Experiment No: 01 Date: 19/08/2025

**Title: A dataset contains the prices of houses in a city. Find the 25th and 75th percentiles and calculate the interquartile range (IQR). How does the IQR help in understanding the price variability?**

**Aim:** Analysing house Price Variability using percentiles and IQR.

**Objective:**

* To calculate the 25th percentile, 75th percentile and Interquartile Range of house price dataset.
* To understand how IQR provides insights into the variability of house prices.

**Procedure:**

1. Import necessary libraries.
2. Generate or load the data.
3. Calculate IQR.
4. Visualize data distribution.
5. Interpret the result.

**Python Code Implementation:**

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

file\_path=input("Enter the Data Set : ")

try:

df=pd.read\_csv(file\_path)

print("\n~~ Dataset loaded successfully~~")

print(df.head())

if df.shape[1]>1:

print("\n available colums",list(df.columns))

column\_name=input("enter the column name containing house prices:")

else:

column\_name=df.columns[0]

house\_prices=df[column\_name].dropna().values

house\_prices=house\_prices[house\_prices>0]

print(f"\n total no of valid house prices {len(house\_prices)}")

q1=np.percentile(house\_prices,25)

q3=np.percentile(house\_prices,75)

print(f"\nThe 25th percentile(Q1) of hosue price is:${q1:,.2f}")

print(f"The 75th percentile (Q3) of house price is:${q3:,.2f}\n")

iqr=q3-q1

print(f"the interquartile (IQR) of house price is ${iqr:,.2f}\n")

except FileNotFoundError:

print("File not found.please check the fiel path")

except pd.errors.EmptyDataError:

print("File is empty or corrupted")

except Exception as e:

print(f"An error occured :{e}")

plt.figure(figsize=(10,6))

sns.boxplot(y=house\_prices,color='skyblue')

plt.hlines(q1,xmin=-0.4,xmax=0.4,colors='red',linestyles='dashed',label=f'Q1(${q1:,.0f})')

plt.hlines(q3,xmin=-0.4,xmax=0.4,colors='green',linestyles='dashed',label=f'Q3(${q3:,.0f})')

plt.text(-0.45,q1,'Q1',va='center',ha='right',color='red',fontsize=12)

plt.text(-0.45,q3,'Q3',va='center',ha='right',color='green',fontsize=12)

plt.title("box plot of house price with q1 and q3")

plt.ylabel("House price ($)")

plt.grid(axis='y',linestyle="--",alpha=0.7)

plt.legend()

plt.show()

plt.figure(figsize=(12,7))

sns.histplot(house\_prices,kde=True,color='purple',bins=30)

plt.axvline(q1,color='red',linestyle='dashed',linewidth=2,label=f'Q1:${q1:.0f}')

plt.axvline(q3,color='green',linestyle='dashed',linewidth=2,label=f'Q3:${q3:.0f}')

plt.title("histogram of house price with q1 and q3")

plt.xlabel("House price ($)")

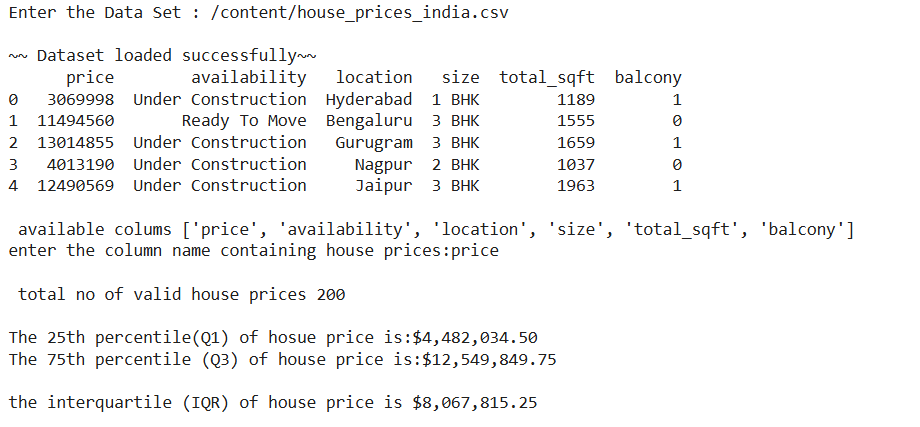
plt.ylabel('frequency')

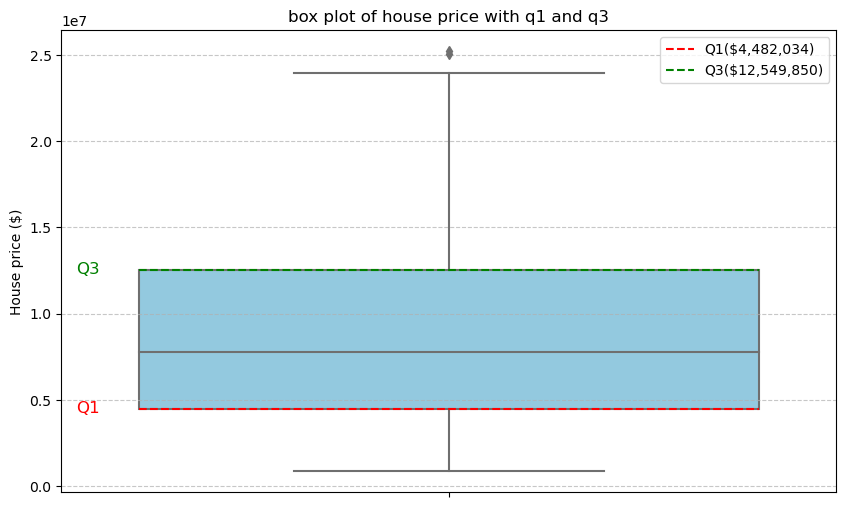
plt.legend()

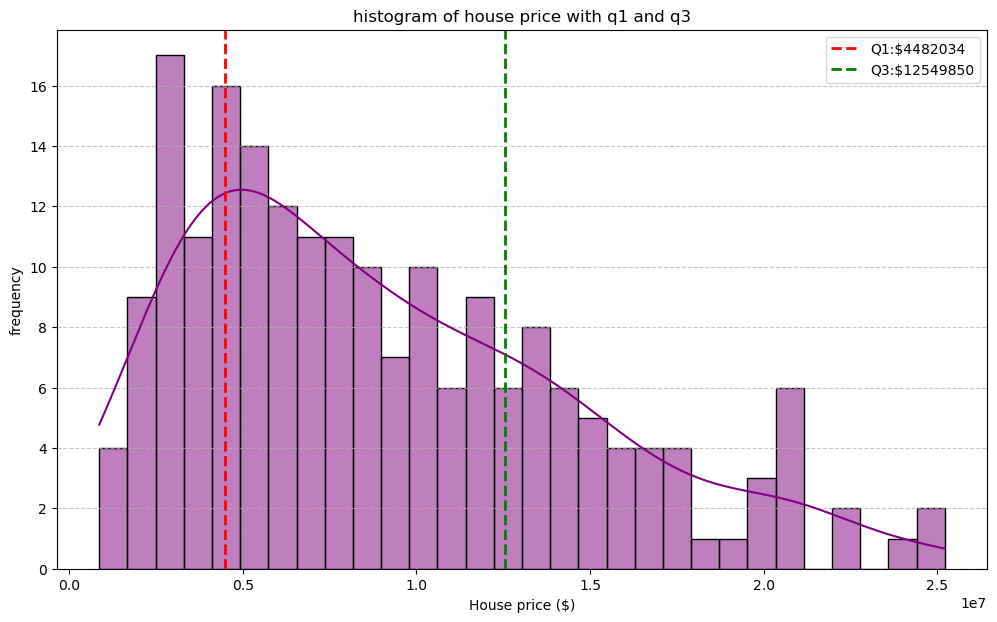
plt.grid(axis='y',linestyle="--",alpha=0.7)

plt.show()

**Result:**







**Conclusion:**

**Understanding price variability with IQR**

The Interquartile Range (IQR) is a measure of statistical dispersion, representing the range between the upper quartile (75th percentile) and the lower quartile (25th percentile). It encompasses the central; 50% of the data. In this context,

* **Robustness to Outliers:**  
  Unlike the full range(max-min), the IQR is not influenced by extreme values providing a more stable measure of spread.
* **Concentration of Data:**  
  A smaller IQR suggest that prices are tightly clustered, indicating lower variability.
* **Spread of Middle Values:**  
  A larger IQR suggest more variability in typical house prices.

Thus, the IQR offers a focused and robust view of variability in housing pricing

Experiment No: 02 Date: 26/08/2025

**Title: You are given a dataset with categorical variables about customer satisfaction levels (Low, Medium, High) and whether customers made repeat purchases (Yes/No). Create visualizations such as bar plots or stacked bar charts to explore the relationship between satisfaction level and repeat purchases. What can you infer from the data?**

**Aim:** To develop a Python program for a dataset with categorical variables about customer satisfaction levels (Low, Medium, High) and whether customers made repeat purchases (Yes/No) and to create visualizations such as bar plots or stacked bar charts to explore the relationship between satisfaction level and repeat purchases.

**Procedure:**

1. Import Necessary Libraries
2. Generate or Load Data
3. Data Preparation
4. Create Visualizations:
   1. Count Plot (Box Plot)
   2. Stacked Bar Chart
   3. Grouped Bar Chart
5. Infer from the Data

**Python Code Implementation:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

np.random.seed(42)

satisfaction\_levels = ['Low', 'Medium', 'High']

repeat\_purchases = ['No', 'Yes']

data = {Satisfaction Level': np.random.choice(satisfaction\_levels, size=500, p=[0.2, 0.5, 0.3]), 'Repeat Purchase': np.random.choice(repeat\_purchases, size=500, p=[0.6, 0.4]) }

df = pd.DataFrame(data)

df.loc[df['Satisfaction Level'] == 'High', 'Repeat Purchase'] = np.random.choice(['Yes', 'No'],

size=len(df[df['Satisfaction Level'] == 'High']), p=[0.8, 0.2])

df.loc[df['Satisfaction Level'] == 'Low', 'Repeat Purchase'] = np.random.choice(['No', 'Yes'],

size=len(df[df['Satisfaction Level'] == 'Low']), p=[0.7, 0.3])

print("--- Customer Data Snippet ---")

print(df.head())

print(f"\nTotal number of customers: {len(df)}\n")

sns.set\_style("whitegrid")

plt.figure(figsize=(8, 5))

sns.countplot(data=df, x='Satisfaction Level', order=satisfaction\_levels, palette='viridis')

plt.title('Distribution of Customer Satisfaction Levels')

plt.xlabel('Satisfaction Level')

plt.ylabel('Number of Customers')

plt.show()

plt.figure(figsize=(6, 4))

sns.countplot(data=df, x='Repeat Purchase', palette='pastel')

plt.title('Distribution of Repeat Purchases')

plt.xlabel('Repeat Purchase')

plt.ylabel('Number of Customers')

plt.show()

satisfaction\_purchase\_counts=df.groupby(['SatisfactionLevel','RepeatPurchase']).size().unstack(fill\_value=0)

satisfaction\_purchase\_proportions =satisfaction\_purchase\_counts.apply(lambda x: x / x.sum(),

axis=1)

fig, ax = plt.subplots(figsize=(10, 6))

satisfaction\_purchase\_proportions.loc[satisfaction\_levels,['No','Yes']].plot(kind='bar',stacked=True, ax=ax, cmap='coolwarm')

plt.title('Repeat Purchase Proportion by Customer Satisfaction Level (Stacked Bar Chart)')

plt.xlabel('Satisfaction Level')

plt.ylabel('Proportion of Customers')

plt.xticks(rotation=0)

plt.legend(title='Repeat Purchase')

plt.tight\_layout()

plt.show()

plt.figure(figsize=(10, 6))

sns.countplot(data=df,x='Satisfaction Level',hue='Repeat Purchase', order=satisfaction\_levels, palette='deep')

plt.title('Repeat Purchase by Customer Satisfaction Level (Grouped Bar Chart)')

plt.xlabel('Satisfaction Level')

plt.ylabel('Number of Customers')

plt.legend(title='Repeat Purchase')

plt.show()

print("--- Inferences from the Data ---")

print("To infer from the data, we examine the created visualizations, particularly the stacked and grouped bar charts.")

cross\_tab = pd.crosstab(df['Satisfaction Level'], df['Repeat Purchase'], margins=True)

print("\nCross-Tabulation of Satisfaction Level vs. Repeat Purchase:")

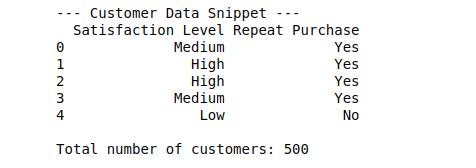
print(cross\_tab)

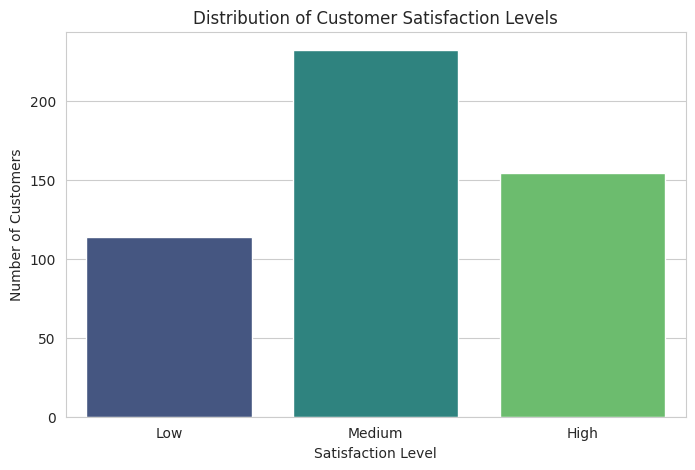
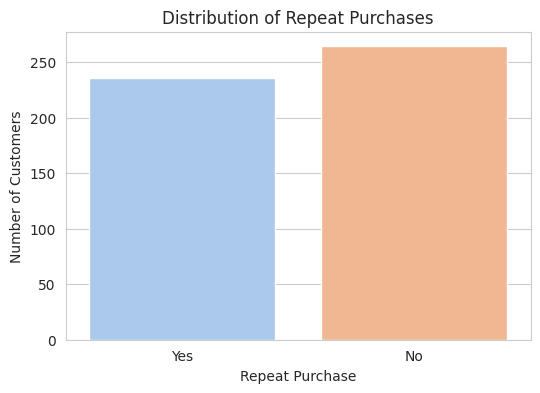
cross\_tab\_prop =pd.crosstab(df['Satisfaction Level'], df['Repeat Purchase'], normalize='index') \* 100

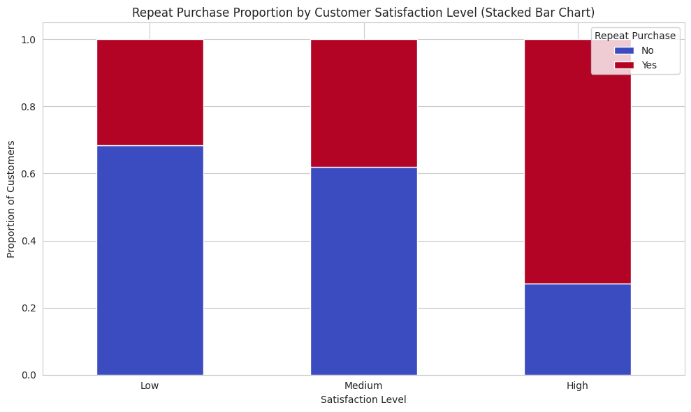
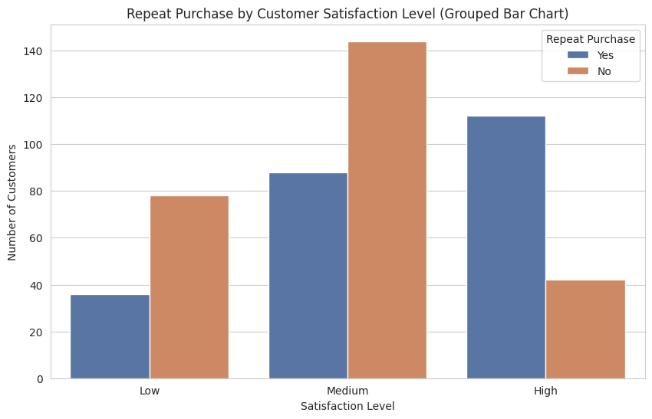
print("\nProportion of Repeat Purchase by Satisfaction Level (in %):")

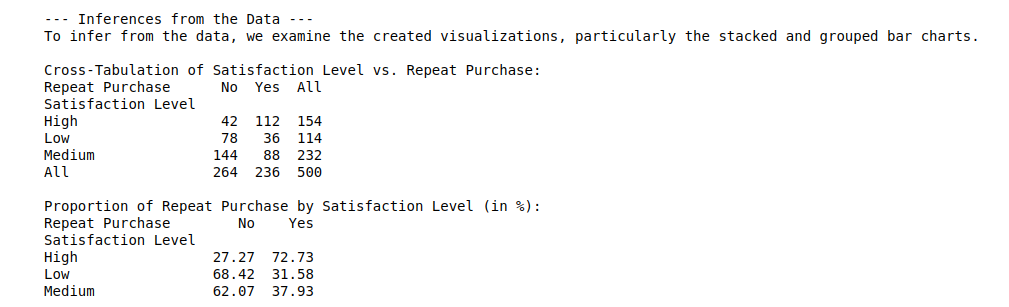
print(cross\_tab\_prop.round(2))

**Result:**

****

** **

** **

****

**Inference:**

Based on the Charts, here is how we could interpret the data

* Customers with High Satisfaction tend to have a higher number of repeat purchases
* Low Satisfaction Customers are mostly not making repeat purchases.
* Medium Satisfaction has mixed behaviour - are roughly even split into Yes/No

Experiment No: 03 Date: 04/09/2025

**Title: A dataset contains information about car models, including the engine size (in Liters), fuel efficiency (miles per gallon), and car price. Use a pair plot or correlation matrix to explore the relationships between these variables. Which variables seem to have the strongest relationships, and what might be the practical significance of these findings?**

**Aim:** To analyse the relationships among engine size (L), fuel efficiency (MPG), and price of cars using pair plots and correlation matrix, and to derive practical insights.

**Objectives:** To explore the relationships between engine size, fuel efficiency, and car price using exploratory data analysis techniques.

**Variables:**

* Engine size (Liters)
* Fuel efficiency (miles per gallon - MPG)
* Car price

**Procedures:**

1. Import necessary libraries.
2. Generate or load the data.
3. Visualization – Pair Plot
4. Interpretation

**Python Code Implementation:**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

df = pd.read\_csv("car\_dataset.csv")

print("First 5 rows of dataset:")

print(df.head())

print("--Generating pair plot--")

sns.pairplot(df,diag\_kind="hist")

plt.suptitle("Pair plot of car dataset",y=1.02)

plt.show()

corr\_matrix=df.corr(numeric\_only=True)

plt.figure(figsize=(8,6))

sns.heatmap(corr\_matrix,annot=True, cmap="coolwarm", fmt=".2f")

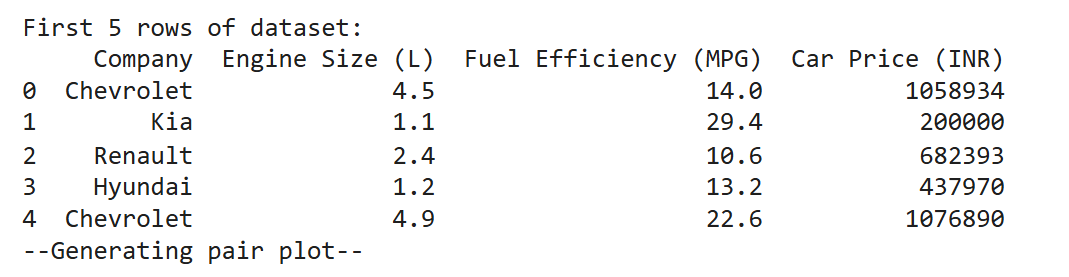
plt.title("Correlation Matrix of car dataset")

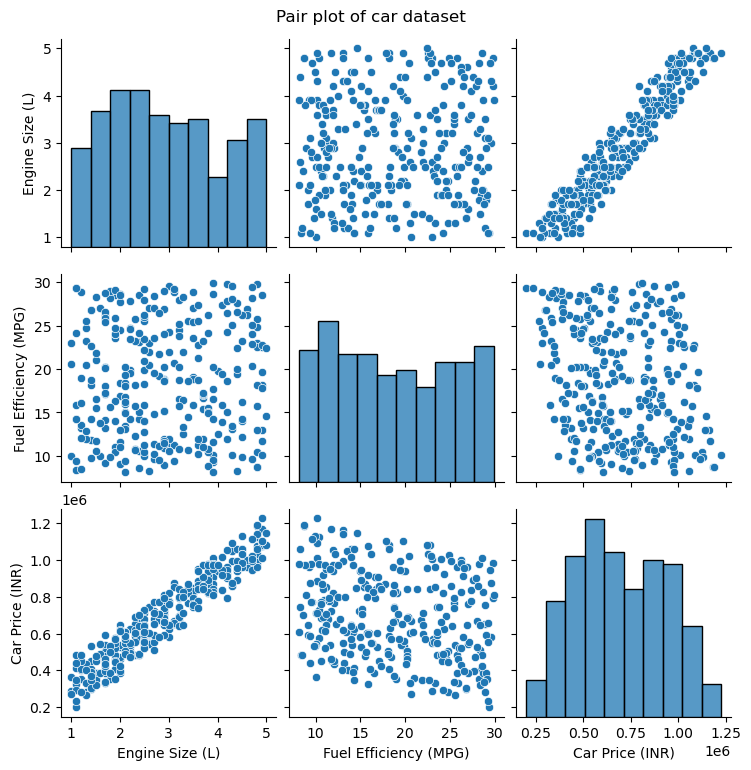
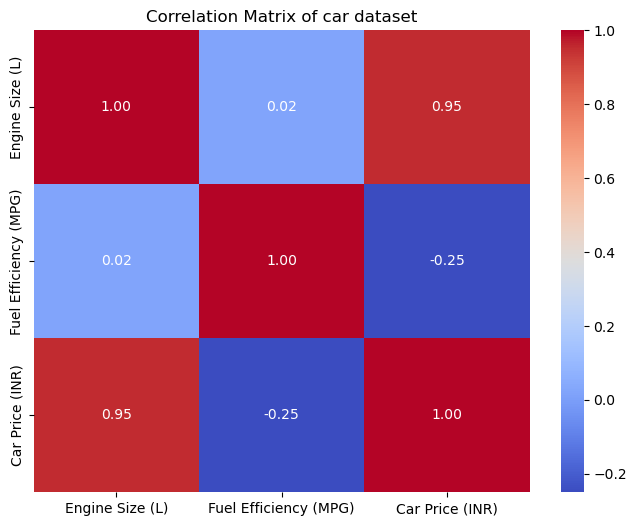
plt.show()

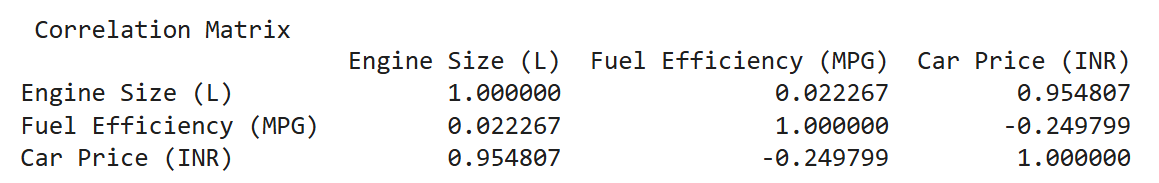
print("\n Correlation Matrix")

print(corr\_matrix)

**Results:**

****

** **

****

**Interpretation**

**Strongest Positive Correlation:** Between Engine Size (L) and Car Price (INR) (0.955), indicating that larger engine sizes are strongly associated with higher car prices.

**Strongest Negative Correlation:** Between Fuel Efficiency (MPG) and Car Price (INR) (-0.250), showing a weak negative relationship where more fuel-efficient cars tend to be slightly less expensive.

**Other Negative Correlation:** There is no other meaningful negative correlation in the matrix; the correlation between Engine Size (L) and Fuel Efficiency (MPG) is very close to zero (0.022), indicating almost no linear relationship.

Experiment No: 04 Date: 13/09/2025

**Title: You want to estimate the mean salary of software engineers in a country. You take 10 different random samples, each containing 50 engineers, and calculate the sample mean for each. Plot the distribution of these sample means. How does the Central Limit Theorem explain the shape of this sampling distribution, even if the underlying salary distribution is skewed?**

**Aim:** To analyse and estimate the mean salary of software engineers in a country.

**Objectives:** To estimate the mean salary of software engineers in a country using random sampling. Demonstrate how the Central Limit Theorem (CLT) shapes the sampling distribution of sample means.

**Procedures:**

1. Assume the population salary distribution is right-skewed (since salaries usually follow a long-tail distribution).
2. Generate a large synthetic dataset of software engineer salaries (e.g., 50,000 salaries) using a log-normal distribution to mimic real-world skewness.
3. Draw 10 random samples, each containing 50 engineers, from this population.
4. Compute the mean salary for each sample.
5. Plot the distribution of sample means (histogram).
6. Compare the shape of the distribution of sample means to the skewed population.
7. Use the Central Limit Theorem to explain why the distribution of means tends toward normality, even though the underlying data is skewed.

**Python Code Implementation:**

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

population\_size = 50000

population\_salaries = np.random.lognormal(mean=10, sigma=0.5, size=population\_size)

sample\_size = 50

num\_samples = 10

sample\_means = []

for \_ in range(num\_samples):

sample = np.random.choice(population\_salaries, size=sample\_size, replace=False)

sample\_means.append(np.mean(sample))

plt.figure(figsize= (12,5))

plt.subplot(1,2,1)

plt.hist(population\_salaries, bins=50, color='skyblue', edgecolor='black')

plt.title("Population Salary Distribution (Skewed)")

plt.xlabel("Salary"); plt.ylabel("Frequency")

plt.subplot(1,2,2)

plt.hist(sample\_means, bins=5, color='lightgreen', edgecolor='black')

plt.title("Distribution of Sample Means (n=50, 10 samples)")

plt.xlabel("Sample Mean Salary"); plt.ylabel("Frequency")

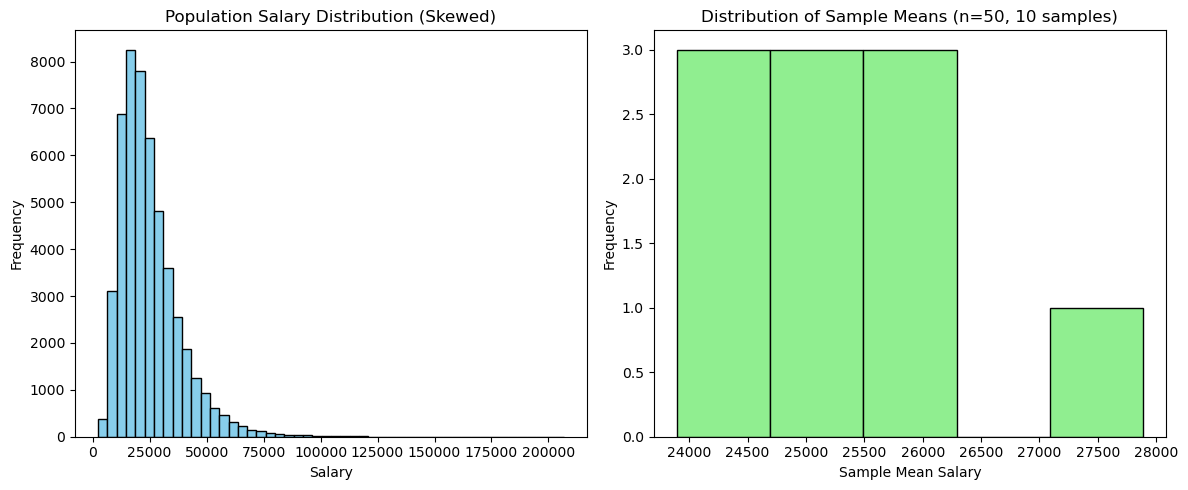
plt.tight\_layout(); plt.show()

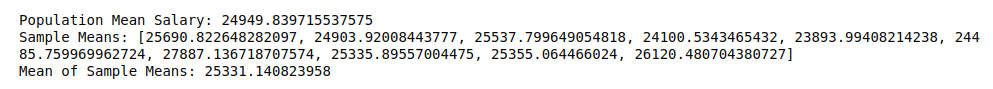
print("Population Mean Salary:", np.mean(population\_salaries))

print("Sample Means:", sample\_means)

print("Mean of Sample Means:", np.mean(sample\_means))

**Result:**





**Interpretation**

* The mean of the sample means (25,331.14) is very close to the true population mean (24,949.84), with only a small difference due to sampling variability.
* This demonstrates that **sample means are unbiased estimators** of the population mean.
* The **Central Limit Theorem** is in action: even though the population distribution is skewed, the sample means form a **normal-like distribution**.