

Random Sampling and sampling Distribution

Aim:

To explore random sampling from a population and understand the concept of sampling distribution using Python in Jupyter Notebook.

Algorithm:

- Create a population of data with a specified distribution
- Perform random sampling from the population to create multiple samples of different sizes. Compute sample statistics for each sample.
- Plot histograms or density plots of sample statistics (e.g., sample means).
- Compare the sampling distribution of the sample statistic (mean) with the population distribution.

Program:

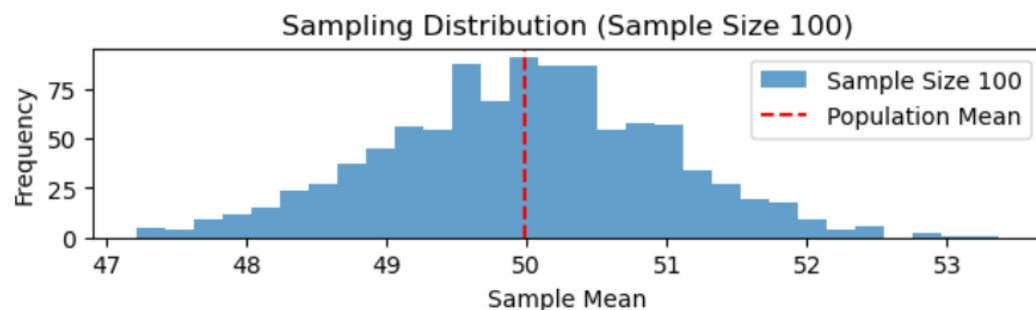
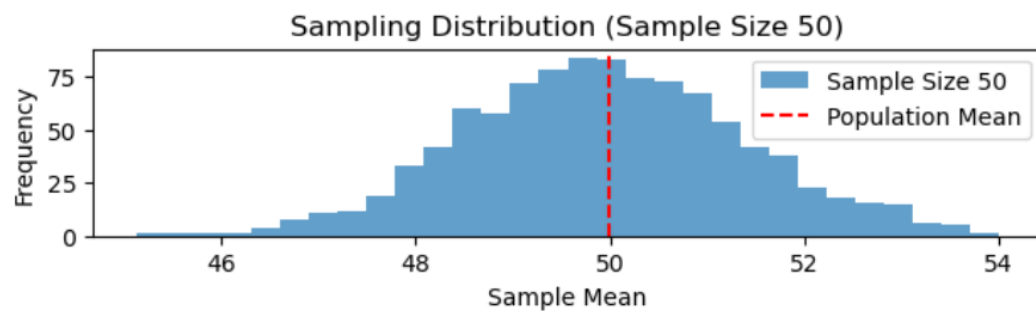
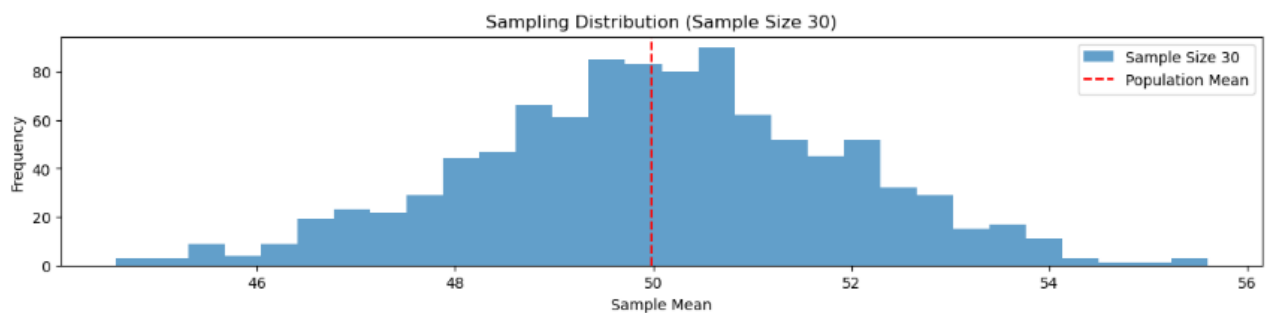
```
[6]: import numpy as np
import matplotlib.pyplot as plt

[7]: population_mean = 50
population_std = 10
population_size = 100000
population = np.random.normal(population_mean, population_std, population_size)

[8]: 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))

[9]: 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, 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 (Sample Size {size})')
    plt.xlabel('Sample Mean')
    plt.ylabel('Frequency')
    plt.legend()

plt.tight_layout()
plt.show()
```



Result:

Thus, the Python code to explore random sampling from a population using Python in Jupyter Notebook has been successfully executed.