

due November 8, 2010

### 1. Mean and Variance Algorithms

Write a program to calculate the mean and variance using the following three different methods:

- (a) Desk Calculator Algorithm
- (b) Method of Provisional Means
- (c) Centering around the First Observation

Test your program using numbers of your choice, and verify your answer using a program such as R, SAS, matlab, or a routine from the `lapack` libraries.

### 2. Ill-conditioned data

Construct a sequence of data sets such that the coefficient of variation (std. deviation / mean) becomes increasingly smaller. Continue your sequence until one or more of the algorithms in part 1 no longer gives correct answers. At what value of the coefficient of variation do you begin to see problems? At what value of the coefficient of variation does the calculation break down completely?

*Hint:* You can generate the sequence you need by adding a progressively larger constant to any set of numbers.

### 3. Uniform Random numbers

Write a function to generate uniform random numbers in the range of 0 to 1. Use a multiplicative congruential generator, with modulus  $2^{32}$ , and a value of your own choosing for  $a$  (but make sure  $\text{mod}(a,8)$  is 3 or 5.) Calculate the mean and variance for several sequences of numbers from your generator, using one of the programs written for part 1. What is the theoretical value of the mean and variance for a uniform distribution from 0 to 1? How do your observed values compare to these values?

### 4. Normal Random Numbers

Write a program to generate normally distributed random numbers with mean 0 and variance 1 by a) using the Polar method, and b) adding together six uniform random numbers. Examine the means, variances, minimums and maximums from each of ten samples from each method to insure that they look reasonable. Compare the time it takes to generate normal random variables by the two methods.

*Note:* In order to time a program, you can use the system call `getrusage()`. To make things easier, look at the file `timer.h` in `../s243/samples`.

### EXTRA CREDIT

Using one or more of the methods described in class, test your random number generator or the random generators from R, matlab, SAS, or some other source.