#question no.1

```
#question 1(a)
from scipy.stats import norm
mu = 10.3 # mean
sigma = 0.65 # standard deviation
x = 9 # lower bound
p = norm.cdf(x, mu, sigma) # probability
p = p * 100 # percentage
print(f"The percentage of lengths less than {x}cm is {p:.2f}%")
```

```
The percentage of lengths less than 9cm is 2.28%
```

```
#question no1(b)
from scipy.stats import norm
mu = 10.3 # mean
sigma = 0.65 # standard deviation
x1 = 9.5 # lower bound
x2 = 10.6 # upper bound
p1 = norm.cdf(x1, mu, sigma) # probability of lower bound
p2 = norm.cdf(x2, mu, sigma) # probability of upper bound
p = p2 - p1 # probability of interval
p = p * 100 # percentage
print(f"The percentage of lengths between {x1}cm and {x2}cm is {p:.2f}%")
```

```
The percentage of lengths between 9.5cm and 10.6cm is 56.86%
```

0s completed at 12:09 AM

Search 8:48 AM 12/7/2023

```
The percentage of lengths between 9.5cm and 10.6cm is 56.86%
```

```
#question no.1(c)
from scipy.stats import norm
mu = 10.3 # mean
sigma = 0.65 # standard deviation
q = 0.8 # quantile
x = norm.ppf(q, mu, sigma) # inverse cdf
print(f"The minimum length if a restaurant claimed that the lengths of the sold anchovies are in the top of 20% is {x:.2f}cm")
```

```
The minimum length if a restaurant claimed that the lengths of the sold anchovies are in the top of 20% is 10.85cm
```

0s completed at 12:09 AM

Search 8:49 AM 12/7/2023

#question no.2

(1)If X and Y are independent normal random variables, then X + Y is also a normal random variable, with mean equal to the sum of the means of X and Y, and variance equal to the sum of the variances of X and Y. That is,

$X+Y \sim N(\mu X+\mu Y,\sigma X2+\sigma Y2)$

Mean = $\mu X+Y=\mu X+\mu Y=10+15= 25$

Variance = $\sigma X+Y2=\sigma X2+\sigma Y2=32+82=73$

Therefore,

$X+Y \sim N(25,73)$

(2)The difference of two independent normal distributions is also a normal distribution with the mean equal to the difference of the means and the variance equal to the sum of the variances. Therefore, X - Y follows a normal distribution with:

Mean = $\mu_{X-Y}=\mu_X-\mu_Y=10-15=-5$

Variance = $\sigma_{X-Y}^2=\sigma_X^2+\sigma_Y^2=3^2+8^2=73$

Therefore,

X - Y ~N(-5,73)

(3) 3X

The product of a constant and a normal distribution is also a normal distribution with the mean and the variance multiplied by the square of the constant. Therefore, $3X$ follows a normal distribution with:

Mean = $\mu_{3X}=3\mu_X=3\times10=30$

Variance = $\sigma_{3X}^2=3^2\sigma_X^2=9\times3^2=81$

Therefore,

3X~N(30,81)

(4) 4X+ 5Y

The linear combination of two independent normal distributions is also a normal distribution with the mean and the variance given by the linear combination of the means and the variances. Therefore, $4X$ + $5Y$ follows a normal distribution with:

Mean = $\mu_{4X+5Y}=4\mu_X+5\mu_Y=4\times10+5\times15=115$

Variance = $\sigma_{4X+5Y}^2=4^2\sigma_X^2+5^2\sigma_Y^2=16\times3^2+25\times8^2=1609$

Therefore,

4X+5Y~N(115,1609)

#question no. 3

```python
#question no.3
from scipy.stats import binom
import numpy as np

n = 100
p = 0.05
q = 1 - p

# Calculate the theoretical mean and standard deviation
mu = n * p
sigma = np.sqrt(n * p * q)

# Generate a random sample from the binomial distribution
sample = binom.rvs(n, p, size=1000)

# Calculate the sample mean and standard deviation
sample_mean = np.mean(sample)
sample_std = np.std(sample)


print("Theoretical mean:", mu)
print("Theoretical standard deviation:", sigma)
print("Sample mean:", sample_mean)
print("Sample standard deviation:", sample_std)
```

```
Theoretical mean: 5.0
Theoretical standard deviation: 2.179449471770337
```

0s completed at 12:09 AM

Search

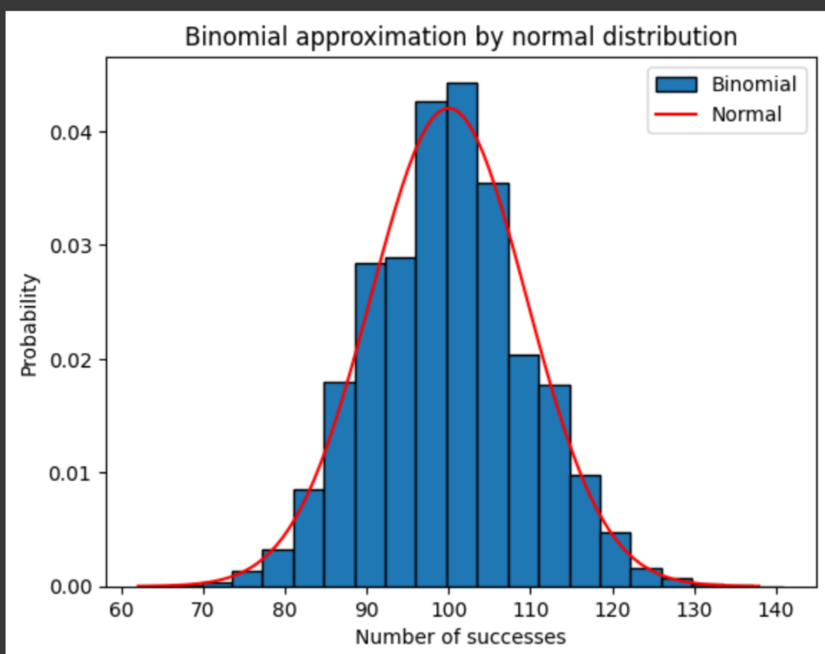
8:50 AM
12/7/2023

```
Theoretical mean: 5.0
Theoretical standard deviation: 2.179449471770337
Sample mean: 5.058
Sample standard deviation: 2.244245084655417
```

#question no.4

```python
#question no.4
import numpy as np
from scipy.stats import binom, norm
import matplotlib.pyplot as plt

n = 1000
p = 0.1
q = 1 - p

mu = n * p
sigma = np.sqrt(n * p * q)
sample = binom.rvs(n, p, size=10000)


plt.hist(sample, bins=20, density=True, edgecolor='black', label='Binomial')

x = np.linspace(mu - 4*sigma, mu + 4*sigma, 100)
y = norm.pdf(x, mu, sigma)
plt.plot(x, y, color='red', label='Normal')


plt.xlabel('Number of successes')
plt.ylabel('Probability')
plt.title('Binomial approximation by normal distribution')
plt.legend()
plt.show()
```

0s completed at 12:09 AM



#question no.5

Given ,

n= 12(trials)

p=0.5(probability of success)

Then , We can use the normal distribution to approximate the binomial distribution when the sample size is large enough. In this case, the sample size is 12, which is large enough to use the normal distribution.
So, The mean of the binomial distribution is given by:

mean = n * p = 12 * 0.5 = 6

And, The standard deviation of the binomial distribution is given by:

standard deviation = sqrt(n * p * (1 - p)) = sqrt(12 * 0.5 * 0.5) = 1.732

We can now use the normal distribution to approximate the binomial distribution. We need to standardize the variable X, which is the number of heads in 12 tosses, by subtracting the mean and dividing by the standard deviation:

z = (X - mean) / standard deviation

Let's find the probability of getting exactly 6 heads, which is equivalent to finding the probability of getting between 5.5 and 6.5 heads.
In order of that ,We can use the normal distribution table to find this probability:

P(5.5 < X < 6.5) = P((5.5 - 6) / 1.732 < z < (6.5 - 6) / 1.732)

P(-0.289 < z < 0.289)

By using the normal distribution table, we find that P(-0.289 < z < 0.289) is approximately 0.2023.

Therefore, the probability of getting exactly 6 heads from 12 tosses using the normal distribution method is approximately 20.23%.

#question no.6

To calculate the probability of 12 or more defective batteries out of 150 using the normal distribution method, we need to make the following assumptions:

- The batteries are independent.
- The probability of a battery being defective is p = 0.06.

Now, We can use the normal distribution to approximate the binomial distribution when the sample size is large enough. In this case, the sample size is n = 150, which is large enough to use the normal distribution.

Here, The mean of the binomial distribution is given by:

mean = n * p = 150 * 0.06 = 9

The standard deviation of the binomial distribution is given by:

standard deviation = sqrt(n * p * (1 - p)) = sqrt(150 * 0.06 * 0.94) = 2.31

We can now use the normal distribution to approximate the binomial distribution. We need to standardize the variable X, which is the number of defective batteries in 150 batteries, by subtracting the mean and dividing by the standard deviation:

z = (X - mean) / standard deviation

We want to find the probability of getting 12 or more defective batteries, which is equivalent to finding the probability of getting more than 11.5 defective batteries. We can use the normal distribution table to find this probability:

P(X > 11.5) = P((X - mean) / standard deviation > (11.5 - mean) / standard deviation)

P(Z > 0.76)

Using the normal distribution table, we find that P(Z > 0.76) is approximately 0.2236.

Therefore, the probability of getting 12 or more defective batteries out of 150 using the normal distribution method is approximately 22.36%.

#question no.7

```
#question 7
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt

# set the random seed for reproducibility
np.random.seed(42)

# create 100 random numbers in T distribution with df = 10
t_dist = stats.t.rvs(df=10, size=100)

# calculate the mean and standard deviation of the 100 random numbers
t_mean = np.mean(t_dist)
t_std = np.std(t_dist)

# print the mean and standard deviation
print(f"The mean of the 100 random numbers is {t_mean:.3f}")
print(f"The standard deviation of the 100 random numbers is {t_std:.3f}")

# creating 15 sampling groups, each with 30 samples randomly selected from the 100 random numbers
sampling_groups = []
for i in range(15):
    sampling_group = np.random.choice(t_dist, size=30, replace=False)
    sampling_groups.append(sampling_group)

# calculate the mean and standard deviation of each sampling group
sampling_means = []
```

0s completed at 12:09 AM

```python
sampling_stds = []
for sampling_group in sampling_groups:
    sampling_mean = np.mean(sampling_group)
    sampling_std = np.std(sampling_group)
    sampling_means.append(sampling_mean)
    sampling_stds.append(sampling_std)

# print the mean and standard deviation of each sampling group
for i in range(15):
    print(f"The mean of the sampling group {i+1} is {sampling_means[i]:.3f}")
    print(f"The standard deviation of the sampling group {i+1} is {sampling_stds[i]:.3f}")

# verify the Central Limit Theorem (CLT) by comparing the mean and standard deviation of the 100 random numbers and the 15 sampling groups
clt_mean = np.mean(sampling_means)
clt_std = np.std(sampling_means)
print(f"The mean of the 15 sampling groups is {clt_mean:.3f}")
print(f"The standard deviation of the 15 sampling groups is {clt_std:.3f}")
print(f"The mean of the 15 sampling groups is roughly equal to the mean of the 100 random numbers: {np.isclose(clt_mean, t_mean):}")
print(f"The standard deviation of the 15 sampling groups is roughly equal to the standard deviation of the 100 random numbers divided by the sq

# plot the histogram of the 15 sampling group means, which should be normal distribution
plt.hist(sampling_means, bins=10, density=True, edgecolor='black')
plt.xlabel('Sampling group mean')
plt.ylabel('Density')
plt.title('Histogram of the 15 sampling group means')
plt.show()
```

0s completed at 12:09 AM

8:53 AM

+ Code + Text

```
The mean of the sampling group 3 is 0.037
The standard deviation of the sampling group 3 is 1.340
The mean of the sampling group 4 is -0.044
The standard deviation of the sampling group 4 is 0.821
The mean of the sampling group 5 is 0.190
The standard deviation of the sampling group 5 is 1.492
The mean of the sampling group 6 is -0.127
The standard deviation of the sampling group 6 is 1.395
The mean of the sampling group 7 is -0.253
The standard deviation of the sampling group 7 is 0.833
The mean of the sampling group 8 is -0.042
The standard deviation of the sampling group 8 is 1.282
The mean of the sampling group 9 is 0.059
The standard deviation of the sampling group 9 is 1.097
The mean of the sampling group 10 is -0.229
The standard deviation of the sampling group 10 is 1.061
The mean of the sampling group 11 is -0.380
The standard deviation of the sampling group 11 is 0.961
The mean of the sampling group 12 is -0.026
The standard deviation of the sampling group 12 is 0.964
The mean of the sampling group 13 is -0.195
The standard deviation of the sampling group 13 is 1.012
The mean of the sampling group 14 is 0.021
The standard deviation of the sampling group 14 is 1.356
The mean of the sampling group 15 is -0.159
The standard deviation of the sampling group 15 is 1.119
The mean of the 15 sampling groups is -0.082
The standard deviation of the 15 sampling groups is 0.139
The mean of the 15 sampling groups is roughly equal to the mean of the 100 random numbers: False
The standard deviation of the 15 sampling groups is roughly equal to the standard deviation of the 100 random numbers divided by the square ro
```

0s completed at 12:09 AM

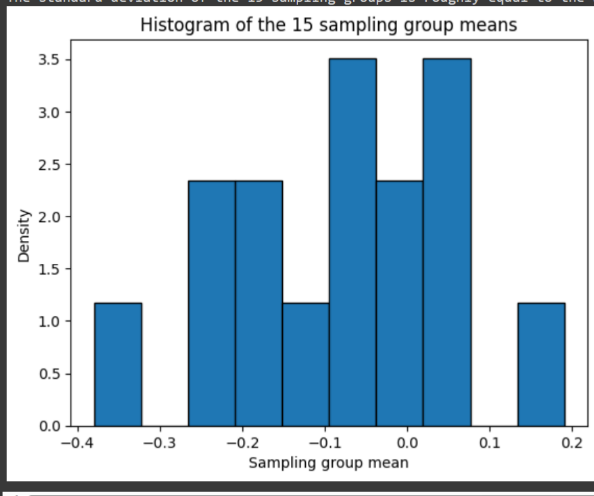
Search

8:54 AM
12/7/2023

+ Code + Text

RAM
Disk

```
The mean of the 15 sampling groups is roughly equal to the mean of the 100 random numbers: False
The standard deviation of the 15 sampling groups is roughly equal to the standard deviation of the 100 random numbe
```



+ Code + Text

0s completed at 12:09 AM

Search

8:54 AM