EXPERIMENT 11

Random Sampling And Sampling Distribution

Aim:

To explore random sampling from a population and understand the concept of sampling distribution using python.

Algorithm:

1. Import necessary libraries — numpy and matplotlib.pyplot.

2. Define the population parameters: mean, standard deviation, and population size.

3. Generate a population dataset using the normal distribution.

4. Choose different sample sizes (e.g., 30, 50, 100) and define number of samples.

5. For each sample size, draw random samples from the population and compute their means.

6. Store all computed sample means for each sample size.

7. Plot histograms of sample means to visualize their sampling distributions.

8. Draw a vertical line showing the population mean and compare how distribution shape changes with larger sample sizes.

Program:

import numpy as np

import matplotlib.pyplot as plt

population\_mean = 50

population\_std = 10

population\_size = 100000

population = np.random.normal(population\_mean, population\_std, population\_size)

sample\_sizes = [30, 50, 100]

num\_samples = 1000

sample\_means = {}

for size in sample\_sizes:

sample\_means[size] = []

for \_ in range(num\_samples):

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

sample\_means[size].append(np.mean(sample))

plt.figure(figsize=(12, 8))

for i, size in enumerate(sample\_sizes):

plt.subplot(len(sample\_sizes), 1, i+1)

plt.hist(sample\_means[size], bins=30, alpha=0.7, edgecolor='black', label=f'Sample Size = {size}')

plt.axvline(np.mean(population), color='red', linestyle='dashed', linewidth=1.5, label='Population Mean')

plt.title(f'Sampling Distribution of Sample Mean (n = {size})')

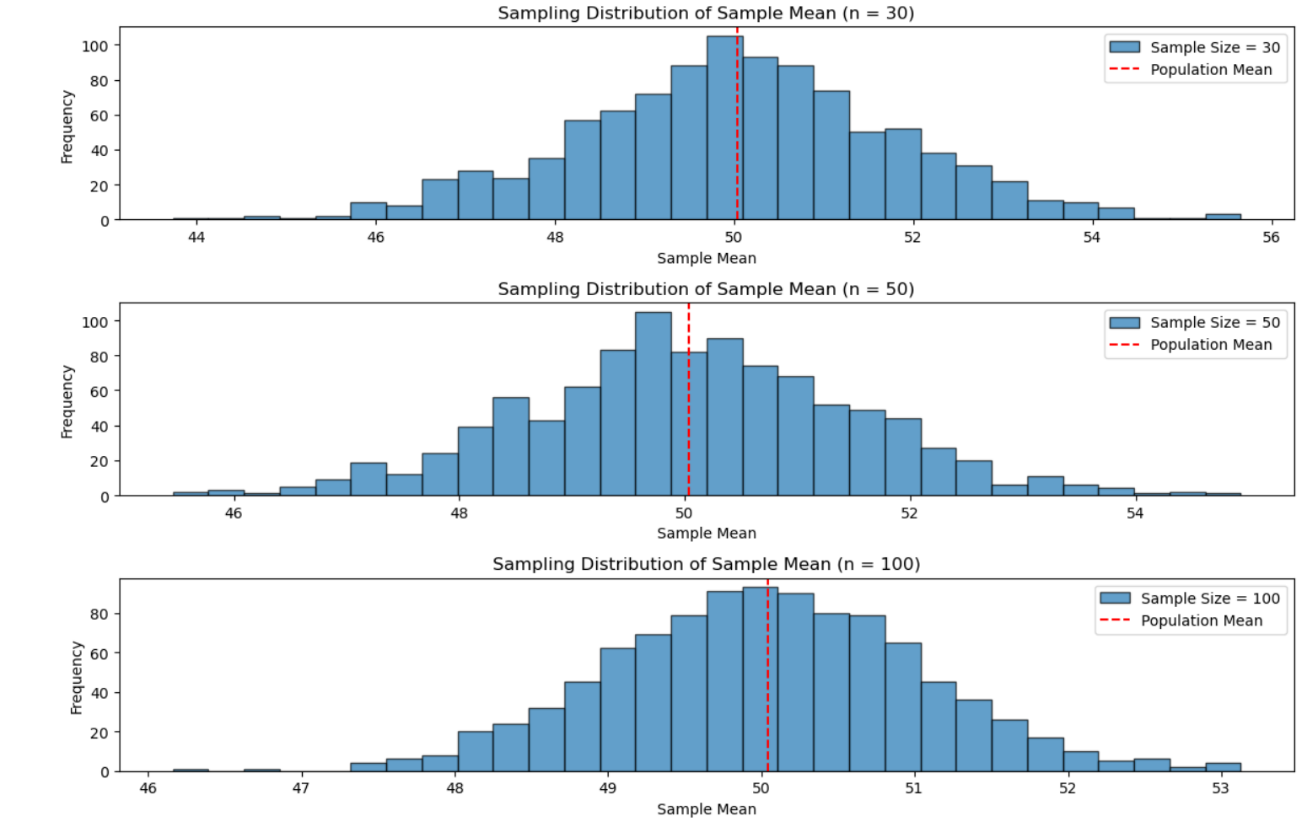
plt.xlabel('Sample Mean')

plt.ylabel('Frequency')

plt.legend()

plt.tight\_layout()

plt.show()

Output:

Result:

Hence a python program for random sampling and sampling distribution is written and executed successfully.