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.

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

+ Code + Text

```
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/192Mj67Cz10InRFsDNrdwX0-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) $= 3$

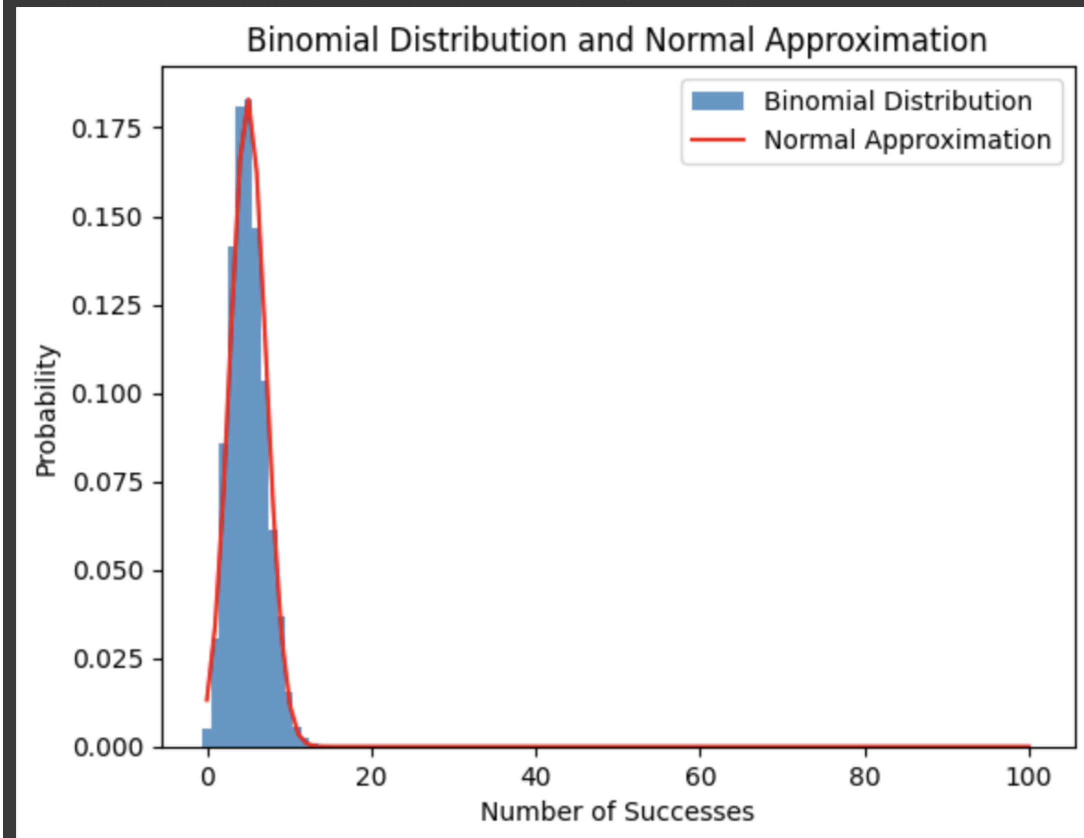
Standard Deviation of Y(σ_Y) $= 8$

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}) $= \sqrt{\sigma^2_{4X+5Y}} = \sqrt{1744} \approx 41.76$

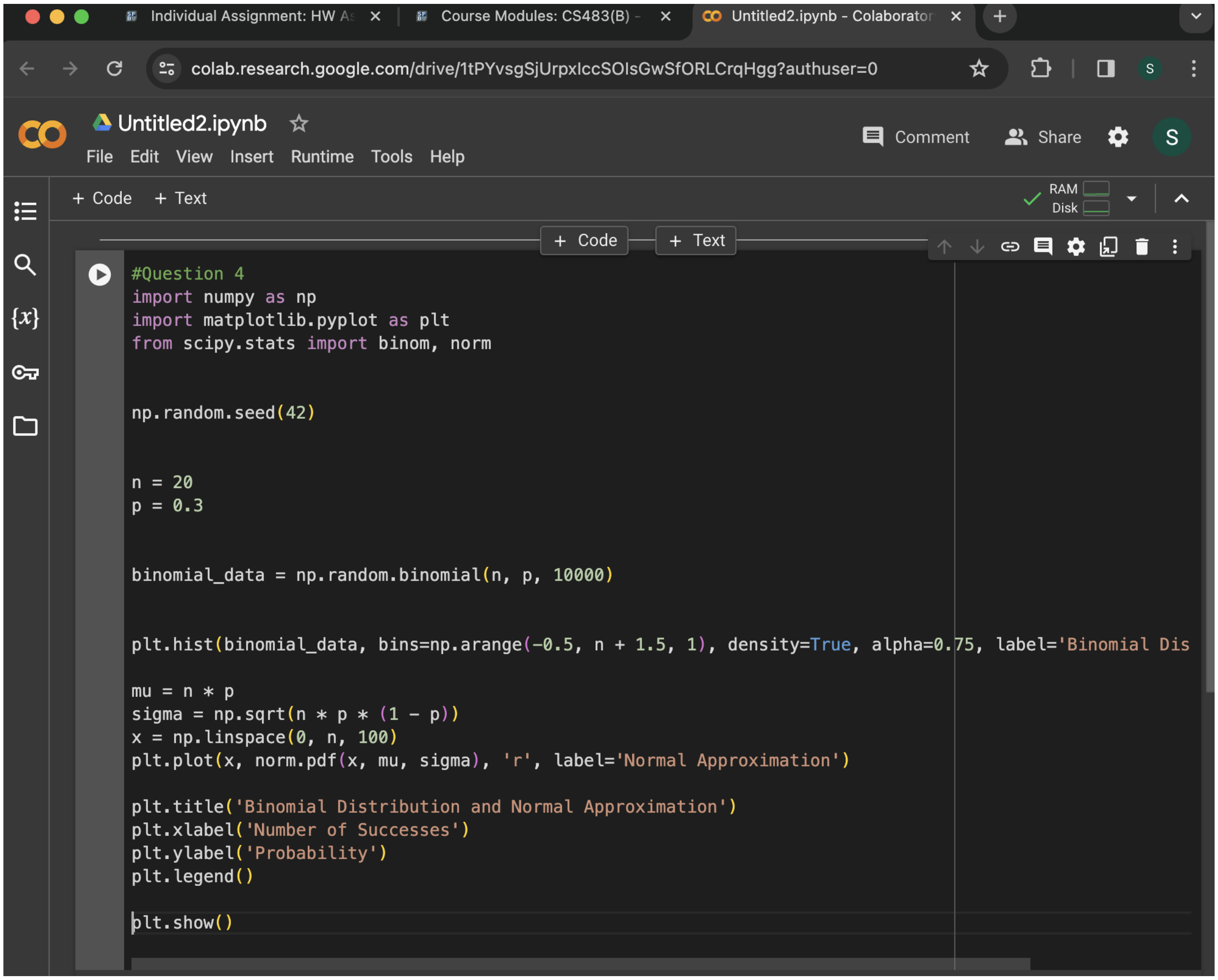
Question 3:
Answer:
https://colab.research.google.com/drive/1qeFz3Z9xBTlydqKj4DogHPKomOnIxoqo?authuser=0#scrollTo=9meychACSSIe

```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()
```

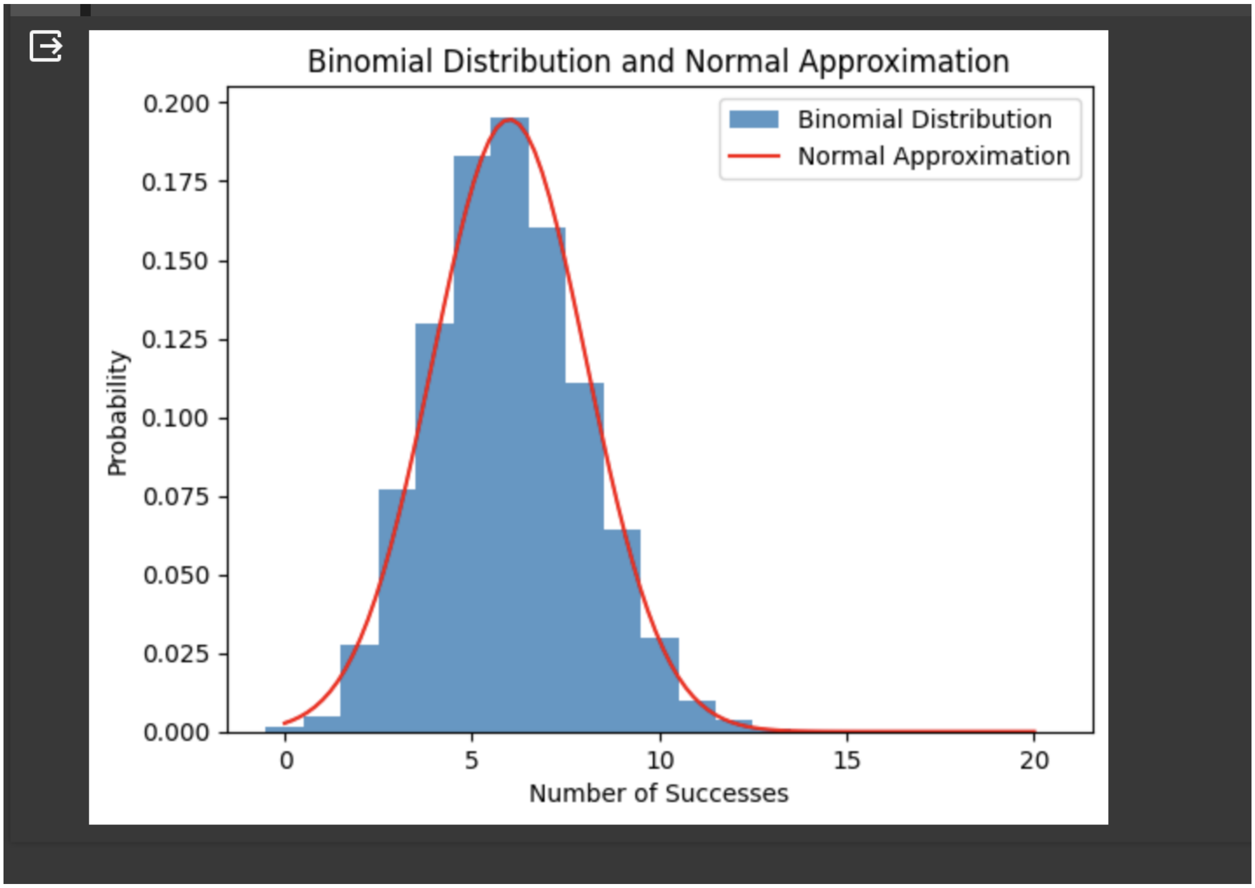
```
Calculated Mean: 4.9557
Theoretical Mean: 5.0
Calculated Standard Deviation: 2.1521936506736563
Theoretical Standard Deviation: 2.179449471770337
```



Question 4:
Answer:
https://colab.research.google.com/drive/1tPYvsgSjUrpxIccSOIsGwSfORLCrqHgg?authuser=0




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


np.random.seed(42)


n = 20
p = 0.3


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


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

mu = n * p
sigma = np.sqrt(n * p * (1 - p))
x = np.linspace(0, n, 100)
plt.plot(x, norm.pdf(x, mu, sigma), 'r', label='Normal Approximation')

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

plt.show()
```



Question number:5
Answer:
Given that:
Let,

n be number of trials i.e. coin tosses
p be probability of success i.e getting heads in a single toss
x be number of heads to get probability

We have,
n=12, p= ½=0.5, x=6
Now,
Mean (μ)=n x p= 12 x 0.5 =6
Standard Deviation (σ)=$\sqrt{n \cdot p \cdot (1 - p)} = \sqrt{12 \cdot 0.5 \cdot 0.5} = \sqrt{3} = 1.73205080757$
Therefore, the required mean is 6 and standard deviation is 1.73205080757
Z-score (Z)=$\frac{x-\mu}{\sigma}$
For $x = 6$:
Z=$\frac{x-\mu}{\sigma}$=$\frac{6-6}{\sqrt{3}}$ =$\frac{0}{\sqrt{3}} = 0$
Therefore, the required Z-score is 0.

Now For normal distribution,

Z-scores	Probabilities
-3	0.001349898032
-2.5	0.006209665326
-2	0.02275013195
-1.5	0.06680720127
-1	0.1586552539
-0.5	0.3085375387
0	0.5
0.5	0.6914624613
1	0.8413447461
1.5	0.9331927987
2	0.9772498681
2.5	0.9937903347
3	0.998650102
3.5	0.9997673709

Question 6:
Answer:
Given that:
Let,

n be number of trials i.e. sample size
p be probability of success i.e defective rate
x be number of defective batteries to get probability

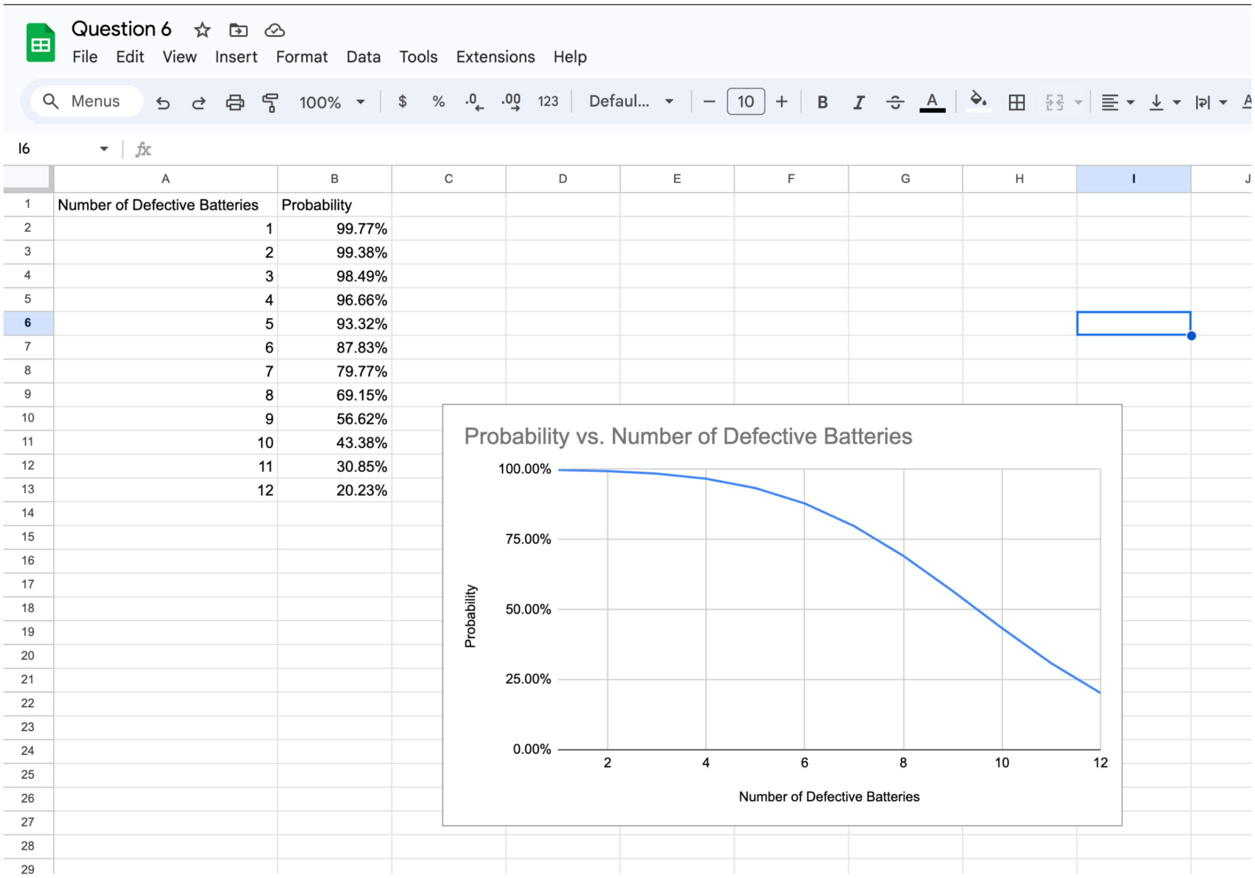
We have,
n=150, p=0.06, x=12
Now,
Mean (μ)=n x p= 150 x 0.06 =9
Standard Deviation (σ)=$\sqrt{n \cdot p \cdot (1 - p)} = \sqrt{150 \cdot 0.06 \cdot 0.94} = \sqrt{8.46} \approx 2.908$
Therefore, the required mean is 6 and standard deviation is 1.73205080757
Z-score (Z)=$\frac{x-\mu}{\sigma}$

For $x = 6$:

$Z = \frac{x-\mu}{\sigma} = \frac{12-9}{2.908} = \frac{3}{2.908} \approx 1.031$

Therefore, the required Z-score is 1.031.

Now, for normal distribution:



Number of Defective Batteries	Probability
1	99.77%
2	99.38%
3	98.49%
4	96.66%
5	93.32%
6	87.83%
7	79.77%
8	69.15%
9	56.62%
10	43.38%
11	30.85%
12	20.23%

Question 7:

Answer:

https://colab.research.google.com/drive/192Mj67Cz10InRFsDNrdwX0-uaoWvICcq?authuser=0#scrollTo=z3C342BhedMK

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

def gen_t_dist_samp(size, df=10):
    return t.rvs(df, size=size)

p = 15
n= 30
total_random_numbers = 100
df = 10
rnd_num = gen_t_dist_samp(total_random_numbers, df)

sampling_groups = [gen_t_dist_samp(n, df) for _ in range(p)]

means = [np.mean(group) for group in sampling_groups]

avg_mean = np.mean(means)
means_std = np.std(means, ddof=1)
print(f"Mean of the 100 random numbers: {np.mean(rnd_num)}")
print(f"Mean of the means from sampling groups: {avg_mean}")
print(f"Standard deviation of the means from sampling groups: {means_std}")

plt.hist(means, bins=15, edgecolor='black')
plt.title('Histogram of Means from Sampling Groups')
plt.xlabel('Means')
plt.ylabel('Frequency')
plt.show()
```

```
Mean of the 100 random numbers: -0.04289098622911745
Mean of the means from sampling groups: -0.04894609562598996
Standard deviation of the means from sampling groups: 0.21082682015163318
```