STAT/BIOSTAT 534 Statistical Computing
Spring Quarter 2015
Homework 7

Adrian Dobra
adobra@uw.edu

This homework is due on Wednesday, June 3, 2015 at 11:00pm. You should use the dropbox to submit your code. Please note that you will be graded not only on how your code works, but also on how readable your code is.

Problem 1 (40 points)

Please write a C/C++ program that performs the following functions.

1. Load the file "erdata.txt". This is the same data file you used in previous assignments. It has $n = 158$ rows and $p = 51$ variables. Use the GSL function `gsl_stat_covariance`
`http://www.gnu.org/software/gsl/manual/html_node/Covariance.html`
to calculate the $p \times p$ sample covariance matrix Σ of these p variables.

2. Write a function that draws independent samples from the multivariate normal distribution $N_p(0, \Sigma)$. To construct the sampler, you need to obtain the Cholesky decomposition of Σ, i.e.

$$\Sigma = \Psi \Psi^T,$$

where Ψ is a $p \times p$ lower triangular matrix. The relevant GSL function for performing the Cholesky decomposition of Σ is `gsl_linalg_cholesky_decomp`:
`http://www.gnu.org/software/gsl/manual/html_node/Cholesky-Decomposition.html#index-gsl_005flinalg_005fcholesky_005fdecomp-1343`

Next you should generate p independent $N(0, 1)$ random numbers $Z_1, Z_2, \ldots, Z_p$ using the GSL function `gsl_ran_ugaussian`:
`http://www.gnu.org/software/gsl/manual/html_node/`

The-Gaussian-Distribution.html#index-gsl_005fran_005fugaussian-1684

Arrange $Z_1, Z_2, \ldots, Z_p$ in a $p \times 1$ column matrix Z. Your random draw from $N_p(0, \Sigma)$ is given by $X = \Psi Z$. The most efficient way to calculate the matrix product between Ψ and Z is to use the GSL BLAS function `gsl_blas_dgemm`. While I would encourage you to make use of this function, you are not required to do so to successfully complete your assignment.

3. Use the sampler you wrote to simulate 10000 random draws from $N_p(0, \Sigma)$. The output of your program should be the sample covariance matrix of the 10000 random draws. Check your code by comparing the elements of this matrix with the elements of Σ.

Problem 2 (60 points)

Consider a discrete random variable X that takes values $\{0, 1, \ldots, n\}$ with probabilities

$$p_i = P(X = i) \propto \binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

Remember that $\Gamma(n+1) = n!$. Your code should work for any $n = 1, 2, \ldots$.

Please write a parallel program that samples from the distribution of X as follows. Use the Master/Slave scheme we studied in class. The Master process should proceed as follows:

1. The Master should ask each Slave process to generate 25000 samples from the distribution of X and return the results associated with the probability p_i. It is assumed that all the processes know the distribution we need to sample from, hence the work request of the Master should include two integer numbers: the number of samples to generate and $i \in \{0, 1, \ldots, n\}$.

2. The Master should wait until each Slave finishes its work and record the corresponding results. Each Slave process returns 3 real (double) numbers: i, an estimate for p_i (the sample mean) and the standard error of this estimate.

3. The Master should output the estimates of $p_0, p_1, \ldots, p_n$ and their standard errors *in this exact order* (not in the order in which the Master has received the results).

Each Slave process should proceed as follows:

1. The Slave should listen for an work order from the Master.

2. Once an order has been received, the Slave should sample the number of samples requested by the Master process using the Metropolis-Hastings algorithm described on Slide 7 of "534sampling.pdf".

3. The Slave calculates the sample mean and the sample standard error only for the probability p_i, where i was the index contained in the request received from the Master.

4. The Slave transmits the results back to the Master.