Instructions

This question set contains two questions, and you are supposed to write Mathcad worksheets on your own to solve the problems. Please hand in a printed version of your answers, showing your Mathcad codes (e.g. using Mathcad screen snapshots) and a sample input/output result, and please give explanations for your work. The help documents of Mathcad are very resourceful, so please make full use of it instead of searching for other people’s work online or offline.

I am pretty sure you will find it yourself how Mathcad itself could be a major problem for you, so saving your work in time and backing up your files properly is highly recommended. Should you meet with technical problems, please feel free to contact with me.

The first question contains two parts, and is meant to familiarize you with programming with Mathcad by creating a simple game. Make sure your code works.

The second one is a typical CHM621 type question, using Mathcad to evaluate concepts. Please always keep in mind that Mathcad’s normal distribution random number generator is more random than its uniform distribution one. So in many problems we have in CHM621, we will use the normal distribution random number generator, e.g. rnorm(m, \mu, \sigma). Also note that rnorm returns a vector, so if you need a single normally distributed number, you need to use rnorm(1, \mu, \sigma)₀.

Points will be given depending on the overall completion of each question.

Good luck!
Question A

(10 points)

Programming is all about having fun, so we will start by creating a game between you and your computer. Please use Mathcad to make a game called Guess the Number.

1) (5 points) First let’s be the guesser. Generate a uniformly distributed random integer within the range of $[1,100]$ first. You “guess” the number by giving the value of your guess to a variable. Then the computer decides if you are correct by comparing the generated integer with your guess variable, and tells you whether your guess is too large, too small, or correct by returning 1, -1, or 0. You keep “guessing” until you have the right answer!
Note: no loop is necessary for this question. You keep guessing by keep setting different values of the guess variable.

2) (5 points) Then let’s fight back by asking the computer to guess! Think of an integer within the range of $[1,100]$, and set a variable to that integer. Use a loop structure (e.g. while loop) to keep the computer guess your number. You are supposed to let Mathcad itself to decide whether its guess is too large, too small, or correct by making it compare its guess variable and your integer variable. Please make a nice strategy for Mathcad so it could always have the right answer in the end.
Note: bisection method is what people usually decide to use.
Question B

(10 points)

It has been discussed in the lecture that if we use sample arithmetic average instead of the actual population mean to estimate the variance, we will have a biased result. Therefore we need to make a small modification to the original variance calculation formula by multiplying by $\frac{N}{N-1}$. Let’s evaluate this bias elimination with some “actual” data.

Generate $N = 10$ normally distributed random numbers with a mean of $\mu = 100$, and a standard deviation of $\sigma = 5$.

1) (2 points) Calculate the arithmetic average $\bar{x}$ of the data, and compare with $\mu = 100$;

2) (2 points) Estimate the population variance using $\mu = 100$, and compare the result with $\sigma^2 = 25$;

3) (2 points) Estimate the population variance by calculating the biased sample variance using $\bar{x}$, and compare the result with $\sigma^2 = 25$;

4) (2 points) Estimate the population variance by calculating the unbiased sample variance using $\bar{x}$, and compare the result with $\sigma^2 = 25$;

5) (2 points) Change $N = 1000$, generate 1000 normally distributed random numbers, and repeat 1) to 4). Explain what you observe.