**ECE3522 Practicum 3: Achieving More Accurate Measurements**

# PART I:

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

Using the formula for expected value:

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

Therefore,

When n=1:

When n=10:

When n=100:

|  |  |  |  |
| --- | --- | --- | --- |
| **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 [

When n=1,000, calculating [

# PART III:

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

Solving for n:

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,

[0.1521

Changing n=1,000 and rerunning the program,

[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 [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')