STA 4503, Spring 2013 — Programming Assignment #3

Due April 10 by 5pm (slip it under my door). Please hand it in on 8 1/2 by 11 inch paper, stapled in the upper left, with no other packaging.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you should not leave any discussion with someone else with any written notes (either on paper or in electronic form).

For this assignment, you will try to use the Stan package for MCMC with Hamiltonian Monte Carlo (from http://mc-stan.org) to sample from the posterior distribution of the Bayesian model you worked with for the first and second programming assignments.

I've installed Stan in my directory /u/radford/stan-src-1.2.0 on mercury.utstat.utoronto.ca, so if you're a Statistics graduate student you can use it from there. Otherwise, you'll need to install Stan on a machine you have access to. Let me know if you have problems doing that.

If you're using mercury.utstat, I recommend that you create a directory for this assignment, called for instance "pa3", and then copy to it two shell files that I have written to make it easier for you to use the Stan installation in my directory. You can do this as follows:

```
mkdir pa3
cd pa3
cp /u/radford/stan-test/stan-compile .
cp /u/radford/stan-test/stan-print .
```

I've also put these shell files on the course web page. If you're using not using mercury.utstat, you might be able to adapt them for use on whatever system you're using.

To use Stan to fit a model, you first need to create a Stan program that specifies the model, which you put in a file ending in ".stan", for instance "pa3.stan". To write your Stan program, you'll need to read the Stan manual, and look at example Stan programs (which may be found in the src/models subdirectory of the Stan source directory).

Once you've written your program, you can compile it with the command

```
./stan-compile pa3
```

You can then run the compiled Stan program to sample from the posterior distribution given data that is in, for instance, the file "stan-data1", as follows:

```
./pa3 --data=stan-data1
```

The data file has to be in the format Stan expects, which is a series of R-like assignment statements. I have converted the three data files you used before to this format and put them on the course web page. The command above runs the Markov chain with default settings, discards an initial "warmup" portion, and writes the remaining sample points to the file "samples.csv" (by default, this can be changed). You can see a summary of the results with the command

```
./stan-print samples.csv
```

You can also read the "samples.csv" file into R, with the R command

```
s <- read.table("samples.csv",head=TRUE,sep=",")
```

The data frame read will have the values of the variables sampled at each iteration saved from the MCMC run, plus some other values concerning what happened each iteration. The "samples.csv" file could instead be read by some program other than R, but note that it has comment lines and a header line in addition to the lines with actual numbers.

A version of Stan meant for use from R, called RStan, is also available from the Stan web site, but I haven't played with it much yet. It might be an easier way to go if you are going to use R, though as far as I can tell the actual MCMC is no different from plain Stan.

For this assignment, you should see how well Stan works with the model and datasets used in the previous assignments, both when using its default MCMC method, the No-U-Turn sampler, and when using standard HMC. You can get standard HMC using the --epsilon and --leafrog_steps options when running the compiled Stan program. The --iter option may also be useful to increase the number of iterations done. The --nondiag_mass argument tells Stan to use a kinetic energy of the form $p^T M^{-1} p/2$, for some "mass matrix" M, that is estimated during the warmup period (this is equivalent to linearly transforming the original variables). You should play around with all these options and others (which you can see with the --help option) to try to get good sampling.

You should hand in your Stan program describing the model, your commands for the runs of your program that you did (so that I can see what options you used), the printed output and/or plots of results, and a discussion of what they mean — that is, what your results say about how well Stan works for this model with the three data files, what options lead to best performance, and whatever you can say that to explain why the results are as they are.