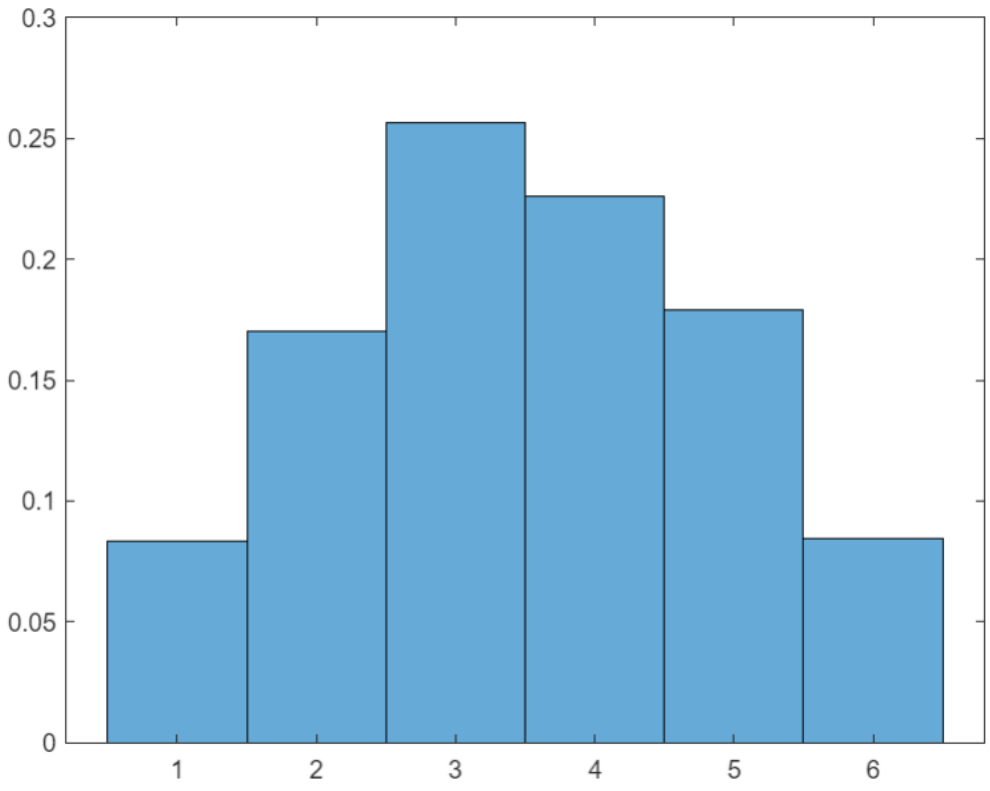
STU33009 – Midterm Assignment - Killian Ronan – 18328687

Q1.

a) The code for my answer can be found below. I initialise the number of items in my dataset (n) to 1017 and convert the imported data to a matrix. I then create an items array which will hold the sum of each row. I loop through the rows in the dataset and add each column to get the total number of items in a basket. I then assign this total in the items array. I use the histogram plot function to plot the PMF. I pass the items array along with 'Normalization' and 'probability' to the histogram function as Matlab allows us to specify normalisation options.

```
n = 1017;
midtermDataset = midterm2021{:,:};
items = zeros(n,1);
for i=1:n
    sumOfItems = midtermDataset(i,1) + midtermDataset(i,2) + midtermDataset(i,3);
    items(i) = sumOfItems;
end
histogram(items,'Normalization','probability')
```



b) To calculate $P(Z_{i,1}= 1)$, we have to obtain the total number of occurrences of 1 at each row where j = 1 (the first column). To do this I wrote a script to count how many occurrences of 1 there was in the first column. The value in column 1 is only ever 0 or 1. This means we can add the value of each column to the count as adding 0's will not affect our result. We then divide the number of occurrences of 1 by n (the total number of rows). This gives us an answer of 0.4887.

```
count = 0;
for i=1:n
```

```
    count = midtermDataset(i,1) + count;
end
finalProb = count/n;
```

c) For CLT:

The 0.95 confidence interval is:

$\mu - 2\sigma \leq P(Zi,1 = 1) \leq \mu + 2\sigma$

We know $\mu = 0.4887$ from our answer in part b). This is our mean or p. p(1-p) is variance. Variance = 0.24987231. We need to calculate 2σ. We can use the following formula to calculate 2σ:

$$2\sigma = 2\sqrt{\frac{\sigma^2}{N}}$$

We also know N = 1017. We can now calculate 2σ = 0.03135. We can now calculate our confidence intervals $[\mu-2\sigma, \mu+2\sigma] = [0.45735, 0.52005]$.

The matlab code I wrote first iterates through the dataset and finds the squared difference from the mean for each data point. The values of the squared differences are summed and then divided by N to calculate the variance σ^2. We then use the equation stated above to calculate 2σ. Then using the first equation mentioned: $\mu - 2\sigma \leq P(Zi,1 = 1)\ X \leq \mu + 2\sigma$, we calculate the confidence interval.

The 0.95 confidence interval using Chebyshev's inequality is $[\mu-\sigma/\sqrt{0.05} *N, \mu+\sigma/\sqrt{0.05} *N]$. We have all the necessary variables to solve this. $[\mu-\sigma/\sqrt{0.05}, \mu+\sigma/\sqrt{0.05}] = [0.4186, 0.5588]$.

The confidence interval obtained from Chebyshev's inequality has a noticeably larger range than the CLT calculated one as it gives an upper bound on the interval.

```
sumOfSquares = 0;
for i=1:n
    valueToSquare = midtermDataset(i,1) - finalProb;
    valueToSquare = valueToSquare * valueToSquare;
    sumOfSquares = sumOfSquares + valueToSquare;
end
variance = sumOfSquares/n;
sigma2 = 2 * sqrt(variance/n);
cltRanges = [finalProb - sigma2; finalProb + sigma2;];
standardDev = sqrt(variance);
chRanges = [finalProb - standardDev/sqrt(0.05 * n); finalProb +
standardDev/sqrt(0.05*n);];
```

d)

In order for the estimate value of P(Zi,1= 1) to have accuracy of +-1% with 95% confidence, we need the formula which we use in part c to equal 0.01.

$$2\sigma = 2\sqrt{\frac{\sigma^2}{N}}$$

This would mean that the estimate is +-0.01 of the mean. This is equivalent to an accuracy of +-1%. We also know this Is with 95% confidence from CLT. $\sigma^2 = 0.24987231$.

$| 2\sigma = 2\sqrt{\frac{\sigma^2}{N}}$:

= 0.1

Therefore, $N = \sigma^2/(0.1/2)^2$.

This gives us N = 9994.8924.

Q2.

a) I began by initialising some variables to count the number of occurrences of 0, 1, 2 or 3 in the 2nd column. I then iterate through the dataset checking the values of the 2nd column. I use a switch statement to check each iteration and increment the relevant counter if the value in column 1 of the same row is set to 1. For example, if the current iterations value for column 2 is 2, the code under case 2 is executed, column 1 is then checked. If column 1 is 1, the count2 variable increments, otherwise it does not. The total count is incremented either way (whether column 1 is 1 or 0). The same process is followed for all code routes. Finally, once the for loop has completed, each mean is calculated by dividing the number of occurrences by the total count. The result is then stored in a 2x4 table where the first column is the value at column 2, and the second column is the mean from the first column.

At J = 2	Mean
0	1
1	0.6917
2	0.2025
3	0

```
count0 = 0;
count1 = 0;
count2 = 0;
count3 = 0;
totalCount = zeros(4,1);
for i=1:n
    switch midtermDataset(i,2)
        case 0
            if(midtermDataset(i,1) == 1)
                count0 = count0+1;
            end
            totalCount(1) = totalCount(1)+1;
        case 1
            if(midtermDataset(i,1) == 1)
                count1 = count1+1;
            end
            totalCount(2) = totalCount(2)+1;
        case 2
            if(midtermDataset(i,1) == 1)
                count2 = count2+1;
            end
            totalCount(3) = totalCount(3)+1;
        case 3
```

```
            if(midtermDataset(i,1) == 1)
                count3 = count3+1;
            end
            totalCount(4) = totalCount(4)+1;
    end
end

mean0 = count0/totalCount(1);
mean1 = count1/totalCount(2);
mean2 = count2/totalCount(3);
mean3 = count3/totalCount(4);
tableResult = table([0;1;2;3], [mean0;mean1;mean2;mean3],'VariableNames',{'At J =
2','Mean'});
```

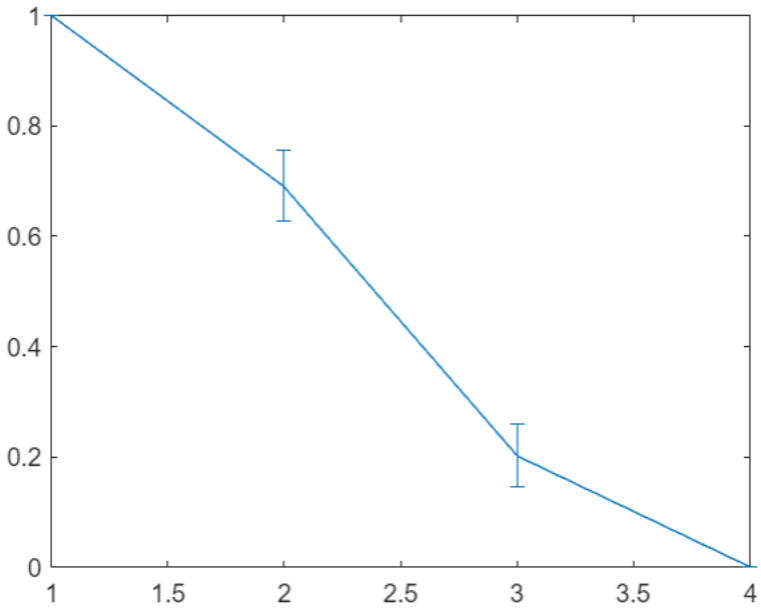
b) For my implementation I am using methods similar to Q1.c). I began by matlab code by initialising the arrays which will hold the lower and upper ranges for both the CLT and Chebyshev's Inequality. I calculate the variance for each of the means taken from part a). I used the formula variance = p(1-p). Using these variances, I then calculate the confidence interval for each value of column 2. The updated table is show below. I use 1.96 as 95% confidence results in this value. With normal distribution P(X > 1.96) = 0.025.

J = 2	Mean	CLT	Chebyshev
0	1	1 ≤ X ≤ 1	1 ≤ X ≤ 1
1	0.6917	0.6633 ≤ X ≤ 0.7201	0.6269 ≤ X ≤ 0.7565
2	0.2025	0.1778 ≤ X ≤ 0.2272	0.1461 ≤ X ≤ 0.2588
3	0	0 ≤ X≤ 0	0 ≤ X≤ 0

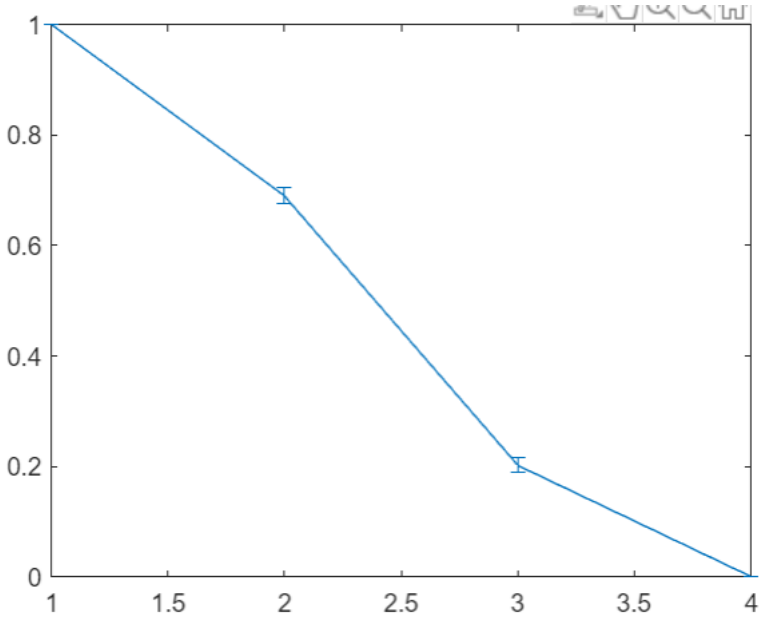
```
cltUpperRanges = [0;0;0;0];
cltLowerRanges = [0;0;0;0];
chUpperRanges = [0;0;0;0];
chLowerRanges = [0;0;0;0];
variances = [0;0;0;0];
means = [mean0; mean1; mean2; mean3];
for i=1:4 % variance = p(1-p).
    variances(i) = sqrt(means(i)*(1 - means(i)));
end

for i=1:4
    clt = variances(i) / sqrt(n);
    ch = variances(i) / sqrt(0.05 * n);
    cltUpperRanges(i) = 1.96 * clt + means(i);
    cltLowerRanges(i) = -1.96 * clt + means(i);
    chUpperRanges(i) = means(i) + ch;
    chLowerRanges(i) = means(i) - ch;
end
```

c) The confidence intervals for the CLT errorbar are a lot closer than those on the Chebyshev errorbar. Both graphs have slope downwards showing a relationship between both columns 1 and 2. The matlab code plots the means against the Chebyshev and CLT equations.

Chebyshev errorbar:



CLT errorbar:




```
% c
cltErr = [0;0;0;0];
chErr = [0;0;0;0];
for i=1:4
    cltErr(i) = variances(i) / sqrt(n);
    chErr(i) = variances(i) / sqrt(0.05 * n);
```

```
end
%errorbar(means, chErr);
errorbar(means, cltErr);
```

d) Initially I calculated that the mean for column 1 was 0.4887. When we look a little deeper it is clear to see some correlations between the first column and second column of items. Whenever the second column is set to 0, the first column is set to 1. This works the same for when there is the max number of items in the second column. If it is set to 3, then the first column will be set to 0. When the second column has a value of 1, most of the time we can expect the value in column 1 to be set to 1. Conversely when the second column is set to 2, most of the time we can expect the value of column 1 to be set to 0. We know this by looking at the confidence intervals for both CLT and Chebyshev's. When we look at the first confidence interval, which we calculated in Q1, it averages closer to 0.5 with smaller gaps between the upper and lower intervals. When we look at the confidence intervals calculated later in Q2, we can see the range is a lot larger, indicating a relationship between column 1 and column 2.

Q3. a) Lowering the dataset size (n) to 100 ended up increasing the size of the range for the CLT in q1 by around 0.05 on both sides. Chebyshev's ranges increased a lot more than CLT. It is harder to pick up trends on smaller datasets, so this is expected. The CLT also increased less than Chebyshev's as we know it concentrates from weak law of large numbers. For Q2, lowering the dataset size caused a large decrease in the CLT case for when column 1 is 1 dropping from 0.6633 to 0.5174. The upper range changed from 0.7083 to 0. 0.7208, which is small. So, when column 1 is 1, with the smaller dataset, the upper range for the CLT increased by a small amount, and the lower range decreased by a large amount.
The CH lower range also changed drastically for most values. For example, when column 1 is 1 the lower range changed from 0.6269 to 0.3950. This makes sense as the CLT changed more than expected too. The upper range changed less significantly but did increase.

b) The results when comparing column 3 with column 1 are shown below. These results do not provide us with reason to think there is a correlation between the value in column 1 and column 3. The results from first table show a clear relationship unlike this. The mean is closer to 0.5 for all values. This means there is a more even spread with no sign of a relationship between the 2 columns. The upper and lower bounds are all at similar positions also which shows there is no trend. In the first table intervals when Column 2 is equal to 0 and 3, indicate a relationship as the interval values are 1 and 0. Here this link cannot be found. The confidence intervals and means trended towards 0 in the first table. In this they do not as there is no clear relationship.

J = 2	Mean	CLT	Chebyshev
0	0.4879	0.4572 ≤ X ≤ 0.5186	0.4178 ≤ X ≤ 0.5580
1	0.5272	0.4965 ≤ X ≤ 0.5579	0.4572 ≤ X ≤ 0.5972
2	0.4760	0.4453 ≤ X ≤ 0.5067	0.4060 ≤ X ≤ 0.5460
3	0.4679	0.4372 ≤ X≤ 0.4985	0.3979 ≤ X≤ 0.5378

```
% Q3
% b
count0 = 0;
count1 = 0;
count2 = 0;
count3 = 0;
totalCount = zeros(4,1);
for i=1:n
    switch midtermDataset(i,3)
        case 0
            if(midtermDataset(i,1) == 1)
                count0 = count0+1;
            end
            totalCount(1) = totalCount(1)+1;
        case 1
            if(midtermDataset(i,1) == 1)
                count1 = count1+1;
            end
            totalCount(2) = totalCount(2)+1;
        case 2
            if(midtermDataset(i,1) == 1)
                count2 = count2+1;
            end
            totalCount(3) = totalCount(3)+1;
        case 3
            if(midtermDataset(i,1) == 1)
                count3 = count3+1;
            end
            totalCount(4) = totalCount(4)+1;
    end
end

mean0 = count0/totalCount(1);
mean1 = count1/totalCount(2);
mean2 = count2/totalCount(3);
mean3 = count3/totalCount(4);
means = [mean0; mean1; mean2; mean3];
for i=1:4 % variance = p(1-p).
    variances(i) = sqrt(means(i)*(1 - means(i)));
end

for i=1:4
    clt = variances(i) / sqrt(n);
```

```
    ch = variances(i) / sqrt(0.05 * n);
    cltUpperRanges(i) = (1.96 * clt) + means(i);
    cltLowerRanges(i) = (-1.96 * clt) + means(i);
    chUpperRanges(i) = means(i) + ch;
    chLowerRanges(i) = means(i) - ch;
end
```

Appendix:

```
% Q1
% a
n = 1017;
midtermDataset = midterm2021{:,:};
items = zeros(n,1);
for i=1:n
    sumOfItems = midtermDataset(i,1) + midtermDataset(i,2) + midtermDataset(i,3);
    items(i) = sumOfItems;
end
histogram(items,'Normalization','probability')

% b
count = 0;
for i=1:n
    count = midtermDataset(i,1) + count;
end
finalProb = count/n;

% c
sumOfSquares = 0;
for i=1:n
    valueToSquare = midtermDataset(i,1) - finalProb;
    valueToSquare = valueToSquare * valueToSquare;
    sumOfSquares = sumOfSquares + valueToSquare;
end
variance = sumOfSquares/n;
sigma2 = 2 * sqrt(variance/n);
cltRanges = [finalProb - sigma2; finalProb + sigma2;];
standardDev = sqrt(variance);
chRanges = [finalProb - standardDev/sqrt(0.05 * n); finalProb +
standardDev/sqrt(0.05*n);];

%d

% Q2
% a
count0 = 0;
count1 = 0;
count2 = 0;
count3 = 0;
totalCount = zeros(4,1);
for i=1:n
    switch midtermDataset(i,2)
        case 0
            if(midtermDataset(i,1) == 1)
                count0 = count0+1;
            end
```

```
            totalCount(1) = totalCount(1)+1;
        case 1
            if(midtermDataset(i,1) == 1)
                count1 = count1+1;
            end
            totalCount(2) = totalCount(2)+1;
        case 2
            if(midtermDataset(i,1) == 1)
                count2 = count2+1;
            end
            totalCount(3) = totalCount(3)+1;
        case 3
            if(midtermDataset(i,1) == 1)
                count3 = count3+1;
            end
            totalCount(4) = totalCount(4)+1;
    end
end

mean0 = count0/totalCount(1);
mean1 = count1/totalCount(2);
mean2 = count2/totalCount(3);
mean3 = count3/totalCount(4);
tableResult = table([0;1;2;3], [mean0;mean1;mean2;mean3],'VariableNames',{'At J =
2','Mean'});

% b
cltUpperRanges = [0;0;0;0];
cltLowerRanges = [0;0;0;0];
chUpperRanges = [0;0;0;0];
chLowerRanges = [0;0;0;0];
variances = [0;0;0;0];
means = [mean0; mean1; mean2; mean3];
for i=1:4 % variance = p(1-p).
    variances(i) = sqrt(means(i)*(1 - means(i)));
end

for i=1:4
    clt = variances(i) / sqrt(n);
    ch = variances(i) / sqrt(0.05 * n);
    cltUpperRanges(i) = (1.96 * clt) + means(i);
    cltLowerRanges(i) = (-1.96 * clt) + means(i);
    chUpperRanges(i) = means(i) + ch;
    chLowerRanges(i) = means(i) - ch;
end

% c
cltErr = [0;0;0;0];
chErr = [0;0;0;0];
for i=1:4
    cltErr(i) = variances(i) / sqrt(n);
    chErr(i) = variances(i) / sqrt(0.05 * n);
end
%errorbar(means, chErr);
errorbar(means, cltErr);

% Q3
% b
count0 = 0;
```

```
count1 = 0;
count2 = 0;
count3 = 0;
totalCount = zeros(4,1);
for i=1:n
    switch midtermDataset(i,3)
        case 0
            if(midtermDataset(i,1) == 1)
                count0 = count0+1;
            end
            totalCount(1) = totalCount(1)+1;
        case 1
            if(midtermDataset(i,1) == 1)
                count1 = count1+1;
            end
            totalCount(2) = totalCount(2)+1;
        case 2
            if(midtermDataset(i,1) == 1)
                count2 = count2+1;
            end
            totalCount(3) = totalCount(3)+1;
        case 3
            if(midtermDataset(i,1) == 1)
                count3 = count3+1;
            end
            totalCount(4) = totalCount(4)+1;
    end
end

mean0 = count0/totalCount(1);
mean1 = count1/totalCount(2);
mean2 = count2/totalCount(3);
mean3 = count3/totalCount(4);
means = [mean0; mean1; mean2; mean3];
for i=1:4 % variance = p(1-p).
    variances(i) = sqrt(means(i)*(1 - means(i)));
end

for i=1:4
    clt = variances(i) / sqrt(n);
    ch = variances(i) / sqrt(0.05 * n);
    cltUpperRanges(i) = (1.96 * clt) + means(i);
    cltLowerRanges(i) = (-1.96 * clt) + means(i);
    chUpperRanges(i) = means(i) + ch;
    chLowerRanges(i) = means(i) - ch;
end
```