PS04

April 28, 2023

1 PS 04

1.1 Name: Xinyu Chang

---

[5]:
```python
# import the packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

1.1.1 1 Explore Central Limit Theorem

Q1 What is Random Variable (RV)? What makes X a RV? A Random Variable (RV) is a central concept that connects probability theory to statistics, enabling the use of mathematical tools to analyze random processes. It is essential to remember that a random variable is not random and not a variable.Essentially, a random variable (RV) is a function that allocates numeric values to the outcomes of a random process. It allows us to represent uncertain experimental outcomes using real numbers. Depending on the type of values they may assume, random variables may be discrete or continuous. **It is a function that assigns a number to each event in the sample space S of a random phenomenon.**

When we conduct an experiment, like flipping a coin, we often use a random variable X to convert the realized outcome into a number, such as 1 or 0 in the case of a coin toss. These numerical values are the possible observed values or realizations of the random variable. While random variables are traditionally denoted by uppercase Latin letters like X or Y, their observed values (realizations) are denoted with corresponding lowercase letters, such as x and y.

For example, the random variable X assigns a numerical value to the outcome of flipping a fair coin (heads or tails). The function X is defined as:

X(E) = { -1 if E = T, 1 if E = H }

Here, E represents the event, T is tails, and H is heads. So, X is a random variable because it maps the outcome of flipping a fair coin to numerical values (-1 and 1) with respective probabilities (0.5 and 0.5).

Q2 Calculate the expected value and variance of this random variable. Explain what is the difference between expected value and the sample mean. The formula for calculating the expected value (E(X)) of a discrete random variable is: $E(X) = \sum_i x_i * P(x_i)$:

```
[6]: expected_value_1 = 1 * 0.5 + (-1) * 0.5
     expected_value_1
```

```
[6]: 0.0
```

The expected value of the random variable is 0.

The variance (Var(X)) is a measure of how much the values of a random variable deviate from its expected value. Using the formula Var(X) = E(X^2) - (E(X))^2:

```
[7]: variance_1 = (0.5 * 1 ** 2 + 0.5 * (-1) ** 2) - expected_value_1 ** 2
     variance_1
```

```
[7]: 1.0
```

The variance of the random variable is 1.

Sample mean is the average of the random realizations in a given sample. To calculate the sample mean, we add up all the outcomes in the random realizations and divide the sum by the total number of realizations. The sample mean is an empirical value obtained from data and takes the sample size into account.Expected value, on the other hand, is a property of the random variable, which represents the random process we are analyzing. It is a theoretical value that takes probability into account, representing the "center" of a probability distribution. The expected value is not random and is not related to the sample.The main difference between the sample mean and the expected value is that the sample mean is an empirical value derived from data and considers the sample size, while the expected value is a theoretical value that considers probabilities and is a property of the random variable itself.

Q3 Choose your number of repetitions R.

```
[8]: R = 1000
```

```
[9]: # Create the function for later computing
     def random_realizations(S, R):
         X = []
         for p in range(R):
             x = np.random.randint(0, 2, size=S) * 2 - 1
             X.append(np.mean(x))
         return X

     def plot_histogram(X, S, R):
         plt.hist(X, bins=100, edgecolor='black')
         plt.xlabel('Mean Value')
         plt.ylabel('Frequency')
         plt.title(f'Histogram of R={R} Random Realizations of X with Sample Size␣
      ↪S={S}')
         plt.show()

     def calculate_sample_stats(X):
```

```
    mean = np.mean(X)
    var = np.var(X)
    return mean, var

def calculate_theoretical_stats(S):
    E_X = 0
    var_X = 1
    E_mean = E_X
    var_mean = var_X / S
    return E_mean, var_mean
```

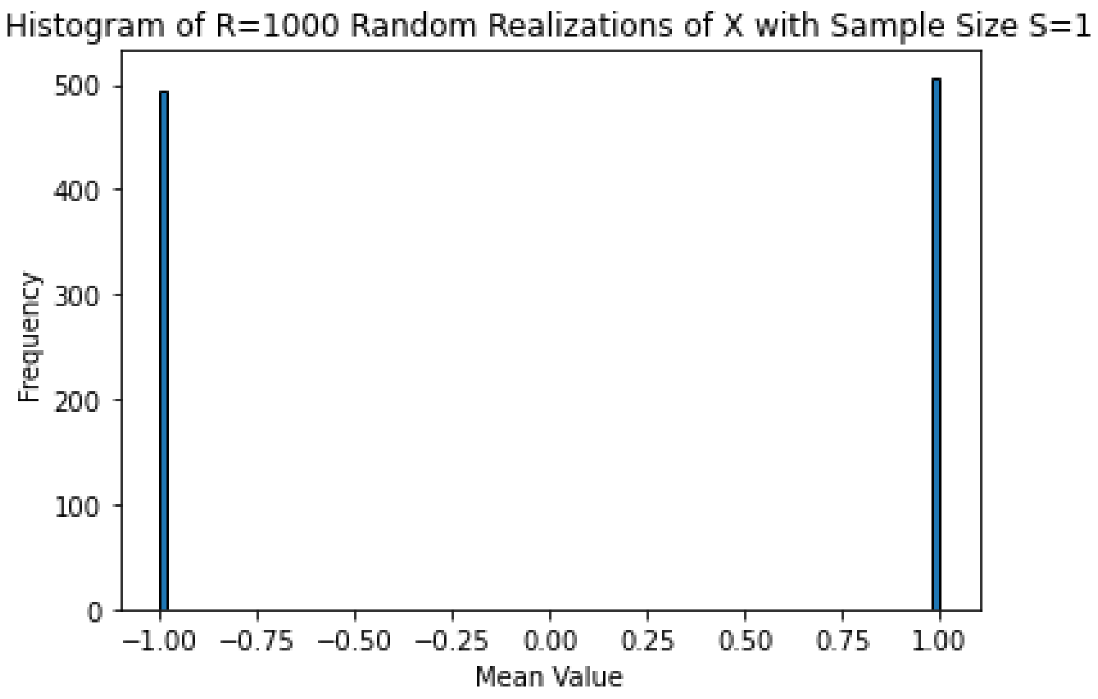
Q4 Create a vector of R random realizations of X. Make a histogram of those. Comment the shape of the histogram.

Q5 Compute and report mean and variance of the sample you created (just use np.mean and np.var)

[10]:
```
S = 1
X = random_realizations(S, R)
plot_histogram(X, S, R)

sample_mean, sample_var = calculate_sample_stats(X)
theoretical_mean, theoretical_var = calculate_theoretical_stats(S)

print(f'Sample Size S = {S}')
print(f'Sample Mean: {sample_mean}, Theoretical Mean: {theoretical_mean}')
print(f'Sample Variance: {sample_var}, Theoretical Variance: {theoretical_var}')
```



```
Sample Size S = 1
Sample Mean: 0.014, Theoretical Mean: 0
Sample Variance: 0.999804, Theoretical Variance: 1.0
```

For sample size S=1:
The histogram shows two peaks at -1 and 1, as there are only two possible values for the mean of a single realization. There will be no central tendency in this case.The sample mean and sample variance(0.014 and 0.9998) are close to the theoretical values computed in question 2 (E(X) = 0 and Var(X) = 1).

Q6 Now create R pairs of random realizations of X (i.e. sample size S = 2).

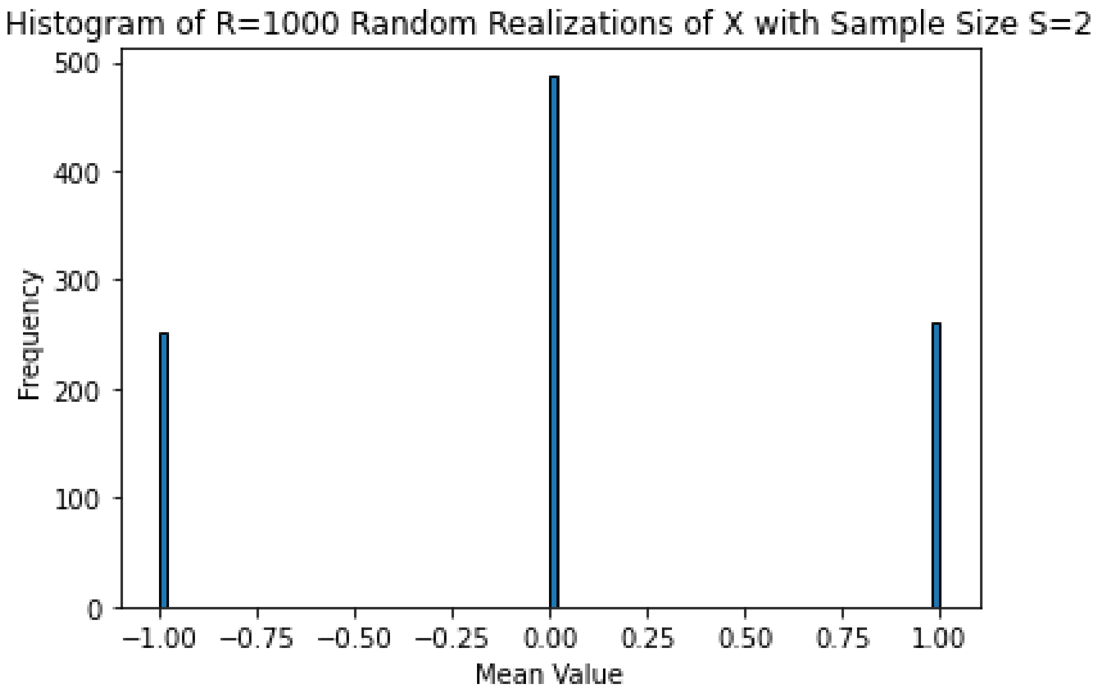
Q7 Compute and report mean of the R pair means, and variance of the means

Q8 Compute the expected value and variance of the pair means, i.e. the theoretical concepts.

[36]:
```
S = 2
X = random_realizations(S, R)
plot_histogram(X, S, R)

sample_mean, sample_var = calculate_sample_stats(X)
theoretical_mean, theoretical_var = calculate_theoretical_stats(S)

print(f'Sample Size S = {S}')
print(f'Sample Mean: {sample_mean}, Theoretical Mean: {theoretical_mean}')
print(f'Sample Variance: {sample_var}, Theoretical Variance: {theoretical_var}')
```



```
Sample Size S = 2
Sample Mean: 0.01, Theoretical Mean: 0
Sample Variance: 0.5119, Theoretical Variance: 0.5
```

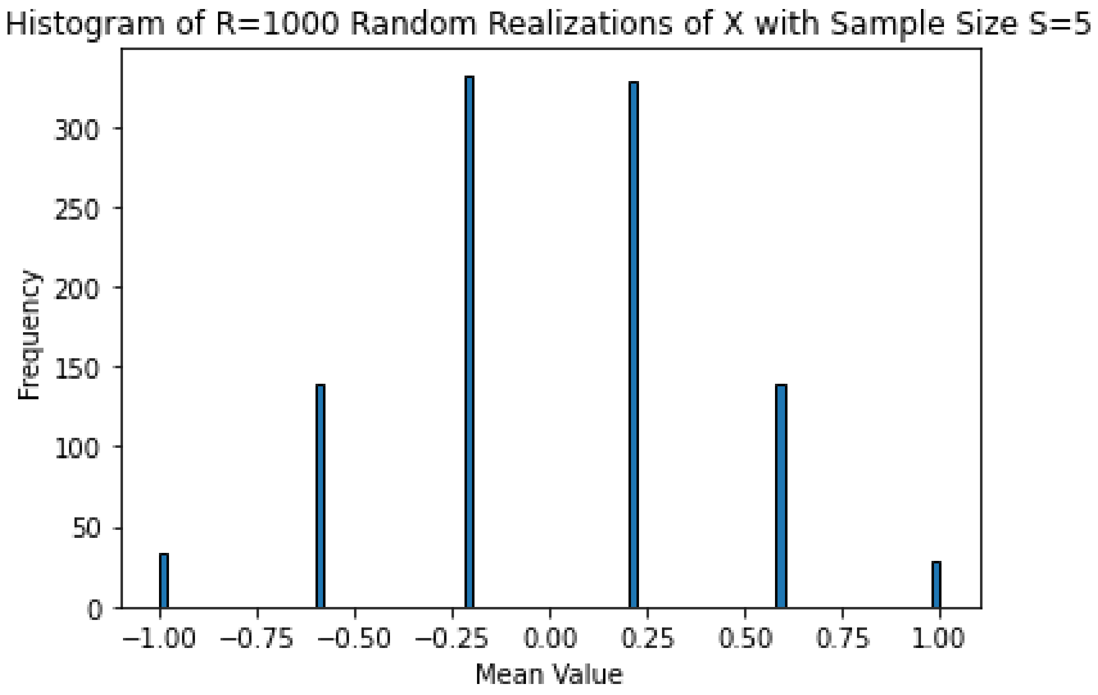
The resulting histogram should show a symmetric distribution centered around the mean value of 0. Compared with the last question, there is a new column for the mean value of 0 which has the highest frequency with almost 500. Now the graph shows three columns instead of two columns. Since in this question, we introduce one more coin, we can get one with head and one with tail. The sample mean and sample variance(0.01, 0.5119) are close to the theoretical values(0 and 0.5).The variance of a pair mean is 1/2 of what I got above S = 1) as for pairs S = 2.

Q9 Now instead of pairs of random numbers, repeat this with 5-tuples of random numbers

[37]:
```
S = 5
X = random_realizations(S, R)
plot_histogram(X, S, R)

sample_mean, sample_var = calculate_sample_stats(X)
theoretical_mean, theoretical_var = calculate_theoretical_stats(S)

print(f'Sample Size S = {S}')
print(f'Sample Mean: {sample_mean}, Theoretical Mean: {theoretical_mean}')
print(f'Sample Variance: {sample_var}, Theoretical Variance: {theoretical_var}')
```



```
Sample Size S = 5
Sample Mean: -0.0048000000000000004, Theoretical Mean: 0
Sample Variance: 0.18845696, Theoretical Variance: 0.2
```

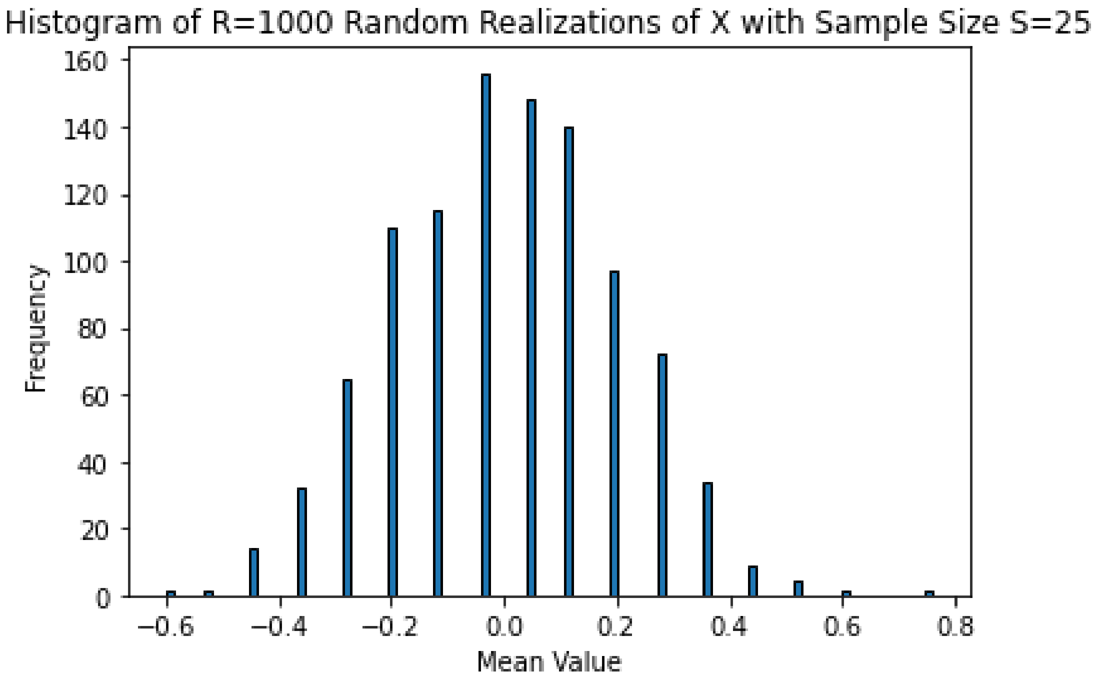
For sample size S=5:
The sample mean(-0.0048) is getting closer to the theoretical mean(0) and the sample variance(0.18846) is also getting closer to the thereotical variance(0.2). The variance also become smaller than the previous variance. The histogram will resemble a more bell-like shape, with the peak at -0.25 and the distribution spreading out symmetrically around the central tendency. The distribution will be narrower than the S=2 case, as the larger sample size reduces variability.

Q10 Repeat with 25-tuples...

[38]:
```
S = 25
X = random_realizations(S, R)
plot_histogram(X, S, R)

sample_mean, sample_var = calculate_sample_stats(X)
theoretical_mean, theoretical_var = calculate_theoretical_stats(S)

print(f'Sample Size S = {S}')
print(f'Sample Mean: {sample_mean}, Theoretical Mean: {theoretical_mean}')
print(f'Sample Variance: {sample_var}, Theoretical Variance: {theoretical_var}')
```



```
Sample Size S = 25
Sample Mean: 0.0028800000000000006, Theoretical Mean: 0
Sample Variance: 0.0388269056, Theoretical Variance: 0.04
```

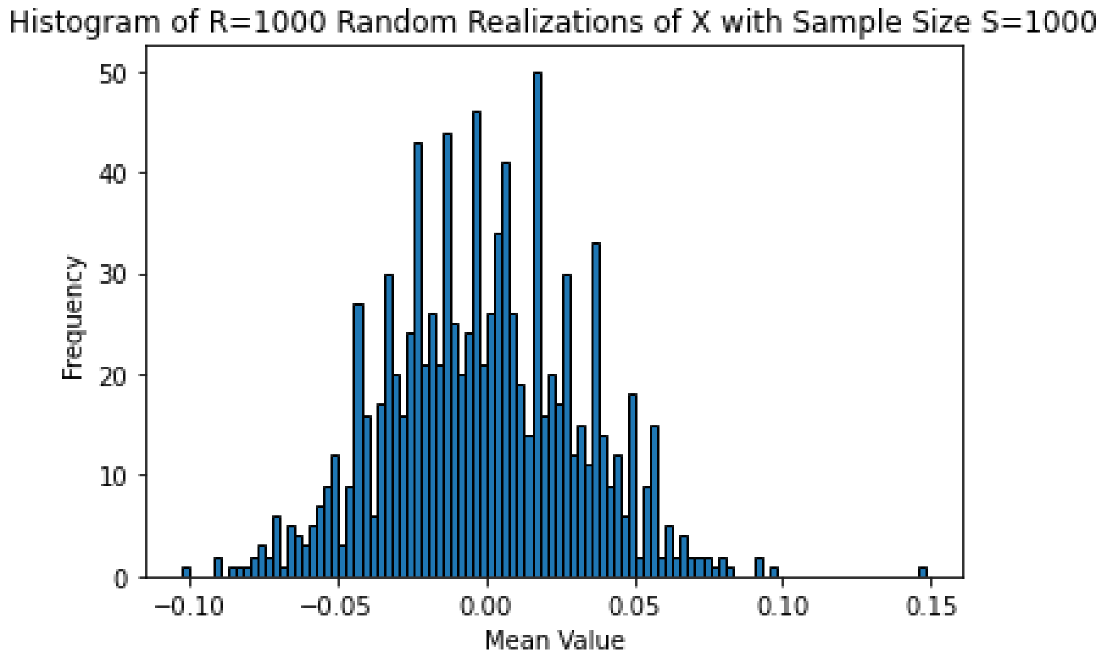
For sample size S=25:
The histogram will be even more bell-shaped and narrower, centered around 0. As the sample size increases, the distribution of the means becomes more concentrated around the theoretical mean, which is about 0 in this case.The sample variance become more smaller than the former cases.The sample mean and sample variance(0.0028, 0.0388) are more closer to the theoretical values(0, 0.04).

Q11 With 1000-tuples

[39]:
```
S = 1000
X = random_realizations(S, R)
plot_histogram(X, S, R)

sample_mean, sample_var = calculate_sample_stats(X)
theoretical_mean, theoretical_var = calculate_theoretical_stats(S)

print(f'Sample Size S = {S}')
print(f'Sample Mean: {sample_mean}, Theoretical Mean: {theoretical_mean}')
print(f'Sample Variance: {sample_var}, Theoretical Variance: {theoretical_var}')
```



```
Sample Size S = 1000
Sample Mean: -0.0009199999999999999, Theoretical Mean: 0
Sample Variance: 0.0010460576000000002, Theoretical Variance: 0.001
```

For a sample size S = 1000:
The histogram distribution of the means of random realizations of the binary variable X with values {-1, 1} will be even more concentrated around the theoretical mean (which is 0 in this case) compared to the distributions for smaller sample sizes. The histogram will resemble a bell-shaped curve (a normal distribution).The sample mean is much closer to theoretical mean with the value of 0.The sample variance tend to be much closer with the theoretical variance with the value of 0.001.

Q12 Comment on the tuple size, and how the shape of the histogram changes when the tuple size increases. As the tuple size (sample size, S) increases, the shape of the histogram changes significantly. When the sample size is small, the histogram displays distinct peaks at specific values, with no clear central tendency and a non-symmetric distribution. However, as the sample size increases, the histogram becomes more symmetric and concentrated around the theoretical mean, eventually resembling a normal distribution. As the sample size S increases, each individual observation has a diminishing effect on the sample mean. In other words, the mean of the sample becomes more stable and less susceptible to random fluctuations in the data. This implies that the variance of the sample mean decreases as sample size increases, as the sources of variability in the sample mean diminish. Also, the increasing sample size leads to a wider range of mean values for the realizations, resulting in a smoother and more continuous distribution.

Q13 Explain why do the histograms resemble normal distribution as S grows.Why did two equal peaks turn into a " "-shaped histogram? The histograms resemble a normal distribution as S grows due to the Central Limit Theorem(The means of a sample of random numbers tend to be normally distributed if the sample gets large.), which states that the distribution of the means of a large number of independent and identically distributed random variables converges to a normal distribution as the sample size increases.The histogram now has three columns instead of two, reflecting the three possible mean values: -1, 0, and 1. The column at 0 corresponds to the cases where one coin shows heads and the other shows tails (or vice versa). This probability is higher than the probabilities of obtaining a mean of -1 (two tails) or 1 (two heads). Therefore, the height of the middle column (0) is greater than the height of the other two columns (-1 and 1).

Q14 Explain what is the difference between R and S. How do changing these values affect the histograms? R represent the number of replications or trials in an experiment, while S could represent the sample size (number of observations) in each trial. Changing the values of R and S can have different effects on the histograms:

Changing R (number of replications or trials):

As we increase the number of replications, the histograms will provide a more accurate representation of the underlying probability distribution. With more replications, the histograms tend to converge to the true distribution, as the law of large numbers comes into play. Conversely, if we decrease the number of replications, the histograms may not accurately represent the true distribution, as the sample might not capture all the possible variations in the random process.In conclusion, increasing R leads to a smoother histogram because the distribution of the averages becomes a better approximation of the underlying distribution, as described by the Central Limit Theorem.

Changing S (sample size):

Increasing the sample size generally results in a more accurate estimation of the population parameters (e.g., mean, variance). As the sample size increases, the Central Limit Theorem (CLT) comes into play, and the distribution of the sample means tends to be normally distributed, regardless of the shape of the original distribution. This allows for more precise inference and more reliable confidence intervals.

1.1.2 2 Is povery in Azraq refugee camp falling?

1.1.3 2.1 Background

Q1 What was the abject poverty in Azraq camp in Q1 and Q2 2022 (when including all assistance)? Lets call these variables p1 and p2. p1 is 66%, and p2 is 51%.The abject poverty in Azraq camp in Q1 2022 including all assistance is 66%. The abject poverty in Azraq camp in Q2 2022 including all assistance is 51%.

Q2 How many households were surveyed in the camp? In the camp, there are 650 households have been surveyed. 325 are from the Azraq camp and 325 are from the Zaatari camp. Considering that we are analyzing the Azraq faction, the sample size is 325 individuals.

1.1.4 2.2 Simulations

Q1 Create a random sample using the correct values of S and p2 you found in 2.1 above.

```
[40]: S = 325
      p1 = 0.66
      p2 = 0.51
```

```
[41]: np.random.seed(42)
      sample_2 = np.random.binomial(1, p2, size=S)
      sample_2[:10]   # just for check
```

```
[41]: array([1, 0, 0, 0, 1, 1, 1, 0, 0, 0])
```

Q2 Compute the sample mean and compare it with p1 and p2 above. How close is it to these figures?

```
[42]: mean_2 = np.mean(sample_2)
      print("Sample mean for p2 is:", mean_2)
```

```
Sample mean for p2 is: 0.48923076923076925
```

The sample means(0.489) is quite close to the true probabilities p2(0.51). However, the sample mean(0.48) is quite far away from the p1(0.66).

Q3 Pick your number of replications R

```
[43]: R = 1000
```

Q4 Repeat the points 1 and 2 for R times: create the sample, compute the average, but also store the average in an array.

```
[44]: sample_means_2 = []
      for i in range(R):
          sample_2 = np.random.binomial(1, p2, size=S)
          sample_means_2.append(np.mean(sample_2))
```

Q5 What is the average of the averages? Which probability from 1 does it resemble? Why?

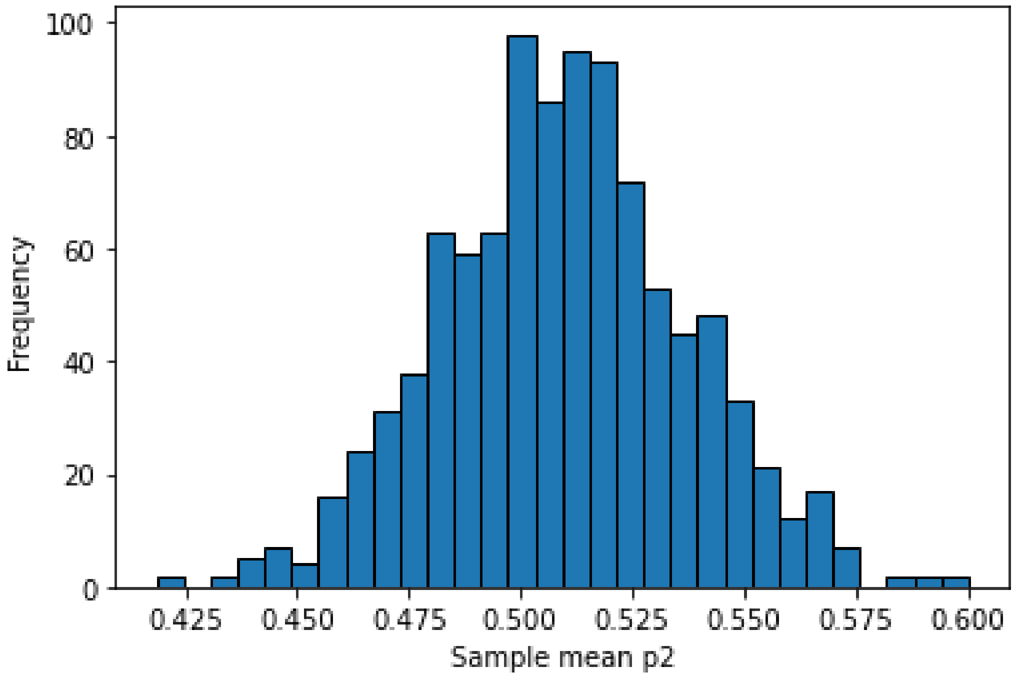
```
[45]: np.random.seed(42)
      avg_of_avgs_2 = np.mean(sample_means_2)
      avg_of_avgs_2
```

```
[45]: 0.5093138461538461
```

Based on the ouput, the average of the averages for p2 (0.509) is very close to the original probability p2 (0.51) but it is not close to p1(0.66).The reason is the average of the averages values resemble its respective probabilities is due to the Law of Large Numbers. As we repeat the sampling process for a large number of replications (R), the average of the sample means converges to the true probability. This means that the more replications we perform, the closer our estimated probabilities (average of the averages) will be to the actual probabilities p2.

Q6 Plot a histogram of the averages.

[46]:
```python
plt.hist(sample_means_2, bins=30, edgecolor='black')
plt.xlabel("Sample mean p2")
plt.ylabel("Frequency")
plt.show()
```



The histogram of the averages should resemble a normal distribution, as the sample means of a large number of Bernoulli-distributed samples tend to be normally distributed according to the Central Limit Theorem. From the graph, the largest frequency is about 0.51 and the smallest frequency is about 0.43 and 0.58.

Q7 Finally, compute 2.5th and 97.5th percentile and the 95% confidence intervals.Does the Q1 poverty value fall into this interval?

[47]:
```python
low_percentile_2 = np.percentile(sample_means_2, 2.5)
high_percentile_2 = np.percentile(sample_means_2, 97.5)
CI_2 = np.percentile(sample_means_2, [2.5, 97.5])
print("2.5th percentile for p2 is:", low_percentile_2)
print("97.5th percentile for p2 is:", high_percentile_2)
print("95% Confidence interval for p2 is:", CI_2)
```

```
2.5th percentile for p2 is: 0.4553846153846154
97.5th percentile for p2 is: 0.5661538461538461
95% Confidence interval for p2 is: [0.45538462 0.56615385]
```

For the 2022-Q2 survey (p2), its 95% confidence interval is [0.45538462 0.56615385]. The 2.5th percentile is 0.455 and the 97.5% percentile is 0.566. Since the 2022-Q1 result (p1 = 0.66) is not within the range [0.45538462 0.56615385], so we can say the 2022-Q1 poverty value does not fall into the 95% confidence interval of the 2022-Q2 survey.

1.1.5 2.3 Theoretical CI

Q1 Compute variance of your sample of X.

[48]:
```
sample_var_2 = p2 * (1 - p2)
print("Variance for p2 is:", sample_var_2)
```

```
Variance for p2 is: 0.2499
```

The variance for p2 is 0.2499.

Q2 But this was variance of X (or sample variance if that was what you computed).What we need is variance of sample mean. What does CLT tell about relationship b/w sample variance and variance of the sample mean?

[49]:
```
sample_mean_var_2 = sample_var_2 / S
print("Variance of the sample mean for p2 is:", sample_mean_var_2)
```

```
Variance of the sample mean for p2 is: 0.0007689230769230 77
```

The variance of sample mean for p2 is 0.0007689. The Central Limit Theorem (CLT) establishes a relationship between the sample variance and the variance of the sample mean. According to the CLT, when we have a large number of independent and identically distributed random variables, their means tend to be normally distributed.

The relationship between the sample variance (Var(X)) and the variance of the sample mean (Var( )) is given by:

Var( ) = Var(X) / n

where n is the sample size.

In other words, the variance of the sample mean is equal to the variance of the individual random variables (sample variance) divided by the sample size. This relationship holds true for large sample sizes and is a key aspect of the Central Limit Theorem.

Q3 Compute the standard deviation of sample mean using CLT.

[25]:
```
sample_mean_std_2 = np.sqrt(sample_mean_var_2)
print("Standard deviation of the sample mean for p2 is:", sample_mean_std_2)
```

```
Standard deviation of the sample mean for p2 is: 0.02772946225448804
```

By using the CLT, the p2 standard deviation of sample mean using CLT is 0.027729.

Q4 Compare the standard deviation you got here with the standard deviation of the sample of averages you computed in 2.2.4.

[26]:
```
sample_std_2 = np.std(sample_means_2)
print("Sample standard deviation:", sample_std_2)
print("Theoretical standard deviation:", sample_mean_std_2)
```

```
Sample standard deviation: 0.028046836735852655
Theoretical standard deviation: 0.02772946225448804
```

The sample standard deviation of the sample mean for p2 is 0.028.The theoretical standard deviation of the sample mean for p2 calculated using the Central Limit Theorem (CLT) is 0.0277.The sample and theoretical standard deviations are similar in this question.The results show that the sample standard deviations of the sample means are very close to the theoretical standard deviations calculated using the CLT. This confirms that the CLT provides an accurate estimation of the standard deviation of the sample mean, especially when the sample size is large and the number of simulations (R) is also large.

Q5 Use this standard deviation to compute the confidence interval.

[27]:
```
CI_p2 = (mean_2 - 1.96 * sample_mean_std_2, mean_2 + 1.96 * sample_mean_std_2)
CI_p2
```

[27]:
```
(0.4348810232119727, 0.5435805152495659)
```

In this method, 2022-Q1 result (p1 = 0.66) does not fall within this 2022-Q2 confidence interval(0.4348810232119727, 0.5435805152495659), which is the same conclusion we drawn from question 2.2.7. In both methods, the 95% confidence interval for p2 does not includes the true value of p1 (0.66).

1.1.6 2.4 Challenge

[28]:
```
# import the packages
import time
from scipy import stats
```

[29]:
```
R2 = 1000000
```

[30]:
```
def simulation_without_storage():
    counter = 0
    for i in range(R2):
        sample_2 = np.random.binomial(1, p2, size=S)
        _ = np.mean(sample_2)
        counter += 1
    return counter
```

[31]:
```
start_time = time.time()
iterations = simulation_without_storage()
elapsed_time = time.time() - start_time

print(f"Elapsed time: {elapsed_time} seconds")
print(f"Iterations: {iterations}")
```

```
Elapsed time: 23.58946990966797 seconds
Iterations: 1000000
```

[32]:
```
simulations_needed = 10**12
time_for_needed_simulations = (elapsed_time / R) * simulations_needed
print(f"Estimated time for {simulations_needed:,} simulations:␣
↪{time_for_needed_simulations} seconds")
```

```
Estimated time for 1,000,000,000,000 simulations: 23589469909.66797 seconds
```

Q2 Find the t-value.

[63]:
```
t = abs(p2 - p1) / sample_std_2
t
```

[63]:
```
5.348196711547615
```

The t-value for the difference assuming the Q1 value is 0.66 with no error is 5.3482.

Q3 Second, what is the probability to receive such t-values?

[65]:
```
norm = stats.norm()
probability = 1 - norm.cdf(t)
probability
```

[65]:
```
4.441744239080947e-08
```

The probability to receive such t-values is 4.441744239080947e-08.

norm.cdf(-1.96) returns the probability of observing a standard normal random variable less than or equal to -1.96. Since the standard normal distribution is symmetric about 0, this also gives the probability of observing a standard normal random variable greater than or equal to 1.96. From the computation, norm.cdf(-1.96) = 0.024997895148220435 and norm.cdf(1.96) = 0.9750021048517795, represent the areas under the standard normal distribution curve to the left of -1.96 and to the left of 1.96, respectively. These probabilities can be interpreted as the proportion of values that fall below these thresholds.To obtain the values below 2.5% and above 97.5%, we need to find the thresholds that correspond to these probabilities. Since the standard normal distribution is symmetric, the thresholds will be equidistant from the mean.

Q4 How many iterations do you need?

[67]:
```
iterations_needed = 3 / probability
print("Iterations needed:", iterations_needed)
```

```
Iterations needed: 67541034.29919094
```

There are 67541034.29919094 iterations we need.

Q5 Based on the timings you did above, how many years do you have to run the simulations? iteration_needs = 67541034.32195295 / 1,000,000 = 67.54103432195295 million Elapsed time for 1 million iteration (R = 1000000) is 23.296139001846313

time = Elapsed time * iterations needed = 23.296139001846313 * 67.5 = 1572.4893826246262
hours = time / (60 * 60) = 1572.4893826246262 / 3600 = 0.4368 hour
years = time / (60 * 60 * 24 * 365.25) = 1572.4893826246262 / (60 * 60 * 24 * 365) = 0.000049863 year

The simulation would need approximately 0.000049863 years. In other words, the simulations would need to run for about 0.4368 hours (assuming 24-hour days) to complete the required iterations.It is possible to complete the simulations by the assignment deadline(April 30).As the total time required for the simulations is only 0.4368 years, and my grandfather's age is more than that (which is very likely), the simulations would have been completed a long time ago if started when my grandfather was born.Similarly, if the first Seattle inhabitants had started the simulations 10,000 years ago, the simulations would have been completed almost instantly in comparison to that timescale, as the total time required is only 0.4368 years.

1.1.7 I spent alomost 7.5 hours in this PS.