ECE3522 Practicum 3: Achieving More Accurate Measurements

PART I:

Since this is a discrete uniform distribution, the function is:

$$f(x) = \frac{1}{b-a} = \frac{1}{56-50} = \frac{1}{6}$$

Using the formula for expected value:

$$E[x] = \frac{a+b}{2} = \frac{50+56}{2} = 53$$

The variance equation was provided and can be used to find sigma:

$$Var[x] = \frac{(b-a)*(b-a+2)}{12} = \frac{(56-50)*(56-50+2)}{12} = 4$$

$$\sigma_x = \sqrt{Var[x]} = \sqrt{4} = 2$$

Therefore,

$$E(M_n) = n * E(M_n) = n * E\left[\frac{1}{n} * x\right] = E[x] = 53$$

$$\sigma_{M_n} = \sqrt{Var[M_n]} = \sqrt{\frac{Var[x]}{n}} = \sqrt{\frac{4}{n}}$$

When n=1:

$$\sigma_{M_1} = \sqrt{Var[M_n]} = \sqrt{\frac{Var[x]}{n}} = \sqrt{\frac{4}{1}} = 2$$

When n=10:

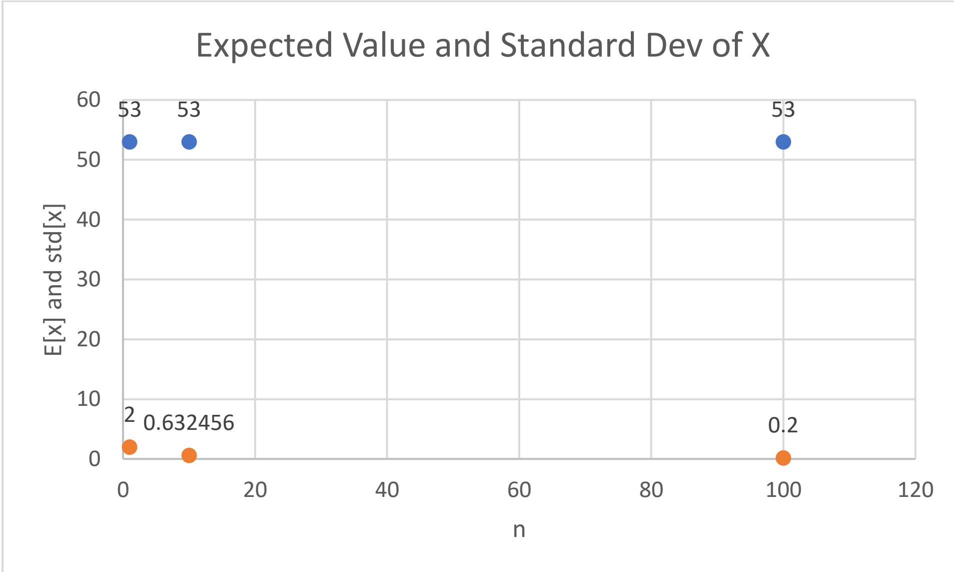
$$\sigma_{M_{10}} = \sqrt{Var[M_n]} = \sqrt{\frac{Var[x]}{n}} = \sqrt{\frac{4}{10}} = 0.632456$$

When n=100:

$$\sigma_{M_{100}} = \sqrt{Var[M_n]} = \sqrt{\frac{Var[x]}{n}} = \sqrt{\frac{4}{100}} = 0.2$$

n=	**1**	**10**	**100**
Expected Value of Mn	53	53	53
Standard Deviation of Mn	2	0.632456	0.2

Using excel to plot:



PART II:

Calculating $P[M_n \geq 53.2] = 1 - P[M_n < 53.2] = 1 - \phi(\frac{n*w - n*M_n}{\sqrt{n*{\sigma_{M_n}}^2}})$

$$=> 1 - \phi\left(\frac{100*53.2 - 100*53}{\sqrt{100*2^2}}\right) = 1 - \phi(1) = 1 - 0.8413 = 0.1587$$

When n=1,000, calculating $P[M_n \geq 53.2] = 1 - P[M_n < 53.2] = 1 - \phi(\frac{n*w - n*M_n}{\sqrt{n*{\sigma_{M_n}}^2}})$

$$=> 1 - \phi\left(\frac{1{,}000*53.2 - 1{,}000*53}{\sqrt{1{,}000*2^2}}\right) = 1 - \phi(3.1623) = 1 - 0.9992 = 0.0007827$$

PART III:

The equation for standard deviation may be rearranged to find the number of sensors

$$\sigma_x = \sqrt{\frac{Var[x]}{n}}$$

Solving for n:

$$n = \frac{Var[x]}{{\sigma_x}^2} = \frac{4}{0.01} \geq 400$$

If you were to plug this value back into the central limit theorem you would find the value given in the manual for part 5.

PART IV:

For n=100

```
N_Sensors = 100;                % number of sensors in each trial
N_Trials = 10000;               % number of trials
range = [50 56];                % range of measured pollution level
hist_range = [50:.1:56];        % range of histogram plot


%Note: Score stores an array of 100 variables during 1 loop iteration, it
%does this 1000 times
for iloop = 1:N_Trials
    score = randi(range, N_Sensors, 1);
    score_ave(iloop) = sum(score)/N_Sensors;
end


%Verifying the results of Part II experimentally P[Mn>=53.2]
P_Mn=sum(score_ave>53.2 & score_ave<56)/N_Trials
```

Running the program,

$P[M_n \geq 53.2]$ =0.1521

Changing n=1,000 and rerunning the program,

$P[M_n \geq 53.2]$ =0.0008

Both values are relatively close to the theoretical value, however the program generates a random set of integers every time it is ran, so my values happened to be quite close, but upon every rerun a new set of integers is stored, so the value will still be close but with a tolerance that will vary with every new set of sample data.

PART V:

Using the same program but now making n=400 sensors based upon my value obtained in part III,the program yields $P[M_n \geq 53.2]$ =0.0228 which is roughly a 0.00005% difference. Therefore, 400 sensors achieves a standard deviation of 0.1, and using the standard deviation function within the command window of score_ave which is the variance, would give a standard deviation of approximately .1, but this also depends on the random set of integers generated.

Appendix:

MATLAB LiveScript code:

```
clear
close all


% For Parts 4 and 5, change N_sensors for verifying part 2 and 3

N_Sensors = 100;                % number of sensors in each trial
N_Trials = 10000;               % number of trials
range = [50 56];                % range of measured pollution level
hist_range = [50:.1:56];        % range of histogram plot


%Note: Score stores an array of 100 variables during 1 loop iteration, it
%does this 1000 times
for iloop = 1:N_Trials
    score = randi(range, N_Sensors, 1);
    score_ave(iloop) = sum(score)/N_Sensors;
end


%Verifying the results of Part II and III experimentally, P[Mn>=53.2]
P_Mn=sum(score_ave>53.2 & score_ave<56)/N_Trials
hist_result = hist(score_ave, hist_range);


%Plotting results
figure(1)
plot(score,'o')
hold on
plot(score_ave(end)*ones(N_Sensors,1),'r');
xlabel('Index of sensors'); ylabel('Measured pollution level')


figure(2)
plot(score_ave)
xlabel('Index of trials');ylabel('Average pollution level')


figure(3)
plot(hist_range, hist_result/N_Trials,'-','LineWidth',2)
xlim(range)
xlabel('Average pollution level'); ylabel('Relative frequency')
```