Computing Lab Section #1

1. *Setting Things Up.*

   Make a course directory.

   ```
   elaine20:~> mkdir s208
   elaine20:~> cd s208
   ```

   Make a file called **standnorm.m** with the following lines:

   ```
   function out=standnorm(n)
   out=randn(1,n);
   ```

   If you are working on a leland machine, you could also simply copy the file from the course directory:

   ```
   cp /usr/class/stat208/lab/standnorm.m .
   ```

2. *Starting Matlab.* Matlab is installed on all leland machines – elaine, tree, etc.
   Just type:

   ```
   elaine20:~> matlab
   ```

   You should see the matlab prompt:

   ```
   >>
   ```

3. *Warming Up.* We will use the **standnorm** function to generate a vector of 15 random numbers.

   ```
   >> standnorm(15)

   ans =

   Columns 1 through 5

  -0.4326   -1.6656    0.1253    0.2877   -1.1465

   Columns 6 through 10

   1.1909    1.1892   -0.0376    0.3273    0.1746

   Columns 10 through 15

   -0.1867    0.7258   -0.5883    2.1832   -0.1364
   ```

   Question 1 *Did you get the same answer as I did? Many of you will. The random number generator is not really random! How could you explain this?*

4. *Distribution of Sample Mean*

   Try typing in:

   ```
   >> sampling=mean(standnorm(15))

   sampling =

       0.2158
   ```

sampling is the mean of a sample of size 15.

Next,

```
>> sampling=[sampling mean(standnorm(15))]

sampling =

   0.2158    -0.1201
```

What does this command do?

Now repeat the previous command 10 more times. Stop when you see **sampling** is a vector of length 12. (**hint**: your ↑ key may save you a lot of typing in matlab).

Question 2 *If the original random variables come from the standard Normal distribution, N(0,1), what is the variance of the means of samples of size 15?*

5. *Distribution of Sample Median*

   How could we get the mean and variance of the *medians* of 100 samples of size 15?

   (a) *A clumsy approach.* Use a **for** loop to repeat commands in 4.
   If we want to discard the sampling vector we have so far, we could do:

   ```
   >> sampling=zeros(0);
   >> for i=1:100
   sampling=[sampling median(standnorm(15))];
   end
   ```

   Question 3 *From a computational point of view, this is lousy. Why?*

   (b) *A much cleaner* **for** *loop.*

   ```
   >> sampling=zeros(100,1);
   >> for i=1:100
   sampling(i)=median(standnorm(15));
   end
   ```

   (c) *Is the loop in (b) actually faster than the one in (a)?*
   Try timing it with

   i. **cputime** – cpu times;
   hint

   ```
   >> start=cputime;
   ...
   >> ending = cputime-start;
   ```

   ii. **etime** – elapsed time;
   iii. **tic toc** – stopwatch timer;
   iv. **flops** – number of floating point operation count;

   hint: Use the **help** command in Matlab.

   Question 4 *Compare time used to complete loops in (a) and (b) with above commands. Do the two loops run equally fast? Explain why, or why not.*

   Question 5 *Using the results from 5(b), how would you estimate the mean and the variance of the medians of samples of size 15?*

6. *Confidence Interval, Quantiles, etc....*

   Question 6 *Using the results from 5(b), how do you find the 95% confidence interval for the sample median? How do you find the quantiles of the sample median?*