Michael Potter

ENGR131A-80

UID 705644667

Final Project (**based on first Class Project upload**)

1.

a.

```
t =   10 , p(odd number)=0.4000
t =   50 , p(odd number)=0.4600
t =  100 , p(odd number)=0.4700
t =  500 , p(odd number)=0.5280
t = 1000 , p(odd number)=0.5270
```

b.

$$S_x = \{1,2,3,4\}$$
$$P(X\ is\ odd) = P\{X = 1\ or\ X = 3\} = \frac{1}{2}$$

c. The experimental result approaches the theoretical result as the number of coin tosses increases (and they are approximately equal as t goes to infinity due to law of large numbers).

d.

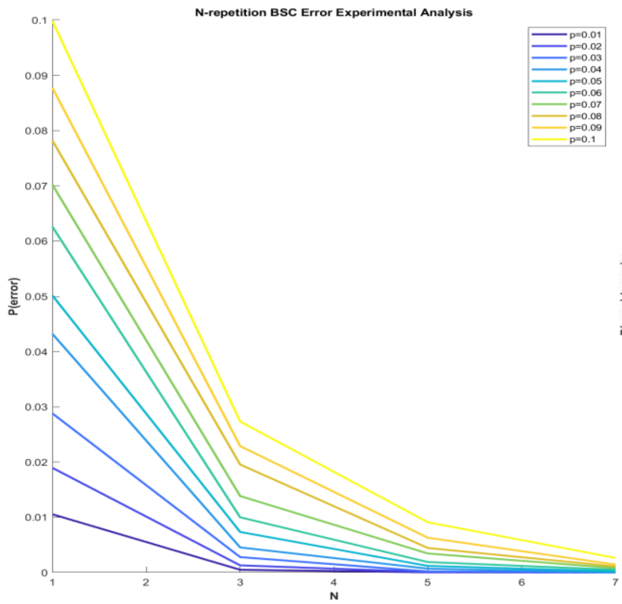
$$x + y = 1$$
$$x = 2y$$
$$2y + y = 1$$
$$y = \frac{1}{3}; x = \frac{2}{3}$$
$$P(odd) = \frac{1}{3}\ ; P(even) = \frac{2}{3}$$

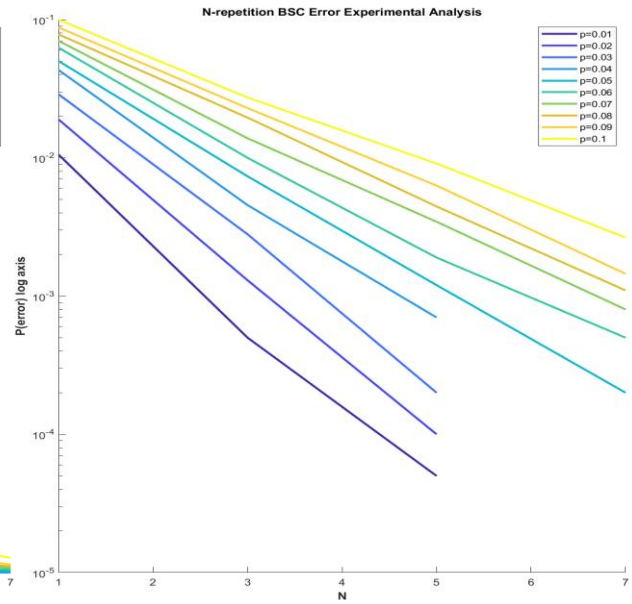
```
t =   10 , p(odd number)=0.2000
t =   50 , p(odd number)=0.2400
t =  100 , p(odd number)=0.3100
t =  500 , p(odd number)=0.3080
t = 1000 , p(odd number)=0.3280
```

Similarly to part c, the experimental result approaches the theoretical result as the number of coin tosses increases (and they are approximately equal as t goes to infinity due to law of large numbers).

2.

a.





	P=0.01	P=0.02	P=0.03	P=0.04	P=0.05	P=0.06	P=0.07	P=0.08	P=0.09	P=0.1
N=1	0.01055	0.01895	0.0288	0.0432	0.05015	0.0626	0.0702	0.0782	0.08775	0.09985
N=3	0.0005	0.0013	0.0028	0.00455	0.00735	0.01	0.01385	0.01955	0.0229	0.02735
N=5	5e-05	0.0001	0.0002	0.0007	0.0012	0.0019	0.00345	0.00445	0.0063	0.0091
N=7	0	0	0	0	0.0002	0.0005	0.0008	0.0011	0.00145	0.00265

b.

$$N = 1,3,5,7, \ldots \, odd\ number$$

$$n = \left\lceil \frac{N}{2} \right\rceil \; ; p(bit = 0) = \frac{1}{2} \; ; p(bit = 1) = \frac{1}{2}$$

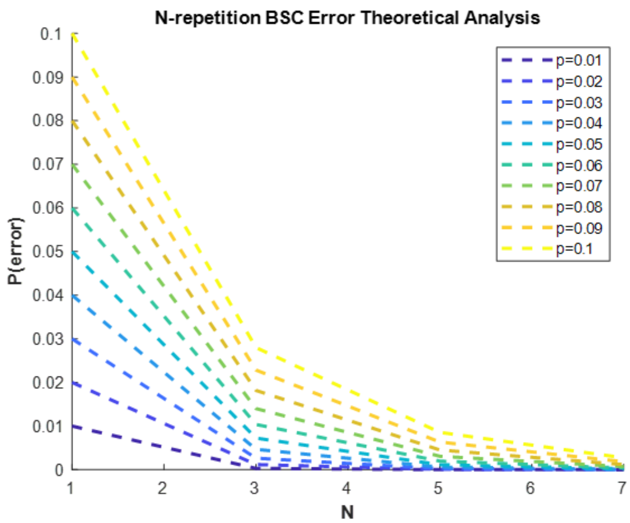
$$p(error \mid bit = 0) = \sum_{i=n}^{N} \binom{N}{i} (1-p)^{N-i} p^{i}$$

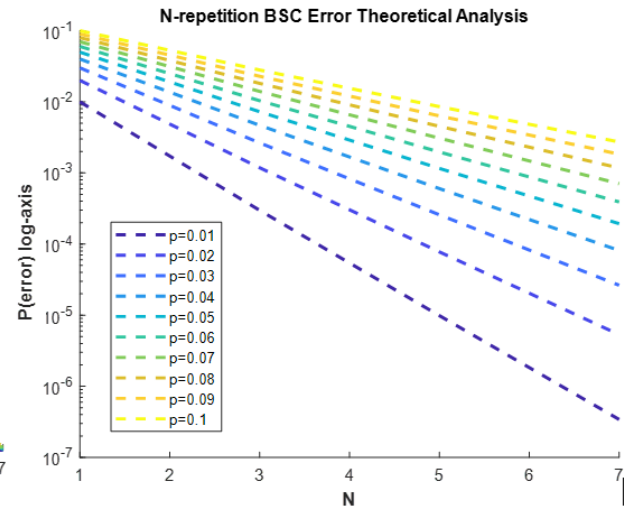
$$p(error \mid bit = 1) = \sum_{i=n}^{N} \binom{N}{i} (1-p)^{N-i} p^{i}$$

$$p(error) = p(bit = 0) * p(error \mid bit = 0) + p(bit = 1) * p(error \mid bit = 1)$$

$$= \sum_{i=n}^{N} \binom{N}{i} (1-p)^{N-i} p^{i}$$

	P=0.01	P=0.02	P=0.03	P=0.04	P=0.05	P=0.06	P=0.07	P=0.08	P=0.09	P=0.1
N=1	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
N=3	0.000298	0.001184	0.002646	0.004672	0.00725	0.010368	0.014014	0.018176	0.022842	0.028
N=5	9.8506e-06	7.7619e-05	0.000258	0.00060221	0.0011581	0.0019703	0.0030799	0.0045253	0.0063413	0.00856
N=7	3.4167e-07	5.3357e-06	2.6359e-05	8.1282e-05	0.00019358	0.00039149	0.00070724	0.0011763	0.0018366	0.002728





c. If the BSC channel has the probability of flipping a bit equal to 0.5, then the BSC channel is uninterpretable (a random channel), as the user cannot determine what bit was sent. To show this, follow the proof below:

$$N = 1,3,5,7, \ldots\, odd\ number$$
$$n = \left\lceil \frac{N}{2} \right\rceil ; p(bit\ sent = 0) = \frac{1}{2} ;\ p(bit\ sent = 1) = \frac{1}{2};$$
$$p(bit\ 1\ flipped) = \frac{1}{2} ;\ p(bit\ 0\ flipped) = \frac{1}{2};$$
$$1 - p(bit\ 1\ flipped) = \frac{1}{2} = p(bit\ 1\ flipped);\ 1 - p(bit\ 0\ flipped) = \frac{1}{2} = \mathrm{p(bit\ 0\ flipped)}$$
$$receive = R, sent = S$$

$$p(R = 0 \mid S = 0) = \sum_{i=n}^{N} \binom{N}{i} \left(p(bit\ 0\ flipped)\right)^{N-i} \left(1 - p(bit\ 0\ flipped)\right)^{i}$$
$$= \sum_{i=n}^{N} \binom{N}{i} \left(1 - p(bit\ 0\ flipped)\right)^{N-i} \left(p(bit\ 0\ flipped)\right)^{i} = p(R = 1 \mid S = 0)$$

Then, by the same logic as the derivation above, p(R=1 | S=1) = p(R=0 | S=1)

$$p(R = 0 \mid S = 0) = P(R = 0 \mid S = 1) = P(R = 1 \mid S = 0) = P(R = 1 \mid S = 1)$$
$$P(S = 1) = P(S = 0)$$

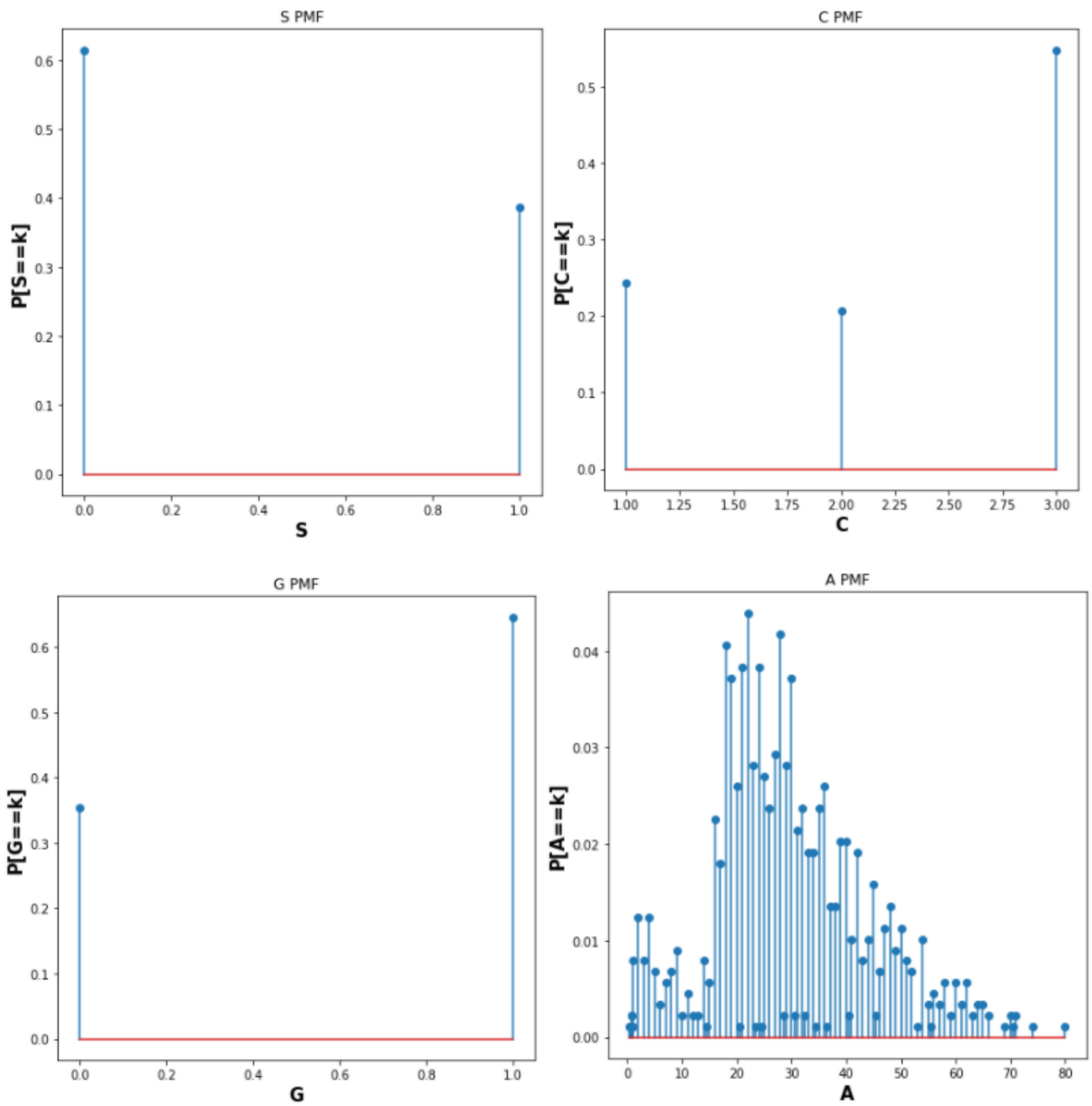
$$p(S = 1 \mid R = 0) = \frac{p(R = 0 \mid S = 1) * p(S = 1)}{\sum_{i=\{0,1\}} P(R = 0 \mid S = i) * P(S = i)} = \frac{p(R = 0 \mid S = 0) * p(S = 0)}{\sum_{i=\{0,1\}} P(R = 0 \mid S = i) * P(S = i)} = P(S = 0 \mid R = 0)$$

and the same logic holds for p(S=0 | R = 1) = p(S=1 | R=1).

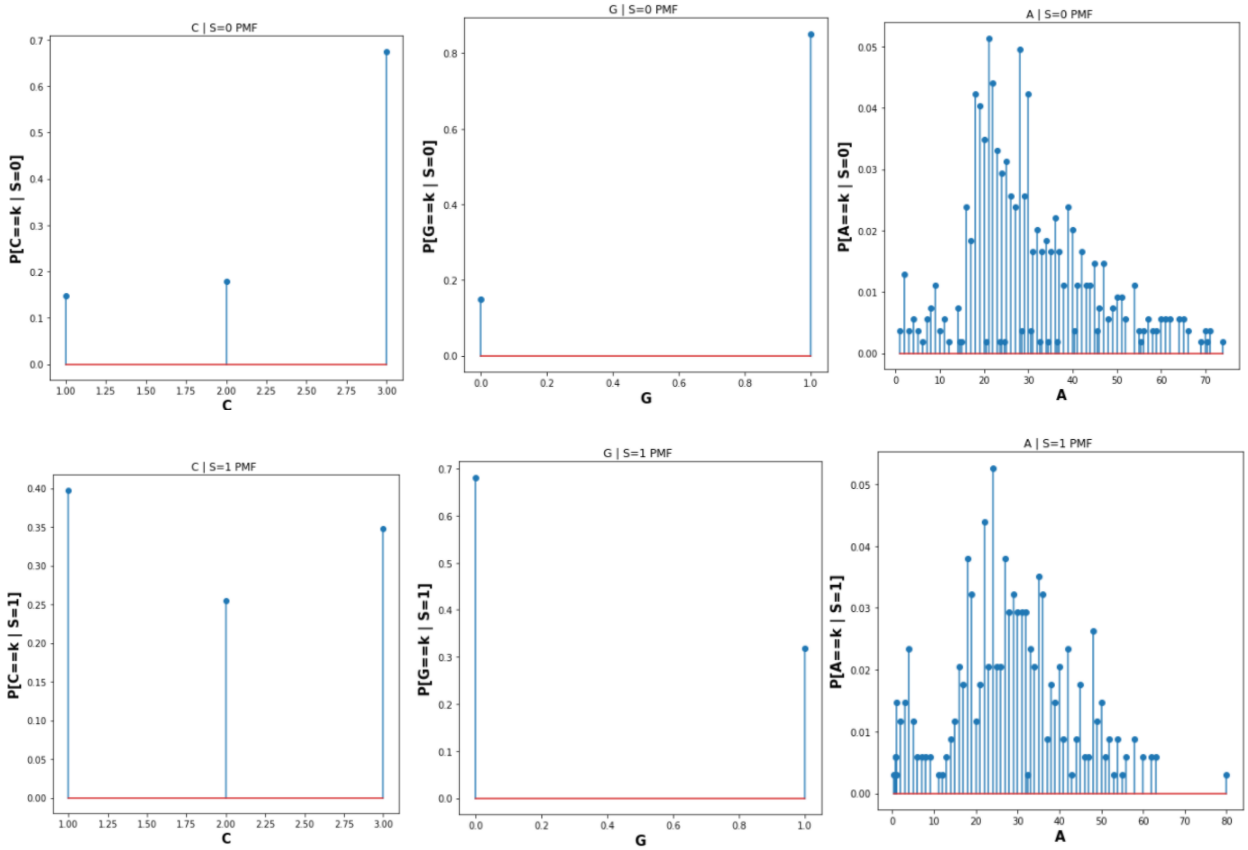
If I knew that 0.5 < p < 1 then I would say that the receiver should swap the 1s with the 0s and vice versa to get an equivalent channel cross probability of (1-p) < ½ .

3.

a.


S PMF
P[S==k]
S
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.2
0.4
0.6
0.8
1.0
C PMF
P[C==k]
C
0.0
0.1
0.2
0.3
0.4
0.5
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
G PMF
P[G==k]
G
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.2
0.4
0.6
0.8
1.0
A PMF
P[A==k]
A
0.00
0.01
0.02
0.03
0.04
0
10
20
30
40
50
60
70
80

b.



c.

$$\boldsymbol{P(S=0, C=1, G=0, A \leq 40)} = P(C=1, G=0, A \leq 40 \mid S=0) * P(S=0)$$
$$= P(C=1 \mid S=0)\, P(G=0 \mid S=0)\, P(A \leq 40 \mid S=0)\, P(S=0)$$
$$= \mathbf{0.010625322687488965}$$

$$\boldsymbol{P(S=1, C=1, G=0, A \leq 40)} = P(C=1, G=0, A \leq 40 \mid S=1) * P(S=1)$$
$$= P(C=1 \mid S=1)\, P(G=0 \mid S=1)\, P(A \leq 40 \mid S=1)\, P(S=1)$$
$$= \mathbf{0.08460553314142813}$$

d.

$$\boldsymbol{P(S=0 \mid C=1, G=0, A \leq 40)} = \frac{P(C=1, G=0, A \leq 40 \mid S=0)\, P(S=0)}{P(C=1, G=0, A \leq 40)}$$
$$= \frac{P(C=1, G=0, A \leq 40 \mid S=0)\, P(S=0)}{\sum_{i=\{0,1\}} P(C=1, G=0, A \leq 40, S=i)}$$

$$= \frac{P(C=1, G=0, A \leq 40 \mid S=0)\, P(S=0)}{P(C=1, G=0, A \leq 40 \mid S=1)\, P(S=1) + P(C=1, G=0, A \leq 40 \mid S=0)\, P(S=0)}$$
$$= \mathbf{0.11157436941003064}$$

$$\boldsymbol{P}(\boldsymbol{S}=\mathbf{1}\ |\boldsymbol{C}=\mathbf{1},\boldsymbol{G}=\mathbf{0},\boldsymbol{A}\leq\mathbf{40})=\frac{P(C=1,G=0,A\leq 40\ |S=1)\ P(S=1)}{P(C=1,G=0,A\leq 40)}$$

$$=\frac{P(C=1,G=0,A\leq 40\ |S=1)\ P(S=1)}{\sum_{i=\{0,1\}}P(C=1,G=0,A\leq 40,S=i)}$$

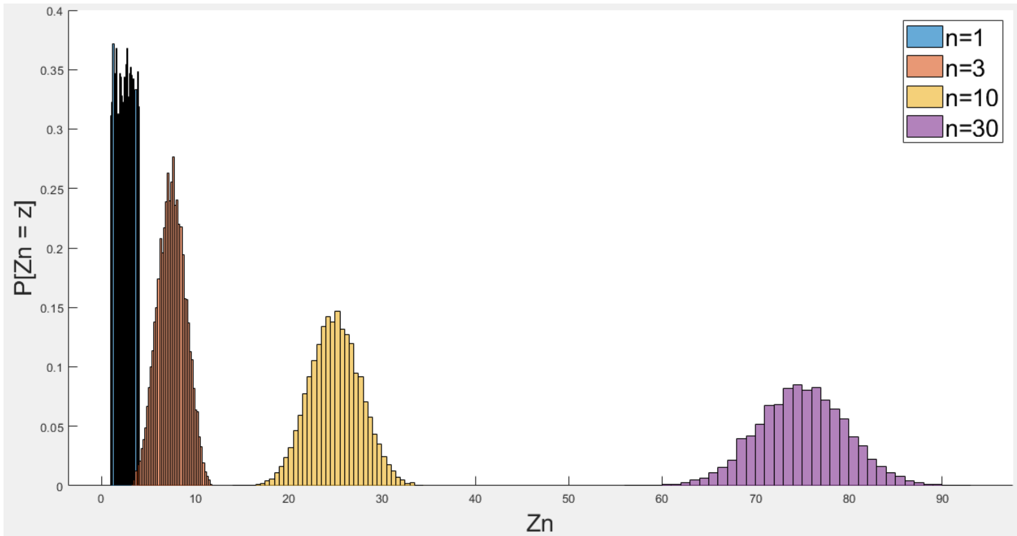
$$=\frac{P(C=1,G=0,A\leq 40\ |S=1)\ P(S=1)}{P(C=1,G=0,A\leq 40\ |S=1)\ P(S=1)+\ P(C=1,G=0,A\leq 40\ |S=0)\ P(S=0)}$$

$=$**0.8884256305899694**

She will survive with probability of approximately 0.888 (most likely will survive…) .

4.

a.



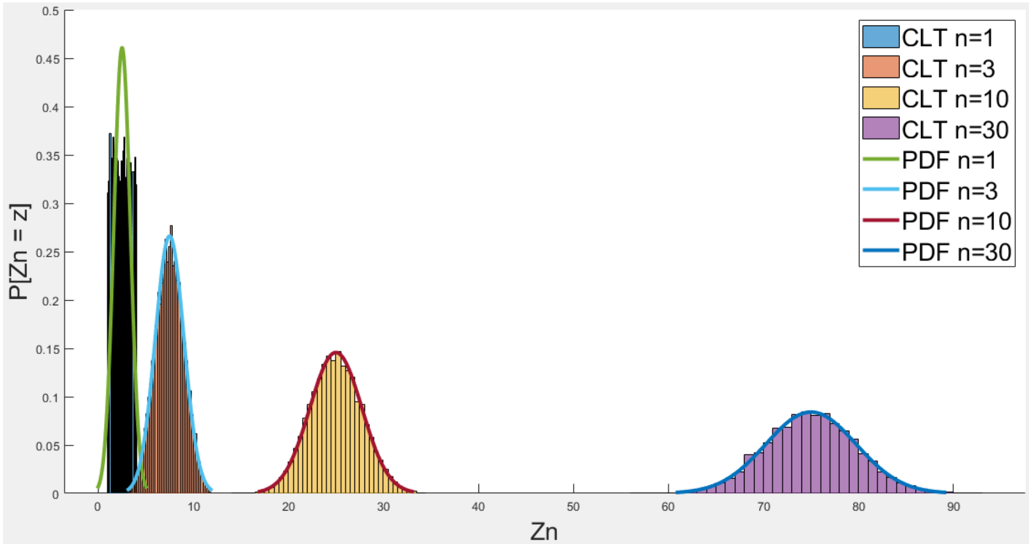
As n increases (and approaches infinity) the distribution of Z_n becomes a Gaussian distribution. This is evident by looking at the difference between n = 1, and n = 30. At n = 1, the distribution of Z_n appears to still be a uniform distribution like the iid X_i , but when n = 30, the distribution of Z_n appears to be a Gaussian distribution. The tails become wider and wider as n increases, with the peak probability at the mean decreasing.

b.

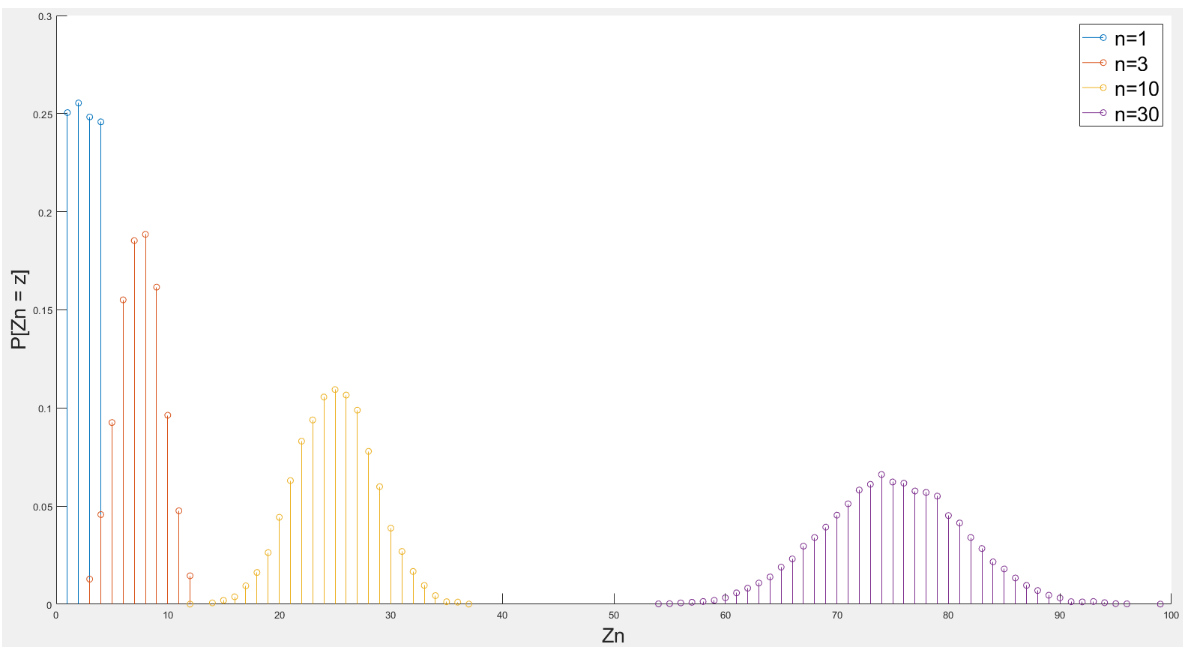
$$E[X_i] = \frac{(a+b)}{2} = \frac{5}{2};\ Var(X_i) = \frac{(b-a)^2}{12} = \frac{3}{4}$$
$$E[Z_n] = n * E[X_i] = \frac{5n}{2}\ ; Var(Z_n) = \frac{n*(b-a)^2}{12} = \frac{3n}{4}$$

n	E[Z_n]	Var(Z_n)
1	2.5	0.75
3	7.5	2.25
10	25.0	7.50
30	75.0	22.5

c.



d.

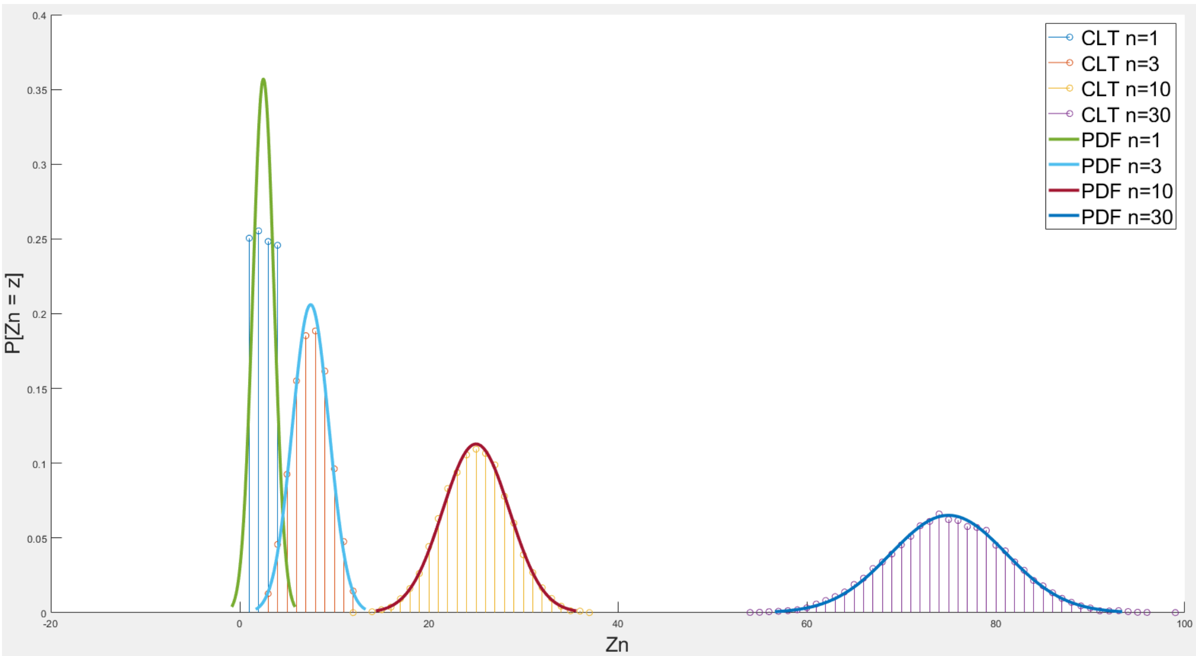


As n increases (and approaches infinity) the distribution of Z_n becomes a Gaussian distribution. This is evident by looking at the difference between n = 1, and n = 30. At n = 1, the distribution of Z_n appears to still be a uniform distribution like the iid X_i , but when n = 30, the distribution of Z_n appears to be a Gaussian distribution. The tails become wider and wider as n increases, with the peak decreasing. A difference between part (a) is that Z_n is discrete random variable.

$$E[X_i] = \frac{(a+b)}{2} = \frac{5}{2}; Var(X_i) = \frac{(b-a+1)^2 - 1}{12} = \frac{15}{12}$$

$$E[Z_n] = n * E[X_i] = \frac{5n}{2}\ ; Var(Z_n) = \frac{n * [(b-a+1)^2 - 1]}{12} = \frac{15n}{12}$$

n	$E[Z_n]$	$Var(Z_n)$
1	2.5	1.25
3	7.5	3.75
10	25.0	12.5
30	75.0	37.5



Appendix (Code)

1.

```
%% 1a
close all; clear all; clc;
tosses = [10,50,100,500,1000];
for t = tosses
    p_odd = mean(mod(randi(4,1,t),2));
    fprintf("t =%5d , p(odd number)=%.4f\n",t,p_odd)
end
%% 1d
close all; clear all; clc;
tosses = [10,50,100,500,1000];
s = RandStream('mlfg6331_64');
for t = tosses
    p_odd = mean(mod(datasample(s,1:4,t,'Weights',[1/6,2/6,1/6,2/6]),2));
    fprintf("t =%5d , p(odd number)=%.4f\n",t,p_odd)
end
```

2.

```
close all; clear all; clc;
%%
p = 0.01:0.01:.1;
N = 1:2:7;

for i = 1:length(N)
    for j = 1:length(p)
        [bits_sent,bits_encoded] = transmitted(p(j),N(i));
        p_error(i,j) = mean(decoder(bits_sent,N(i)) ~= bits_encoded);
        theoretical_error(i,j) = p_error_theor(p(j),N(i));
    end
end
subplot(1,2,1)
hold on
cmap = colormap(parula(length(p)));
for i = 1:length(p)
    plot(N,p_error(:,i),'Color',cmap(i,:),'LineWidth',2)
end
legend("p="+string(p))
title("N-repetition BSC Error Experimental Analysis")
xlabel("N",'FontWeight','bold')
ylabel("P(error)",'FontWeight','bold')

figure(1)
subplot(1,2,2)
hold on
cmap = colormap(parula(length(p)));
for i = 1:length(p)
    plot(N,p_error(:,i),'Color',cmap(i,:),'LineWidth',2)
end
set(gca,'yscale','log')
legend("p="+string(p))
title("N-repetition BSC Error Experimental Analysis")
xlabel("N",'FontWeight','bold')
ylabel("P(error) log axis",'FontWeight','bold')

array2table(p_error,'RowNames',"N="+string(N),'VariableNames',"P="+string(p))

%%
figure(2)
subplot(1,2,2)
hold on
for i = 1:length(p)
    plot(N,theoretical_error(:,i),'Color',cmap(i,:),'LineStyle','--
','LineWidth',2)
end
legend("p="+string(p))
set(gca,'yscale','log')
legend("p="+string(p))
title("N-repetition BSC Error Theoretical Analysis")
xlabel("N",'FontWeight','bold')
ylabel("P(error) log axis",'FontWeight','bold')

subplot(1,2,1)
```

```
hold on
for i = 1:length(p)
    plot(N,theoretical_error(:,i),'Color',cmap(i,:),'LineStyle','--
','LineWidth',2)
end
legend("p="+string(p))
legend("p="+string(p))
title("N-repetition BSC Error Theoretical Analysis")
xlabel("N",'FontWeight','bold')
ylabel("P(error)",'FontWeight','bold')

array2table(theoretical_error,'RowNames',"N="+string(N),'VariableNames',"P="+
string(p))



%%
function [bits_sent,bits_encoded] = transmitted(p,N)
    bits_encoded = rand(20000,1) < .5; % <.5 is 1, >= .5 is 0 transmitted
    bits_sent = rand(20000,N);
    bits_sent = (bits_sent < p) .* (1-bits_encoded) + (1-(bits_sent < p)) .*
bits_encoded;
end

function bits_received = decoder(transmission,N)
    bits_received = sum(transmission,2)>N/2;
end

function theoretical_error = p_error_theor(p,N)
    n = ceil(N/2);
    theoretical_error = 0;
    for i = n:N
        theoretical_error = theoretical_error + nchoosek(N,i)*p^i*(1-p)^(N-
i);
    end
end
```

3. (in python)

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import os


titanic = pd.read_csv("titanic.csv").rename(columns={'Survived':'S','Pclass':'C','Sex':'G','Age':'A'})

titanic.head()

titanic.describe()

titanic.isna().sum()

# a

N = titanic.shape[0]

PMF = {}

for col in titanic.columns:

    plt.figure(figsize=(7,7))

    ps = titanic[col].value_counts()/N

    plt.stem(ps.index,ps)

    plt.title("{} PMF".format(col))

    plt.xlabel(col,fontsize=15,weight='bold')

    plt.ylabel("P[{}==k]".format(col),fontsize=15,weight='bold')

    PMF[col] = ps

# b

survived = titanic.pop("S")

survived.unique()

PMF_Conditioned = {}

for survival in survived.unique():

    for col in titanic.columns:

        plt.figure(figsize=(7,7))

        N = (survived==survival).sum()

        ps = (titanic[col][survived==survival]).value_counts()  / N
```

```
    plt.stem(ps.index,ps)
    plt.title("{} | S={} PMF ".format(col,survival))
    plt.xlabel(col,fontsize=15,weight='bold')
    plt.ylabel("P[{}==k | S={}]".format(col,survival),fontsize=15,weight='bold')
    PMF_Conditioned["{}|S={}".format(col,survival)] = ps
# c
p0 = PMF['S'][0]*((PMF_Conditioned["A|S=0"][PMF_Conditioned["A|S=0"].index<=40]).sum())*(PMF_Conditioned["C|S=0"][1])*(PMF_Conditioned["G|S=0"][0])

p1 = PMF['S'][1]*((PMF_Conditioned["A|S=1"][PMF_Conditioned["A|S=1"].index<=40]).sum())*(PMF_Conditioned["C|S=1"][1])*(PMF_Conditioned["G|S=1"][0])
# d (she will survive)
p0 / (p0 + p1)
p1 / (p0 + p1)
```

4.

```matlab
%% a
close all; clc; clear all;
samples = [1,3,10,30];

i = 1;
for n = samples
    xs = rand(10000,n)*3 + 1;
    Z = sum(xs,2);
    fprintf("Z @ n =%3d = %4.5f ; =%4.5f\n",n,mean(Z),var(Z))
    z(i,:) = Z;
    i = i + 1;
end

figure(1)
hold on
for i = 1:length(samples)
   histogram(z(i,:),'Normalization','pdf')
end
legend("n="+string(samples),'FontSize',20)
fprintf("\n\n")
ylabel("P[Zn = z]",'FontSize',20)
xlabel("Zn",'FontSize',20)
%% b
clearvars -except samples

i = 1;
for n = samples
    mu = (4+1)/2;
    var = (4-1)^2 / 12;
    z_mu(i) = (mu*n);
    z_var(i) = (var*n);
    fprintf("Z_mu @ n = %3d = %4.5f\n",n,z_mu(i))
    fprintf("Z_var @ n = %3d = %4.5f\n\n",n,z_var(i))
    i = i + 1;
end
%% c
figure(1)
hold on
for i = 1:length(samples)
x = linspace(z_mu(i) - sqrt(z_var(i))*3,z_mu(i) + sqrt(z_var(i))*3,1000);
y = normpdf(x,z_mu(i),sqrt(z_var(i)));
plot(x,y,'linewidth',3)
end
legend('CLT n=1','CLT n=3','CLT n=10','CLT n=30','PDF n=1','PDF n=3','PDF
n=10','PDF n=30')


%% a
close all; clc; clear all;
samples = [1,3,10,30];

i = 1;
for n = samples
    xs = randi([1,4],10000,n);
    Z = sum(xs,2);
    fprintf("Z @ n =%3d = %4.5f ; =%4.5f\n",n,mean(Z),var(Z))
```

```
    z(i,:) = Z;
    i = i + 1;
end
%%
figure(1)
hold on
for i = 1:length(samples)
   [C,ia,ic] = unique(z(i,:));
    a_counts = accumarray(ic,1);
    value_counts = [C', a_counts];
    stem(C',a_counts/sum(a_counts));
end
legend("n="+string(samples),'FontSize',20)
fprintf("\n\n")
ylabel("P[Zn = z]",'FontSize',20)
xlabel("Zn",'FontSize',20)
%% b
clearvars -except samples

i = 1;
for n = samples
    mu = (4+1)/2;
    var = ((4-1+1)^2 - 1) / 12;
    z_mu(i) = (mu*n);
    z_var(i) = (var*n);
    fprintf("Z_mu @ n = %3d = %4.5f\n",n,z_mu(i))
    fprintf("Z_var @ n = %3d = %4.5f\n\n",n,z_var(i))
    i = i + 1;
end
%% c
figure(1)
hold on
for i = 1:length(samples)
x = linspace(z_mu(i) - sqrt(z_var(i))*3,z_mu(i) + sqrt(z_var(i))*3,1000);
y = normpdf(x,z_mu(i),sqrt(z_var(i)));
plot(x,y,'linewidth',3)
end
legend('CLT n=1','CLT n=3','CLT n=10','CLT n=30','PDF n=1','PDF n=3','PDF
n=10','PDF n=30')
```