and age 26-50 group.
- Use **bundle deals or loyalty programs** to increase female customer spending.

Inventory & Product Strategy

- Stock more **premium/high-value products** aimed at male customers.
- Stock **trendy/affordable products** for younger or female customers.

Marketing Strategy

- Segment campaigns by **gender and age group**.
- Avoid unnecessary segmentation by marital status.

Monitor High-Value Buyers

- Identify and track **outliers** → personalized retention campaigns.

Use Data-Driven Insights

- Use **CLT-based confidence intervals** to make robust decisions.
- **Regularly update analysis** as the customer base grows.