Samir Khadka (19701)

Question 1:
Answer:
Given:
Mean(μ)=10.3 cm
Standard Deviation$(\sigma) = 0.65cm$

a. Less than 9 cm

Z-score(Z) =$\frac{X-\mu}{\sigma} = \frac{9-10.3}{0.65} = \frac{-1.3}{0.65} = -2$

b. Between 9.5 cm to 10.6 cm

$Z_{lower} = \frac{X-\mu}{\sigma} = \frac{9.5-10.3}{0.65} = \frac{-0.8}{0.65} \approx -1.23$

$Z_{upper} = \frac{X-\mu}{\sigma} = \frac{10.6-10.3}{0.65} = \frac{0.3}{0.65} \approx 0.461$

$P(9.5 < X < 10.6) = P(Z_{upper}) - P(Z_{lower})$

c.

$X = \mu + Z_{top\ 20\%} + \sigma$

```
import scipy.stats as stats

mean = 10.3
sd = 0.65

z_score_a = (9 - mean) / sd
probability_a = stats.norm.cdf(z_score_a) * 100

z_score_b_lower = (9.5 - mean) / sd
z_score_b_upper = (10.6 - mean) / sd
probability_b = (stats.norm.cdf(z_score_b_upper) - stats.norm.cdf(z_score_b_lower)) * 100

z_score_c_top_20 = stats.norm.ppf(0.8)
length_c_top_20 = mean + z_score_c_top_20 * sd

print("a. Probability of anchovies length less than 9 cm:", round(probability_a, 2), "%")
print("b. Probability of anchovies length between 9.5 cm and 10.6 cm:", round(probability_b, 2), "%")
print("c. Minimum length for the top 20%:", round(length_c_top_20, 2), "cm")
```

```
a. Probability of anchovies length less than 9 cm: 2.28 %
b. Probability of anchovies length between 9.5 cm and 10.6 cm: 56.86 %
c. Minimum length for the top 20%: 10.85 cm
```

https://colab.research.google.com/drive/192Mj67Cz10lnRFsDNrdwX0-uaoWvICcq?authuser=0

Question 2:
Answer:
Given that:
Mean of X(μ_X)=10
Mean of Y(μ_Y)=15

Standard Deviation of X(σ_X) $= 3$

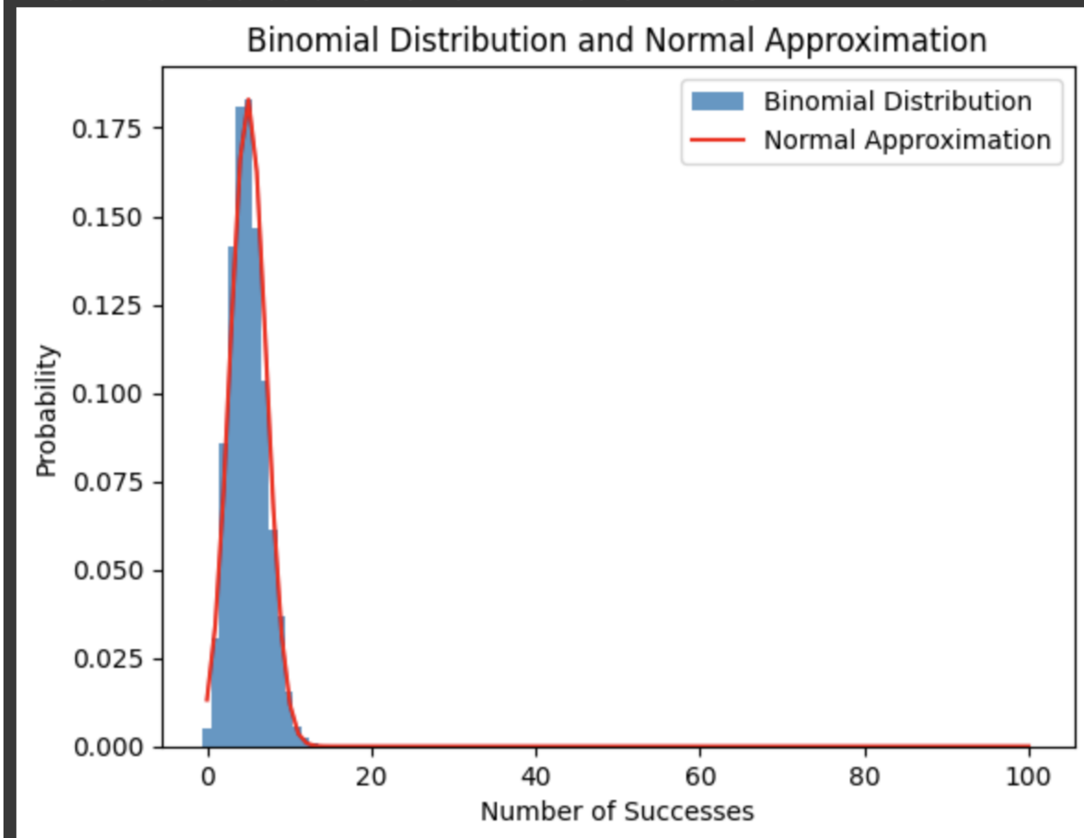
Standard Deviation of Y(σ_Y) $= 8$

1. X+Y:
   Mean(μ_{X+Y}) $=\mu_X + \mu_Y$=10+15=25
   Variance(σ^2_{X+Y}) $= \sigma^2_X + \sigma^2_Y = 3^2 + 8^2 = 9 + 64 = 73$
   Standard Deviation(σ_{X+Y}) $= \sqrt{\sigma^2_{X+Y}} = \sqrt{73} \approx 8.55$
2. X-Y:
   Mean(μ_{X-Y}) $=\mu_X - \mu_Y$=10-15=-5
   Variance(σ^2_{X-Y}) $= \sigma^2_X + \sigma^2_Y = 3^2 + 8^2 = 9 + 64 = 73$
   Standard Deviation(σ_{X+Y}) $= \sqrt{\sigma^2_{X+Y}} = \sqrt{73} \approx 8.55$
3. 3X:
   Mean(μ_{3X}) $=3\mu_X =3 \cdot 10 = 30$
   Variance(σ^2_{3X}) $= 3^2 \cdot \sigma^2_X = 3^2 \cdot 3^2 = 81$
   Standard Deviation(σ_{3X}) $= 3 \cdot \sigma_X = 9$
4. 4X+5Y:
   Mean(μ_{4X+5Y}) $=4 \cdot \mu_X + 5 \cdot \mu_Y = 4 \cdot 10+5 \cdot 15 = 40 + 75 = 115$
   Variance(σ^2_{4X+5Y}) $= 4^2 \cdot \sigma^2_X + 5^2 \cdot \sigma^2_Y = 16 \cdot 3^2 + 25 \cdot 8^2 = 144 + 1600 = 1744$
   Standard Deviation(σ_{4X+5Y}) $= \sqrt{\sigma^2_{4X+5Y}} = \sqrt{1744} \approx 41.76$

Question 3:
Answer:
https://colab.research.google.com/drive/1qeFz3Z9xBTlydqKj4DogHPKomOnIxoqo?authuser=0#scrollTo=9meychACSSIe

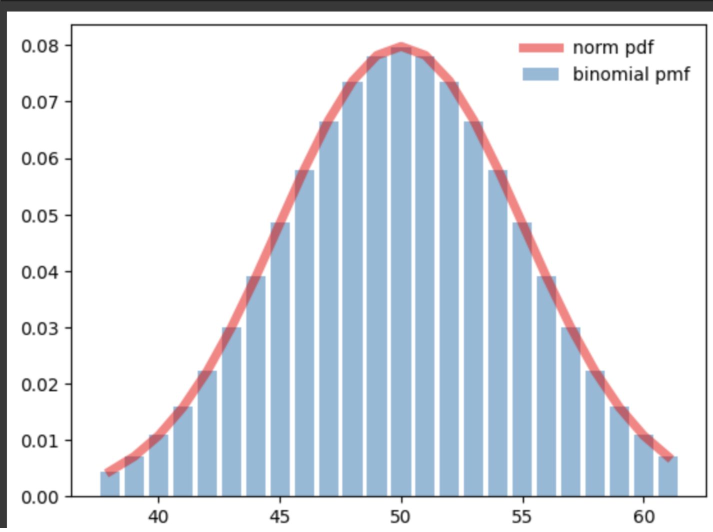
```python
#question 3
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)

n = 100
p = 0.05
q = 1 - p

binomial_data = np.random.binomial(n, p, 10000)

mean_cal = np.mean(binomial_data)
std_dev_cal = np.std(binomial_data)

mean_theoretical = n * p
std_dev_theoretical = np.sqrt(n * p * q)

print("Calculated Mean:", mean_cal)
print("Theoretical Mean:", mean_theoretical)
print("Calculated Standard Deviation:", std_dev_cal)
print("Theoretical Standard Deviation:", std_dev_theoretical)

plt.hist(binomial_data, bins=np.arange(-0.5, n + 1.5, 1), density=True, alpha=0.75, label='Binomial Distribution')

x = np.linspace(0, n, 100)
normal_approximation = (1 / (std_dev_theoretical * np.sqrt(2 * np.pi))) * np.exp(-(x - mean_theoretical)**2 / (2 * std_dev_theoretical**2))
plt.plot(x, normal_approximation, 'r', label='Normal Approximation')

plt.title('Binomial Distribution and Normal Approximation')
plt.xlabel('Number of Successes')
plt.ylabel('Probability')
plt.legend()

plt.show()
```

```
Calculated Mean: 4.9557
Theoretical Mean: 5.0
Calculated Standard Deviation: 2.1521936506736563
Theoretical Standard Deviation: 2.179449471770337
```



Question 4:
Answer:
https://colab.research.google.com/drive/1Y_sT5TPGUTAG7Rz0EhmMPayn0Jhhf2JY

```python
#Question 4
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom, norm

n = 100
p = 0.5

binomial = binom(n, p)

mu = n * p
sigma = np.sqrt(n * p * (1 - p))
normal = norm(mu, sigma)

x = np.arange(binomial.ppf(0.01), binomial.ppf(0.99))

plt.bar(x, binomial.pmf(x), alpha=0.5, label='binomial pmf')

plt.plot(x, normal.pdf(x), 'r-', lw=5, alpha=0.5, label='norm pdf')

plt.legend(loc='best', frameon=False)
plt.show()
```

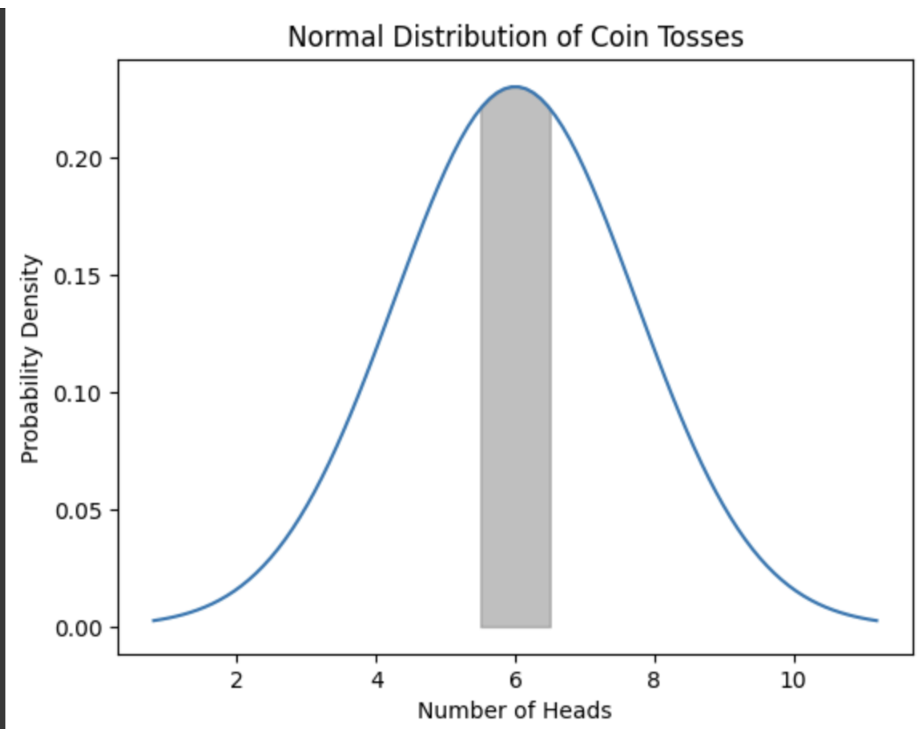


Question number:5
Answer:

+ Code + Text

```python
#Question num 5
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

n = 12
p = 0.5

mu = n * p
sigma = (n * p * (1 - p)) ** 0.5

lower_bound = 5.5
upper_bound = 6.5
z_lower = (lower_bound - mu) / sigma
z_upper = (upper_bound - mu) / sigma

prob_lower = norm.cdf(z_lower)
prob_upper = norm.cdf(z_upper)
probability = prob_upper - prob_lower

print(f"The approximate probability of getting exactly 6 heads in 12 tosses is: {probability:.4f}")

x = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
y = norm.pdf(x, mu, sigma)
plt.plot(x, y)

px = np.linspace(lower_bound, upper_bound, 100)
py = norm.pdf(px, mu, sigma)
plt.fill_between(px, py, color='grey', alpha=0.5)

plt.title('Normal Distribution of Coin Tosses')
plt.xlabel('Number of Heads')
plt.ylabel('Probability Density')
plt.show()
```

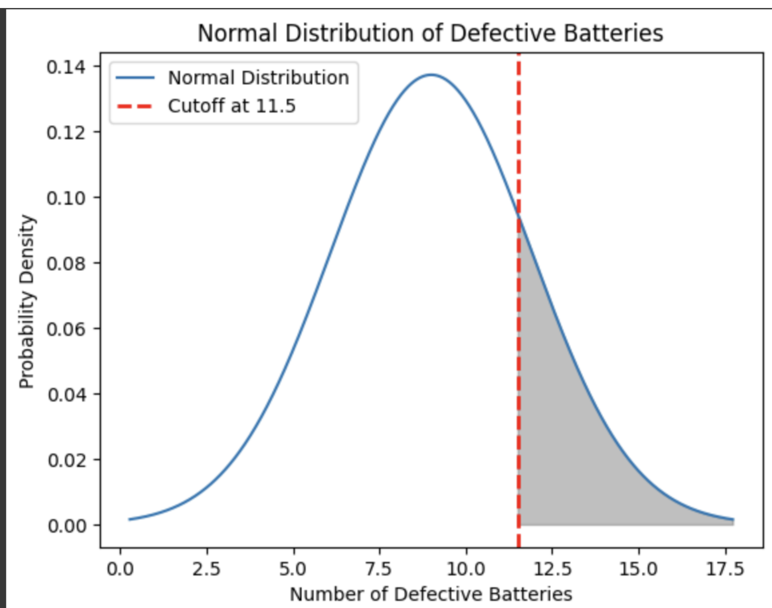
```
The approximate probability of getting exactly 6 heads in 12 tosses is: 0.2272
```



Question 6:
Answer:

```
#Question num 6
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

n = 150
p = 0.06

mu = n * p
sigma = (n * p * (1 - p)) ** 0.5

cutoff = 11.5
z_score = (cutoff - mu) / sigma

probability = 1 - norm.cdf(z_score)

print(f"The approximate probability of having 12 or more defective batteries is: {probability:.4f}")

x = np.linspace(mu - 3*sigma, mu + 3*sigma, 1000)
y = norm.pdf(x, mu, sigma)
plt.plot(x, y, label='Normal Distribution')

px = np.linspace(cutoff, mu + 3*sigma, 1000)
py = norm.pdf(px, mu, sigma)
plt.fill_between(px, py, color='grey', alpha=0.5)

plt.axvline(x=cutoff, color='red', linestyle='dashed', linewidth=2, label=f'Cutoff at {cutoff}')

plt.title('Normal Distribution of Defective Batteries')
plt.xlabel('Number of Defective Batteries')
plt.ylabel('Probability Density')
plt.legend()

plt.show()
```

```
The approximate probability of having 12 or more defective batteries is: 0.1950
```



Question 7:
Answer:

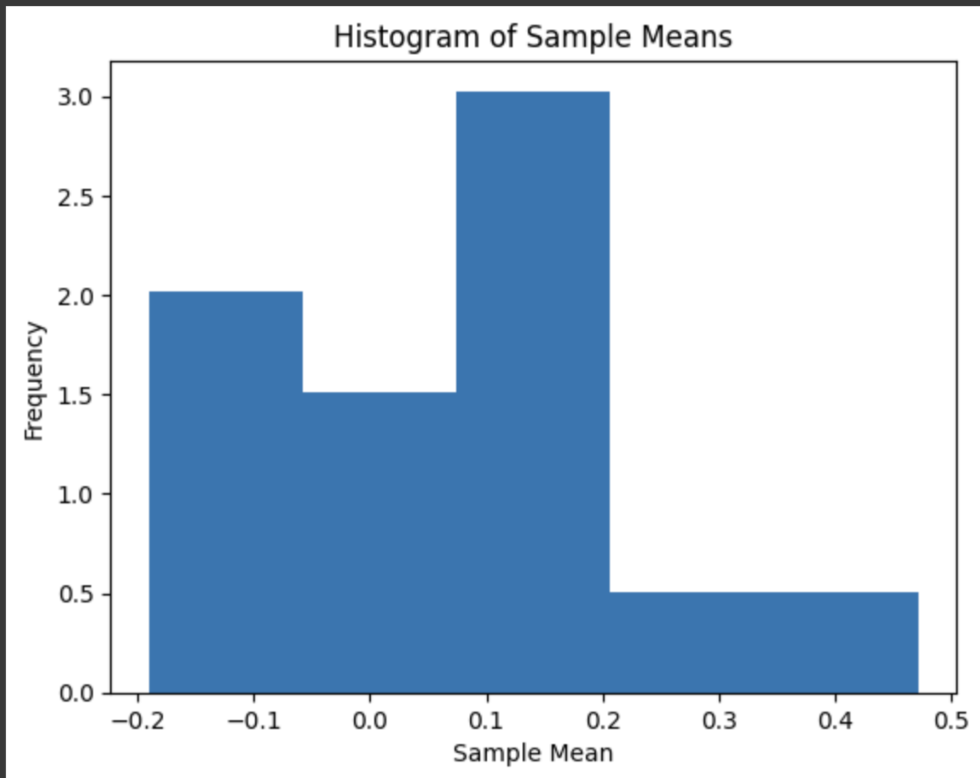
https://colab.research.google.com/drive/1keiCSRHcOEuwev0uZ_2k1PZkF9jMvK9F

Q7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
#Answer of 7
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import t

df = 10
random_numbers = t.rvs(df, size=100)

mu = np.mean(random_numbers)
sigma = np.std(random_numbers)

n_samples = 30
n_groups = 15
samples = [np.random.choice(random_numbers, n_samples) for _ in range(n_groups)]

means = [np.mean(sample) for sample in samples]

mu_x = np.mean(means)
sigma_x = sigma / np.sqrt(n_samples)

plt.hist(means, bins='auto', density=True)
plt.title('Histogram of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()

print(f"Mean of 100 random numbers (mu): {mu}")
print(f"Standard deviation of 100 random numbers (sigma): {sigma}")
print(f"Mean of sample means (mu_x): {mu_x}")
print(f"Standard deviation of sample means (sigma_x): {sigma_x}")
```




```
Mean of 100 random numbers (mu): 0.1228976515777304
Standard deviation of 100 random numbers (sigma): 1.0378748493545245
Mean of sample means (mu_x): 0.0673377088035075
Standard deviation of sample means (sigma_x): 0.18948915561958304
```