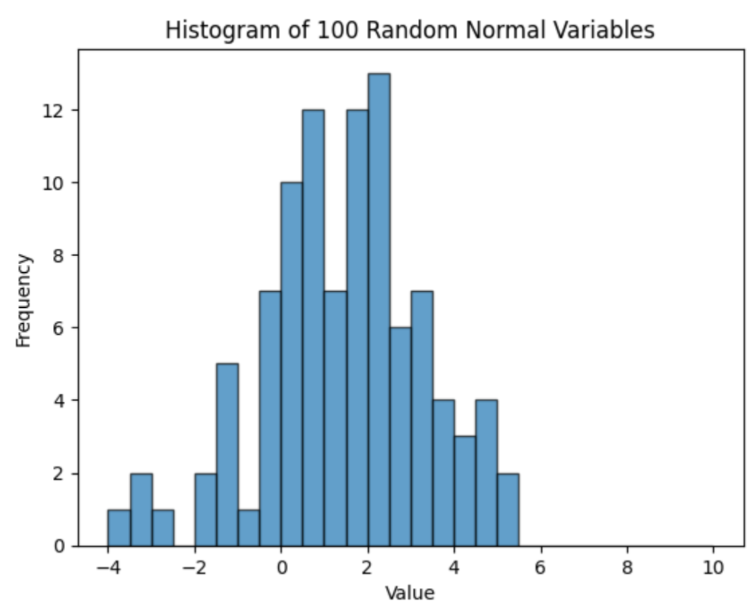
Computer Project 1

MATH 408

Assia Khader

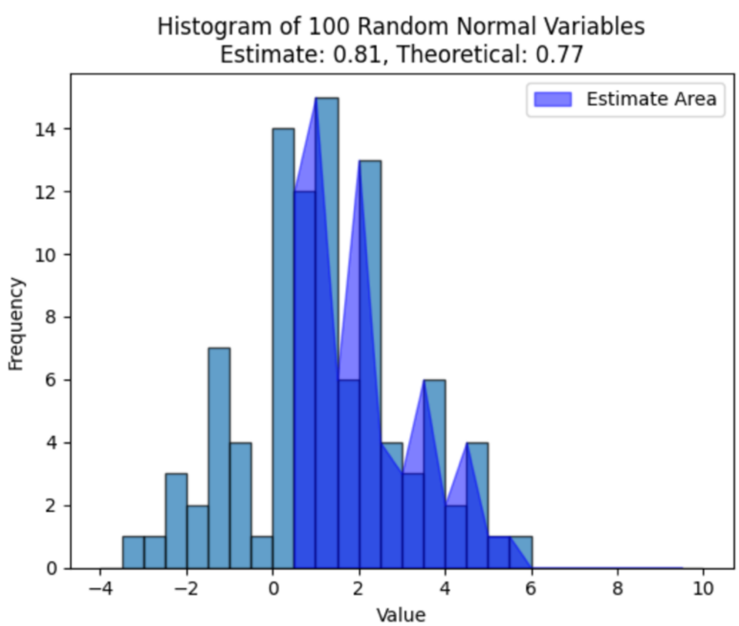
1)

```python
import numpy as np
import matplotlib.pyplot as plt

# Set parameters
mean = 1.5
std_dev = 2
num_samples = 100

# Simulate 100 random normal variables
data = np.random.normal(mean, std_dev, num_samples)

# Set the range and interval for the histogram
bin_edges = np.arange(-4, 10.5, 0.5)  # Class intervals of length 0.5 from -4 to 10

# Plot the histogram
plt.hist(data, bins=bin_edges, edgecolor='black', alpha=0.7)

# Set labels and title
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of 100 Random Normal Variables')

# Show the plot
plt.show()
```



2)

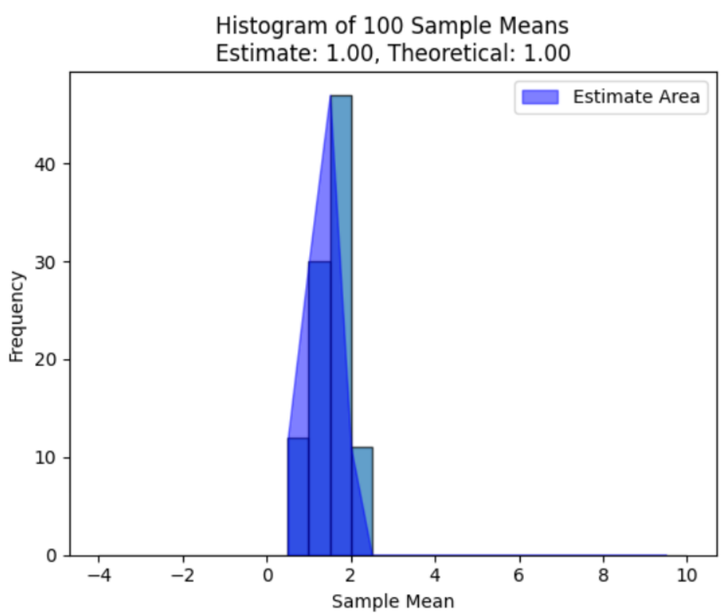
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Set parameters
mean = 1.5
std_dev = 2
num_samples = 100

# Simulate 100 random normal variables
data = np.random.normal(mean, std_dev, num_samples)

# Count how many variables are bigger than 0
num_above_zero = np.sum(data > 0)

# Estimate the probability
estimate_prob = num_above_zero / num_samples

# Theoretical probability using CDF for normal distribution
theoretical_prob = 1 - norm.cdf(0, mean, std_dev)

# Set the range and interval for the histogram
bin_edges = np.arange(-4, 10.5, 0.5)

# Plot the histogram
plt.hist(data, bins=bin_edges, edgecolor='black', alpha=0.7)

# Mark the area for the estimate
plt.fill_between(bin_edges[:-1], 0, np.histogram(data, bins=bin_edges)[0], where=(bin_edges[:-1] > 0), color='blue', alpha=0.5, label="Estimate Area")

# Set labels and title
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title(f'Histogram of 100 Random Normal Variables\nEstimate: {estimate_prob:.2f}, Theoretical: {theoretical_prob:.2f}')

# Show the plot
plt.legend()
plt.show()

# Output the results
print(f'Number of values greater than 0: {num_above_zero}')
print(f'Estimated probability: {estimate_prob:.2f}')
print(f'Theoretical probability: {theoretical_prob:.2f}')
```




```
Number of values greater than 0: 81
Estimated probability: 0.81
Theoretical probability: 0.77
```

3)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Set parameters
mean = 1.5
std_dev = 2
num_samples = 100
sample_size = 20

# Simulate 100 sample means by averaging 20 random normal variables each time
sample_means = np.array([np.mean(np.random.normal(mean, std_dev, sample_size)) for _ in range(num_samples)])

# Count how many sample means are bigger than 0
num_above_zero = np.sum(sample_means > 0)

# Estimate the probability
estimate_prob = num_above_zero / num_samples

# Theoretical probability using Central Limit Theorem
sample_std_dev = std_dev / np.sqrt(sample_size)  # Standard deviation of the sample mean
theoretical_prob = 1 - norm.cdf(0, mean, sample_std_dev)

# Set the range and interval for the histogram
bin_edges = np.arange(-4, 10.5, 0.5)

# Plot the histogram
plt.hist(sample_means, bins=bin_edges, edgecolor='black', alpha=0.7)

# Mark the area for the estimate (sample means > 0)
plt.fill_between(bin_edges[:-1], 0, np.histogram(sample_means, bins=bin_edges)[0], where=(bin_edges[:-1] > 0), color='blue', alpha=0.5, label="Estimate Area")

# Set labels and title
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.title(f'Histogram of 100 Sample Means\nEstimate: {estimate_prob:.2f}, Theoretical: {theoretical_prob:.2f}')

# Show the plot
plt.legend()
plt.show()

# Output the results
print(f'Number of sample means greater than 0: {num_above_zero}')
print(f'Estimated probability: {estimate_prob:.2f}')
print(f'Theoretical probability: {theoretical_prob:.2f}')
```



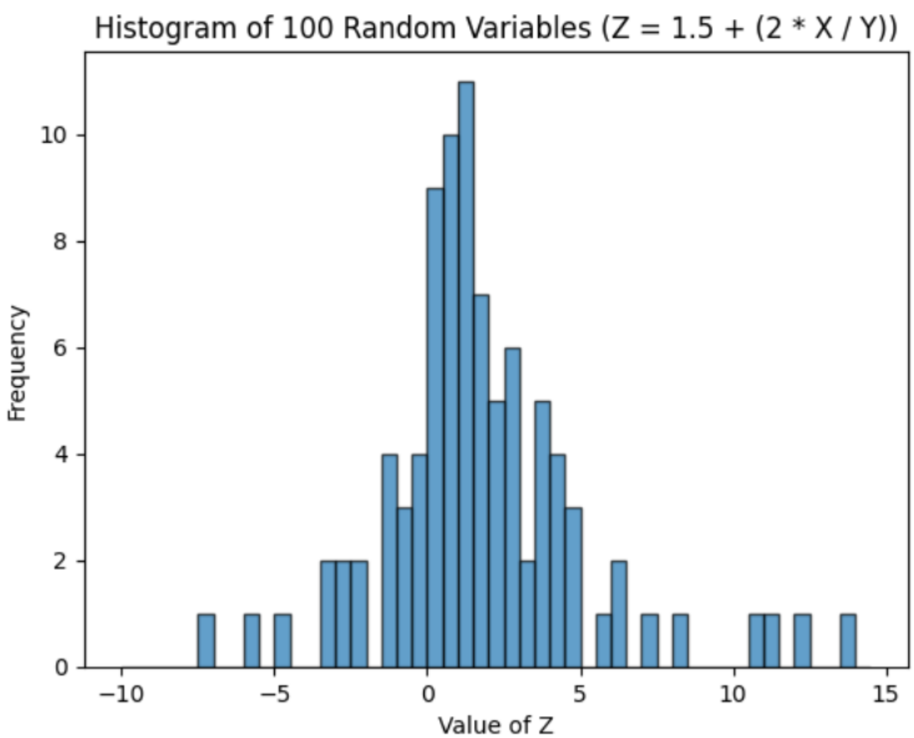

```
Number of sample means greater than 0: 100
Estimated probability: 1.00
Theoretical probability: 1.00
```

4) The two histograms—one for the 100 random normal variables and the other for the 100 sample means—reveal distinct characteristics that illustrate the effect of averaging. The histogram for the sample means is much narrower and more concentrated around the mean (1.5), as predicted by the Central Limit Theorem (CLT). The CLT asserts that, regardless of the original distribution, the distribution of sample means will approach a normal distribution as the sample size increases, with a reduced spread compared to the original data. This narrower spread results in a higher estimated probability of obtaining a value greater than 0, which is 1.00 for the sample means, indicating that the averaging process has created a more predictable distribution.

In contrast, the histogram for the random normal variables shows a broader spread, reflecting the higher variability inherent in individual values. This is expected since the individual random variables are drawn from a normal distribution with a mean of 1.5 and a standard deviation of 2. As a result, the estimated probability for the random normal variables being greater than 0 is lower, at 0.81. Both histograms show that the theoretical probability of a value being above 0 for the random normal variables and the sample means is 0.77. However, the sample means' histogram more closely approximates this theoretical value, confirming the more predictable nature of averaging multiple variables and the effect of the CLT in reducing variability.

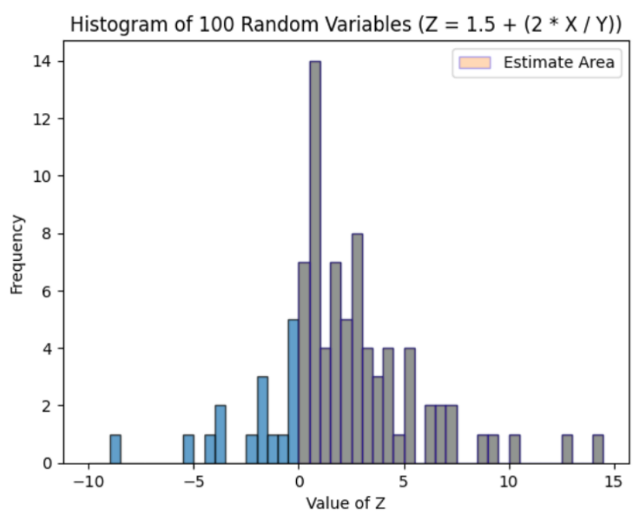
5)

```python
import numpy as np
import matplotlib.pyplot as plt

# Set parameters
num_samples = 100

# Generate independent standard normal random variables X and Y
X = np.random.normal(0, 1, num_samples)
Y = np.random.normal(0, 1, num_samples)

# Calculate Z = 1.5 + (2 * X / Y)
Z = 1.5 + (2 * X / Y)

# Plot the histogram
plt.hist(Z, bins=np.arange(-10, 15, 0.5), edgecolor='black', alpha=0.7)

# Set labels and title
plt.xlabel('Value of Z')
plt.ylabel('Frequency')
plt.title('Histogram of 100 Random Variables (Z = 1.5 + (2 * X / Y))')

# Show the plot
plt.show()

# Print the first few values of Z
print(Z[:10])
```



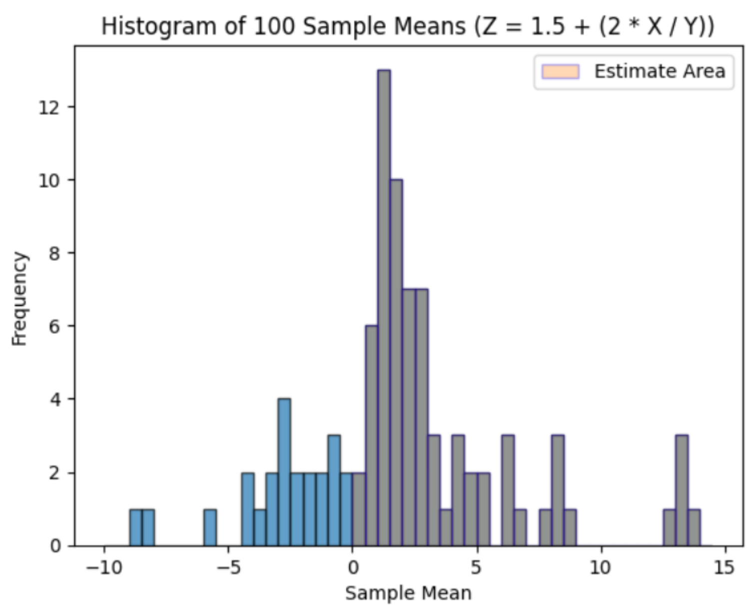

```
[-1.25159765 -2.84345901  1.88809065  0.53646633  2.48542474  3.49966751
  0.42926358  1.30340754  2.34744947  1.44666828]
```

```python
import numpy as np
import matplotlib.pyplot as plt

# Set parameters
num_samples = 100

# Generate independent standard normal random variables X and Y
X = np.random.normal(0, 1, num_samples)
Y = np.random.normal(0, 1, num_samples)

# Calculate Z = 1.5 + (2 * X / Y)
Z = 1.5 + (2 * X / Y)

# Estimate the probability that Z > 0
probability_estimate = np.sum(Z > 0) / num_samples

# Theoretical probability for a Cauchy distribution with location 1.5 and scale 2
theoretical_probability = 0.5  # As explained earlier

# Plot the histogram
plt.hist(Z, bins=np.arange(-10, 15, 0.5), edgecolor='black', alpha=0.7)

# Add the shaded area indicating the estimate
plt.hist(Z[Z > 0], bins=np.arange(0, 15, 0.5), edgecolor='blue', alpha=0.3, label='Estimate Area')

# Set labels and title
plt.xlabel('Value of Z')
plt.ylabel('Frequency')
plt.title(f'Histogram of 100 Random Variables (Z = 1.5 + (2 * X / Y))')

# Show the plot with the legend
plt.legend()
plt.show()

# Print the estimate and theoretical probability
print(f"Estimated probability (Z > 0): {probability_estimate:.2f}")
print(f"Theoretical probability (Z > 0): {theoretical_probability:.2f}")
```




```
Estimated probability (Z > 0): 0.78
Theoretical probability (Z > 0): 0.50
```

```python
import numpy as np
import matplotlib.pyplot as plt

# Set parameters
num_samples = 100
sample_size = 20

# Generate independent standard normal random variables X and Y
X = np.random.normal(0, 1, (num_samples, sample_size))
Y = np.random.normal(0, 1, (num_samples, sample_size))

# Calculate the sample means Z = 1.5 + (2 * X / Y)
Z_sample_means = 1.5 + (2 * X / Y)
sample_means = Z_sample_means.mean(axis=1)  # Taking the average across each sample

# Estimate the probability that the sample mean is greater than 0
probability_estimate = np.sum(sample_means > 0) / num_samples

# Theoretical probability for a Cauchy distribution with location 1.5 and scale 2
theoretical_probability = 0.5  # As explained earlier

# Plot the histogram
plt.hist(sample_means, bins=np.arange(-10, 15, 0.5), edgecolor='black', alpha=0.7)

# Add the shaded area indicating the estimate
plt.hist(sample_means[sample_means > 0], bins=np.arange(0, 15, 0.5), edgecolor='blue', alpha=0.3, label='Estimate Area')

# Set labels and title
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.title(f'Histogram of 100 Sample Means (Z = 1.5 + (2 * X / Y))')

# Show the plot with the legend
plt.legend()
plt.show()

# Print the estimate and theoretical probability
print(f"Estimated probability (Sample Mean > 0): {probability_estimate:.2f}")
print(f"Theoretical probability (Sample Mean > 0): {theoretical_probability:.2f}")
```




```
Estimated probability (Sample Mean > 0): 0.71
Theoretical probability (Sample Mean > 0): 0.50
```

Upon analyzing the two histograms, a noticeable difference in their shapes illustrates the effects of averaging on the sample means. The histogram for the random variables generated by the Cauchy distribution ($Z = 1.5 + (2 * X / Y)$) shows a high level of variability because the Cauchy distribution has heavy tails and does not settle around a central value. This characteristic is due to the fact that the Cauchy distribution lacks a well-defined mean and variance, which is typical of distributions with infinite variance. In contrast, the histogram of the sample means, which averages 20 random variables in each sample, demonstrates reduced variability. According to the Central Limit Theorem, averaging random variables leads to a distribution that is more concentrated around the mean and becomes more normal-shaped, even if the original distribution is not. This reduction in variability for the sample means is evident in the histogram, where the distribution of means is tighter and more predictable.

When comparing the estimated probabilities, we see that the probability of $Z > 0$ for the random variables is approximately 0.78, while the probability for the sample mean being greater than 0 is 0.71. Both of these values are higher than the theoretical probability of 0.50 for a Cauchy distribution with a location parameter of 1.5 and scale parameter of 2. This suggests that the random variables tend to favor positive values more, and by averaging multiple observations, the sample means show an even stronger tendency toward positive values. The slight difference in the estimated probabilities can be attributed to the inherent characteristics of the Cauchy distribution and the effect of averaging in reducing extreme values for the sample means, which aligns with the predictions of the Central Limit Theorem.