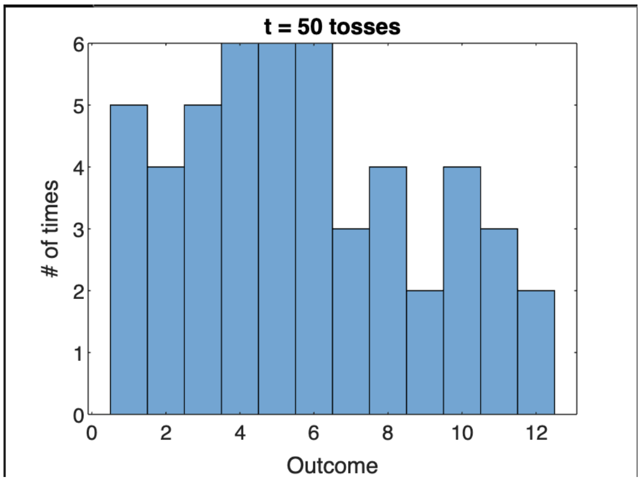
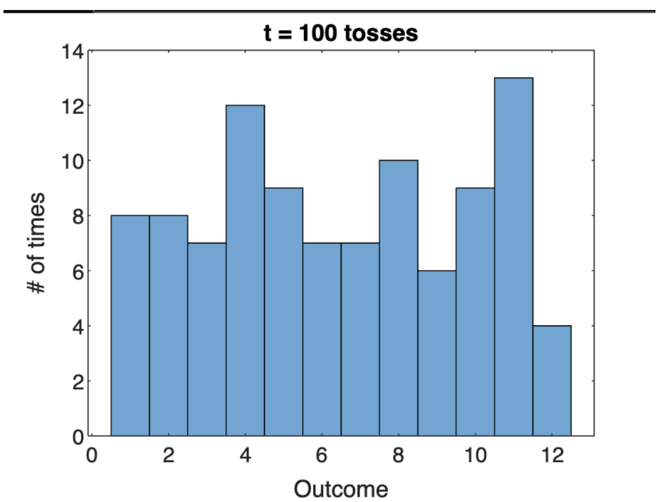
Alexis Lee

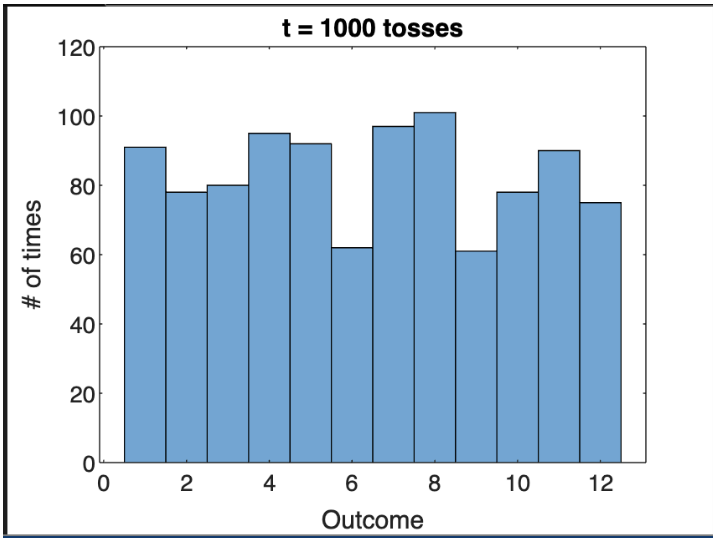
ECE 131A Project

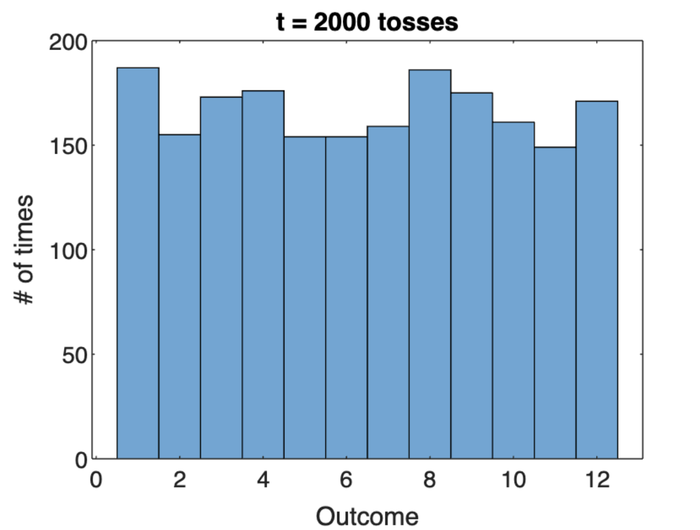
Question 1: *Tossing a fair and unfair die*

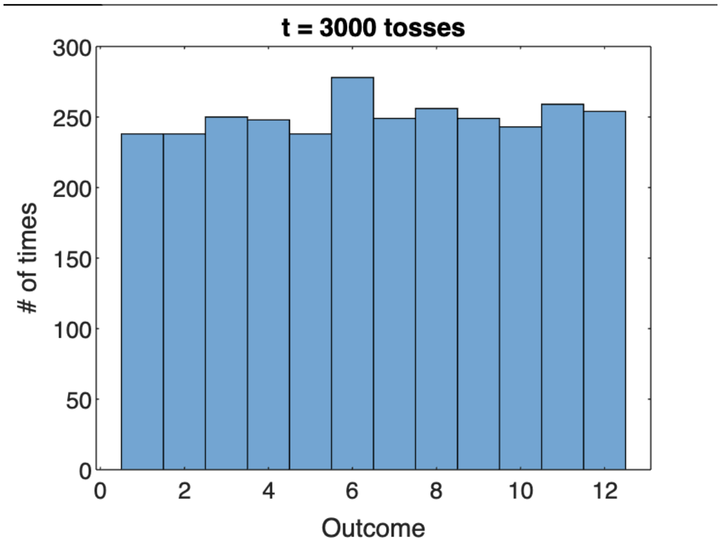
a) Histograms of each simulation are shown below, where each outcome has an equal probability. (see appendix for full code)

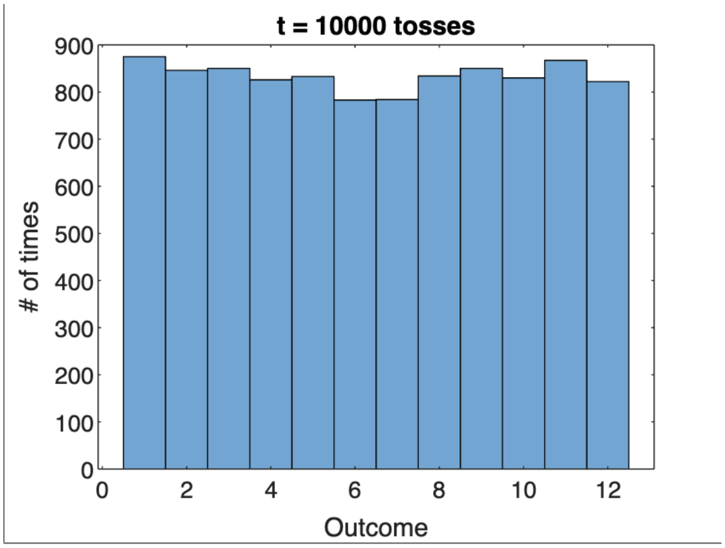


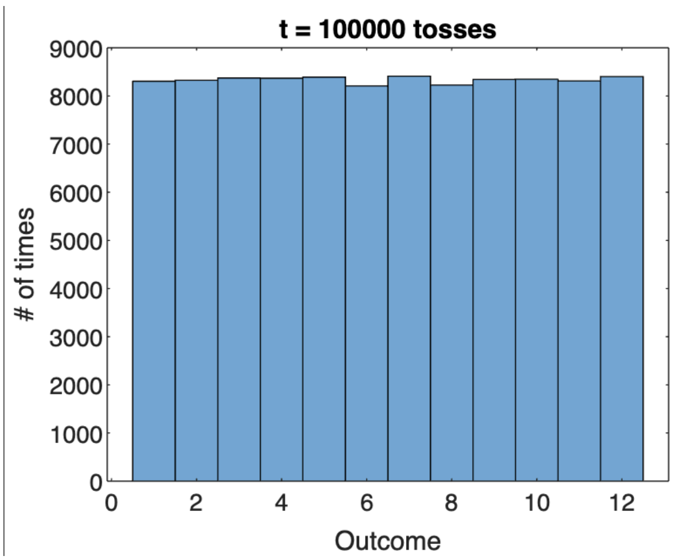



t = 1000 tosses
# of times
Outcome
0
20
40
60
80
100
120
0
2
4
6
8
10
12


t = 2000 tosses
# of times
Outcome
0
50
100
150
200
0
2
4
6
8
10
12


t = 3000 tosses
# of times
Outcome
0
50
100
150
200
250
300
0
2
4
6
8
10
12



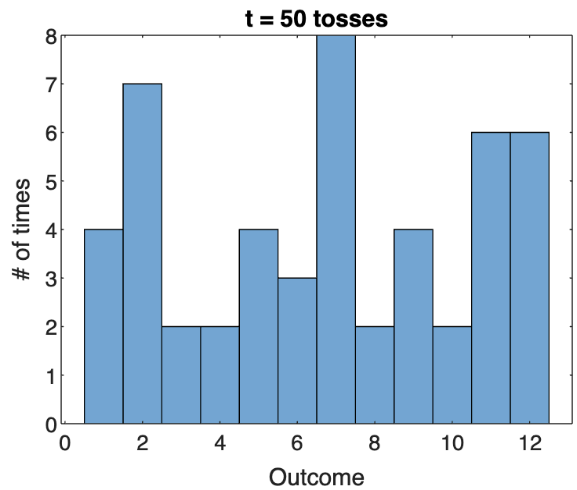


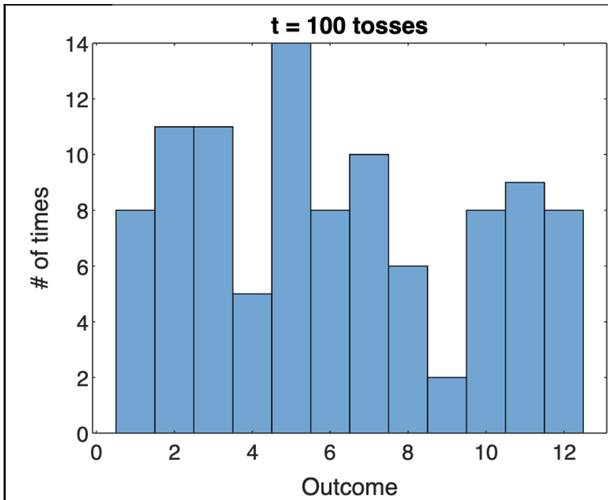
Based on the simulation, I'd estimate a ½ probability of obtaining an odd number since it seems that an odd number is obtained half of the time as the number of tosses (samples) increases; in other words, the probabilities for each outcome are roughly the same and as the number of tosses increases they're expected to be exactly the same.

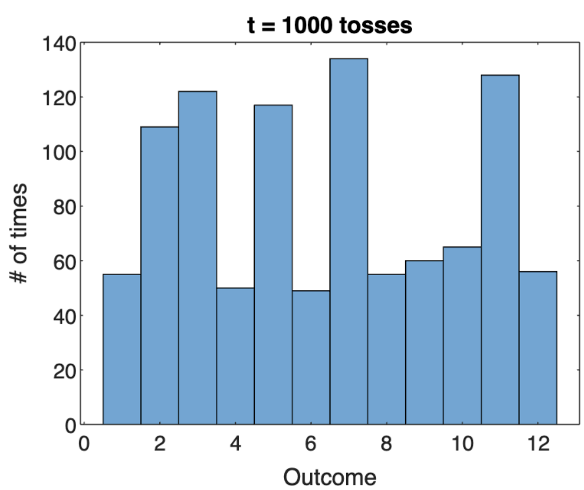
b) P(X odd) = P(X = 1) + P(X = 3) + P(X = 5) + P(X = 7) + P(X = 9) + P(X = 11)
so P(X odd) = 6 * (1/12) = ½.

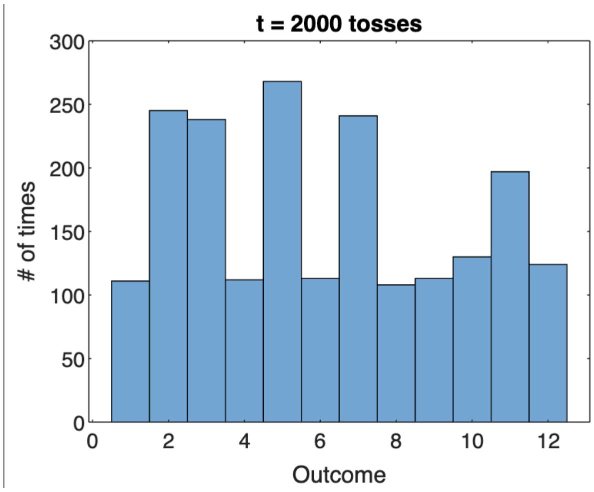
c) Yes, ½ = ½ so the experimental and theoretical results agree.

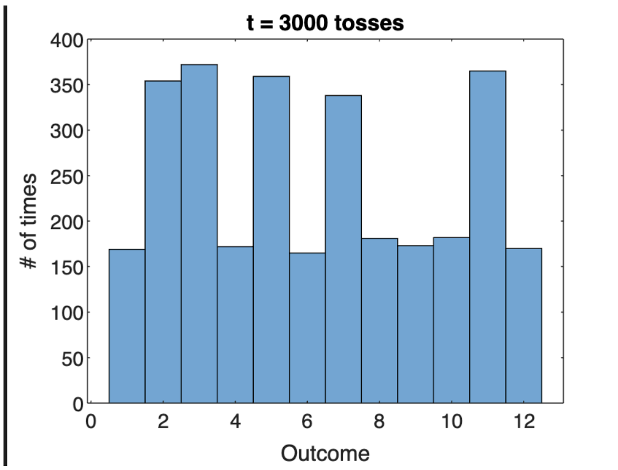
d) a. I implemented this scenario by defining an array of 17 outcomes, where the prime numbers appear twice as frequently as the non-time numbers. Then I used the randi function in MATLAB to pick integers between 1 and 17 to pick an outcome from the array based on its index, t times. Histograms of each simulation are shown below. (see appendix for full code)

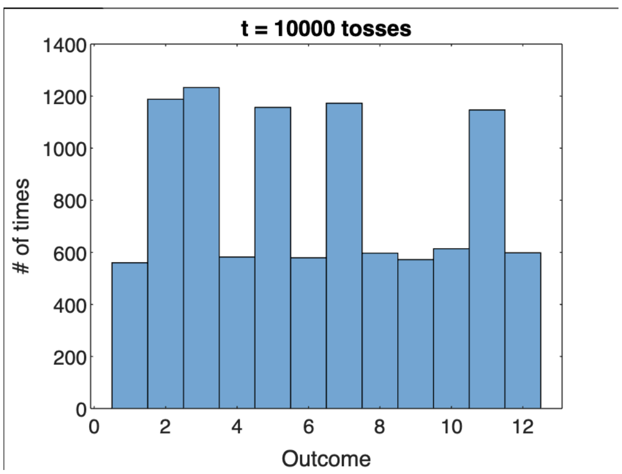

t = 50 tosses
# of times
0 1 2 3 4 5 6 7 8
0 2 4 6 8 10 12
Outcome

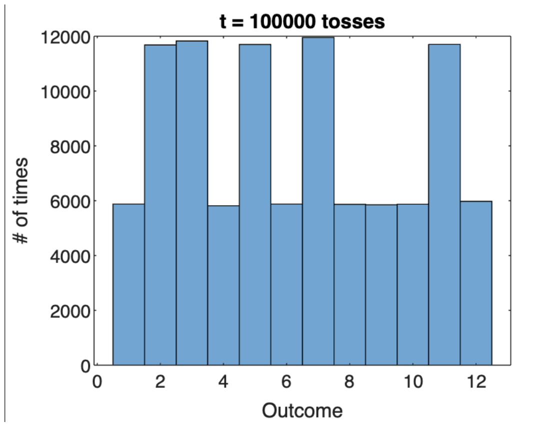

t = 100 tosses
# of times
0 2 4 6 8 10 12 14
0 2 4 6 8 10 12
Outcome


t = 1000 tosses
# of times
0 20 40 60 80 100 120 140
0 2 4 6 8 10 12
Outcome


t = 2000 tosses
# of times
Outcome
0
50
100
150
200
250
300
2
4
6
8
10
12


t = 3000 tosses
# of times
Outcome
0
50
100
150
200
250
300
350
400
2
4
6
8
10
12


t = 10000 tosses
# of times
Outcome
0
200
400
600
800
1000
1200
1400
2
4
6
8
10
12



Based on the simulations, I'd expect that the probabilities of getting a prime number are 2/17 and of not getting a prime number are 1/17. As such, the probabilities of getting an odd number would be P(X = 1) + P(X = 3) + P(X = 5) + P(X = 7) + P(X = 9) + P(X = 11) = (1/17) + (2/17) + (2/17) + (2/17) + (1/17) + (2/17) = 10/17.

b.

The prime numbers are 2, 3, 5, 7, and 11.
The non-prime numbers are 1, 4, 6, 8, 9, 10, and 12.

$P(X=1) = P(X=4) = P(X=6) = P(X=8) = P(X=9) = P(X=10) = P(X=12)$, and
$P(X=2) = P(X=3) = P(X=5) = P(X=7) = P(X=11) = 2P(X=1)$.

So, $7 \cdot P(X=1) + 5 \cdot (2P(X=1)) = 1$
$P(X=1) = 1/17$ and $P(X=2) = 2/17$

$P(X \text{ odd}) = P(X=1) + P(X=3) + P(X=5) + P(X=7) + P(X=9) + P(X=11)$
$P(X=\text{odd}) = 1/17 + 2/17 + 2/17 + 2/17 + 1/17 + 2/17 = 10/17$

So there is a 10/17 probability that X is odd.

c. 10/17 = 10/17 so the expected and theoretical probabilities agree.

Question 2: *Maximum Likelihood Estimation*

a)

Since $X_1, X_2, \dots, X_n$ are independent,

$$f_{X_1, X_2, \dots X_n}(x_1, x_2, \dots, x_n) = f_{X_1}(x_1) \cdot f_{X_2}(x_2) \cdots f_{X_n}(x_n)$$

where $f_{X_i}(x_i) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$. Therefore,

$$f_{X_1, X_2, \dots X_n}(x_1, x_2, \dots, x_n) = \frac{1}{(\sqrt{2\pi})^n \sigma^n} e^{-\frac{(x_1-\mu)^2 + \dots + (x_n-\mu)^2}{2\sigma^2}}$$

So $f_{X_1, X_2, \dots X_n}(x_1, x_2, \dots, x_n \mid \mu, \sigma) = (2\pi\sigma^2)^{-\frac{n}{2}} e^{-\frac{(x_1-\mu)^2 + \dots + (x_n-\mu)^2}{2\sigma^2}}$

so $\log\left(f_{X_1, X_2, \dots X_n}(x_1, x_2, \dots, x_n \mid \mu, \sigma)\right) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}$. ✓

b)

$$\mu_{MLE} = \underset{\mu}{\mathrm{argmax}}\left[-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}\right]$$

Want to find μ that maximizes $-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}$.

Take derivative with respect to μ and set $=0$;

$$\frac{\partial}{\partial\mu}\left[-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}\right] = 0$$

$$\frac{\partial}{\partial\mu}\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right] = 0$$

$$\frac{\partial}{\partial\mu}\left[\sum_{i=1}^{n}(x_i-\mu)^2\right] = 0$$

$$-2\sum_{i=1}^{n}(x_i-\mu) = 0 \implies \sum_{i=1}^{n}(x_i-\mu) = 0$$

So, $\left[\sum_{i=1}^{n}x_i\right] - n\mu = 0$ so $\boxed{\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n}x_i}$

$$\sigma_{MLE} = \underset{\sigma}{\mathrm{argmax}}\left[-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-\mu_{MLE})^2}{2\sigma^2}\right]$$

To find σ_{MLE}, take derivative with respect to σ:

$$\frac{\partial}{\partial\sigma}\left[-\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n}\frac{(x_i-\mu_{MLE})^2}{2\sigma^2}\right] = 0$$

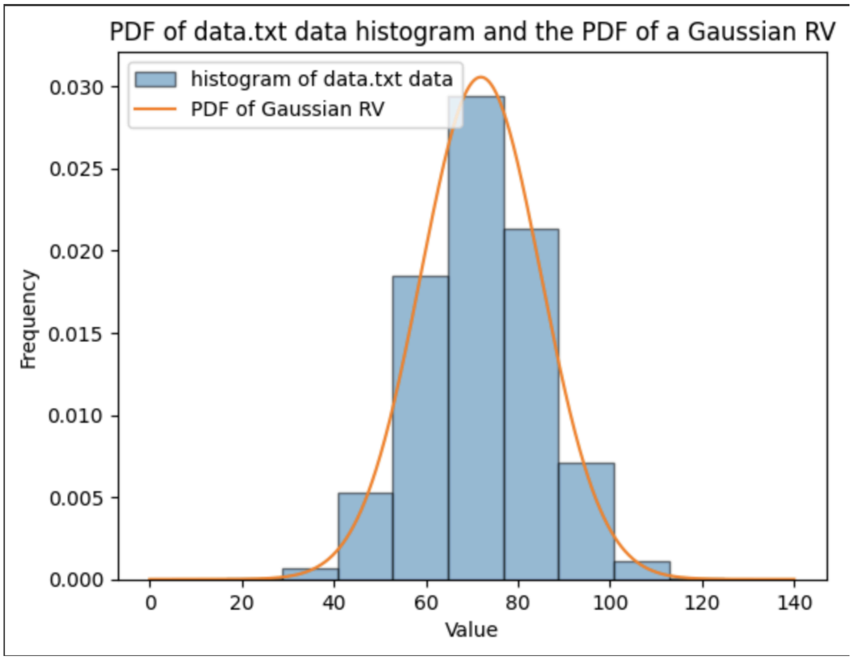
$$-\frac{n}{2}\left(\frac{4\pi\sigma}{2\pi\sigma^2}\right) + (2)\sum_{i=1}^{n}\frac{(x_i-\mu_{MLE})^2}{2\sigma^3} = 0$$

$$\frac{-n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i-\mu_{MLE})^2 = 0$$

$n = \frac{\sum_{i=1}^{n}(x_i-\mu_{MLE})^2}{\sigma^2}$ so $\boxed{\sigma_{MLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_{MLE})^2}}$

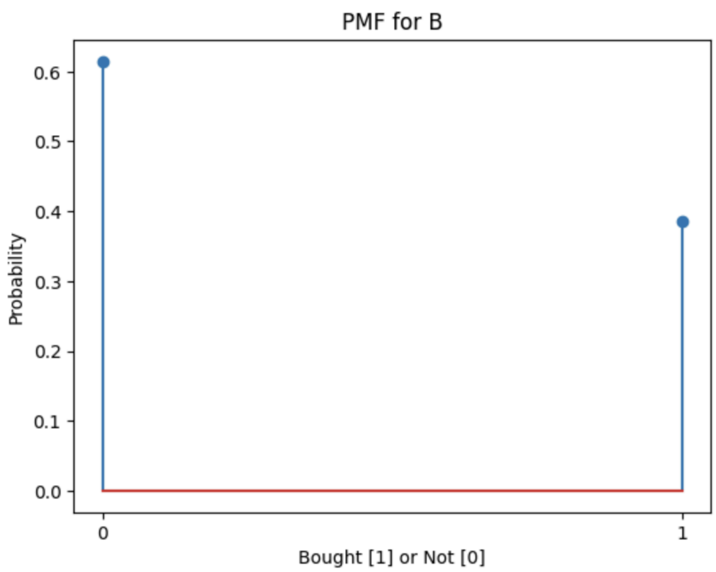
Using Python as a calculator, for the data in data.txt provided I found that $\mu_{MLE} = 71.94377524297302$ and $\sigma_{MLE} = 13.05470055004703$.

c) As we can see below, the μ_{MLE} and σ_{MLE} calculated above fairly accurately approximate the observed distribution from the data.txt data into a normal distribution. (see appendix for code for the following plot)

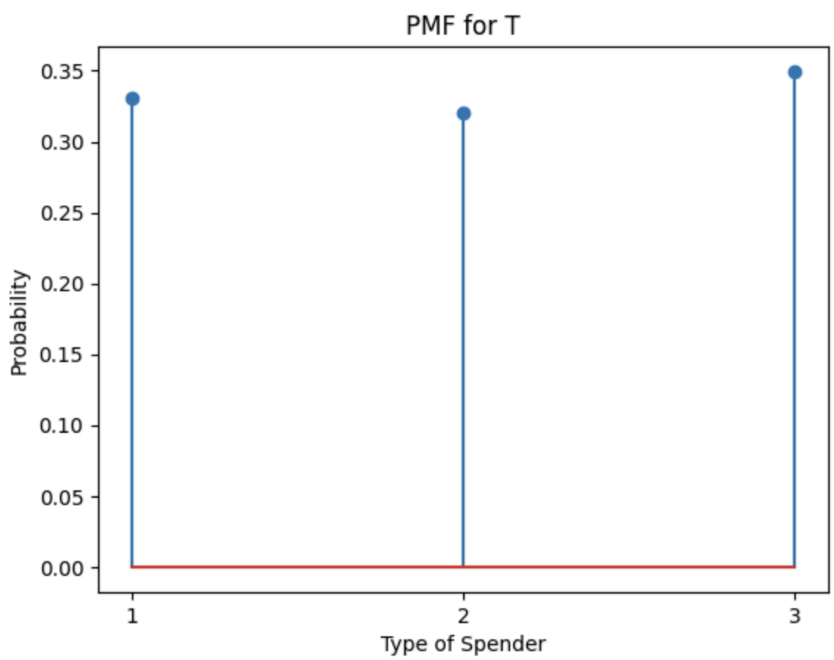


Question 3: *Naïve Bayes Classifier*

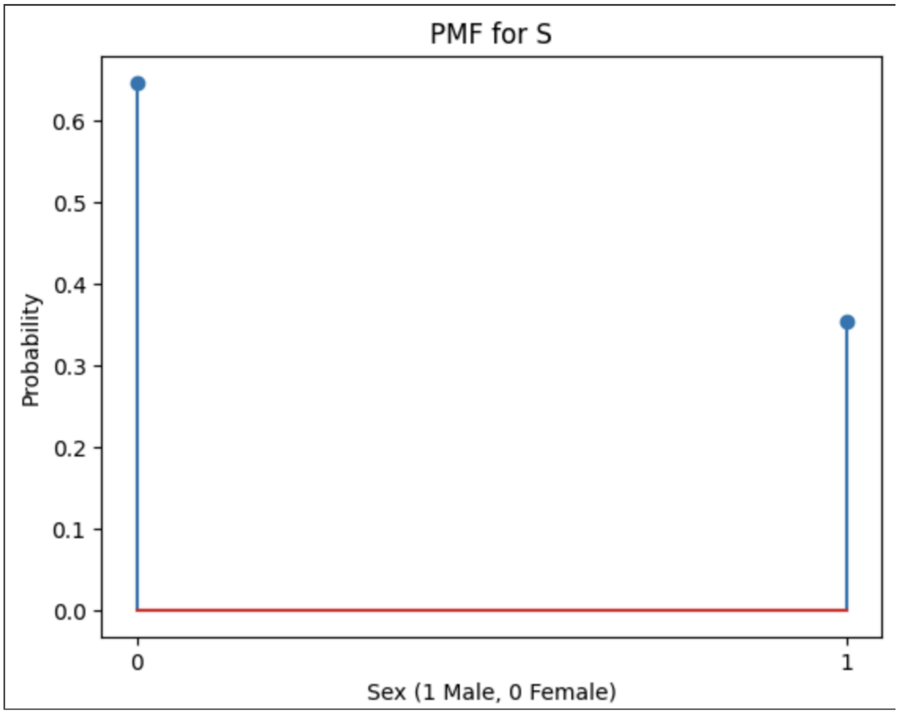
a) By using Python and the pandas library to traverse the user_data.csv file, and by dividing the number of users that have/have not bought the item before by the total number of users, I found that $P(B = 0) = 0.6144306651634723$ and $P(B = 1) = 0.3855693348365276$. The PMF plot is shown below. (see appendix for full code)



Additionally, I found that $P(T = 1) = 0.33032694475760993$, $P(T = 2) = 0.3201803833145434$, and $P(T = 3) = 0.34949267192784667$.



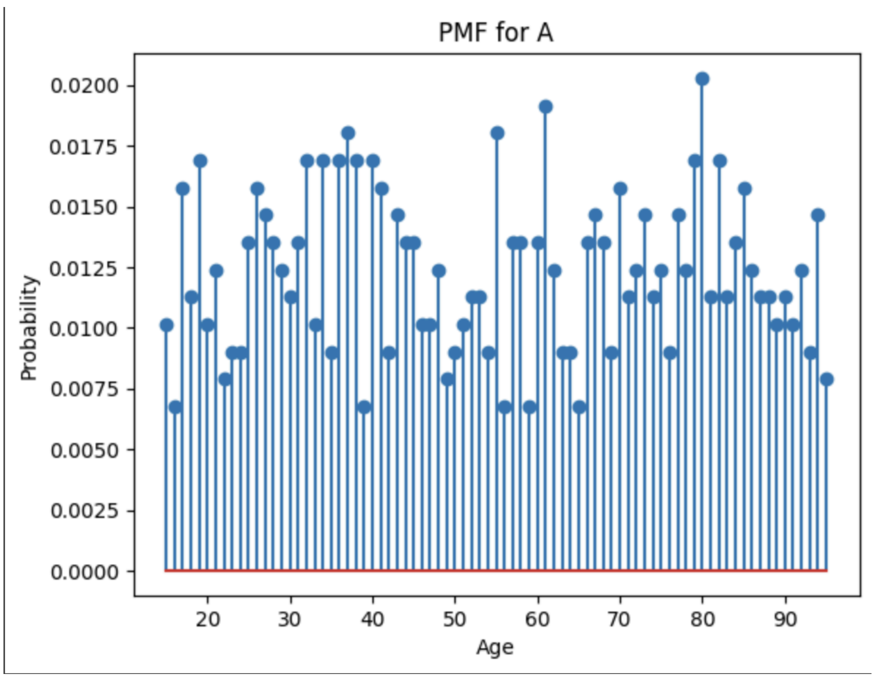
I found that $P(S = 0) = 0.6459977452085682$ and $P(S = 1) = 0.35400225479143177$.



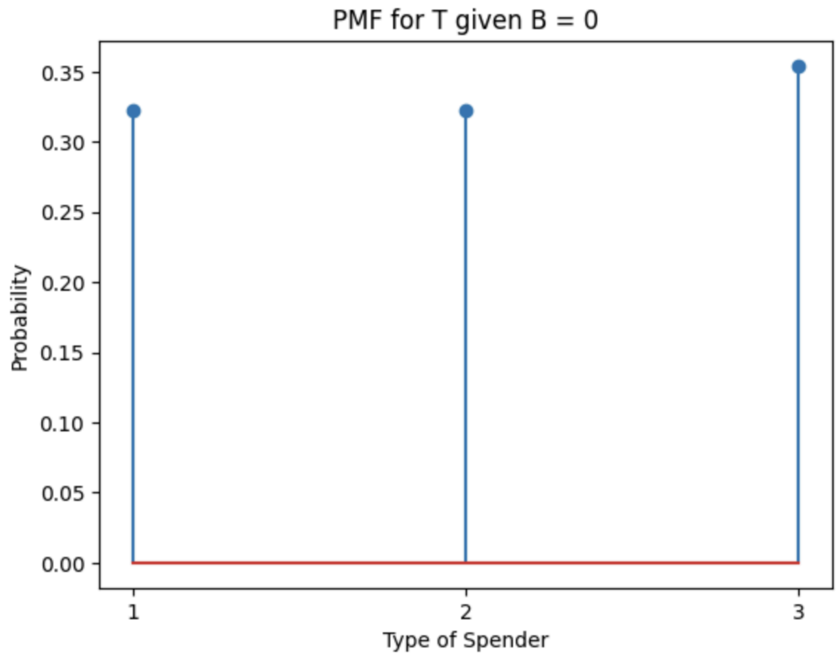
Finally, I found the PMF of the ages from the minimum age $A = 15$ to the maximum age $A = 95$ in the following array, where the value at index 0 is $P(A = 15)$ and the last value of the array is $P(A = 95)$:

```
[0.010146561443066516, 0.006764374295377677,
0.015783540022547914, 0.011273957158962795,
0.016910935738444193, 0.010146561443066516,
0.012401352874859075, 0.007891770011273957,
0.009019165727170236, 0.009019165727170236,
0.013528748590755355, 0.015783540022547914,
0.014656144306651634, 0.013528748590755355,
0.012401352874859075, 0.011273957158962795,
0.013528748590755355, 0.016910935738444193,
0.010146561443066516, 0.016910935738444193,
0.009019165727170236, 0.016910935738444193,
0.018038331454340473, 0.016910935738444193,
0.006764374295377677, 0.016910935738444193,
0.015783540022547914, 0.009019165727170236,
0.014656144306651634, 0.013528748590755355,
0.013528748590755355, 0.010146561443066516,
0.010146561443066516, 0.012401352874859075,
0.007891770011273957, 0.009019165727170236,
0.010146561443066516, 0.011273957158962795,
0.011273957158962795, 0.009019165727170236,
0.018038331454340473, 0.006764374295377677,
0.013528748590755355, 0.013528748590755355,
0.006764374295377677, 0.013528748590755355,
```

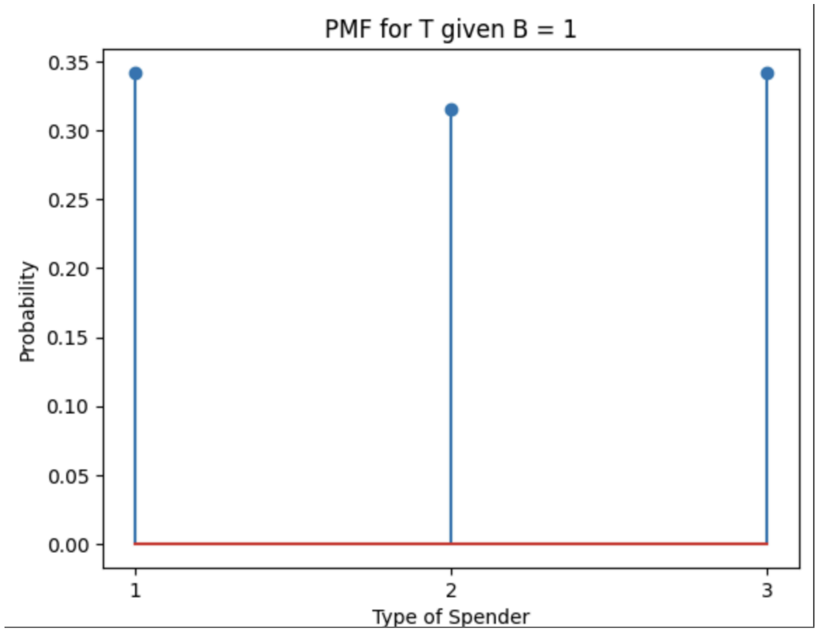
```
0.019165727170236752, 0.012401352874859075,
0.009019165727170236, 0.009019165727170236,
0.006764374295377677, 0.013528748590755355,
0.014656144306651634, 0.013528748590755355,
0.009019165727170236, 0.015783540022547914,
0.011273957158962795, 0.012401352874859075,
0.014656144306651634, 0.011273957158962795,
0.012401352874859075, 0.009019165727170236,
0.014656144306651634, 0.012401352874859075,
0.016910935738444193, 0.020293122886133032,
0.011273957158962795, 0.016910935738444193,
0.011273957158962795, 0.013528748590755355,
0.015783540022547914, 0.012401352874859075,
0.011273957158962795, 0.011273957158962795,
0.010146561443066516, 0.011273957158962795,
0.010146561443066516, 0.012401352874859075,
0.009019165727170236, 0.014656144306651634,
0.007891770011273957].
```



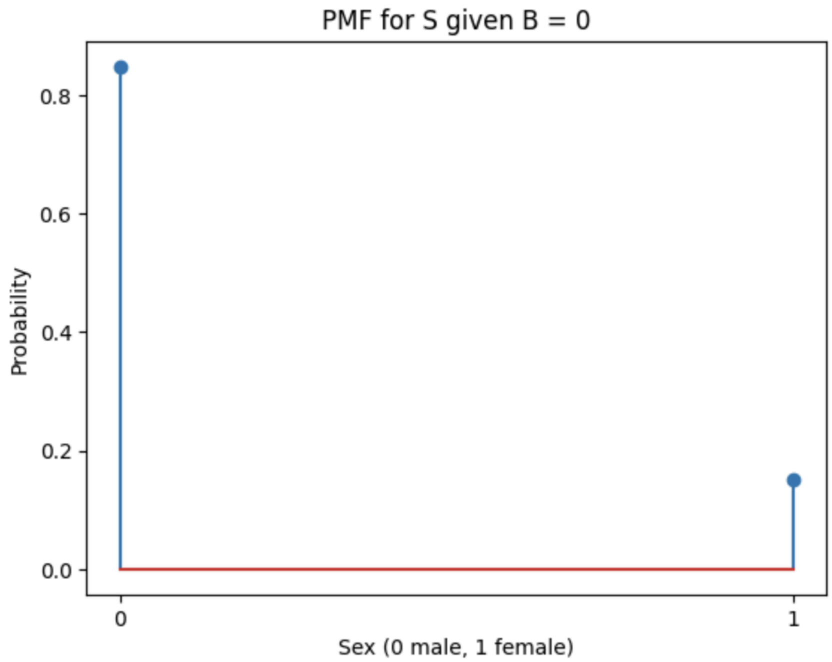
b) From traversing the csv file in Python and finding the probabilities that B = 0 and T = whichever type I was focusing on (or P(B = 0, T)), then dividing these probabilities by the probability that B = 0 as according to the conditional probability law, I found that $P(T = 1|B = 0) = 0.3229357798165138$, $P(T = 2|B = 0) = 0.3229357798165138$, and $P(T = 3|B = 0) = 0.3541284403669725$. (see appendix for full code)



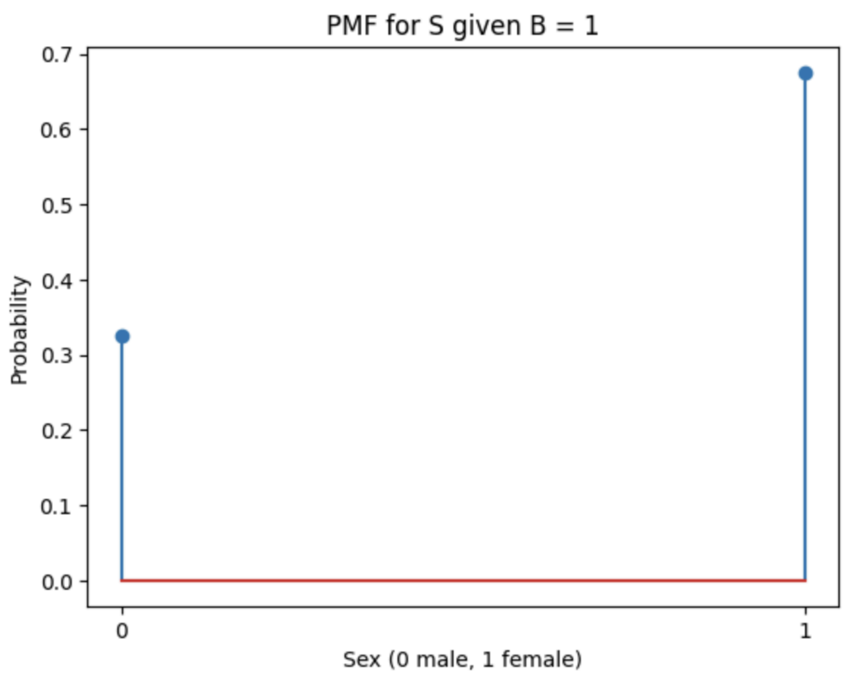
I similarly found that $P(T = 1|B = 1) = 0.34210526315789475$, $P(T = 2|B = 1) = 0.3157894736842105$, and $P(T = 3|B = 1) = 0.34210526315789475$.



I estimated that $P(S = 0|B = 0) = 0.8477064220183487$ and $P(S = 1|B = 0) = 0.15229357798165138$.



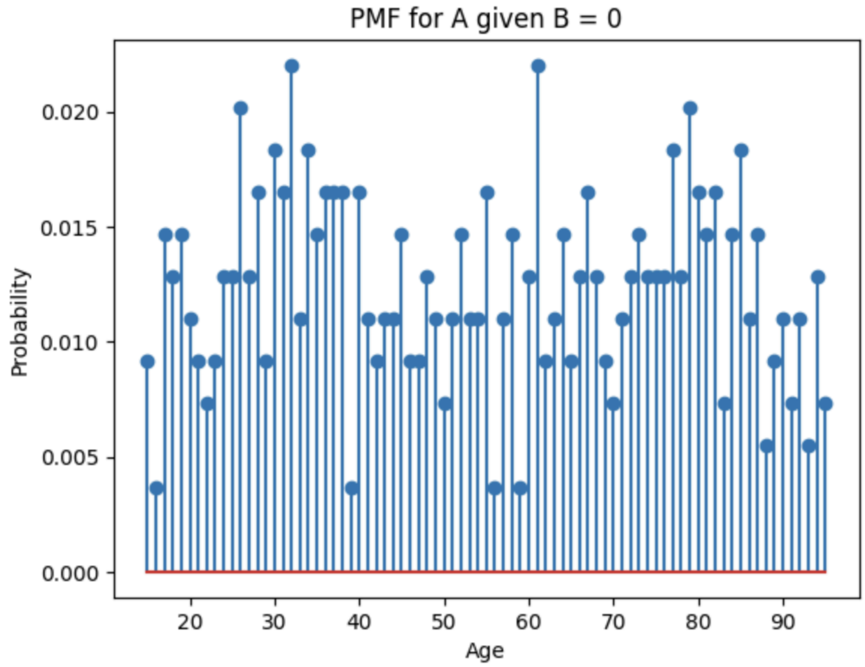
I also estimated that $P(S = 0|B = 1) = 0.32456140350877194$ and $P(S = 1|B = 1) = 0.6754385964912281$.



The probabilities that the person would be aged 15-95 (in order) given that B = 0 are given in the array below:

```
[0.009174311926605505 , 0.003669724770642202 ,
0.014678899082568808 , 0.012844036697247707 ,
0.014678899082568808 , 0.011009174311926606 ,
0.009174311926605505 , 0.007339449541284404 ,
```

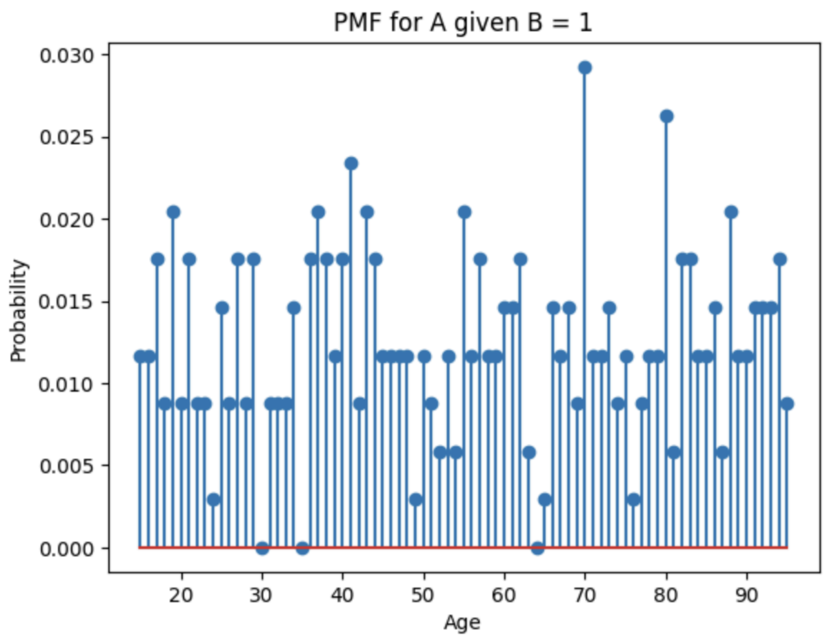
0.009174311926605505 , 0.012844036697247707 ,
0.012844036697247707 , 0.02018348623853211 ,
0.012844036697247707 , 0.01651376146788991 ,
0.009174311926605505 , 0.01834862385321101 ,
0.01651376146788991 , 0.022018348623853212 ,
0.011009174311926606 , 0.01834862385321101 ,
0.014678899082568808 , 0.01651376146788991 ,
0.01651376146788991 , 0.01651376146788991 ,
0.003669724770642202 , 0.01651376146788991 ,
0.011009174311926606 , 0.009174311926605505 ,
0.011009174311926606 , 0.011009174311926606 ,
0.014678899082568808 , 0.009174311926605505 ,
0.009174311926605505 , 0.012844036697247707 ,
0.011009174311926606 , 0.007339449541284404 ,
0.011009174311926606 , 0.014678899082568808 ,
0.011009174311926606 , 0.011009174311926606 ,
0.01651376146788991 , 0.003669724770642202 ,
0.011009174311926606 , 0.014678899082568808 ,
0.003669724770642202 , 0.012844036697247707 ,
0.022018348623853212 , 0.009174311926605505 ,
0.011009174311926606 , 0.014678899082568808 ,
0.009174311926605505 , 0.012844036697247707 ,
0.01651376146788991 , 0.012844036697247707 ,
0.009174311926605505 , 0.007339449541284404 ,
0.011009174311926606 , 0.012844036697247707 ,
0.014678899082568808 , 0.012844036697247707 ,
0.012844036697247707 , 0.012844036697247707 ,
0.01834862385321101 , 0.012844036697247707 ,
0.02018348623853211 , 0.01651376146788991 ,
0.014678899082568808 , 0.01651376146788991 ,
0.007339449541284404 , 0.014678899082568808 ,
0.01834862385321101 , 0.011009174311926606 ,
0.014678899082568808 , 0.005504587155963303 ,
0.009174311926605505 , 0.011009174311926606 ,
0.007339449541284404 , 0.011009174311926606 ,
0.005504587155963303 , 0.012844036697247707 ,
0.007339449541284404].



Similarly, for B = 1:

```
[0.011695906432748537 , 0.011695906432748537 ,
0.017543859649122806 , 0.008771929824561403 ,
0.02046783625730994 , 0.008771929824561403 ,
0.017543859649122806 , 0.008771929824561403 ,
0.008771929824561403 , 0.0029239766081871343 ,
0.014619883040935672 , 0.008771929824561403 ,
0.017543859649122806 , 0.008771929824561403 ,
0.017543859649122806 , 0.0 , 0.008771929824561403 ,
0.008771929824561403 , 0.008771929824561403 ,
0.014619883040935672 , 0.0 , 0.017543859649122806 ,
0.02046783625730994 , 0.017543859649122806 ,
0.011695906432748537 , 0.017543859649122806 ,
0.023391812865497075 , 0.008771929824561403 ,
0.02046783625730994 , 0.017543859649122806 ,
0.011695906432748537 , 0.011695906432748537 ,
0.011695906432748537 , 0.011695906432748537 ,
0.0029239766081871343 , 0.011695906432748537 ,
0.008771929824561403 , 0.005847953216374269 ,
0.011695906432748537 , 0.005847953216374269 ,
0.02046783625730994 , 0.011695906432748537 ,
0.017543859649122806 , 0.011695906432748537 ,
0.011695906432748537 , 0.014619883040935672 ,
0.014619883040935672 , 0.017543859649122806 ,
0.005847953216374269 , 0.0 , 0.0029239766081871343 ,
0.014619883040935672 , 0.011695906432748537 ,
```

```
0.014619883040935672 , 0.008771929824561403 ,
0.029239766081871343 , 0.011695906432748537 ,
0.011695906432748537 , 0.014619883040935672 ,
0.008771929824561403 , 0.011695906432748537 ,
0.0029239766081871343 , 0.008771929824561403 ,
0.011695906432748537 , 0.011695906432748537 ,
0.02631578947368421 , 0.005847953216374269 ,
0.017543859649122806 , 0.017543859649122806 ,
0.011695906432748537 , 0.011695906432748537 ,
0.014619883040935672 , 0.005847953216374269 ,
0.02046783625730994 , 0.011695906432748537 ,
0.011695906432748537 , 0.014619883040935672 ,
0.014619883040935672 , 0.014619883040935672 ,
0.017543859649122806 , 0.008771929824561403].
```



---

c) By summing the probabilities that the ages are less than or equal to 67 given B = 0, I found that $P(A \leq 67|B = 0) = 0.6587156$ and similarly, $P(A > 67|B = 0) = 0.3412844$. $P(A \leq 67|B = 1) = 0.62865497$ and $P(A > 67|B = 1) = 0.37134503$. Therefore,

$$P(T=1, S=0, A\leq 67 \mid B=0) = \frac{P(B=0,\ T=1, S=0, A\leq 67)}{P(B=0)}$$

by Conditional Probability Law.

So $P(B=0, T=1, S=0, A\leq 67) = P(T=1, S=0, A\leq 67 \mid B=0) \cdot P(B=0)$.

By conditional independence,

$$P(T=1, S=0, A\leq 67 \mid B=0) = P(T=1 \mid B=0)\,P(S=0 \mid B=0)\,P(A\leq 67 \mid B=0)$$

So,

$$P(B=0, T=1, S=0, A\leq 67) = P(T=1 \mid B=0)\,P(S=0 \mid B=0)\,P(A\leq 67 \mid B=0)\,P(B=0)$$

So $\boxed{P(B=0, T=1, S=0, A\leq 67) = 0.11079814}$

Similarly,

$$\boxed{P(B=1, T=1, S=0, A\leq 67) = 0.02691358}$$

d)

$$P(T=1, S=0, A\leq 67) = P(B=0, T=1, S=0, A\leq 67) + P(B=1, T=1, S=0, A\leq 67)$$

$= 0.13771172$

$$\boxed{P(B=0 \mid T=1, S=0, A\leq 67) = \frac{P(B=0,\ T=1, S=0, A\leq 67)}{P(T=1, S=0, A\leq 67)} = 0.8045658}$$

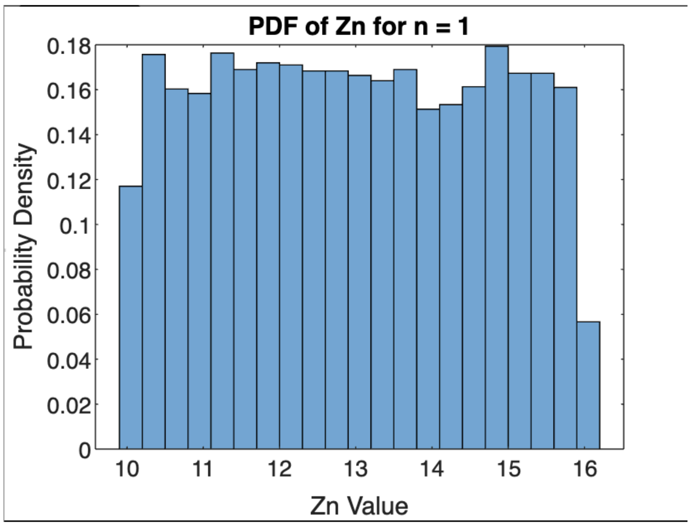
Similarly,

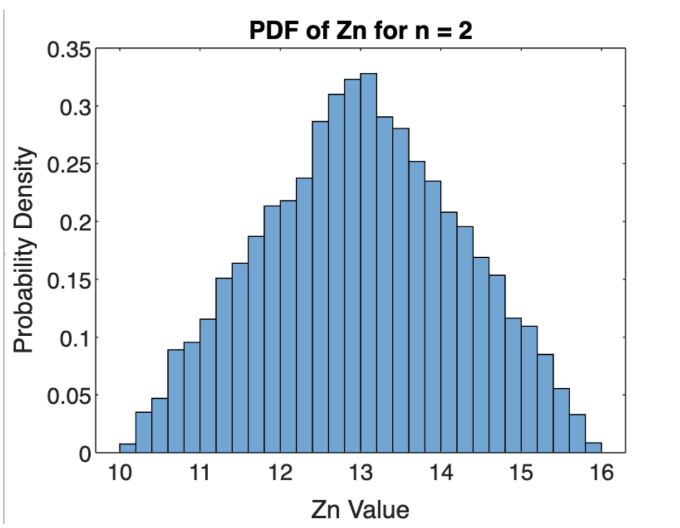
$$\boxed{P(B=1 \mid T=1, S=0, A\leq 67) = 0.1954342}$$

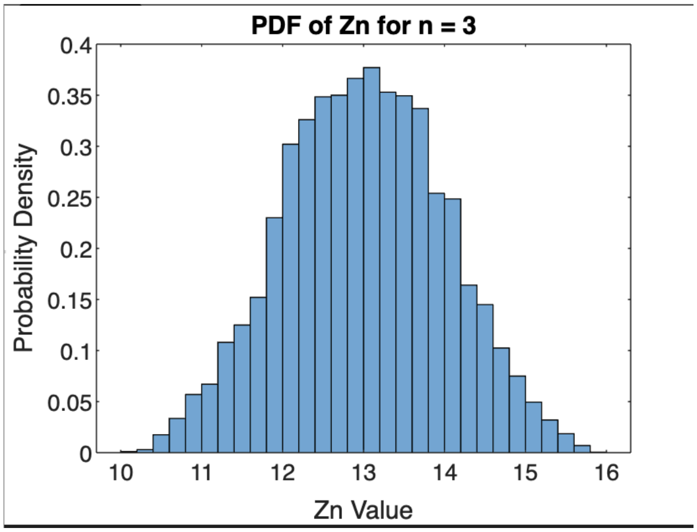
Therefore, the probability that someone already bought the item before given that they were female, a large spender, and less than or equal to the age of 67 is 0.1954342, while the probability that someone hasn't bought the item before given that they were female, a large spender, and less than or equal to the age of 65 is 0.8045658. As such, there is a 0.1954342 chance that a female whose age is below or equal to 67 and who is a large spender will buy this product, based on historical data.

Question 4: *Central Limit Theorem*

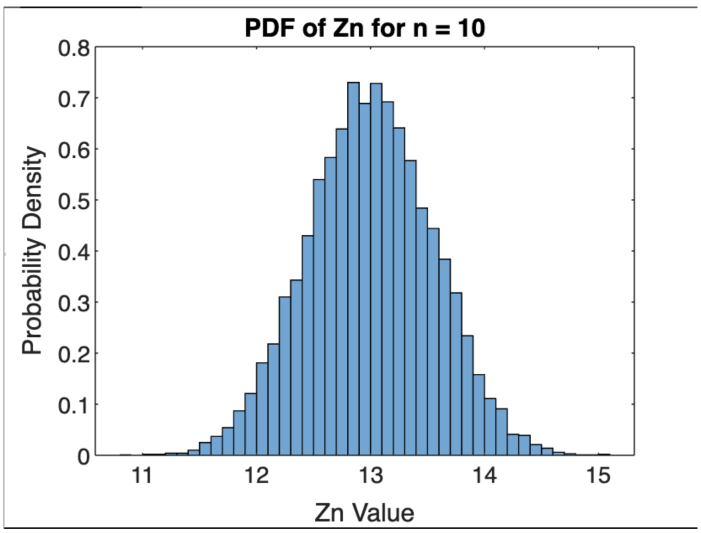
a) (see appendix for full code) To plot the following, I sampled n random numbers between 10–16; each random number is an outcome of the random variable X_i; then I divided the sum of these random numbers by n, and did this t = 10000 times to get the samples of Z_n. Then I plotted the normalized pdf of the histogram of Z_n.



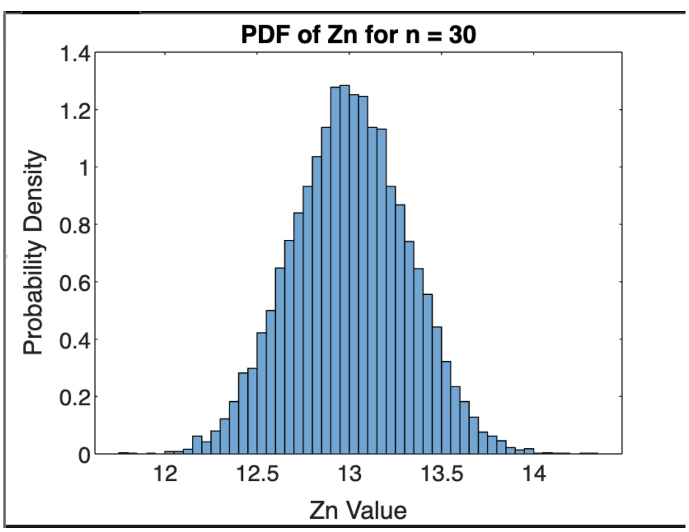




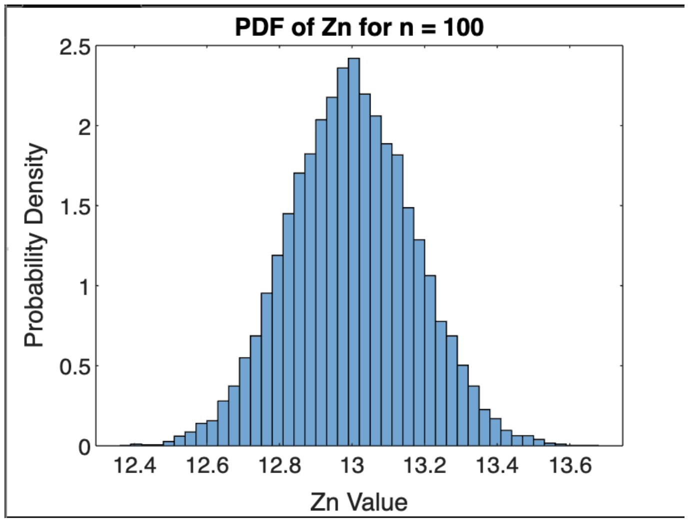
PDF of Zn for n = 3
Probability Density
Zn Value
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
10
11
12
13
14
15
16



PDF of Zn for n = 10
Probability Density
Zn Value
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
11
12
13
14
15



PDF of Zn for n = 30
Probability Density
Zn Value
0
0.2
0.4
0.6
0.8
1
1.2
1.4
12
12.5
13
13.5
14



As n gets larger, the PDF of Zn converges to a Gaussian distribution, which aligns with the Central Limit Theorem that states that as n approaches infinity, the normalized sum of n i.i.d. random variables approaches a gaussian distribution.

b)

X_i is uniform continuous on $(10, 16)$ so

mean of X_i = $\boxed{\mathbb{E}[X_i] = \frac{(10+16)}{2} = 13}$

Variance of X_i: $\mathbb{E}[X_i^2] = \frac{10^2 + (10 \cdot 16) + 16^2}{3} = 172$

$\mathbb{E}[X_i]^2 = 169$ so $\boxed{VAR(X_i) = 172 - 169 = 3}$

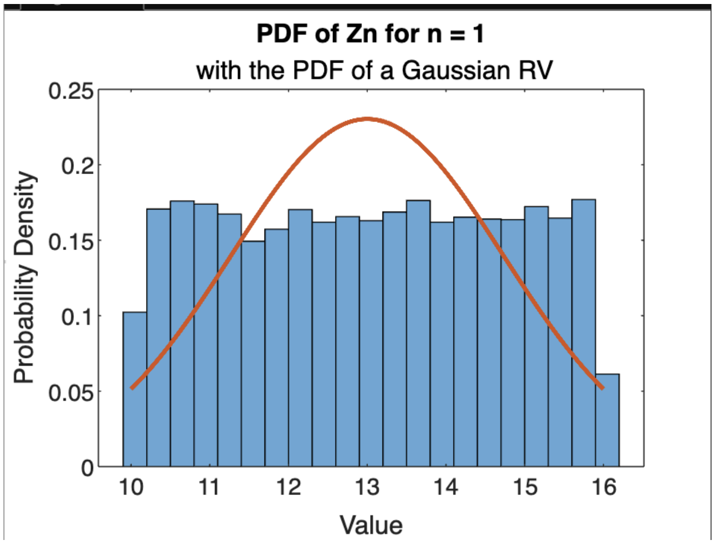
Mean of Z_n: $E[Z_n] = \mathbb{E}\left[\frac{X_1 + \cdots + X_n}{n}\right] = \frac{1}{n}\left(\mathbb{E}[X_1] + \cdots + \mathbb{E}[X_n]\right)$

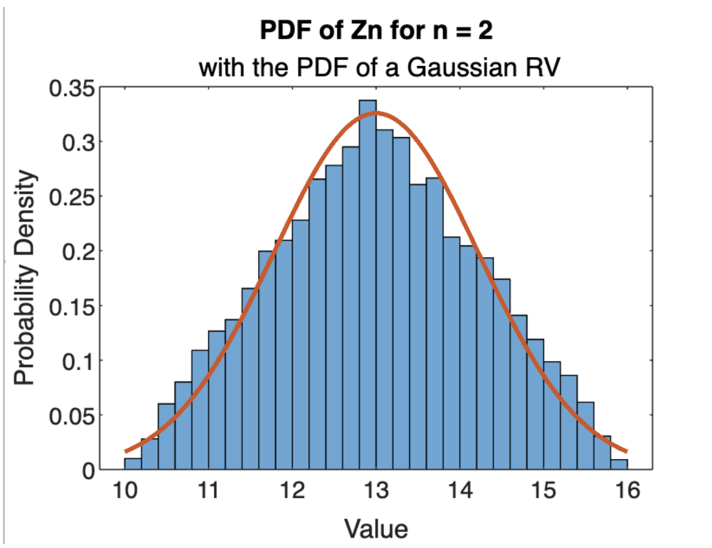
$\boxed{\mathbb{E}[Z_n] = \frac{13n}{n} = 13}$

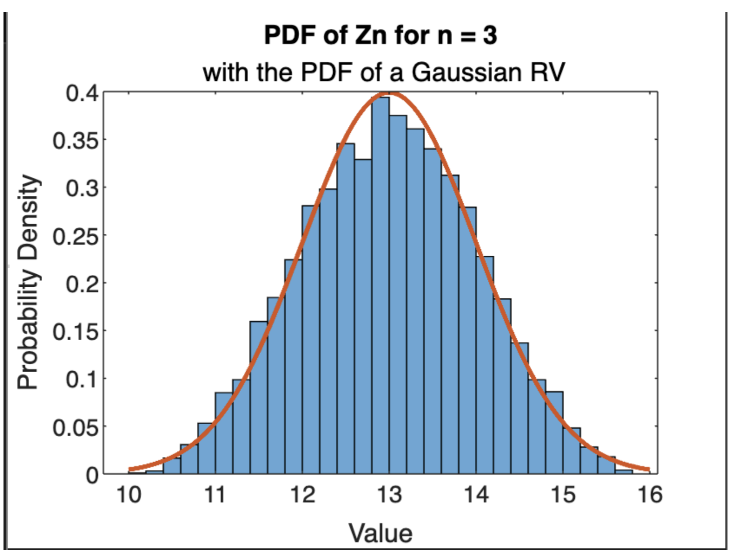
Variance of Z_n: since $X_1, \ldots X_n$ are i.i.d,

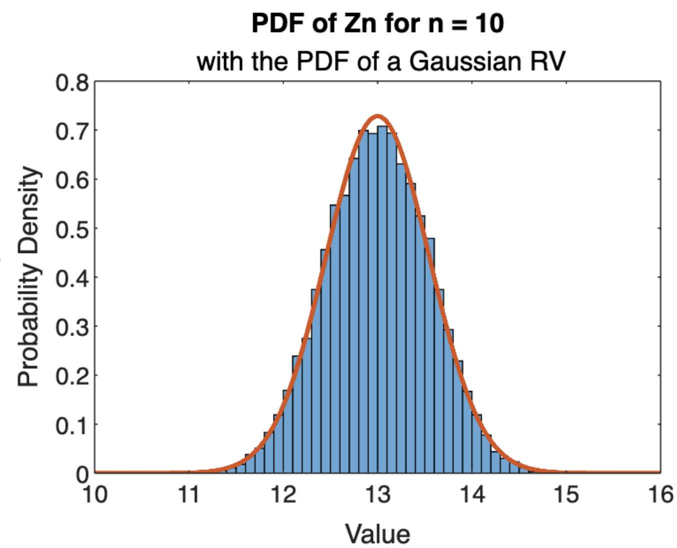
$\boxed{VAR(Z_n) = \frac{VAR(X_i)}{n} = \frac{3}{n}}$.

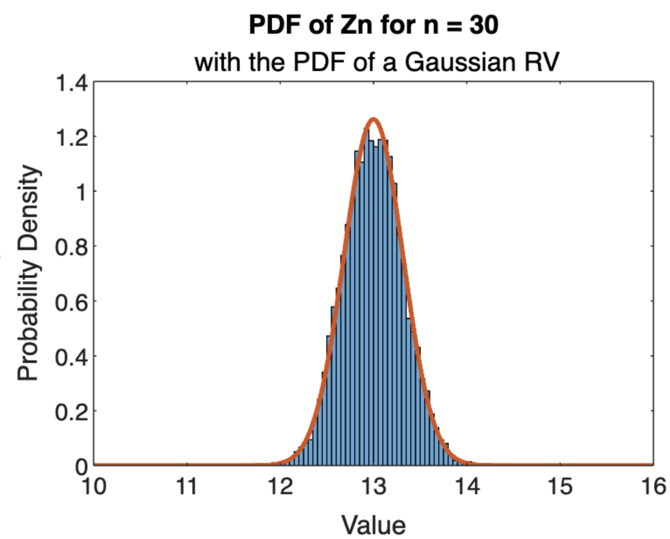
c) The plots of the PDFs of sampled Z_n, along with the theoretical Gaussian pdf with the mean and variance calculated above, are shown below.

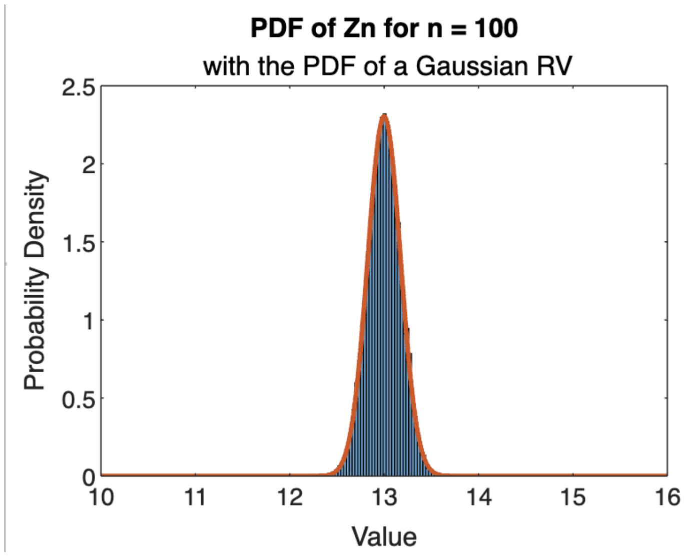

PDF of Zn for n = 1
with the PDF of a Gaussian RV
Probability Density
Value


PDF of Zn for n = 2
with the PDF of a Gaussian RV
Probability Density
Value


PDF of Zn for n = 3
with the PDF of a Gaussian RV
Probability Density
Value

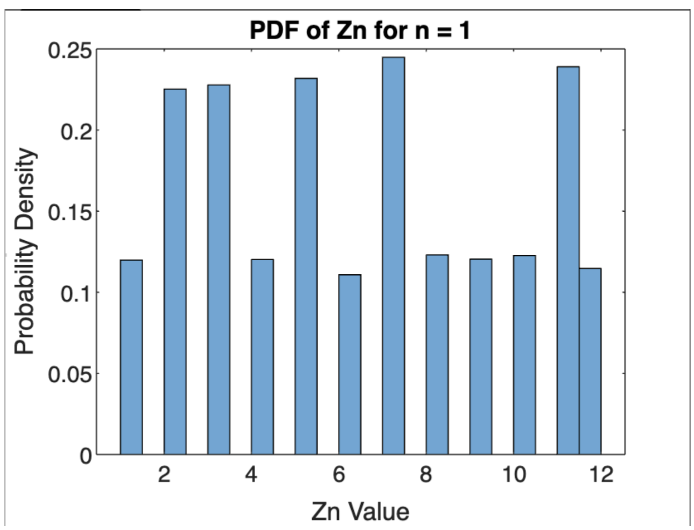

PDF of Zn for n = 10
with the PDF of a Gaussian RV
Probability Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
11
12
13
14
15
16
Value

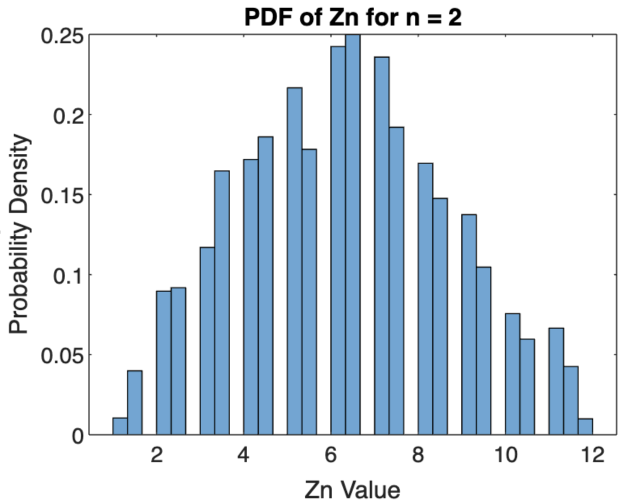

PDF of Zn for n = 30
with the PDF of a Gaussian RV
Probability Density
1.4
1.2
1
0.8
0.6
0.4
0.2
0
10
11
12
13
14
15
16
Value

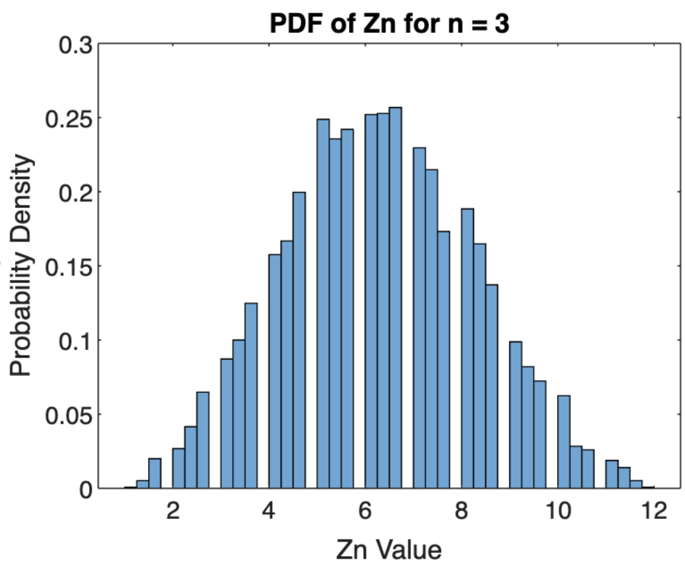


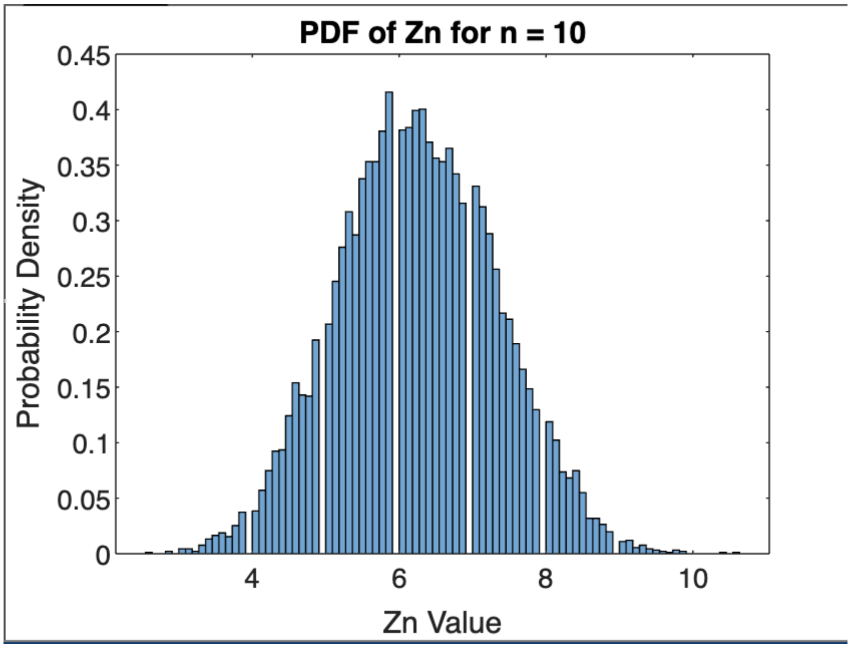
d)

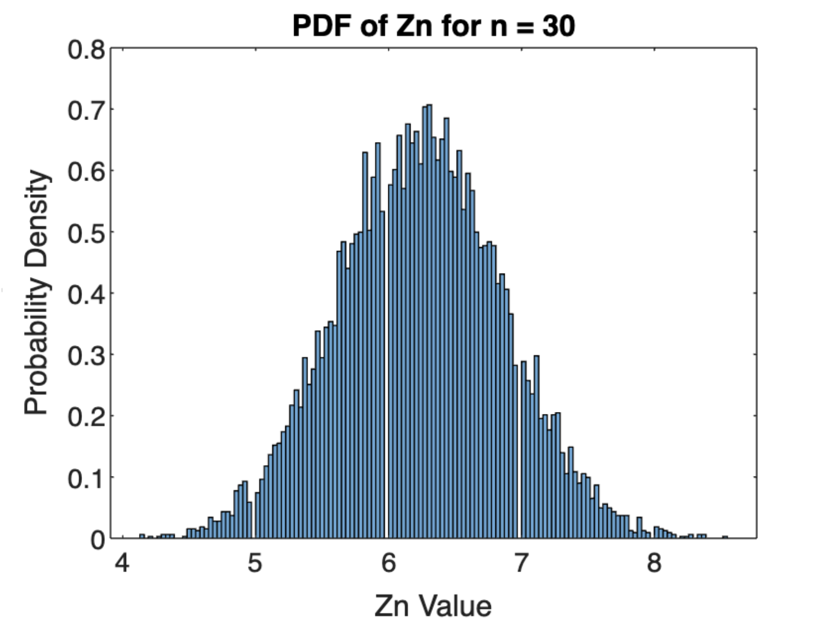
a. The plots for the discrete case of X_i described are shown below.

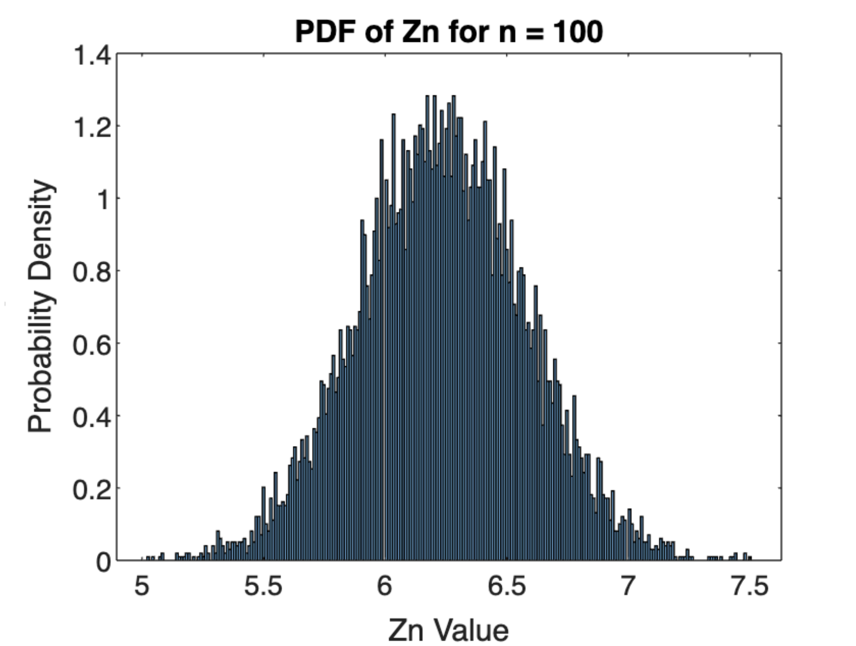



PDF of Zn for n = 2
Probability Density
Zn Value
0.25
0.2
0.15
0.1
0.05
0
2
4
6
8
10
12


PDF of Zn for n = 3
Probability Density
Zn Value
0.3
0.25
0.2
0.15
0.1
0.05
0
2
4
6
8
10
12


PDF of Zn for n = 10
Probability Density
Zn Value
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
4
6
8
10


PDF of Zn for n = 30
Probability Density
Zn Value
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
4
5
6
7
8



Here, too, we observe that as n gets larger the PDF of Zn converges to that of a Gaussian RV with the same mean and variance, as per the Central Limit Theorem. Therefore, the Central Limit Theorem applies to not only continuous i.i.d. random variables; it also applies to discrete random variables.

b.

Mean of X_i: $\mathbb{E}[X_i] = 1 \cdot P(X=1) + 2 \cdot P(X=2) + \cdots + 12 \cdot P(X=12)$

$$\mathbb{E}[X_i] = 1\cdot\left(\tfrac{1}{17}\right) + 2\cdot\left(\tfrac{2}{17}\right) + 3\left(\tfrac{2}{17}\right) + 4\left(\tfrac{1}{17}\right) + 5\left(\tfrac{2}{17}\right) + 6\left(\tfrac{1}{17}\right) + 7\left(\tfrac{2}{17}\right) + 8\left(\tfrac{1}{17}\right) + 9\left(\tfrac{1}{17}\right)$$
$$+ 10\left(\tfrac{1}{17}\right) + 11\left(\tfrac{2}{17}\right) + 12\left(\tfrac{1}{17}\right)$$

$$\boxed{\mathbb{E}[X_i] = \frac{106}{17} \approx 6.235}$$

Variance of X_i: $\mathbb{E}[X_i^2] = 1^2 \cdot P(X=1) + \cdots + 12^2 \cdot P(X=12)$

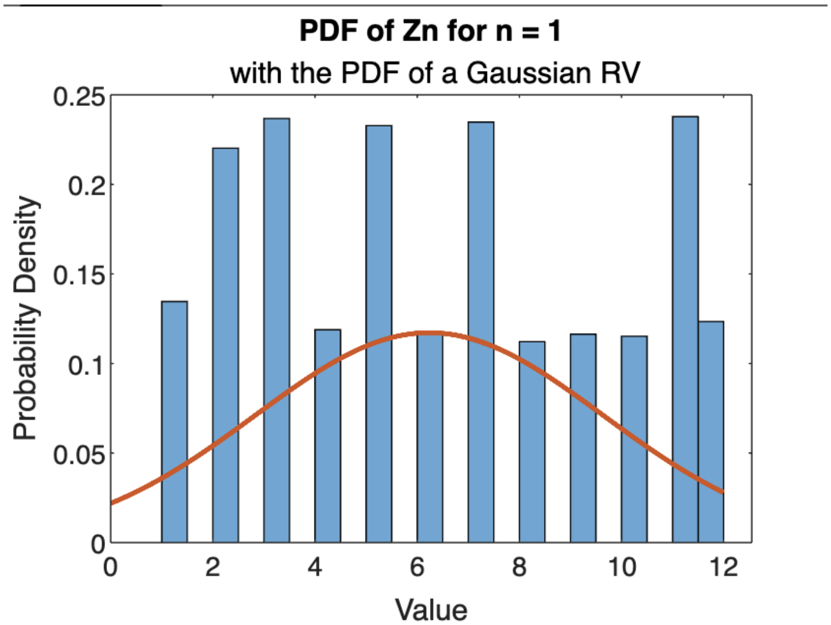
$$\mathbb{E}[X_i^2] = 1^2\cdot\left(\tfrac{1}{17}\right) + 2^2\cdot\left(\tfrac{2}{17}\right) + 3^2\cdot\left(\tfrac{2}{17}\right) + 4^2\cdot\left(\tfrac{1}{17}\right) + 5^2\cdot\left(\tfrac{2}{17}\right) + 6^2\cdot\left(\tfrac{1}{17}\right) + 7^2\cdot\left(\tfrac{2}{17}\right) + 8^2\cdot\left(\tfrac{1}{17}\right)$$
$$+ 9^2\cdot\left(\tfrac{1}{17}\right) + 10^2\cdot\left(\tfrac{1}{17}\right) + 11^2\cdot\left(\tfrac{2}{17}\right) + 12^2\cdot\left(\tfrac{1}{17}\right)$$

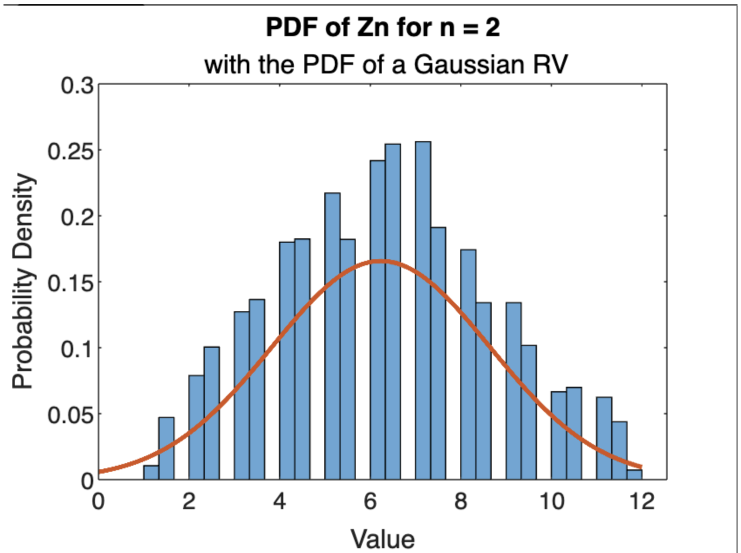
$$\mathbb{E}[X_i^2] = \frac{858}{17} \approx 50.471$$

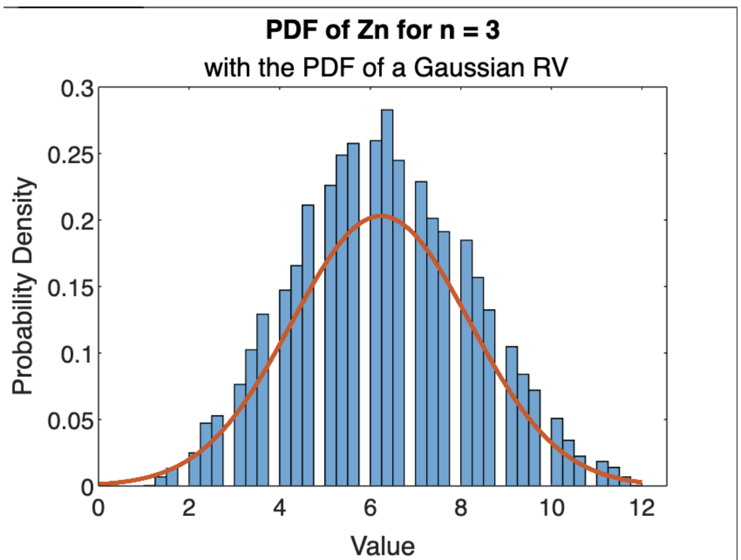
so $\boxed{\mathrm{VAR}(X_i) = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2 \approx 11.592}$

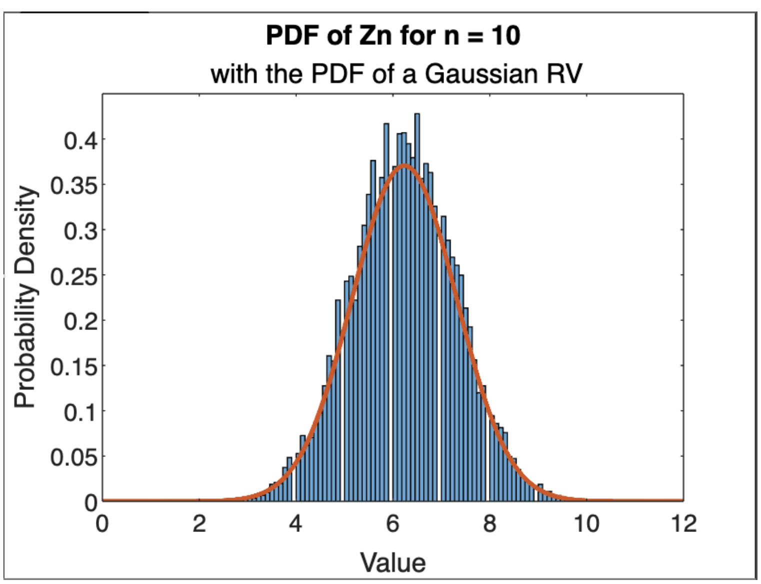
Since X_i's are i.i.d., $\boxed{\mathbb{E}[Z_n] = 6.235 \text{ and } \mathrm{VAR}(Z_n) = \frac{11.592}{n}.}$

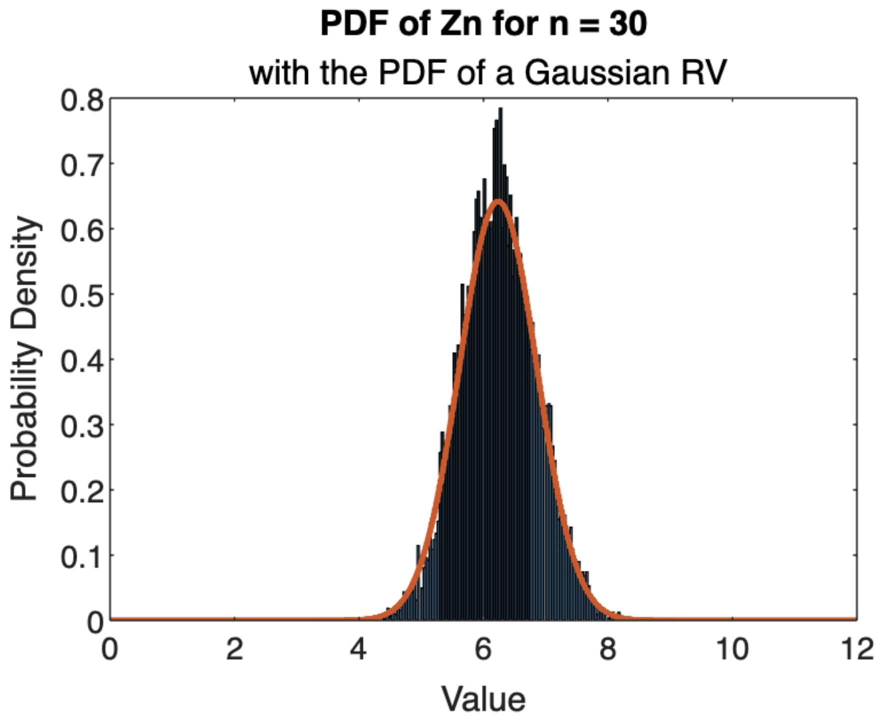
c.

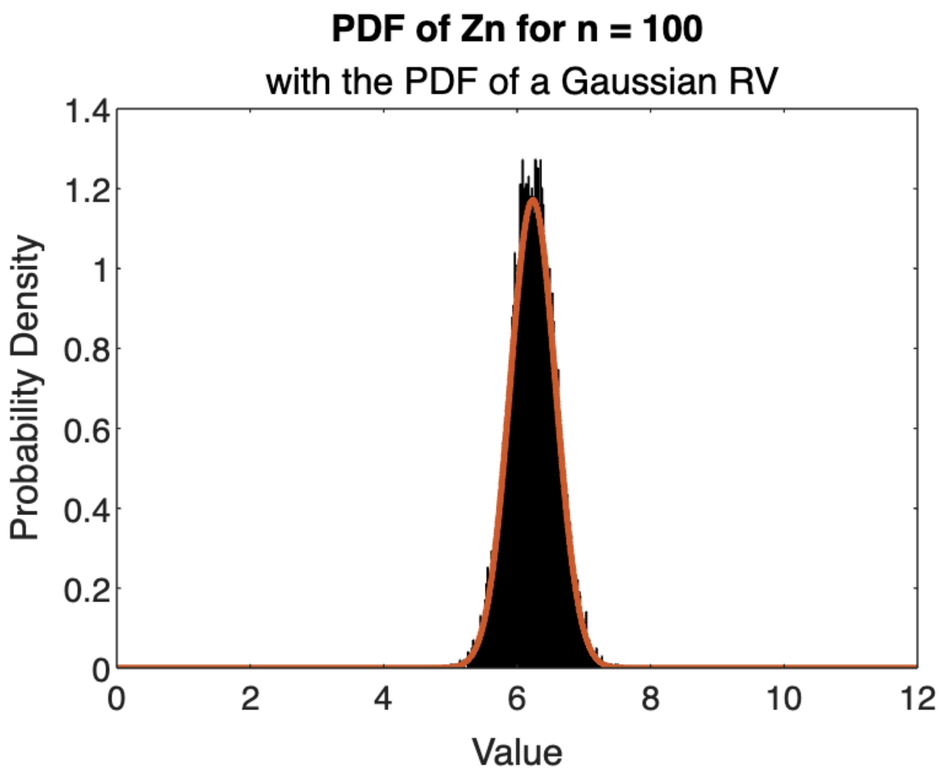



PDF of Zn for n = 2
with the PDF of a Gaussian RV
Probability Density
Value
0
0.05
0.1
0.15
0.2
0.25
0.3
2
4
6
8
10
12


PDF of Zn for n = 3
with the PDF of a Gaussian RV
Probability Density
Value
0
0.05
0.1
0.15
0.2
0.25
0.3
2
4
6
8
10
12

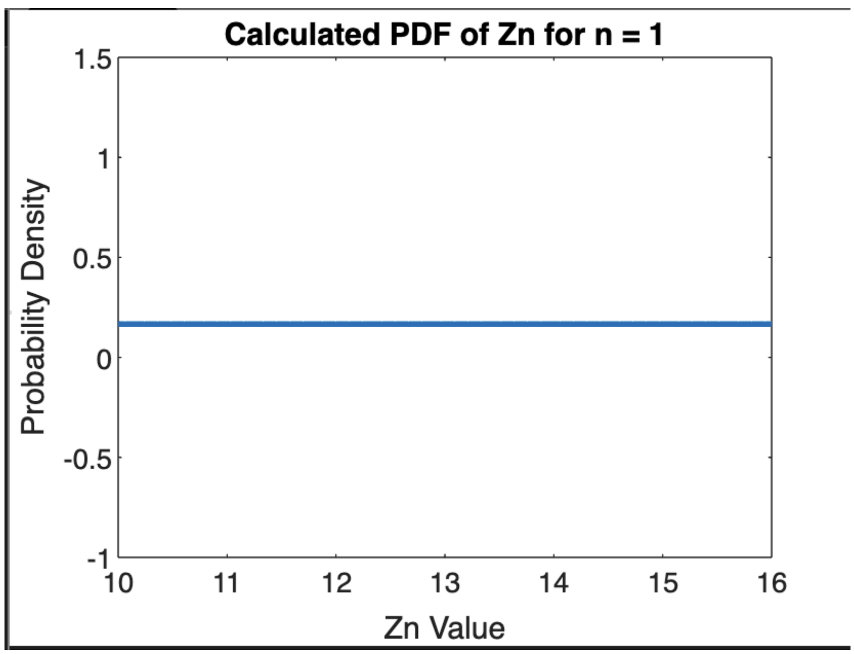

PDF of Zn for n = 10
with the PDF of a Gaussian RV
Probability Density
Value
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
2
4
6
8
10
12

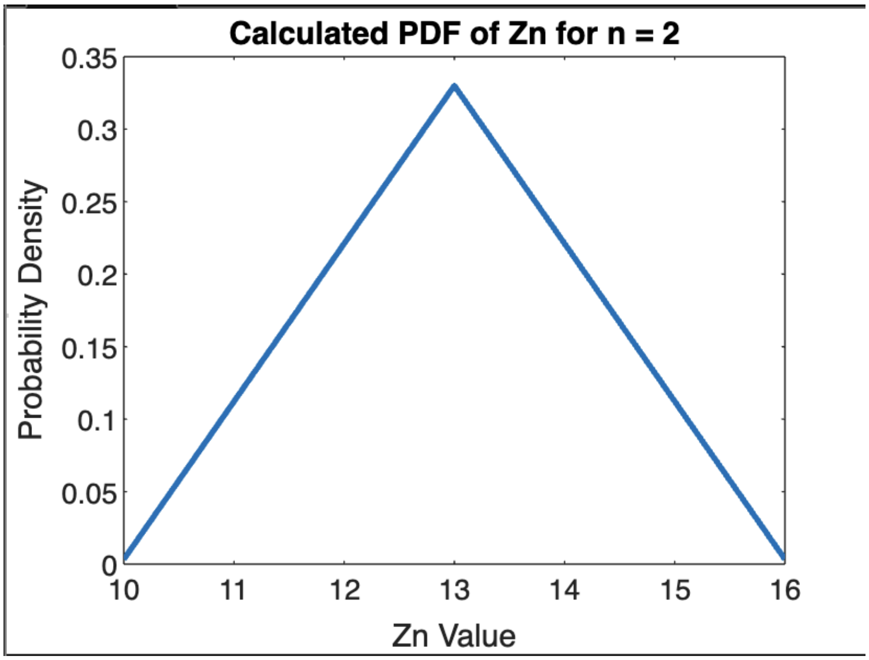

PDF of Zn for n = 30
with the PDF of a Gaussian RV
Probability Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
Value

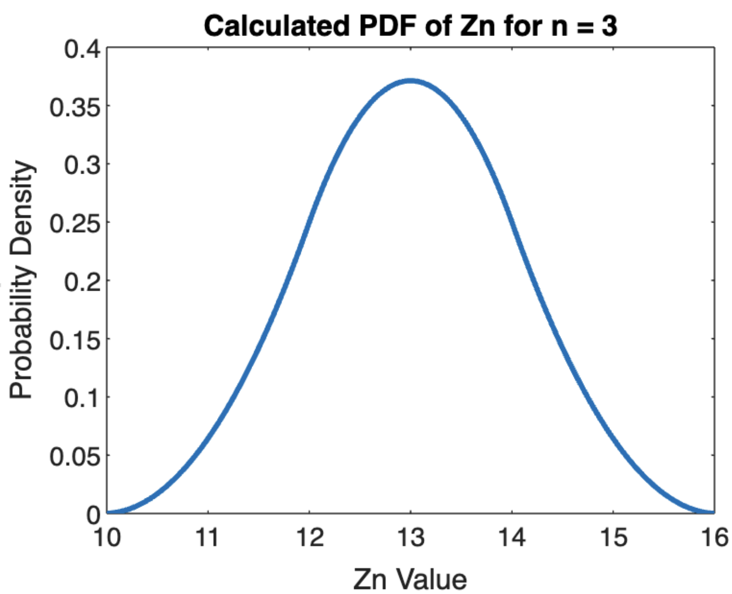

PDF of Zn for n = 100
with the PDF of a Gaussian RV
Probability Density
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
Value

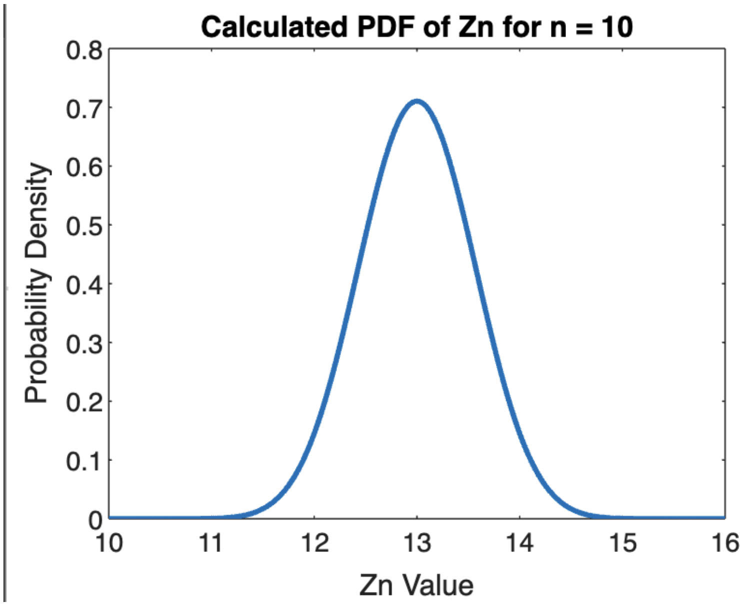
e) Since, as mentioned in lecture, the PDF of $S_n = X_1 + \ldots + X_n$ is the convolution of the PDFs of $X_1, \ldots X_n$, the PDF of Z_n is the normalized $[f_{X_1}(x_1) * f_{X_2}(x_2) * \ldots * f_{X_n}(x_n)]$. Since X_i is uniformly and continuously distributed between (10, 16), $f_{X_i}(x_i) = \frac{1}{16-10} = \frac{1}{6}$ for $10 < x_i < 16$. In order to get the graph for the pdf of X_i, I took 100 samples of random numbers between 10-16 as the x-axis values, and ⅙ as each x-axis value's y-axis value. We can continuously convolve $f_{X_i}(x_i)$ n times, then normalize the resulting graph by dividing the result of all the convolutions by the area under the graph to get the pdf of Z_n.

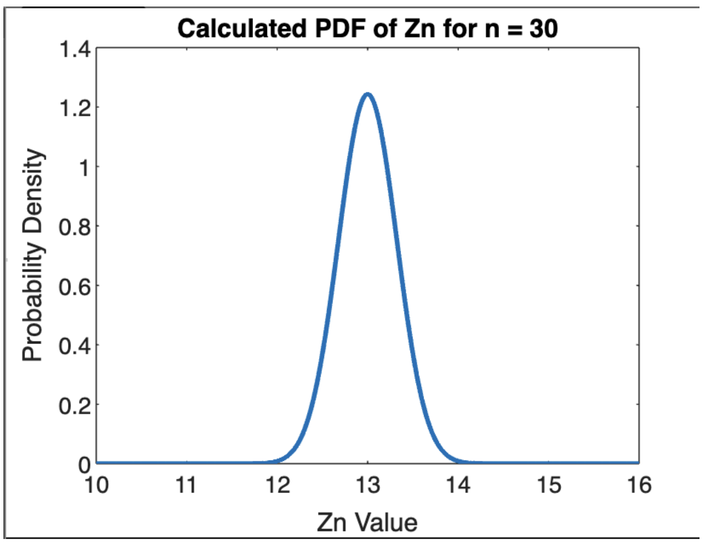
a. The graphs for part a) are below.

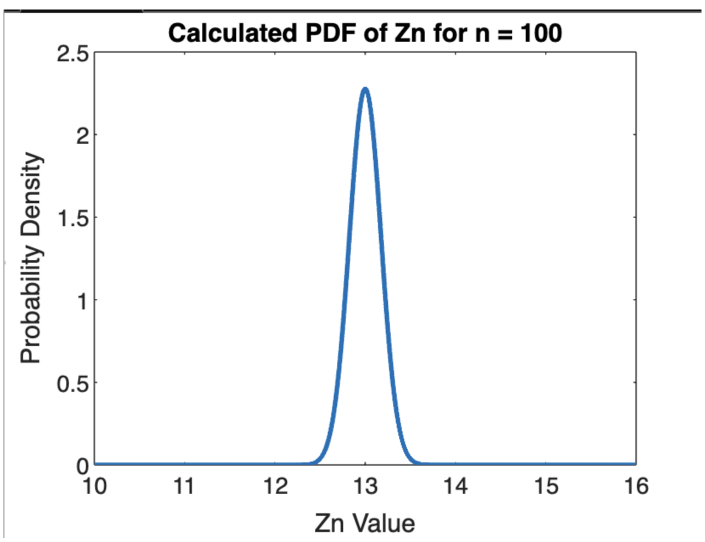



Calculated PDF of Zn for n = 2
Probability Density
Zn Value
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
10
11
12
13
14
15
16


Calculated PDF of Zn for n = 3
Probability Density
Zn Value
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
10
11
12
13
14
15
16

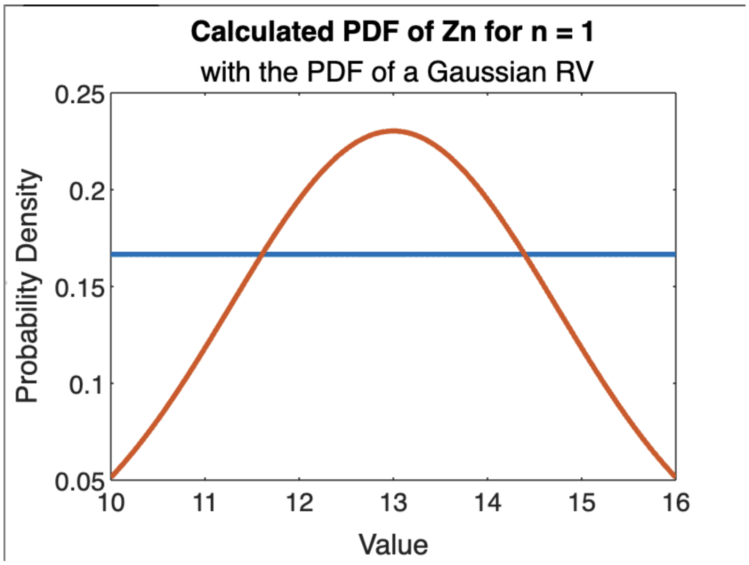

Calculated PDF of Zn for n = 10
Probability Density
Zn Value

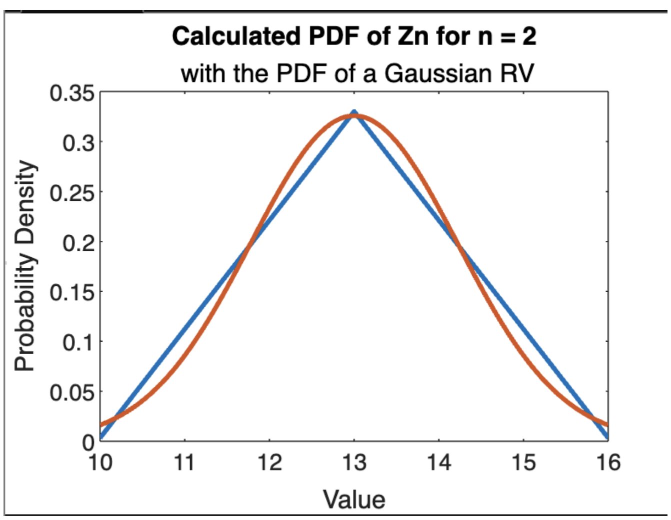

Calculated PDF of Zn for n = 30
Probability Density
Zn Value

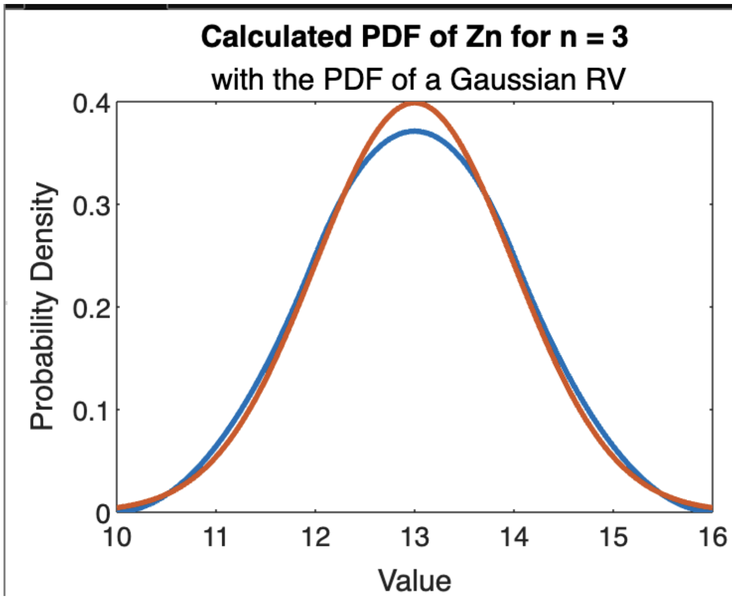

Calculated PDF of Zn for n = 100
Probability Density
Zn Value

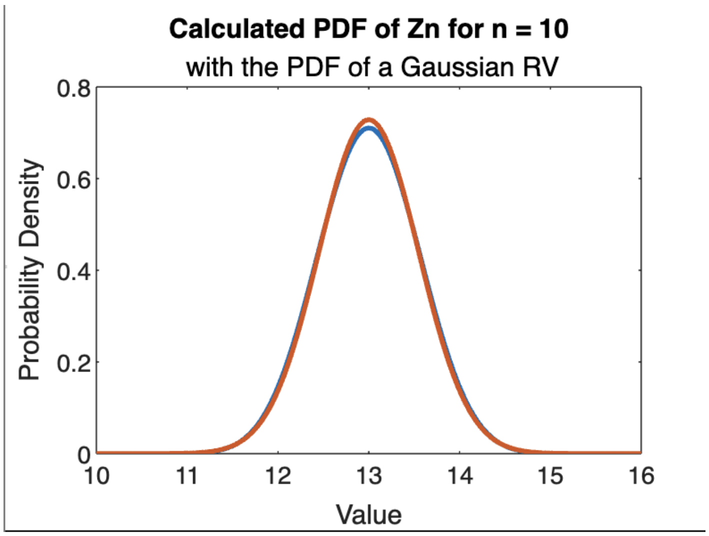
Here, too, we observe the Central Limit Theorem where as n increases, the pdf of Zn converges to that of a Gaussian distribution with the appropriate mean and variance.

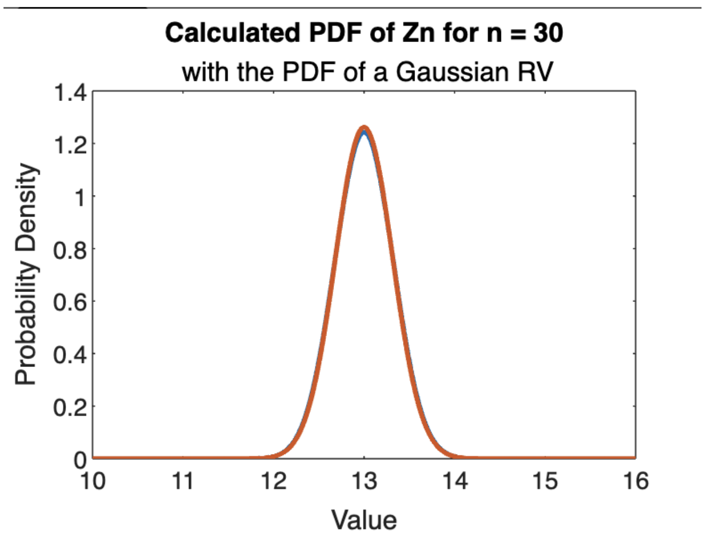
c.

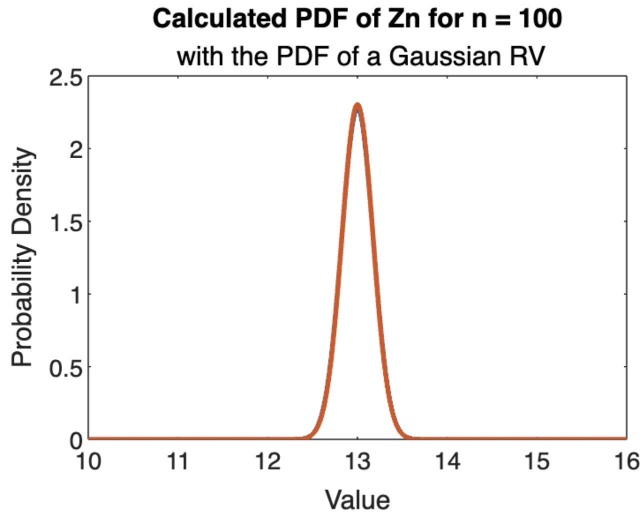





Calculated PDF of Zn for n = 3
with the PDF of a Gaussian RV
Probability Density
0.4
0.3
0.2
0.1
0
10
11
12
13
14
15
16
Value


Calculated PDF of Zn for n = 10
with the PDF of a Gaussian RV
Probability Density
0.8
0.6
0.4
0.2
0
10
11
12
13
14
15
16
Value


Calculated PDF of Zn for n = 30
with the PDF of a Gaussian RV
Probability Density
1.4
1.2
1
0.8
0.6
0.4
0.2
0
10
11
12
13
14
15
16
Value


Calculated PDF of Zn for n = 100
with the PDF of a Gaussian RV
Probability Density
0
0.5
1
1.5
2
2.5
10
11
12
13
14
15
16
Value

Appendix

Question 1 Code (MATLAB):

a) <u>ece131aproject1a.m</u>: (change value of *numTosses* for each value of t tosses)

```
numTosses = 100000;
X1=randi([1, 12], 1, numTosses);
histogram(X1);
title(['t = ', num2str(numTosses), ' tosses']);
xlabel('Outcome');
ylabel('# of times');
```

d) <u>ece131aproject1d.m</u> (change value of *numTosses* for each value of t tosses)

```
pickFrom=[1, 2, 2, 3, 3, 4, 5, 5, 6, 7, 7, 8, 9, 10, 11, 11, 12];
numTosses = 100000;
X1=pickFrom(randi([1, 17], 1, numTosses));
histogram(X1);
title(['t = ', num2str(numTosses), ' tosses']);
xlabel('Outcome');
ylabel('# of times');
```

Question 2 Code (Python):

```
import numpy as np
import matplotlib.pyplot as plt
import random
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
# 2
# 2b)
data = np.loadtxt('/content/drive/My Drive/Academics/2023-2024/EC
ENGR 131A/data.txt')
n = np.size(data)

meanMLE = np.sum(data) / n
print("Mean MLE: ", meanMLE)
sumSquares = 0
for xi in data:
 sumSquares += (xi - meanMLE) * (xi - meanMLE)
```

```python
sigmaMLE = np.sqrt(sumSquares / n)
print("Sigma MLE: ", sigmaMLE)

#2c
plt.hist(data, edgecolor='black', alpha=0.5, density = True, label =
"histogram of data.txt data")
x = np.linspace(0, 140, 10000)
pdf = (1/(2 * np.pi * sigmaMLE**2)**0.5) * (np.exp(-(x -
meanMLE)**2/(2 * sigmaMLE**2)))
plt.plot(x, pdf, label = "PDF of Gaussian RV")
plt.title("PDF of data.txt data histogram and the PDF of a Gaussian
RV")
plt.legend()
plt.xlabel("Value")
plt.ylabel("Frequency")
```

Question 3 Code (Python):

```python
#3
# 3a
user_data = pd.read_csv('/content/drive/My
Drive/Academics/2023-2024/EC ENGR 131A/user_data.csv')
print(user_data)
numUsers, numCols = user_data.shape

# find PMF for B
B0 = (user_data['Bought'] == 0).sum() / numUsers
B1 = (user_data['Bought'] == 1).sum() / numUsers
B_in = [0, 1]
B_out = [B0, B1]
print("Never bought before: ", B0)
print("Bought before: ", B1)
plt.stem([0, 1], B_out)
plt.xticks([0, 1])
plt.title("PMF for B")
plt.xlabel("Bought [1] or Not [0]")
plt.ylabel("Probability")
```

```python
# find PMF for T
T1 = (user_data['Spender Type'] == 1).sum() / numUsers
T2 = (user_data['Spender Type'] == 2).sum() / numUsers
T3 = (user_data['Spender Type'] == 3).sum() / numUsers
print("Spender Type 1: ", T1)
print("Spender Type 2: ", T2)
print("Spender Type 3: ", T3)
plt.stem([1, 2, 3], [T1, T2, T3])
plt.xticks([1, 2, 3])
plt.title("PMF for T")
plt.xlabel("Type of Spender")
plt.ylabel("Probability")
```

```python
# find PMF for S
S0 = (user_data['Sex'] == 0).sum() / numUsers
S1 = (user_data['Sex'] == 1).sum() / numUsers
print("Female: ", S0)
print("Male: ", S1)
plt.stem([0, 1], [S0, S1])
plt.xticks([0, 1])
plt.title("PMF for S")
plt.xlabel("Sex (1 Male, 0 Female)")
plt.ylabel("Probability")
```

```python
# find PMF for A
min_age = user_data['Age'].min()
max_age = user_data['Age'].max()
print(min_age)
print(max_age)

age = min_age
agesPMF = []
agesArray = []
while (age <= max_age):
 agesPMF.append((user_data['Age'] == age).sum() / numUsers)
 agesArray.append(age)
 age += 1
```

```
print("PMF of ages: ", agesPMF)

plt.stem(agesArray, agesPMF)
plt.title("PMF for A")
plt.xlabel("Age")
plt.ylabel("Probability")
```

```
# 3b

# conditional PMFs for T, S, and A
T1B0 = 0
T2B0 = 0
T3B0 = 0

T1B1 = 0
T2B1 = 0
T3B1 = 0


S0B0 = 0
S1B0 = 0

S0B1 = 0
S1B1 = 0


agesPMFB0 = np.zeros((len(agesArray), 1))
agesPMFB1 = np.zeros((len(agesArray), 1))

for index, row in user_data.iterrows(): # iterate over each row
 if (row['Bought'] == 0):
   if (row['Spender Type'] == 1):
     T1B0 += 1
   elif (row['Spender Type'] == 2):
     T2B0 += 1
   else:
     T3B0 += 1
```

```
    if (row['Sex'] == 0):
      S0B0 += 1
    else:
      S1B0 += 1

    age = row['Age']
    agesPMFB0[age - 15] += 1

  else:
    if (row['Spender Type'] == 1):
      T1B1 += 1
    elif (row['Spender Type'] == 2):
      T2B1 += 1
    else:
      T3B1 += 1

    if (row['Sex'] == 0):
      S0B1 += 1
    else:
      S1B1 += 1

    age = row['Age']
    agesPMFB1[age - 15] += 1

T1B0 /= (numUsers * B0)
T2B0 /= (numUsers * B0)
T3B0 /= (numUsers * B0)

T1B1 /= (numUsers * B1)
T2B1 /= (numUsers * B1)
T3B1 /= (numUsers * B1)


S0B0 /= (numUsers * B0)
S1B0 /= (numUsers * B0)

S0B1 /= (numUsers * B1)
S1B1 /= (numUsers * B1)
```

```
agesPMFB0 /= (numUsers * B0)
agesPMFB1 /= (numUsers * B1)

print("Condition that B = 0: ")
print("Spender Type 1: ", T1B0)
print("Spender Type 2: ", T2B0)
print("Spender Type 3: ", T3B0)
plt.stem([1, 2, 3], [T1B0, T2B0, T3B0])
plt.xticks([1, 2, 3])
plt.title("PMF for T given B = 0")
plt.xlabel("Type of Spender")
plt.ylabel("Probability")
```

```
print("Condition that B = 1: ")
print("Spender Type 1: ", T1B1)
print("Spender Type 2: ", T2B1)
print("Spender Type 3: ", T3B1)
plt.stem([1, 2, 3], [T1B1, T2B1, T3B1])
plt.xticks([1, 2, 3])
plt.title("PMF for T given B = 1")
plt.xlabel("Type of Spender")
plt.ylabel("Probability")
```

```
print("Condition that B = 0: ")
print("Sex 0 (Male): ", S0B0)
print("Sex 1 (Female): ", S1B0)
plt.stem([0, 1], [S0B0, S1B0])
plt.xticks([0, 1])
plt.title("PMF for S given B = 0")
plt.xlabel("Sex (0 male, 1 female)")
plt.ylabel("Probability")
```

```
print("Condition that B = 1: ")
print("Sex 0 (Male): ", S0B1)
print("Sex 1 (Female): ", S1B1)
plt.stem([0, 1], [S0B1, S1B1])
```

```
plt.xticks([0, 1])
plt.title("PMF for S given B = 1")
plt.xlabel("Sex (0 male, 1 female)")
plt.ylabel("Probability")
```

```
print("Given B = 0: ")
print("PMF of ages: [", end = "")
for element in agesPMFB0:
 print(element[0], ", ", end="")
print("]")

plt.stem(agesArray, agesPMFB0)
plt.title("PMF for A given B = 0")
plt.xlabel("Age")
plt.ylabel("Probability")
```

```
print("Given B = 1: ")
print("PMF of ages: [", end = "")
for element in agesPMFB1:
 print(element[0], ", ", end="")
print("]")

plt.stem(agesArray, agesPMFB1)
plt.title("PMF for A given B = 1")
plt.xlabel("Age")
plt.ylabel("Probability")
```

```
# 3c
Ageless67B0 = 0
Agegreater67B0 = 0

Ageless67B1 = 0
Agegreater67B1 = 0

i = 0
while (i < 53):
 Ageless67B0 += agesPMFB0[i]
 Ageless67B1 += agesPMFB1[i]
```

```
 i+=1

while (i < len(agesArray)):
 Agegreater67B0 += agesPMFB0[i]
 Agegreater67B1 += agesPMFB1[i]
 i+=1

print("Age ≤ 67 given B = 0: ", Ageless67B0)
print("Age > 67 given B = 0: ", Agegreater67B0)
print("Age ≤ 67 given B = 1: ", Ageless67B1)
print("Age > 67 given B = 1: ", Agegreater67B1)

print("P(B=0, T=1, S=0, A≤67) = ", T1B0 * S0B0 * Ageless67B0 * B0)
print("P(B=1, T=1, S=0, A≤67) = ", T1B1 * S0B1 * Ageless67B1 * B1)
```

```
#3d

print("P(T=1, S=0, A≤67) = ", T1B0 * S0B0 * Ageless67B0 * B0 + T1B1 *
S0B1 * Ageless67B1 * B1)
print("P(B=0|T=1, S=0, A≤67) = ", (T1B0 * S0B0 * Ageless67B0 *
B0)/(T1B0 * S0B0 * Ageless67B0 * B0 + T1B1 * S0B1 * Ageless67B1 *
B1))
print("P(B=1|T=1, S=0, A≤67) = ", (T1B1 * S0B1 * Ageless67B1 *
B1)/(T1B0 * S0B0 * Ageless67B0 * B0 + T1B1 * S0B1 * Ageless67B1 *
B1))
```

Question 4 Code (MATLAB):

a) ece131aproject4a.m (change the value of *n* accordingly)

```
t = 10000; % 10^4 samples of Zn
n = 100; % change value of n accordingly
samples = zeros(t, 1);
for i=1:t
   Xn=rand(n, 1);
   Xn = 10 + Xn * 6; % to make it between 10 and 16
   Zn = sum(Xn) / n;
   % this is one sample of Zn
   samples(i) = Zn;
end
histogram(samples, 'Normalization', 'pdf');
```

```
title(['PDF of Zn for n = ', num2str(n)]);
xlabel("Zn Value")
ylabel("Probability Density")
```

c) <u>ece131aproject4c.m</u> (change the value of *n* accordingly)

```
t = 10000; % 10^4 samples of Zn
n = 100; % change value of n accordingly
samples = zeros(t, 1);
for i=1:t
   Xn=rand(n, 1);
   Xn = 10 + Xn * 6; % to make it between 10 and 16
   Zn = sum(Xn) / n;
   % this is one sample of Zn
   samples(i) = Zn;
end
mu = 13;
variance = 3/n;
sigma = variance^0.5;
xi = linspace(10, 16, t); % Range of x values
pdf = normpdf(xi, mu, sigma);
histogram(samples, 'Normalization', 'pdf');
hold on;
plot(xi, pdf, 'Linewidth', 2);
title(['PDF of Zn for n = ', num2str(n)], ' with the PDF of a
Gaussian RV');
xlabel("Value");
ylabel("Probability Density");
%legend("Histogram of Zn", "PDF of a Gaussian RV");
hold off;
```

d)

a. <u>ece131aproject4da.m:</u> (change values of *n* accordingly)

```
pickFrom=[1, 2, 2, 3, 3, 4, 5, 5, 6, 7, 7, 8, 9, 10, 11, 11, 12];
t = 10000; % 10^4 samples of Zn
n = 100; % change value of n accordingly
samples = zeros(t, 1);
for i=1:t
   Xi=pickFrom(randi([1, 17], 1, n));
   Zn = sum(Xi) / n;
   % this is one sample of Zn
   samples(i) = Zn;
end
histogram(samples, 'Normalization', 'pdf', 'BinWidth', (1/(n + 1)));
title(['PDF of Zn for n = ', num2str(n)]);
```

```
xlabel("Zn Value")
ylabel("Probability Density")
```

c. <u>ece131aproject4dc.m:</u> (change values of *n* accordingly)

```
pickFrom=[1, 2, 2, 3, 3, 4, 5, 5, 6, 7, 7, 8, 9, 10, 11, 11, 12];
t = 10000; % 10^4 samples of Zn
n = 100; % change value of n accordingly
samples = zeros(t, 1);
for i=1:t
   Xi=pickFrom(randi([1, 17], 1, n));
   Zn = sum(Xi) / n;
   % this is one sample of Zn
   samples(i) = Zn;
end
mu = 106/17;
variance = (3350/289)/n;
sigma = variance^0.5;
xi = linspace(0, 12, t); % Range of x values
pdf = normpdf(xi, mu, sigma);
histogram(samples, 'Normalization', 'pdf', 'BinWidth', (1/(n + 1)));
hold on;
plot(xi, pdf, 'Linewidth', 2);
title(['PDF of Zn for n = ', num2str(n)], ' with the PDF of a
Gaussian RV');
xlabel("Value");
ylabel("Probability Density");
%legend("Histogram of Zn", "PDF of a Gaussian RV");
hold off;
```

e)

a. <u>ece131aproject4ea.m:</u> (change values of *n* accordingly)

```
n = 100; % change value of n accordingly
t = 100;
Xi = ones(t, 1);
Xi = Xi / 6; %pdfs of Xi's
Zn = Xi;
if n > 1
   for i=2:n
       Zn = conv(Zn, Xi);
   end
end
xi = linspace(10, 16, length(Zn));
areaUnderGraph = trapz(xi, Zn);
Zn = Zn / areaUnderGraph; % normalize Zn
```

```
plot(xi, Zn, 'Linewidth', 2);
title(['Calculated PDF of Zn for n = ', num2str(n)]);
xlabel("Zn Value")
ylabel("Probability Density")
```

c. <u>ece131aproject4ec.m:</u> (change values of *n* accordingly)

```
n = 100; % change value of n accordingly
t = 100;
Xi = ones(t, 1);
Xi = Xi / 6; %pdfs of Xi's
Zn = Xi;
if n > 1
   for i=2:n
       Zn = conv(Zn, Xi);
   end
end
xi = linspace(10, 16, length(Zn));
areaUnderGraph = trapz(xi, Zn);
Zn = Zn / areaUnderGraph; % normalize Zn
mu = 13;
variance = 3/n;
sigma = variance^0.5;
pdf = normpdf(xi, mu, sigma);
plot(xi, Zn, 'Linewidth', 2);
hold on;
plot(xi, pdf, 'Linewidth', 2);
title(['Calculated PDF of Zn for n = ', num2str(n)], ' with the PDF
of a Gaussian RV');
xlabel("Value");
ylabel("Probability Density");
hold off;
```