BRSM In-Class Assignment

25.01.24

Srujana Vanka - 2020102005

In [25]:
```
# Importing Libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import beta
```

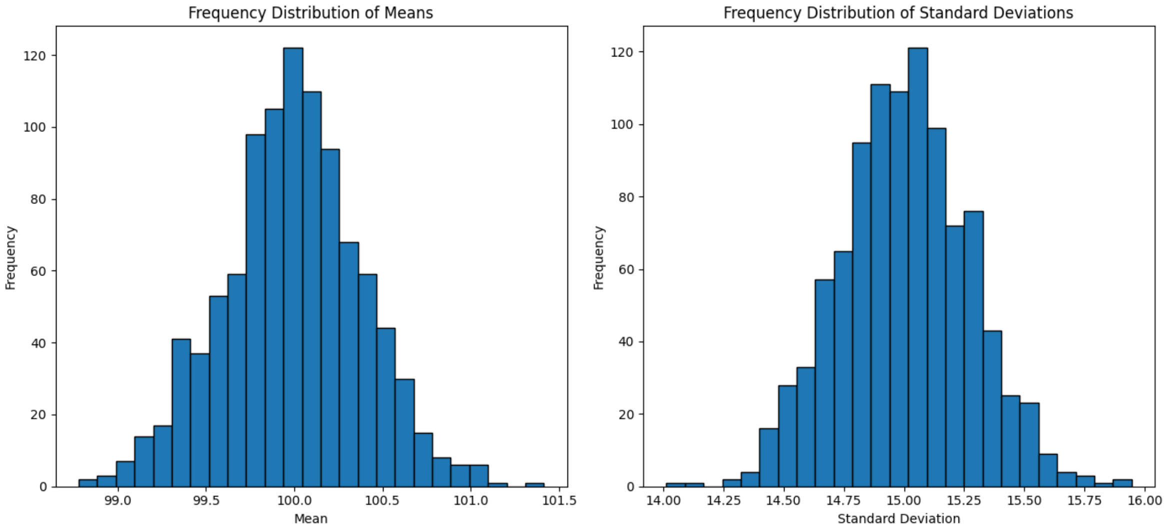
Question 1

In [8]:
```
# a. Assume that your population distribution is N(100,15).
# Given mean = 100, std deviation = 15
population_mean = 100
population_stddev = 15
sample_size = 10

# b. Sample 10 random numbers from N(100,15) and calculate the mean and s
# Generating a random sample of size 10
population_sample = np.random.normal(population_mean, population_stddev,

# Mean and standard deviation
sample_mean = np.mean(population_sample)
sample_stddev = np.std(population_sample, ddof=1)

print("Random sample of size 10:", population_sample)
print("Mean of the sample:", sample_mean)
print("Standard deviation of the sample:", sample_stddev)
```

```
Random sample of size 10: [103.05027796 105.98188458 111.52294392 104.6552
4372 100.28642972
 101.68806911 110.65496342 111.95739479 107.53614895  96.91088168]
Mean of the sample: 105.42442378546032
Standard deviation of the sample: 5.066976782763711
```

In [21]:
```
# c. Repeat this for 1000 trials, and plot the frequency distribution of

# Function to generate samples, calculate means and standard deviations
def init_sample(population_mean, population_stddev, sample_size, trials):
    means = np.zeros(trials)
    stddevs = np.zeros(trials)

    for i in range(trials):
        random_sample = np.random.normal(population_mean, population_stdd
        means[i] = np.mean(random_sample)
        stddevs[i] = np.std(random_sample, ddof=1)

    return means, stddevs


trials = 1000
```

```
means, std_devs = init_sample(population_mean, population_stddev, sample_

plt.figure(figsize=(13, 6))
plt.subplot(1, 2, 1)
plt.hist(means, bins=25, edgecolor='black')
plt.title('Frequency Distribution of Means')
plt.xlabel('Mean')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
plt.hist(std_devs, bins=25, edgecolor='black')
plt.title('Frequency Distribution of Standard Deviations')
plt.xlabel('Standard Deviation')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

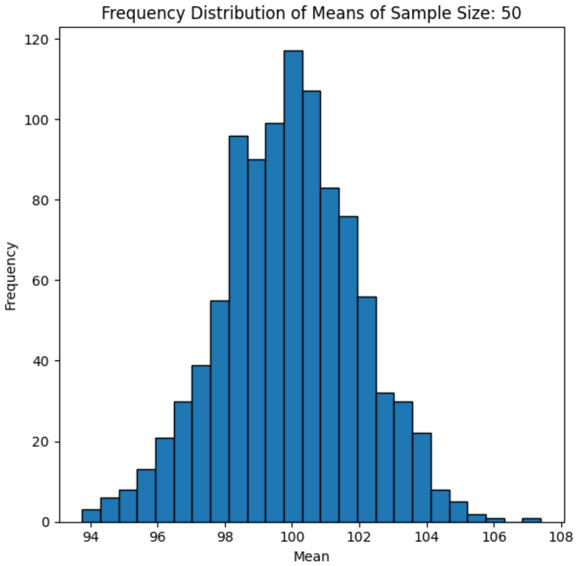


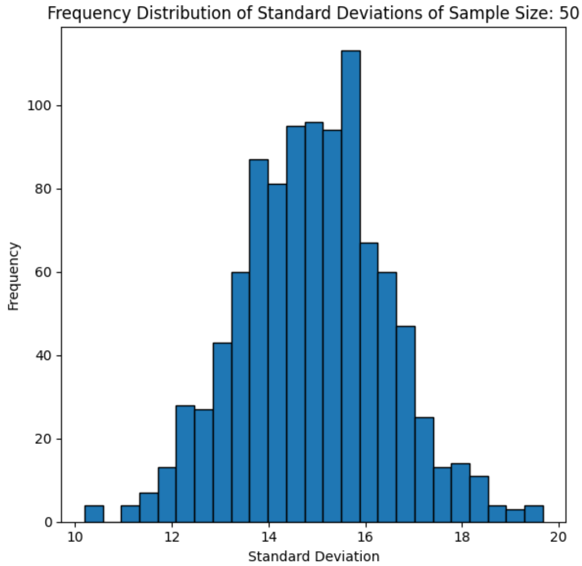
In [22]:

```
# d. Repeat steps b and c for 50, 100, 500, and 1500 numbers.
sample_sizes = [50, 100, 500, 1500]

for sample_size in sample_sizes:
    means, std_devs = init_sample(population_mean, population_stddev, sam

    # Mean and standard deviation for each sample size
    print(f"Random sample of size: {sample_size}")
    print(f"Mean: {np.mean(means)}")
    print(f"Standard Deviation: {np.mean(std_devs)}")
    print("\n")

    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.hist(means, bins=25, edgecolor='black')
    plt.title(f'Frequency Distribution of Means of Sample Size: {sample_s
    plt.xlabel('Mean')
    plt.ylabel('Frequency')

    plt.subplot(1, 2, 2)
    plt.hist(std_devs, bins=25, edgecolor='black')
    plt.title(f'Frequency Distribution of Standard Deviations of Sample S
    plt.xlabel('Standard Deviation')
    plt.ylabel('Frequency')

    plt.tight_layout()
```

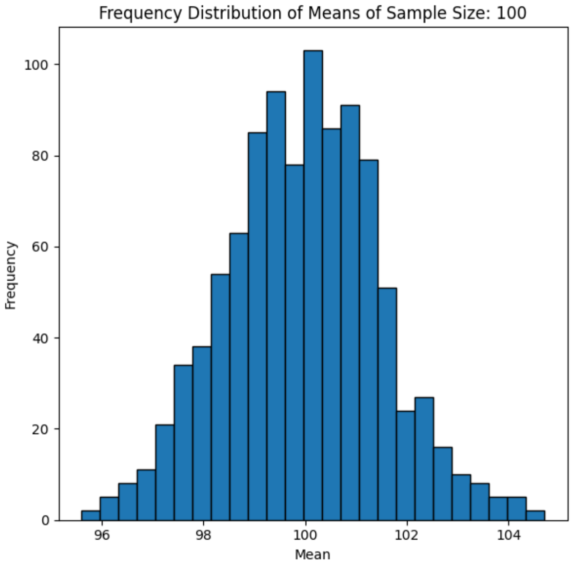
```
    plt.show()
    print("\n")
```

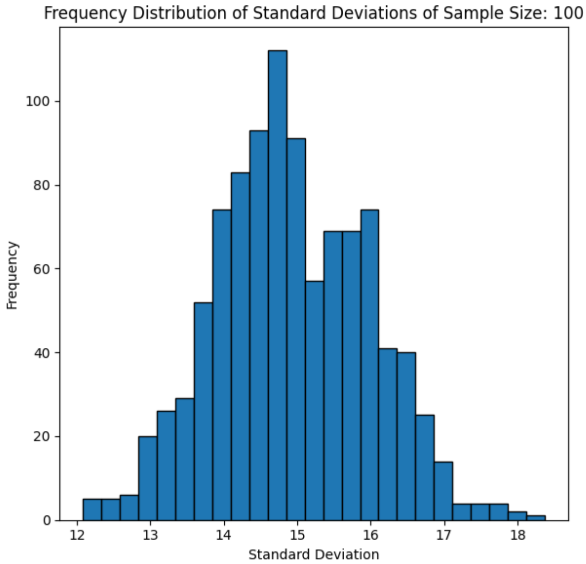
```
Random sample of size: 50
Mean: 99.90999883130944
Standard Deviation: 14.92257471533251
```



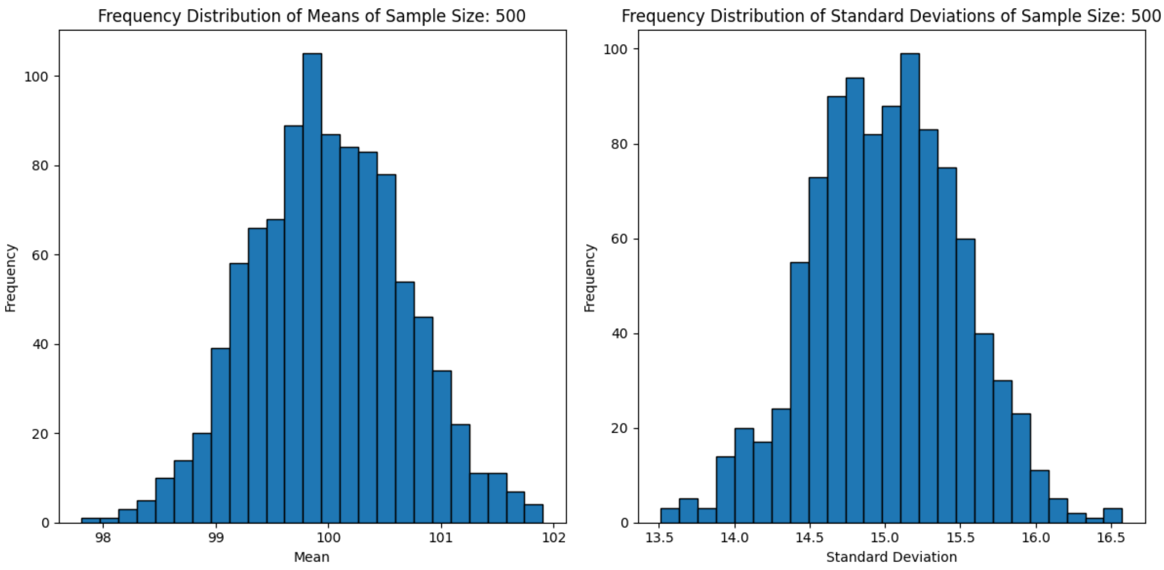



```
Random sample of size: 100
Mean: 99.95573243707041
Standard Deviation: 14.927350930239212
```

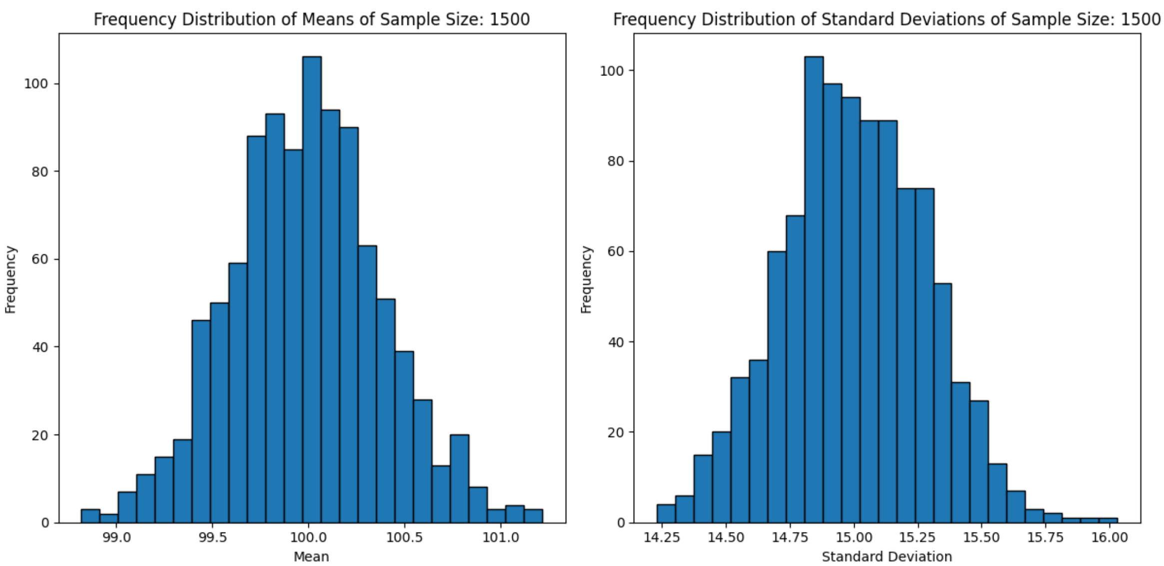





```
Random sample of size: 500
Mean: 99.99964314889905
Standard Deviation: 15.004937940761971
```




```
Random sample of size: 1500
Mean: 99.9848559335035
Standard Deviation: 15.001064131989253
```



e. Observations

For smaller sample sizes (10):

- The distribution of means for a sample size of 10 does not appear perfectly normal but tends to show a more symmetric shape than the underlying population.

For sample sizes of 50 and 100:

- As the sample size increases to 50 and 100, the distribution of means becomes more bell-shaped and symmetric, approaching a normal distribution.
- The standard deviation of the means further decreases, demonstrating the stabilizing effect of larger sample sizes.

For larger sample sizes (500 and 1500):

- The distribution of means becomes even more normal and centered around the true population mean (100).

As N increases, the distribution of sample standard deviations tends to become more centered around the population standard deviation (smaller N exhibits greater variability). With increase in the sample size, the shape of the histogram of sample standard deviations becomes more symmetric and bell shaped (similar to distribution of sample means). With larger sample sizes, the sample standard deviation becomes a more consistent and unbiased estimator of the population standard deviation.

Inferences

The Central Limit Theorem explains that, regardless of the shape of the original population distribution, the distribution of sample means will tend to be normal with a mean close to the population mean. Larger sample sizes lead to a more accurate estimation of the population mean and a narrower spread of the distribution of sample means. The Central Limit Theorem works best for larger sample sizes. With smaller sample sizes, the distribution of means may not perfectly resemble a normal distribution.

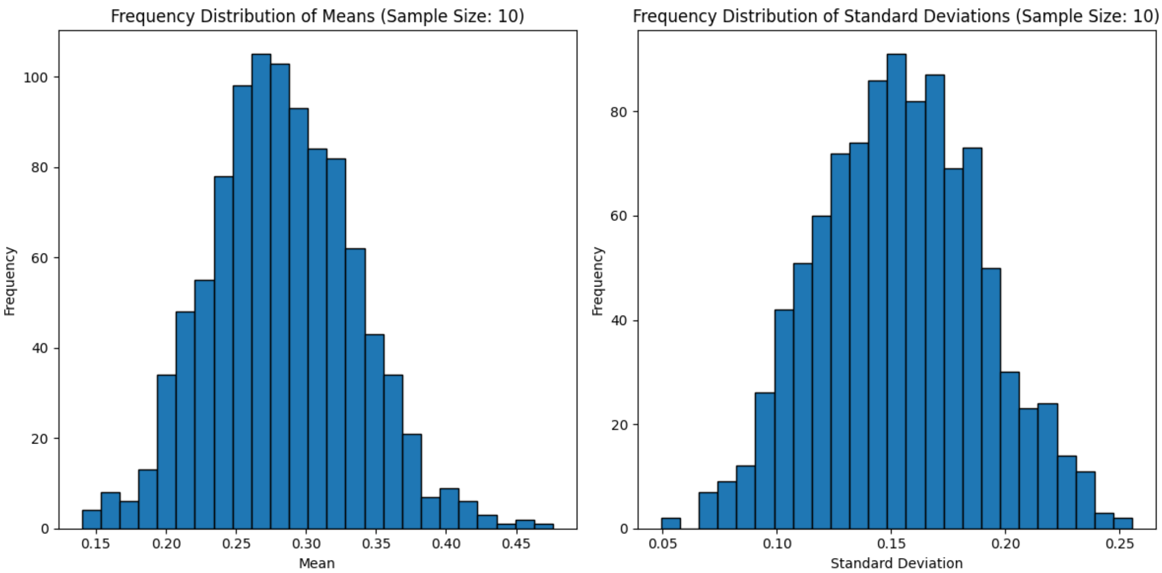
Question 2

In [30]:

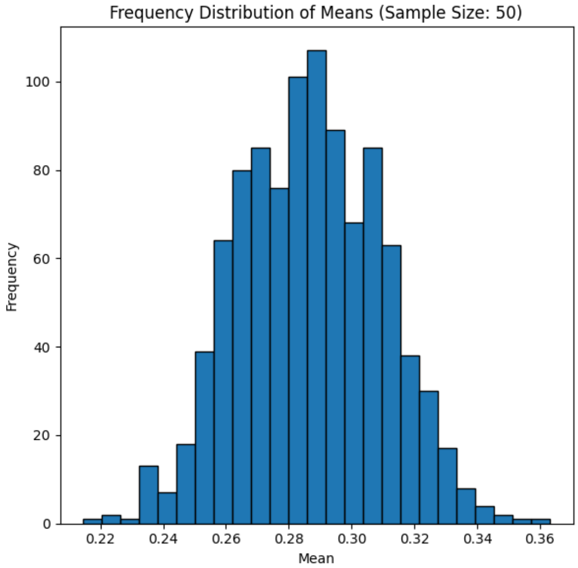
```
# Function to generate samples, calculate means and standard deviations
def init_sample_beta(alpha_population, beta_population, sample_size, tria
    means = np.zeros(trials)
    stddevs = np.zeros(trials)

    for i in range(trials):
        random_sample = beta.rvs(alpha_population, beta_population, size=
        means[i] = np.mean(random_sample)
        stddevs[i] = np.std(random_sample, ddof=1)

    return means, stddevs

# Parameters for the Beta distribution
alpha_population = 2
beta_population = 5
trials = 1000

sample_sizes = [10, 50, 100, 500, 1500]

for sample_size in sample_sizes:
    means, std_devs = init_sample_beta(alpha_population, beta_population,

    print(f"Sample Size: {sample_size}")
    print(f"Mean: {np.mean(means)}")
    print(f"Standard Deviation: {np.mean(std_devs)}")
    print("\n")

    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
```

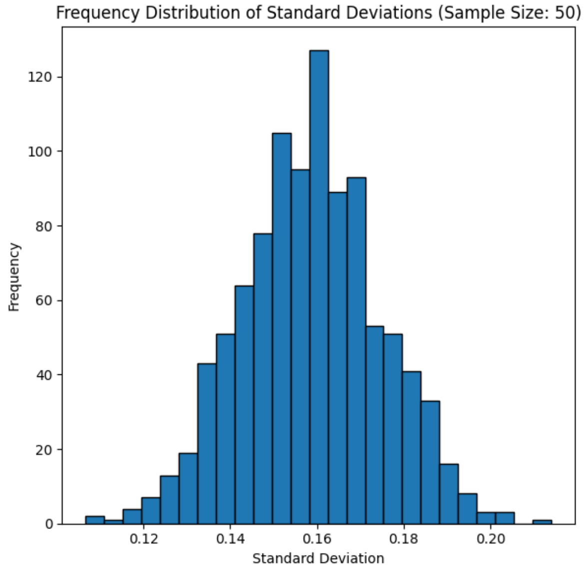
```
    plt.hist(means, bins=25, edgecolor='black')
    plt.title(f'Frequency Distribution of Means (Sample Size: {sample_siz
    plt.xlabel('Mean')
    plt.ylabel('Frequency')

    plt.subplot(1, 2, 2)
    plt.hist(std_devs, bins=25, edgecolor='black')
    plt.title(f'Frequency Distribution of Standard Deviations (Sample Siz
    plt.xlabel('Standard Deviation')
    plt.ylabel('Frequency')
    print("\n")

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

```
Sample Size: 10
Mean: 0.2829837621392738
Standard Deviation: 0.1540473128879329
```

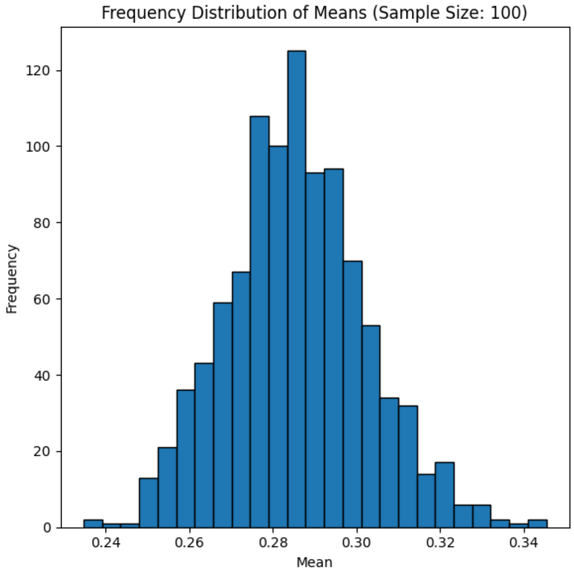


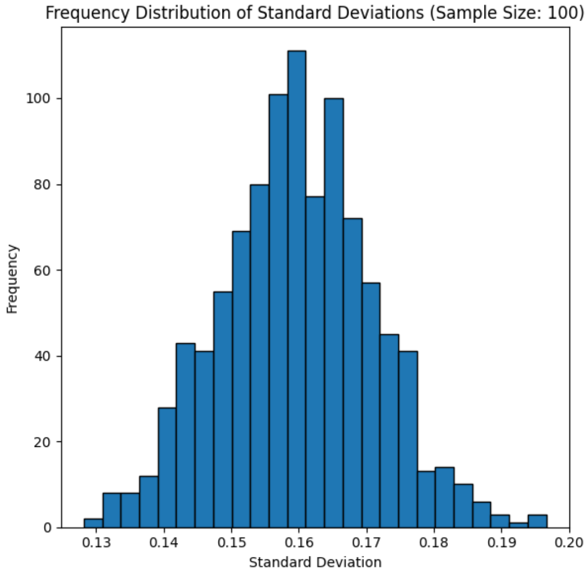

```
Sample Size: 50
Mean: 0.28670907875931373
Standard Deviation: 0.15864209616734845
```



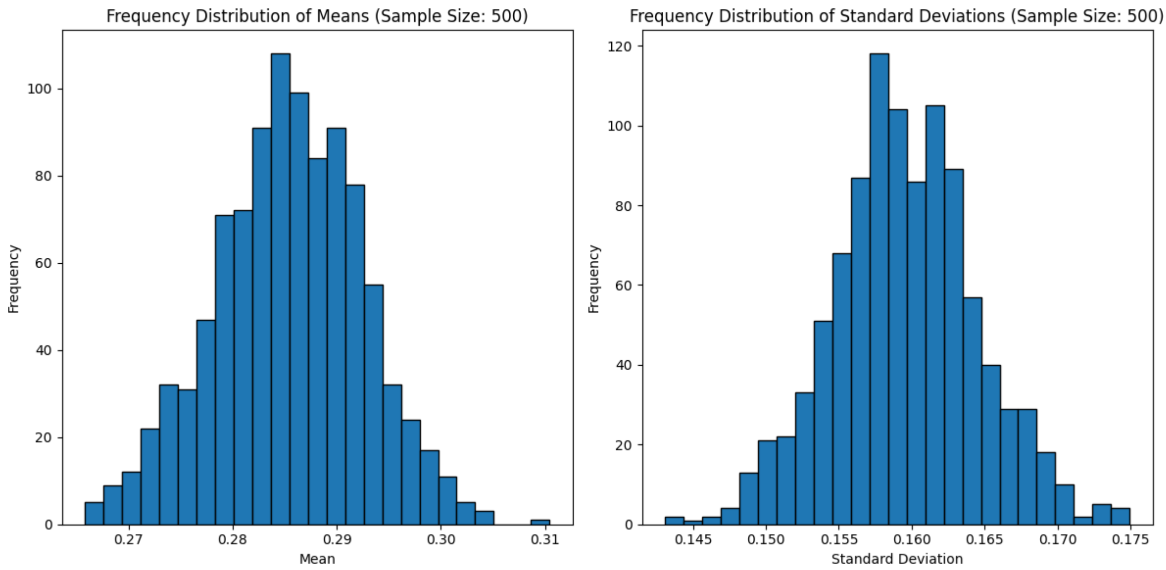



```
Sample Size: 100
Mean: 0.28570010256230616
Standard Deviation: 0.15965199997759005
```

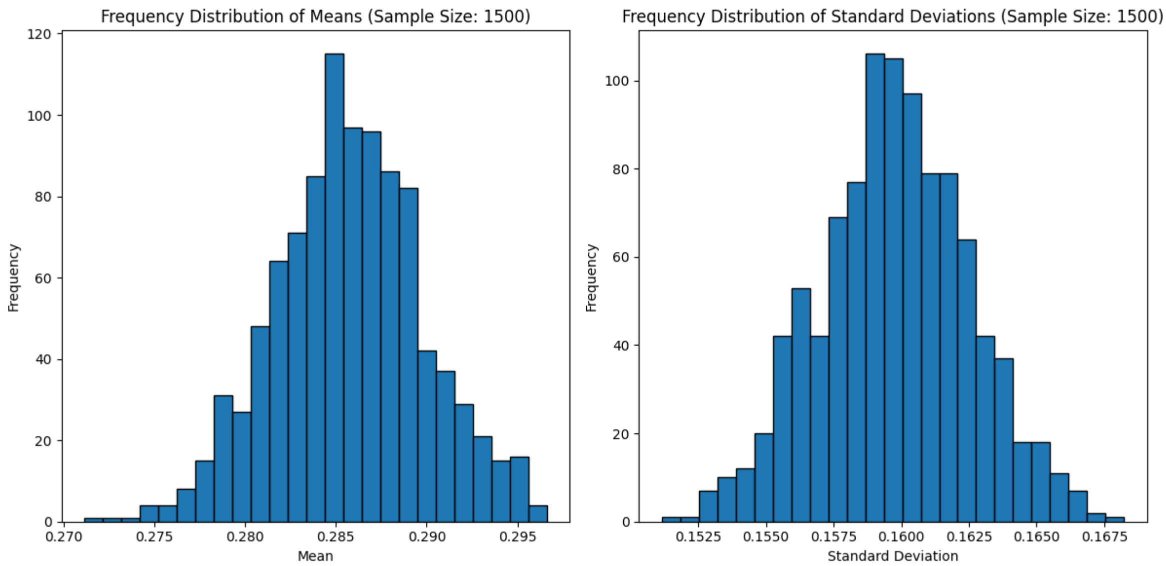





```
Sample Size: 500
Mean: 0.2853800530531955
Standard Deviation: 0.15953418491427768
```




```
Sample Size: 1500
Mean: 0.2857110214351718
Standard Deviation: 0.1597540977198217
```



Observations and Inferences

What do you observe now that is different from Question 2, w.r.t to the central limit theorem.

Unlike the normal distribution, the Beta distribution is not symmetric and may have a skewed shape. The Beta distribution has a limited range (between 0 and 1) compared to the unbounded range of the normal distribution. Despite the differences between normal and beta distribution, the central limit theorem states that, if one repeatedly draws sufficiently large samples from this distribution and calculates an mean for every sample drawn, then the sample means will follow a Normal distribution.

Do you need a larger or smaller sample size now so that your sample estimate of the population mean is accurate?

The convergence to normality might be slower, as the distribution might not be as well behaved as the normal distribution. With a non-normal population like the Beta distribution, a larger sample size may be needed to obtain an accurate estimate of the population mean.