Statistical Methods for Computer Science

Assignment Mid-Term — STU33009

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

DECLARATION

I understand that this is an individual assessment and that collaboration is not permitted. I have not received any assistance with my work for this assessment. Where I have used the published work of others, I have indicated this with appropriate citation.

I have not and will not share any part of my work on this assessment, directly or indirectly, with any other student.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

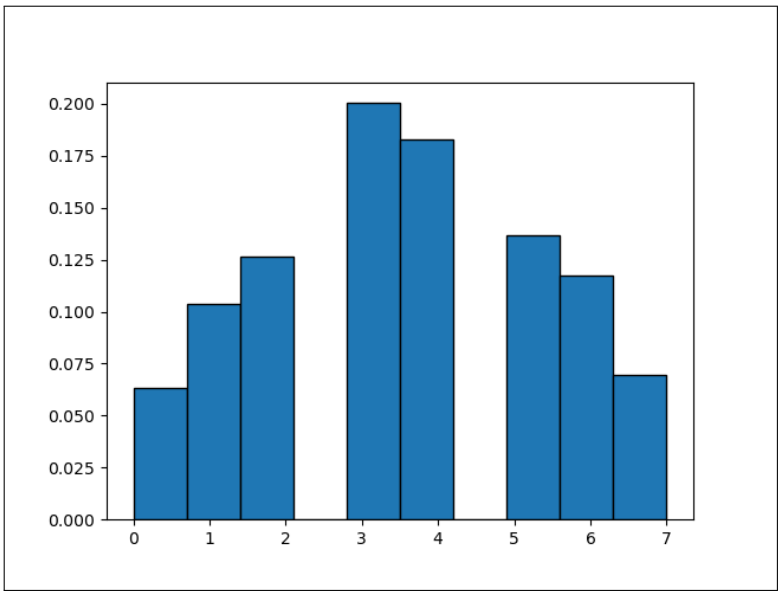
I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write."

I understand that by returning this declaration with my work, I am agreeing with the above statement. ☑

Name: Hamza Mughees

Date: 05/04/2021

1. (a) In order to plot the PMF, we must first obtain the number of items in each basket. If a given basket contains x items, we can obtain the number of baskets that contain x items and divide this quantity by the total number of baskets. By doing this for all the different frequencies (i.e. values of x), we can obtain the PMF.



(b) In order to obtain the probability, we can first calculate N by getting the length of the first column. Since this column only contains 0's and 1's, summing would give us the amount of 1's. Hence, we can divide the sum of the values in the first column by N. Below is the output of the python program.

```
N = 1901
probability = 0.5039452919516044
```

(c) In order to derive the confidence interval, we must first calculate the mean (μ) and the variance (σ^2). The mean is simply equal to $P(Z_{i,1} = 1)$. We know this because,

$$\mu = E[X] = \sum_{i=1}^{n} x_i p(x_i)$$

$$\mu = 1(0.5039452919516044) + 0(1 - 0.5039452919516044) = 0.5039452919516044$$

We also know that,

$$\sigma^2 = Var(X) = \sum_{i=1}^{n} (x_i - \mu)^2 p(x_i)$$

However, since column 1 is a Bernoulli distribution, we can use the following, simpler formula,

$$\sigma^2 = \mu(1 - \mu)$$

$$\sigma^2 = 0.5039452919516044(1 - 0.5039452919516044) = 0.2499844347$$

CLT tells us that $X \sim N(\mu, \frac{\sigma^2}{N})$. A normal distribution follows the "68-95-99.7" rule.

$$P(-2\sigma \leq X - \mu \leq 2\sigma) \approx 0.95$$

where,

$$2\sigma = 2\sqrt{\frac{\sigma^2}{N}} = 2\sqrt{\frac{0.2499844347}{1901}} = 0.0229348245$$

This gives us the following confidence interval,

$$-0.0229348245 \leq X - \mu \leq 0.0229348245$$

$$\mu - 0.0229348245 \leq X \leq \mu + 0.0229348245$$

$$0.5039452920 - 0.0229348245 \leq X \leq 0.5039452920 + 0.0229348245$$

$$0.4810104675 \leq X \leq 0.5268801165$$

Chebyshev inequality tells us,

$$P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{N\epsilon^2}$$

Since we are dealing with the 95% confidence case, we can say that for some value of ϵ,

$$P(|X - \mu| \geq \epsilon) \leq 0.05$$

Therefore,

$$\frac{\sigma^2}{N\epsilon^2} = 0.05$$

In this equation, ϵ represents the confidence bound. This is what we are attempting to calculate. We can make this the subject of the equation, like so,

$$\epsilon = \sqrt{\frac{\sigma^2}{0.05N}}$$

Subbing in the values, we get,

$$\epsilon = \sqrt{\frac{0.2499844347}{0.05 \cdot 1901}} = 0.05128382664$$

We can now say,

$$P(|X - \mu| \geq 0.05128382664) \leq 0.05$$

We can re-write this as,

$$P(|X - \mu| \leq 0.05128382664) \geq 0.95$$

$$P(-0.05128382664 \leq X - \mu \leq 0.05128382664) \geq 0.95$$

Following the same simplification steps as before, we get the below confidence interval,

$$0.4526614653 \leq X \leq 0.5552291186$$

(d) Chebyshev inequality tells us,

$$P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{N\epsilon^2}$$

Since we want a confidence of 95%, we can say,

$$\frac{\sigma^2}{N\epsilon^2} = 0.05$$

We can make N the subject of the equation.

$$N = \frac{\sigma^2}{0.05\epsilon^2}$$

In order for the accuracy to be $\pm 1\%$, the value of ϵ must be 0.01. Subbing in the values, we get,

$$N = \frac{0.2499844347}{0.05 \cdot 0.01^2} = 49996.88693$$

Therefore, according to Chebyshev Inequality, we would need at least 49997 baskets to estimate the value of $P(Z_{i,1} = 1)$ to an accuracy of $\pm 1\%$. However, we can improve on this number. As we have seen already, the CLT tends to be more optimistic in it's confidence intervals. This means that the CLT would give a much smaller value of N to achieve an accuracy of $\pm 1\%$. For the CLT, we know that the formula for a 95% error bound is the following,

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

Same as before, we can make N the subject of the formula.

$$N = \frac{4\sigma^2}{(2\sigma)^2}$$

2σ is the error bound, which we know is 0.01. Subbing in the values, we get,

$$N = \frac{4 \cdot 0.2499844347}{0.01^2} = 9999.377387$$

Therefore, according to the CLT, we would need at least 10000 baskets to estimate the value of $P(Z_{i,1} = 1)$ to an accuracy of $\pm 1\%$.

2. (a) In order to obtain the set of possible values of z, we can put out second column into a set, which gives us,

$$z \in \{0, 1, 2, 3\}$$

We know that the sample mean of $Z_{i,1}$ conditioned on $Z_{i,2} = z$ for each z is,

$$\frac{1}{N} \sum_{j \in \{i: Z_{i,2} = z\}} Z_{j,1}$$

where $\{i : Z_{i,2} = z\}$ is the set of baskets for which $Z_{i,2} = z$, and $N = |\{i : Z_{i,2} = z\}|$ is the size of this set. We must calculate the sample mean for each value of z. The following is the output of our python program,

```
For z = 0: N = 465, b = 0, sample mean = 0.0
For z = 1: N = 439, b = 97, sample mean = 0.22095671981776766
For z = 2: N = 514, b = 378, sample mean = 0.7354085603112841
For z = 3: N = 483, b = 483, sample mean = 1.0
```

The following is the table with the results,

$\boldsymbol{z}$	0	1	2	3
$\bar{\boldsymbol{x}}$	0.0	0.22095671981776766	0.7354085603112841	1.0

Table 1: Sample means for each z

(b) Since column 1 is a Bernoulli distribution, we can calculate the variance for each value of z with the following formula,

$$\sigma^2 = Var(X) = \mu(1 - \mu)$$

For each z, $\mu = \bar{x}$. We can change up our code from part (a) to obtain the variances. We want a confidence of 95%. For the CLT, we would need an error bound of 2σ. The formula for this is,

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

For Chebyshev Inequality, we can get the value of ϵ with the following formula,

$$\epsilon = \sqrt{\frac{\sigma^2}{0.05N}}$$

Since we have the two formulas for the CLT and Chebyshev Inequality, we can create functions to calculate these in our code. We can then add a function that takes these error bounds and outputs a confidence interval for each value of z. The following is the output of our code,

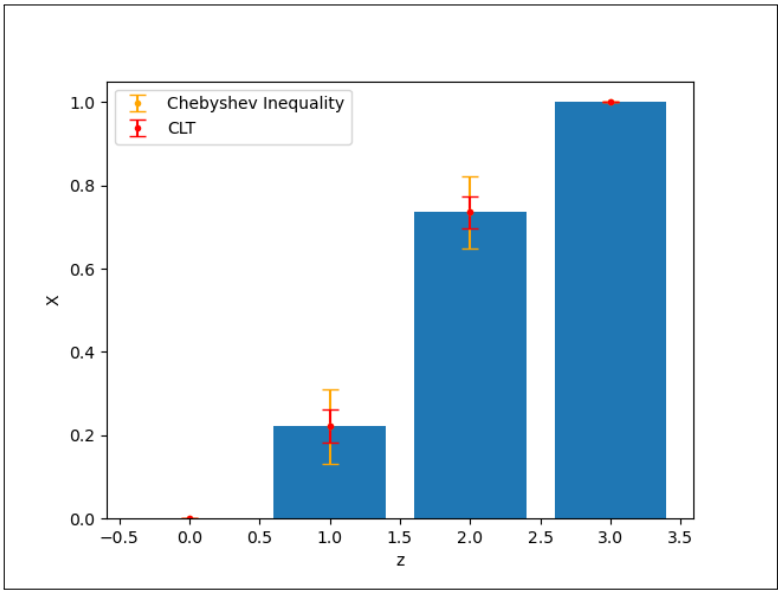
```
For z = 0:
CLT:
0.0 <= X <= 0.0
Chebyshev:
0.0 <= X <= 0.0

For z = 1:
CLT:
0.1813533505281464 <= X <= 0.2605600891073889
Chebyshev:
0.132400893948147 <= X <= 0.3095125456873883

For z = 2:
CLT:
0.6964950265375933 <= X <= 0.7743220940849749
Chebyshev:
0.6483952535485775 <= X <= 0.8224218670739907

For z = 3:
CLT:
1.0 <= X <= 1.0
Chebyshev:
1.0 <= X <= 1.0
```

Adding these confidence intervals (in 4-point precision) to the table, we get,

$\boldsymbol{z}$	0	1	2	3
$\bar{\boldsymbol{x}}$	0.0	0.22095671981776766	0.7354085603112841	1.0
CLT	$X = 0.0$	$0.1814 \leq X \leq 0.2606$	$0.6965 \leq X \leq 0.7743$	$X = 1.0$
Chebyshev	$X = 0.0$	$0.1324 \leq X \leq 0.3095$	$0.6484 \leq X \leq 0.8224$	$X = 1.0$

Table 2: Sample means and confidence intervals for each z

(c) The output graph shows clearly the correlation between the number of item 2s in the basket and the presence of item 1. We can see that when there are $z = 0$ item 2s in the basket, we are guaranteeing that item 1 is not present in the basket. When there are $z = 3$ item 2s in the basket, we are guaranteeing that item 1 is present in the basket. This is clear due to the 0 error bound. The probability of item 1 being present in the basket for each value of z shows that as we increase z, we become more confident that item 1 is present in the basket.



(d) We can see that as the number of item 2s in the basket (z) increases, the probability that item 1 is in the basket also increases. It is noticeable that at both ends of the spectrum ($z = 0$ and $z = 3$), our error bounds are 0. This means that we can say with 95% confidence that if there are $z = 0$ item 2s in a basket, item 1 is definitely not present in the basket. Similarly, if there are $z = 3$ item 2s in a basket, item 1 is definitely present in the basket. For $z = 1$ and $z = 2$, we can see, as expected, a gradual increase of the probability of the presence of item 1. For these reasons, we can conclude that the presence of item 2 in the basket is predictive of item 1 being in the basket.

3. (a) For this part, we can reuse all the code from the previous questions. However, we must slightly change the way we import our dataset into the pandas dataframe to ensure that we obtain only the first 100 rows.

```
df = pd.read_csv('./data.csv', header=None)[:100]
```

With the above alteration, we can use the code from question 1, part (b) to obtain the probability of the first item being in the basket from the first 100 rows. As we have seen already, this would be equal to μ. This gives the following output,

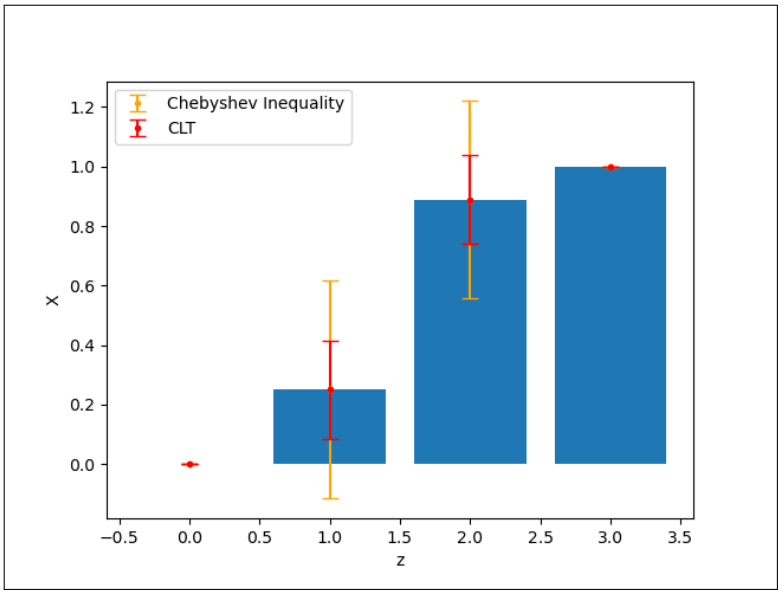
```
N = 100
probability = 0.47
```

We can use the code from question 2, part (c) to obtain our new confidence intervals and plot our new bar chart with these confidence intervals. Below are the resulting confidence intervals.

```
For z = 0:
CLT:
0.0 <= X <= 0.0
Chebyshev:
0.0 <= X <= 0.0

For z = 1:
CLT:
0.08633658232300573 <= X <= 0.41366341767699427
Chebyshev:
-0.11596252735569995 <= X <= 0.6159625273556999

For z = 2:
CLT:
0.7407407407407407 <= X <= 1.037037037037037
Chebyshev:
0.5576195588889199 <= X <= 1.2201582188888578

For z = 3:
CLT:
1.0 <= X <= 1.0
Chebyshev:
```

```
1.0 <= X <= 1.0
```

Here is the output bar chart.



It can be noticed that the confidence intervals have widened and become less optimistic. The error bound is inversely proportional to the size of the sample. Although the presence of item 2 in the basket can still be predictive of item 1 being in the basket, it isn't reliable. Our confidence intervals are too wide and even reach below zero and above one.

(b) In our code from question 2, part (c), we have a variable with the following value,

```
conditioned_on = 1
```

This variable holds the 0-indexed position of the column on which we are conditioning $E[X_{i,1} = 1]$. For this part, we can simply change the value of this variable to 2.

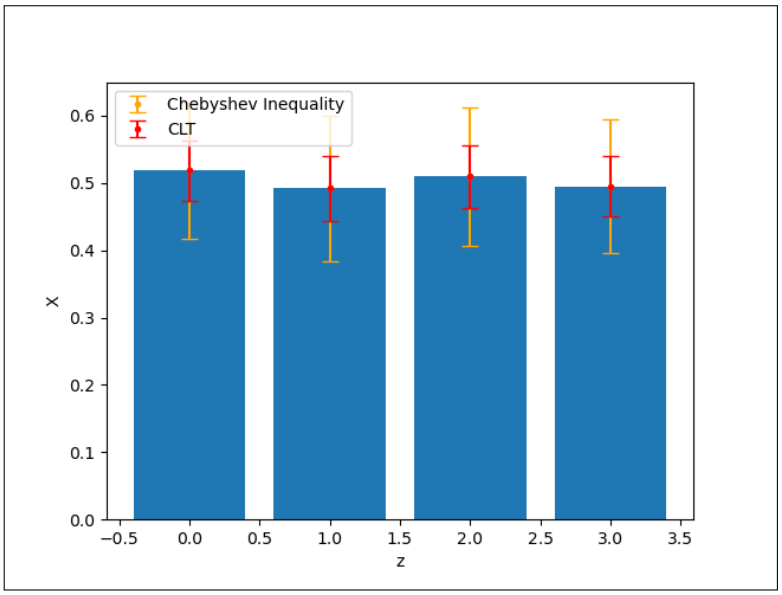
```
conditioned_on = 2
```

This results in our calculations and our new graph to plot $E[Z_{i,1}|Z_{i_3} = z]$. Again, as before, we can use the code from question 2, part (c) to obtain our confidence intervals for each value of z and plot them on a bar chart. Here are the output confidence intervals.

```
For z = 0:
CLT:
```

```
0.473256373750724 <= X <= 0.5631808732128387
Chebyshev:
0.41767997666181544 <= X <= 0.6187572703017473

For z = 1:
CLT:
0.4432641608345197 <= X <= 0.5402652509301862
Chebyshev:
0.383314190209607 <= X <= 0.6002152215550989

For z = 2:
CLT:
0.4634849852046125 <= X <= 0.5556232950501645
Chebyshev:
0.4065403780540923 <= X <= 0.6125679022006848

For z = 3:
CLT:
0.45087235429169875 <= X <= 0.5393429098961681
Chebyshev:
0.3961945439245492 <= X <= 0.5940207202633176
```

Here is the resulting bar chart.



It can be noticed that our confidence intervals for every value of z revolve around

the 0.5 mark. This suggests that $Z_{i,1} = 1$ and $Z_{i,3} = z$ are independent events and so $Z_{i,3} = z$ for any z is not predictive on $Z_{i,1}$.

APPENDIX

1. (a)

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

# 'data.csv' containing baskets and items
df = pd.read_csv('./data.csv', header=None)

# summing each row to get number of items in each basket
x = df.sum(axis=1)
# changing from frequencies to probabilities
weights = np.ones_like(x) / len(x)

# plotting the histogram
plt.hist(x, weights=weights, edgecolor='black')
plt.show()
```

(b)

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

df = pd.read_csv('./data.csv', header=None)

# obtaining first column (i.e. frequency of first item in each
# basket)
x = df[0]
# obtaining number of baskets
N = len(x)

# calculating probability
probability = sum(x) / N
print(f'N = {N}')
print(f'probability = {probability}')
```

2. (a)

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

df = pd.read_csv('./data.csv', header=None)

# column on which E[Z_i1] is conditioned (0-indexed)
conditioned_on = 1

# looping through the possible values of z
for z in set(df[conditioned_on]):
    # obtaining set of baskets where Z_i,2 = z
    baskets = df.loc[df[conditioned_on] == z]
    # getting size of this set
    N = len(baskets)
    # getting number of baskets which contain item 1
    b = sum(baskets[0])
    # calculating sample mean
    sample_mean = b / N
    print(f'For z = {z}: N = {N}, b = {b}, sample mean = {sample_mean}')
```

(b)

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

df = pd.read_csv('./data.csv', header=None)

def clt_95(N, variance):
    return 2*((variance/N)**0.5)

def chebyshev_95(N, variance):
    return (variance/(0.05*N))**0.5

def print_confidence_interval(sample_mean, bound_offset):
    print(f'{sample_mean-bound_offset} <= X <= {sample_mean+bound_offset}')
```

```
conditioned_on = 1

for z in set(df[conditioned_on]):
    baskets = df.loc[df[conditioned_on] == z]
    N = len(baskets)
    b = sum(baskets[0])
    sample_mean = b / N
    # calculating variance
    variance = sample_mean*(1-sample_mean)
    # clt error bound
    clt = clt_95(N, variance)
    # chebyshev error bound
    chebyshev = chebyshev_95(N, variance)

    print(f'For z = {z}:\nCLT:')
    print_confidence_interval(sample_mean, clt)
    print('Chebyshev:')
    print_confidence_interval(sample_mean, chebyshev)
    print()
```

(c)

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

df = pd.read_csv('./data.csv', header=None)

def clt_95(N, variance):
    return 2*((variance/N)**0.5)

def chebyshev_95(N, variance):
    return (variance/(0.05*N))**0.5

def print_confidence_interval(sample_mean, bound_offset):
    print(f'{sample_mean-bound_offset} <= X <= {sample_mean+bound_offset}')

sample_means = []
clt_errors = []
```

```
chebyshev_errors = []
conditioned_on = 1

for z in set(df[conditioned_on]):
    baskets = df.loc[df[conditioned_on] == z]
    N = len(baskets)
    b = sum(baskets[0])
    sample_mean = b / N
    variance = sample_mean*(1-sample_mean)
    clt = clt_95(N, variance)
    chebyshev = chebyshev_95(N, variance)

    # appending values and errors
    sample_means.append(sample_mean)
    clt_errors.append(clt)
    chebyshev_errors.append(chebyshev)

    print(f'For z = {z}:\nCLT:')
    print_confidence_interval(sample_mean, clt)
    print('Chebyshev:')
    print_confidence_interval(sample_mean, chebyshev)
    print()

# getting z values
z = np.arange(len(sample_means))

# plotting sample means
plt.bar(z, sample_means)

# clt and chebyshev errors
plt.errorbar(z, sample_means,
             yerr=chebyshev_errors,
             fmt='.',
             color='orange',
             capsize=5,
             label='Chebyshev Inequality')
plt.errorbar(z, sample_means,
             yerr=clt_errors,
```

```
            fmt='.',
            color='red',
            capsize=5,
            label='CLT')

plt.xlabel('z')
plt.ylabel('X')
plt.legend(loc='upper left')

plt.show()
```

3. All questions have very slight alterations to the code for the prior questions. The alterations are mentioned in the answers.