Question (Geyer) Consider simulated data for logistic regression found in the URL

`http://www.stat.umn.edu/geyer/PhD/F03/logit.txt`

which can be read into R by the following commands

```
foo <- read.table(
    url("http://www.stat.umn.edu/geyer/PhD/F03/logit.txt"),
    header = TRUE)
```

assuming the computer is connected to the internet. Since the file is plain text with variables in white-space-separated columns headed by variable names, it can be read into any other computer package with minimal effort. There are one response variable `y` and four predictor variables named `x1` through `x4`.

We assume the data follow the usual logistic regression model, the response variables y_i, $i = 1, \ldots, n$ are independent Bernoulli(p_i) random variables with the success probabilities being defined by

$$\eta_i = \operatorname{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right)$$

(logit link function) and the linear predictor vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$ is defined by

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$$

where $\mathbf{X}$ (the "design matrix" or "model matrix") is $n \times 5$ with one column of 1's (the constant predictor) and the other four columns being the vectors `x1` through `x4` of the data set.

In short, we assume the model whose frequentist analysis is done by the following R commands (assuming the data have been read into the data frame `foo` as shown above)

```
out <- glm(y ~ x1 + x2 + x3 + x4, data = foo,
    family = binomial())
summary(out)
```

But this problem isn't about that frequentist analysis, we want a Bayesian analyis. For our Bayesian analysis we assume the same data model as the frequentist, and we assume the prior distribution of the five parameters (the regression coefficients) makes them independent and identically normally distributed with mean 0 and standard deviation 2.

(a) Construct a Markov chain Monte Carlo sampler for the posterior distribution of these parameters. Describe your sampler in sufficient detail so that an expert could duplicate your results.

If you wish, you may use the `metropolis` function written for the 8701 course. The current version is installed on the system. The R command

```
library(mcmc)
```

makes it available. **Caution:** The main change to the function from previous versions is that the supplied R function now must calculate *log* unnormalized density, where the previous versions wanted plain (no log) unnormalized density.

(b) Provide Monte Carlo estimates for

  (i) the posterior mean of each of the five regression coefficients, and

  (ii) the posterior variance of each of the five regression coefficients.

(c) Provide Monte Carlo standard errors (MCSE) for the ten posterior expectations reported in part (b), where MCSE are estimated standard errors due to Monte Carlo sampling in these quantities. Describe your method of calculating MCSE in sufficient detail so that an expert could duplicate your results.

Use a long enough run of your Markov chain sampler so that the MCSE are less than 0.01.

(d) Provide a plain text file of R commands that when batched reproduces your results, that is, supposing your file is named `foo.R` the command

```
R CMD BATCH foo.R
```

produces a file `foo.Rout` that contains *all* the results you report in your write-up. In order to get the same results every time, include the command

```
set.seed(42)
```

at the top of your `foo.R` file (you may choose another number if you don't like 42).

You may use any other widely available computer package so long as you can batch process a plain text file in a manner similar to that described here.

To preserve anonymity, submit your plain text computer file by e-mail to `dana@stat.umn.edu`.

(e) Also submit a brief write-up describing what you did in plain English (no computer code) and summarizing the results.

Solution (Geyer) This problem is not "Gibbs friendly" because the one-dimensional conditionals are not "brand name" distributions. Hence I used a Metropolis "random walk" sampler with a multivariate normal proposal centered at the current position having variance a constant times the identity. After some experimentation trying 0.1, 0.2, 0.3, and 0.4 for the constant, I settled on the latter, which gave acceptance rates of 0.239, 0.2365, 0.2346333, and 0.23484 in various runs.

The results were for $E(\beta \mid \text{data})$ in a run of length 2×10^5 were

	intercept	x_1	x_2	x_3	x_4
mean	0.6584	0.8008	1.1706	0.5016	0.7269
MCSE	0.0026	0.0034	0.0033	0.0031	0.0038

and the results were for $\text{var}(\beta \mid \text{data})$ in the same run were

	intercept	x_1	x_2	x_3	x_4
mean	0.0926	0.1362	0.1304	0.1249	0.1616
MCSE	0.0010	0.0016	0.0015	0.0013	0.0019

The MCSE were determined by the method of overlapping batch means using batches of length 50. Autocorrelation plots (not shown) indicated no significant autocorrelation past lag 35, so batch length 50 should be safe.

My batch file and the results of running it are at

```
http://www.stat.umn.edu/geyer/PhD/F03/foo.R
http://www.stat.umn.edu/geyer/PhD/F03/foo.Rout
```

Details about MCSE of Posterior Variance A variance is just an expectation, $\text{var}(X) = E\{(X - \mu)^2\}$. If

$$m_i = \frac{1}{n} \sum_{j=1}^{n} \beta_{i,j}$$

is your estimate of the posterior mean of β_i (where β_{ij} is the value of β_i in the j-th iteration of the Markov chain) then

$$v_i = \frac{1}{n} \sum_{j=1}^{n} (\beta_{i,j} - m_i)^2$$

is your estimate of the posterior variance of β_i. And this is "just" an average (variance is a special case of expectation, sample variance is a special case of sample average).

So you calculate MCSE for v_i exactly the same way as you calculate it for m_i. To calculate it for m_i you hand the sequence

$$j \mapsto \beta_{i,j}$$

to your favorite method of Markov chain variance estimation, say OLBM, and to calculate it for v_i you hand the sequence

$$j \mapsto (\beta_{i,j} - m_i)^2$$

to your favorite method of Markov chain variance estimation, say OLBM.

You don't need to invent something new because variance is a special case of expectation.

Additional Stuff Not asked, but of some interest, is the posterior standard deviation (that is, the square root of the reported posterior variance). The reason it was not asked is that its MCSE would involve the delta method. Anyway it was

	intercept	x_1	x_2	x_3	x_4
mean	0.3044	0.3690	0.3611	0.3535	0.4020
MCSE	0.0016	0.0021	0.0020	0.0019	0.0024