TASK 1 : POPULATION & SAMPLING

```
import pandas as pd

# Load dataset
df = pd.read_csv("/content/Sample - Superstore.csv", encoding='latin-1')

# Display basic information
print("Dataset Shape:", df.shape)
df.head()
```

```
Dataset Shape: (9994, 21)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Produ
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-B 100017
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-C 100004
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-L 100002
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-T 100005
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-S 100007

5 rows × 21 columns

```
# Treat full dataset as Population
population = df.copy()
print("Population Size:", len(population))
```

```
Population Size: 9994
```

```
# Simple Random Sampling
sample = population.sample(n=500, random_state=42)
print("Sample Size:", len(sample))
```

```
Sample Size: 500
```

TASK 2 : SAMPLING TECHNIQUES

```
# Random Sample
random_sample = population.sample(n=500, random_state=42)
```

```
# Systematic Sample
n = 500
step = len(population) // n
systematic_sample = population.iloc[::step][:n]
```

```
# Compare Means
print("Population Sales Mean:", population["Sales"].mean())
```

```
print("Random Sample Sales Mean:", random_sample["Sales"].mean())
print("Systematic Sample Sales Mean:", systematic_sample["Sales"].mean())
```

```
Population Sales Mean: 229.85800083049833
Random Sample Sales Mean: 224.14023900000007
Systematic Sample Sales Mean: 215.72915560000004
```

Differences (Short Note):

- Population mean is the true average of all data.
- Sample means are close but not exactly the same due to sampling variability.
- Random vs. systematic sampling can yield slightly different means depending on data ordering.

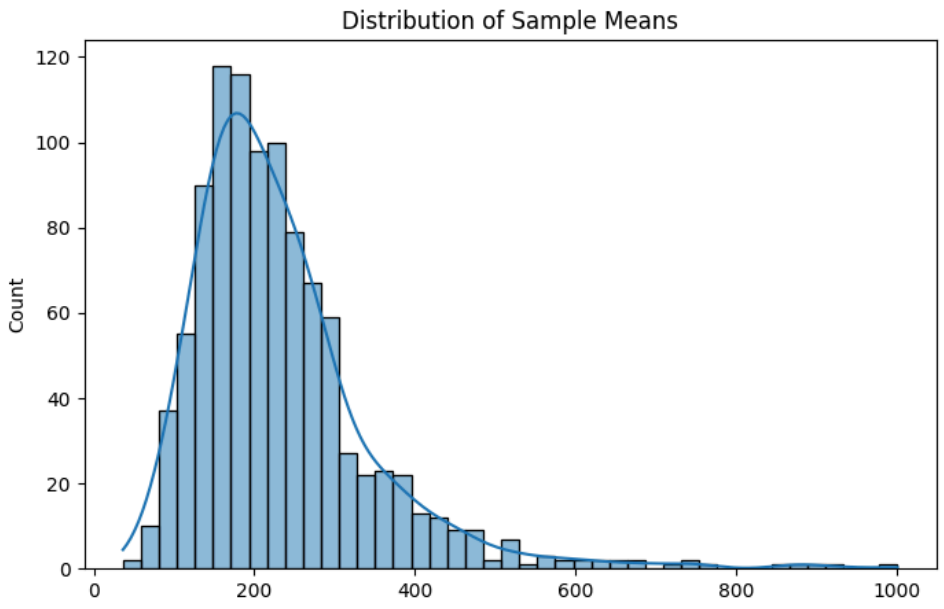
TASK 3 : CENTRAL LIMIT THEOREM (CLT)

```
import numpy as np

sample_means = []
for i in range(1000):
    sample_i = population["Sales"].sample(n=30)
    sample_means.append(sample_i.mean())
```

```
# Plot Distribution of Sample Means
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 5))
sns.histplot(sample_means, kde=True)
plt.title("Distribution of Sample Means")
plt.show()
```



Observation:

The distribution of sample means forms a bell curve (normal shape) regardless of the original distribution — this is the CLT in action

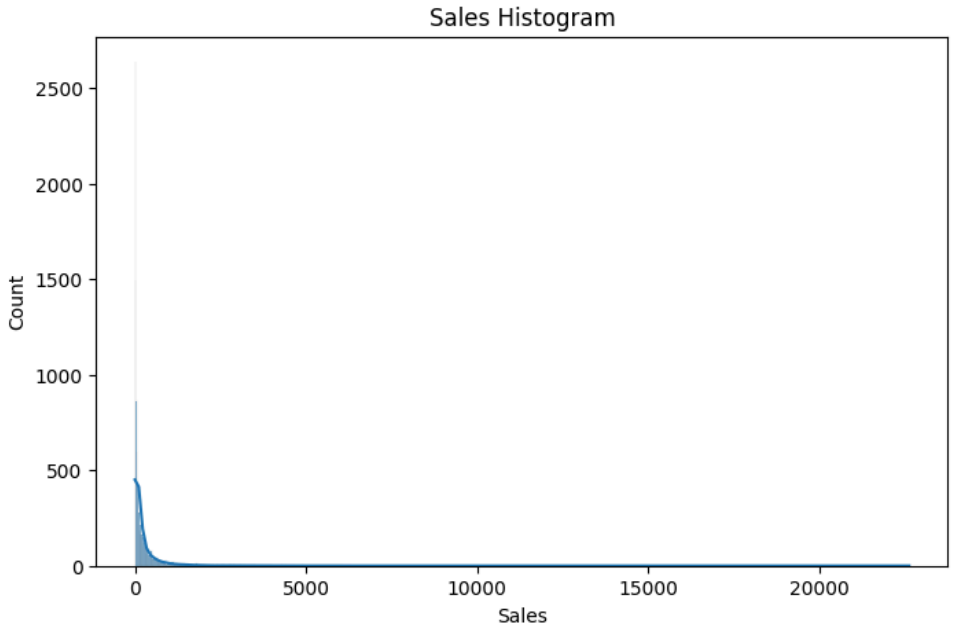
TASK 4 : NORMAL DISTRIBUTION ANALYSIS

```
num_col = population["Sales"]
mean_val = num_col.mean()
std_val = num_col.std()

print("Mean:", mean_val)
print("Standard Deviation:", std_val)
```

```
plt.figure(figsize=(8, 5))
sns.histplot(num_col, kde=True)
plt.title("Sales Histogram")
plt.show()
```

```
Mean: 229.85800083049833
Standard Deviation: 623.2451005086818
```




```
within_1 = num_col[(num_col >= mean_val - std_val) & (num_col <= mean_val + std_val)]
within_2 = num_col[(num_col >= mean_val - 2*std_val) & (num_col <= mean_val + 2*std_val)]
within_3 = num_col[(num_col >= mean_val - 3*std_val) & (num_col <= mean_val + 3*std_val)]

print("% within 1 SD:", len(within_1) / len(num_col) * 100)
print("% within 2 SD:", len(within_2) / len(num_col) * 100)
print("% within 3 SD:", len(within_3) / len(num_col) * 100)
```

```
% within 1 SD: 93.9763858314989
% within 2 SD: 97.52851711026615
% within 3 SD: 98.72923754252551
```

TASK 5 : Z-SCORE CALCULATION

```
population["Z_Score"] = (population["Sales"] - mean_val) / std_val

outliers = population[(population["Z_Score"] > 3) | (population["Z_Score"] < -3)]
print("Outliers:", outliers.shape[0])
outliers.head()
```

Outliers: 127

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Region	Product ID	Ca
27	28	US-2015-150630	9/17/2015	9/21/2015	Standard Class	TB-21520	Tracy Blumstein	Consumer	United States	Philadelphia	...	East	FUR-BO-10004834	Fu
165	166	CA-2014-139892	9/8/2014	9/12/2014	Standard Class	BM-11140	Becky Martin	Consumer	United States	San Antonio	...	Central	TEC-MA-10000822	Tech
251	252	CA-2016-145625	9/11/2016	9/17/2016	Standard Class	KC-16540	Kelly Collister	Consumer	United States	San Diego	...	West	TEC-AC-10003832	Tech
262	263	US-2014-106992	9/19/2014	9/21/2014	Second Class	SB-20290	Sean Braxton	Corporate	United States	Houston	...	Central	TEC-MA-10000822	Tech
263	264	US-2014-106992	9/19/2014	9/21/2014	Second Class	SB-20290	Sean Braxton	Corporate	United States	Houston	...	Central	TEC-MA-10003353	Tech

5 rows × 22 columns

TASK 6 : BUSINESS INSIGHTS

Q. Why is sampling required

-> Sampling allows us to make estimates of a large dataset without processing the entire population, saving time and resources.

Q. How does CLT help?

-> CLT says that the distribution of sample means will be approximately normal if the sample size is large enough. This lets us make inferential statistics.

Q. Why is normal distribution important before hypothesis testing?

-> Many statistical tests assume normality so that the results are valid and accurate.

Q. How does Z-Score help?

-> Z-score measures how far a data point is from the mean, enabling us to spot unusual (outlier) values.