Dave Kleinschmidt
BST 413 HW 5
April 17, 2013

1. Just myself.
2. Just Gelman et al.
3. Seemed pretty reasonable. Don't quite remember how long I worked on it.
4. (a) The number of bicycles on each block y_j (out of n_j total vehicles), $j = 1 \dots J$ is modeled with a binomial distribution with a rate of θ_j.

$$p(y_j|\theta_j, n_j) = \binom{n_j}{y_j}\theta_j^{y_j}(1-\theta_j)^{n_j-y_j} \tag{1}$$

The rates θ_j are linked via a Beta population distribution:

$$p(\theta_j|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta_j^{\alpha-1}(1-\theta_j)^{\beta-1} \tag{2}$$

As suggested by Gelman et al. for the rats problem, a reasonable hyperior is uniform on $(\frac{\alpha}{\alpha+\beta}, (\alpha+\beta)^{-1/2})$, which has density

$$p(\alpha,\beta) \propto (\alpha+\beta)^{-5/2} \tag{3}$$

The hyperparameters themselves are somewhat difficult to interpret, but can be transformed to a more natural scale, the (log) *prior sample size* $\log(\alpha+\beta)$ and the *expected log-odds*, $\log(\alpha/\beta)$. On this scale, the prior has density

$$p(\log(\alpha/\beta), \log(\alpha+\beta)) \propto \alpha\beta(\alpha+\beta)^{-5/2} \tag{4}$$

Under this model, the joint posterior distribution is

$$\begin{aligned}
p(\theta,\alpha,\beta|y) &\propto p(\alpha,\beta)\prod_j p(y_j|\theta_j)p(\theta_j|\alpha,\beta) && (5)\\
&\propto p(\alpha,\beta)\prod_j \binom{n_j}{y_j}\theta_j^{y_j}(1-\theta_j)^{n_j-y_j}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta_j^{\alpha-1}(1-\theta_j)^{\beta-1} && (6)\\
&\propto p(\alpha,\beta)\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^J\prod_{j=1}^J\binom{n_j}{y_j}\theta_j^{y_j+\alpha-1}(1-\theta_j)^{n_j-y_j+\beta-1} && (7)
\end{aligned}$$

leaving the prior as $p(\alpha,\beta)$ because the parametrization used will change the actual prior density function, but no other parts of the joint posterior.

(b) The marginal posterior is

$$\begin{aligned}
p(\alpha,\beta|y) &\propto p(\alpha,\beta)\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^J\prod_j\binom{n_j}{y_j}\frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)} && (8)\\
&\propto (\alpha+\beta)^{-5/2}\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^J\prod_j\binom{n_j}{y_j}\frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)} && (9)
\end{aligned}$$

$$p(\log(\alpha+\beta), \log(\alpha/\beta)|y) \propto \alpha\beta(\alpha+\beta)^{-5/2}\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^J\prod_j\binom{n_j}{y_j}\frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)} \tag{10}$$

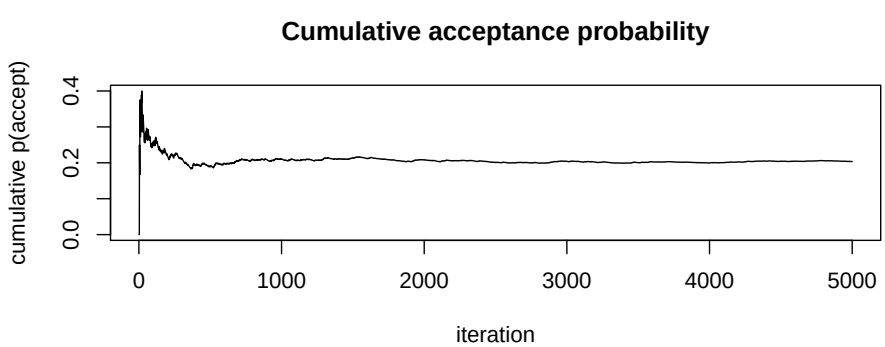
The conditional posterior of the parameters θ_j is Beta$(\alpha+y_j, \beta+n_j-y_j)$, so

$$p(\theta_j|\alpha,\beta,y) = \frac{\Gamma(\alpha+\beta+n_j)}{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}\theta_j^{\alpha+y_j-1}(1-\theta_j)^{\beta+n_j-y_j-1} \tag{11}$$

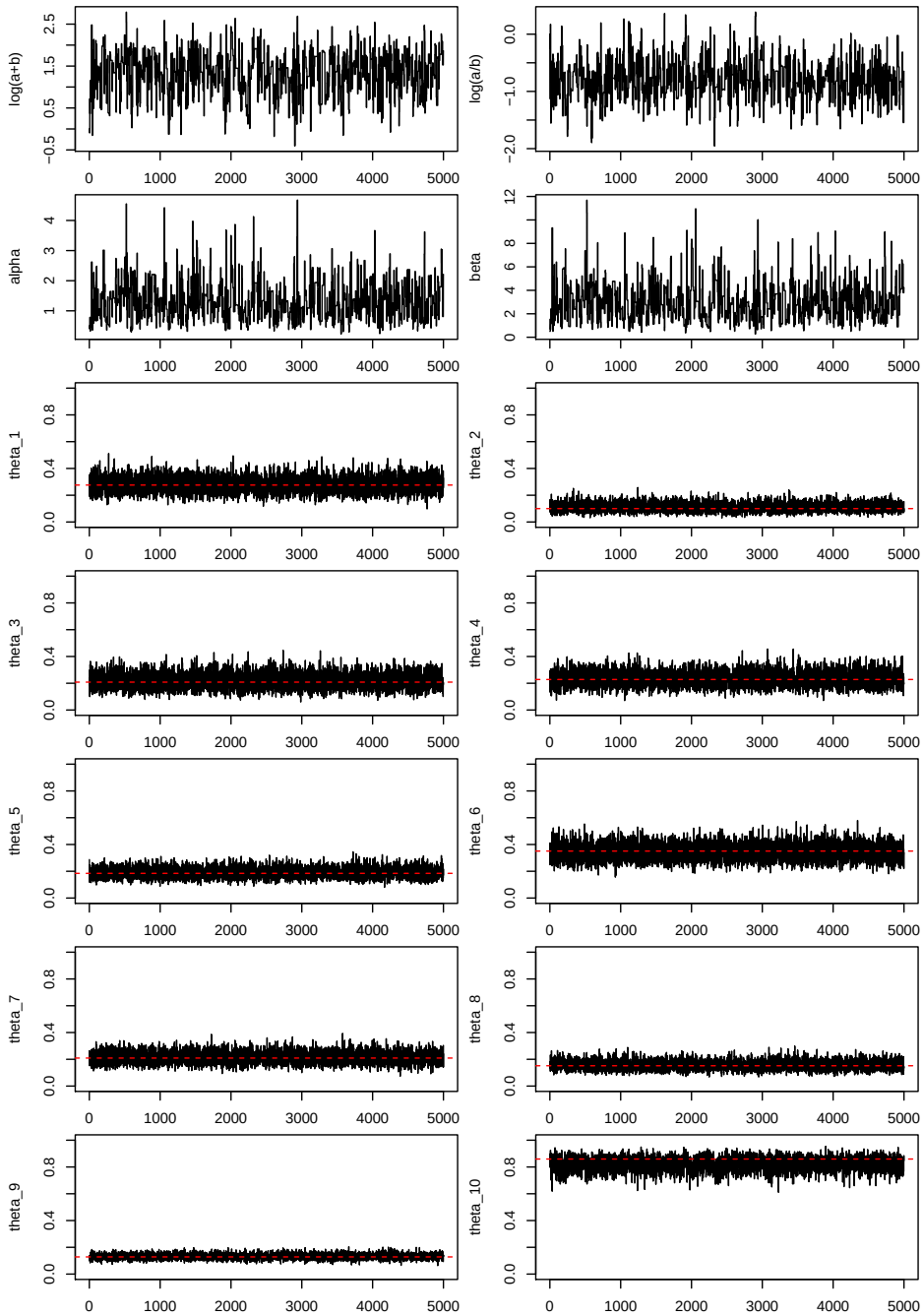
(c)
```
> require(mvtnorm)
>
> # two ways of calculating the log prior: one transforms back to a,b, the other
> # just uses the log sample size and the log odds:
> log.hyper.prior <- function(log.samp.size, logodds) {
+   return(logodds - 0.5 * log.samp.size - 2 * log(1+exp(logodds)))
+ }
> log.hyper.prior.ab <- function(a, b) {
+   return(log(a) + log(b) - 2.5 * log(a+b))
+ }
> # convert from log sample size/log odds parametrization to alpha/beta
> log.odds.ss.to.ab <- function(log.samp.size, logodds) {
+   odds <- exp(logodds)
+   samp.size <- exp(log.samp.size)
+   list(a=odds*samp.size / (1+odds),
+        b=samp.size / (1+odds))
+ }
>
> log.post.hyper <- function(log.samp.size, logodds, data=bikes) {
+   # convert from log sample size/log odds parametrization to alpha/beta
+   params <- log.odds.ss.to.ab(log.samp.size, logodds)
+   a <- params$a
+   b <- params$b
+   # initialize log density with log-prior and the normalization constant constant in y
+   J <- nrow(data)
+   L <- log.hyper.prior.ab(a, b) + (lgamma(a+b) - lgamma(a) - lgamma(b)) * J
+   # add log-density contribution from each block
+   for (j in 1:J) {
+     L <- L + lgamma(a+data[j, 'y']) + lgamma(b+data[j, 'n']-data[j, 'y']) - lgamma(a+b+data[j
+   }
+   return(L)
+ }
>
> # x[1] is log sample size, and x[2] is log-odds
> Q.bikes <- function(x) rmvnorm(1, x, diag(1, 2))
> q.bikes <- function(y, x) dmvnorm(x, y, diag(1,2))
> p.bikes <- function(x) exp(log.post.hyper(log.samp.size=x[1], logodds=x[2]))
>
> met.hast <- function(p, q, Q, x0, n.iter=1000) {
+   results <- list(iter=rep(NA, n.iter),
+                   x=matrix(NA, nrow=n.iter, ncol=length(x0)),
+                   x.prop=matrix(NA, nrow=n.iter, ncol=length(x0)),
+                   alpha=rep(NA, n.iter),
+                   u=rep(NA, n.iter),
+                   acc=rep(NA, n.iter))
+
+   x.cur <- x0
+   for (i in 1:n.iter) {
+     results$iter[i] <- i
+     # draw from proposal distribution, and store
+     x.prop <- Q(x.cur)
+     results$x.prop[i, ] <- x.prop
+     # calculate acceptance probability
```

```
+     results$alpha[i] <- min(1, p(x.prop) * q(x.cur, x.prop) / p(x.cur) / q(x.prop, x.cur))
+     results$u[i] <- runif(1)
+     if (results$u[i] <= results$alpha[i]) {
+       results$acc[i] <- TRUE
+       results$x[i, ] <- x.prop
+       x.cur <- x.prop
+     } else {
+       results$acc[i] <- FALSE
+       results$x[i, ] <- x.cur
+     }
+   }
+   return(results)
+ }
>
> mh.test <- met.hast(p=function(x) dnorm(x, 0, 3),
+                     q=function(x, y) dnorm(x, y, 3),
+                     Q=function(x) rnorm(1, x, 3),
+                     x0=0)
>
>
> set.seed(12345)
> mh.bikes <- met.hast(p=p.bikes,
+                      q=q.bikes,
+                      Q=Q.bikes,
+                      x0=c(0,0),
+                      n.iter=5000)
>
> # convert to alpha/beta
> ab.samp <- t(apply(mh.bikes$x, 1,
+                    function(x) unlist(log.odds.ss.to.ab(log.samp.size=x[1], logodds=x[2]))))
> # sample parameters for each draw of alpha/beta
> theta.samp <- t(apply(ab.samp, 1, function(ab) rbeta(nrow(bikes),
+                                                      ab[1] + bikes$y,
+                                                      ab[2] + bikes$n - bikes$y)))
> dimnames(theta.samp) <- list(samp=1:nrow(theta.samp), block=1:ncol(theta.samp))
```

(d) The overall acceptance rate is 0.20.

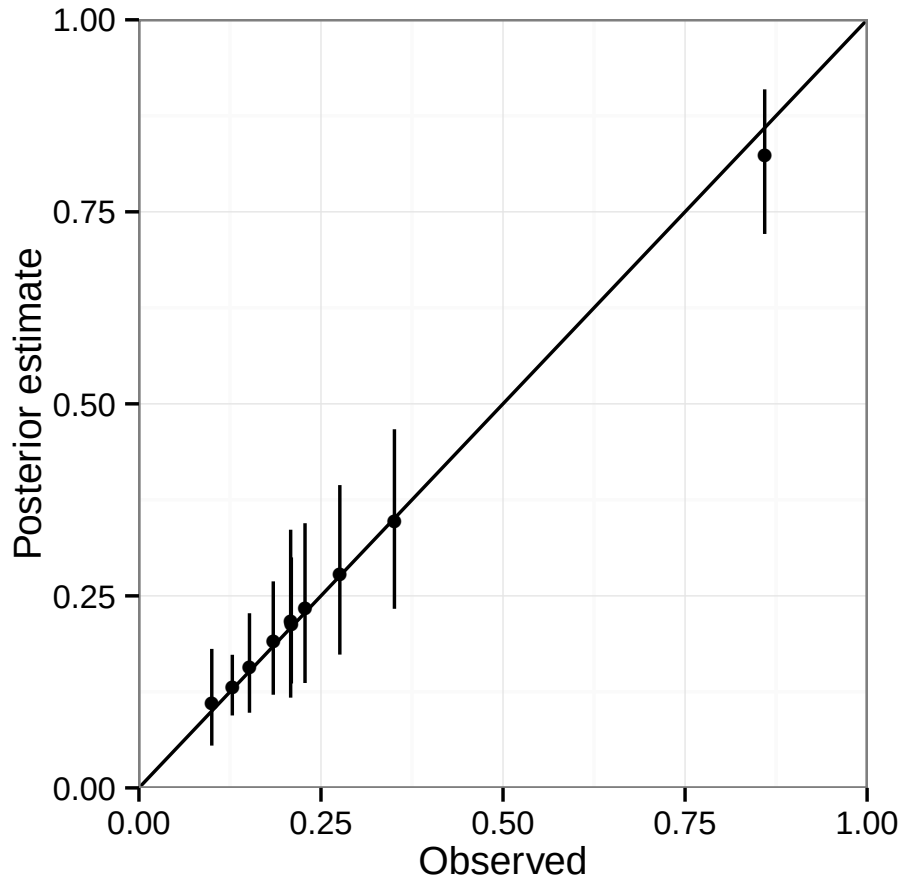


(e) The red dashed lines show the point estimates based on the observed frequencies of bikes:



(f)

	2.5%	mean	97.5%
1	0.17	0.28	0.39
2	0.06	0.11	0.18
3	0.12	0.22	0.34
4	0.14	0.23	0.34
5	0.12	0.19	0.27
6	0.23	0.35	0.47
7	0.14	0.21	0.30
8	0.10	0.16	0.23
9	0.09	0.13	0.17
10	0.72	0.82	0.91



(g)

(h) The 95% posterior interval for the number of bikes, $\frac{\alpha}{\alpha+\beta}$ is $[0.19, 0.47]$. This is basically the same as the interval obtained from the grid sampling method used in HW4, which was $[0.19, 0.46]$.

(i) For this application, the MCMC approach doesn't offer any substantial advantage. Both methods require deriving the marginal posterior distribution function for α and β, which is the hardest part. Grid sampling woud be reasonably easy to implement in reusable code, as the Metropolis-Hastings sampler used here is. For a problem where it was not possible to derive the marginal posterior of the hyperparameters analytically, MCMC approaches would be substantially better, because when the group-level parameters θ_j are considered, the dimensionality of the full, joint posterior becomes too high to realistically compute the joint posterior density over a fine enough grid.

5. (a) The posterior distribution of the regression parameters is

$$p(\beta|y, X, \sigma^2) \propto p(y|\beta, X, \sigma^2)p(\beta|\sigma^2) \tag{12}$$

$$\propto \mathcal{N}(y|X\beta, \sigma^2 I)\mathcal{N}(\beta|\beta_0, \sigma^2\Sigma_0) \tag{13}$$

$$\propto \exp\left(-\frac{1}{2}(y - X\beta)^T(\sigma^2 I)^{-1}(y - X\beta)\right)\exp\left(-\frac{1}{2}(\beta - \beta_0)^T(\sigma^2\Sigma_0)^{-1}(\beta - \beta_0)\right) \tag{14}$$

For the purposes of finding the posterior, the kernel of the likelihood can be rewritten as the kernel of a $\mathcal{N}(\beta|\hat{\beta}, \sigma^2 V_\beta)$ distribution, with mean $\hat{\beta} = (X^T X)^{-1}X^T y$ and variance $\sigma^2 V_\beta = \sigma^2(X^T X)^{-1}$.

Then,

$$p(\beta|y,X,\sigma^2) \propto \exp\left(-\frac{1}{2}(\beta-\hat{\beta})^T(\sigma^2 V_\beta)^{-1}(\beta-\hat{\beta})\right)\exp\left(-\frac{1}{2}(\beta-\beta_0)^T(\sigma^2\Sigma_0)^{-1}(\beta-\beta_0)\right) \tag{15}$$

$$\propto \exp\left(-\frac{1}{2\sigma^2}\left((\beta-\hat{\beta})^T V_\beta^{-1}(\beta-\hat{\beta}) + (\beta-\beta_0)^T\Sigma_0^{-1}(\beta-\beta_0)\right)\right) \tag{16}$$

Distributing all the multiplication dropping terms that are constants with respect to β, and combining the common β terms, and letting $V = (V_\beta^{-1} + \Sigma_0^{-1})^{-1}$, this becomes

$$\propto \exp\left(-\frac{1}{2\sigma^2}\left(\beta^T(V_\beta^{-1}+\Sigma_0^{-1})\beta - \beta^T(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0) - (V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0)^T\beta\right)\right) \tag{17}$$

$$\propto \exp\left(-\frac{1}{2\sigma^2}\left(\beta^T V^{-1}\beta - \beta^T V^{-1}V(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0) - (V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0)^T V V^{-1}\beta\right.\right. \tag{18}$$

$$\left.\left. +(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0)^T V V^{-1} V(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0)\right)\right) \tag{19}$$

$$\propto \exp\left(-\frac{1}{2\sigma^2}\left((\beta - V(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0))^T V^{-1}(\beta - V(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0))\right)\right) \tag{20}$$

$$\propto \mathcal{N}(\beta|V(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0), \sigma^2(V_\beta^{-1}+\Sigma_0^{-1})^{-1}) \tag{21}$$

$$= \mathcal{N}(\beta|M,\sigma^2 V) \tag{22}$$

Where the posterior precision is the sum of the prior and likelihood precisions, $\sigma^{-2}V^{-1} = \sigma^{-2}(V_\beta^{-1}+\Sigma_0^{-1}) = \sigma^{-2}(X^TX+\Sigma_0^{-1})$, and the posterior mean is the precision-weighted average of the maximum likelihood parameters $\hat{\beta} = (X^TX)^{-1}X^Ty$ and the prior estimate β_0, $M = V(V_\beta^{-1}\hat{\beta}+\Sigma_0^{-1}\beta_0) = V(X^Ty+\Sigma_0^{-1}\beta_0)$.

(b) The marginal posterior $p(\sigma^2|y,X)$ can be derived analytically by noting that $p(\sigma^2|y,X) = \frac{p(\sigma^2,\beta|y,X)}{p(\beta|\sigma^2,y,X)}$. The first step is to derive the joint posterior, expressed in terms of the posterior mean and variance of β as derived above, but leaving in all the terms that were dropped when completing the square above:

$$p(\beta,\sigma^2|y,X) \propto p(y|X,\beta,\sigma^2)p(\beta|\sigma^2)p(\sigma^2) \tag{23}$$

$$\propto (\sigma^2)^{-n/2}\exp\left(-\frac{1}{2\sigma^2}(y-X\beta)^T(y-X\beta)\right) \tag{24}$$

$$(\sigma^2)^{-1/2}\exp\left(-\frac{1}{2\sigma^2}(\beta-\beta_0)^T\Sigma_0^{-1}(\beta-\beta_0)\right) \tag{25}$$

$$(\sigma^2)^{-(a+1)}\exp\left(-\frac{1}{\sigma^2}b\right) \tag{26}$$

$$\propto (\sigma^2)^{-(n+2a+3)/2}\exp\left(-\frac{1}{2\sigma^2}(\beta-M)^TV^{-1}(\beta-M)\right) \tag{27}$$

$$\exp\left(-\frac{1}{2\sigma^2}\left(\hat{\beta}^TV_\beta^{-1}\hat{\beta}+\beta_0^T\Sigma_0^{-1}\beta_0 - M^TV^{-1}M + 2b\right)\right) \tag{28}$$

For the purposes of computing the posterior of σ^2, the denominator is

$$p(\beta|\sigma^2,y,X) = \mathcal{N}(\beta|M,\sigma^2V) \tag{29}$$

$$\propto (\sigma^2)^{-(d+1)/2}\exp\left(-\frac{1}{2\sigma^2}(\beta-M)^TV^{-1}(\beta-M)\right) \tag{30}$$

where $d+1$ is the number of predictors (including the intercept), the dimensionality of β. Noting that this cancels the first exponential term from $p(\beta,\sigma^2|y,X)$ (27), the marginal posterior of σ^2

is

$$p(\sigma^2|y,X) = \frac{p(\sigma^2,\beta|y,X)}{p(\beta|\sigma^2,y,X)} \tag{31}$$

$$\propto (\sigma^2)^{-(\frac{n-d+2a}{2}+1)} \exp\left(-\frac{1}{2\sigma^2}\left(\hat{\beta}^T V_\beta^{-1}\hat{\beta} + \beta_0^T \Sigma_0^{-1}\beta_0 - M^T V^{-1} M + 2b\right)\right) \tag{32}$$

That is,

$$\sigma^2|y,X \sim \mathrm{IG}\left(a + \frac{n-d}{2}, b + \frac{1}{2}\left(\hat{\beta}^T V_\beta^{-1}\hat{\beta} + \beta_0^T \Sigma_0^{-1}\beta_0 - M^T V^{-1} M\right)\right) \tag{33}$$

(c) The conditional posterior of σ^2 is simply

$$p(\sigma^2|\beta,X,y) \propto p(y|X,\beta,\sigma^2)p(\sigma^2) \tag{34}$$

$$\propto (\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}(y-X\beta)^T(y-X\beta)\right)(\sigma^2)^{-(a+1)} \exp\left(-\frac{1}{\sigma^2}b\right) \tag{35}$$

$$\propto (\sigma^2)^{-(a+n/2+1)} \exp\left(-\frac{1}{\sigma^2}\left(b + \frac{1}{2}(y-X\beta)^T(y-X\beta)\right)\right) \tag{36}$$

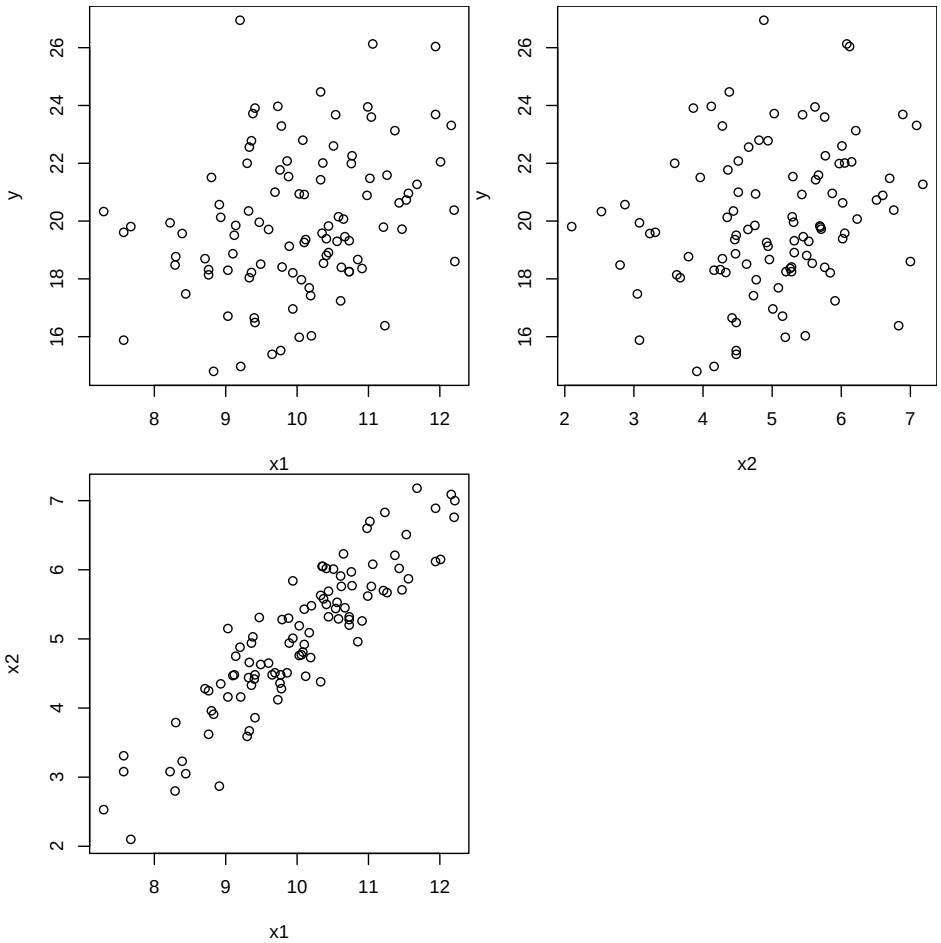
and so $\sigma^2|\beta,y,X \sim \mathrm{IG}(a+\frac{n}{2}, b+\frac{1}{2}(y-X\beta)^T(y-X\beta))$.

6. (a)

```
> dat <- read.table('hw5.txt')
> dat.lm <- lm(data=dat, y ~ x1 + x2)
> print(dat.lm.summ <- summary(dat.lm))

##
## Call:
## lm(formula = y ~ x1 + x2, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5298 -1.4690 -0.2705  1.7186  7.7025
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.8519     3.0340   3.906 0.000171 ***
## x1            0.9825     0.5159   1.904 0.059749 .
## x2           -0.3368     0.5239  -0.643 0.521838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.418 on 99 degrees of freedom
## Multiple R-squared: 0.08998,Adjusted R-squared: 0.07159
## F-statistic: 4.894 on 2 and 99 DF,  p-value: 0.009399

>
> print(beta.hat <- dat.lm.summ$coefficients[, 'Estimate'])

## (Intercept)          x1          x2
##  11.8518960   0.9824974  -0.3367573

> print(sigmasq.hat <- dat.lm.summ$sigma^2)

## [1] 5.844507

> beta.se <- dat.lm.summ$coefficients[, 'Std. Error']
> print(beta.hat + beta.se %o% c(-1.96, 1.96))
```

```
##                     [,1]       [,2]
## (Intercept)  5.90523375 17.7985582
## x1          -0.02863648  1.9936313
## x2          -1.36358073  0.6900662
```

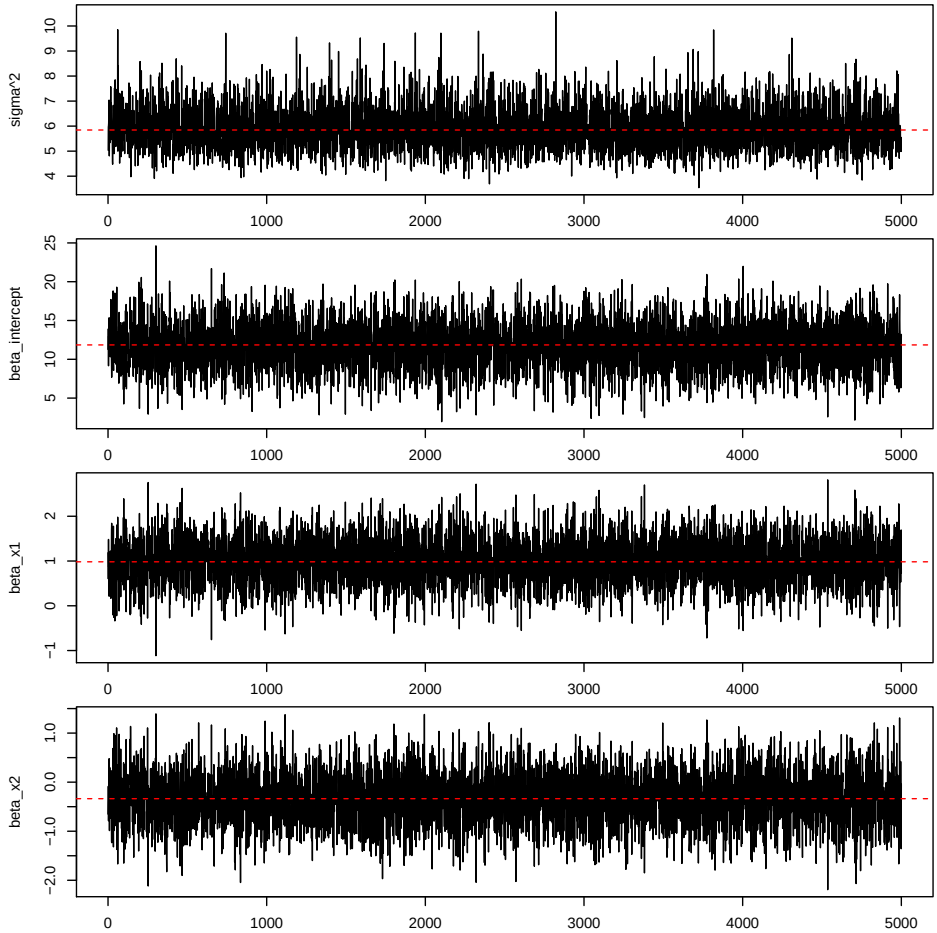
(b) Each predictor is only weakly correlated with the outcome variable: $\text{cor}(x_1, y) = 0.29$, and $\text{cor}(x_2, y) = 0.24$. However, the two predictors are themselves highly correlated: $\text{cor}(x_1, x_2) = 0.90$



(c)
```
>
>
> # draw sample of beta ~ mvnorm(M, sigma^2 V)
> #   M = V * (X^T * y + Sigma_0^-1 * beta_0),
> #   V = (X^T * X + Sigma_0^-1)^-1
> # draw sample of sigma^2 ~ IG(a+n/2, b+1/2 inner(y-X\beta))
>
> X <- with(dat, cbind(rep(1, nrow(dat)), x1, x2))
> y <- matrix(dat$y, ncol=1)
> n <- nrow(X)
>
> # hyperparameters
> beta0 <- matrix(c(0,0,0), ncol=1)
> Sigma0 <- solve(diag(rep(0.01, 3)))
>
> # pre-compute parameters for coefficient posterior
```

```
> V <- solve(t(X) %*% X + solve(Sigma0))
> M <- V %*% (t(X) %*% y + solve(Sigma0) %*% beta0)
>
> # hyperparameters for IG prior on sigma^2
> a <- b <- 0.5
>
> a.post <- a + n/2
> b.post <- function(beta) b + 0.5 * sum((y-X %*% beta)^2)
>
> # set number of iterations and pre-allocate samples
> niter <- 5000
> beta.samp <- matrix(NA, nrow=niter, ncol=length(beta0))
> sigmasq.samp <- matrix(NA, nrow=niter, ncol=1)
>
> # initialize parameters
> beta <- beta0
> sigmasq <- b.post(beta.hat) / a.post
>
> # draw samples with Gibbs sampling
> for (i in 1:niter) {
+   # draw sample of beta ~ mvnorm(M, sigma^2 V)
+   #   M = V * (X^T * y + Sigma_0^-1 * beta_0),
+   #   V = (X^T * X + Sigma_0^-1)^-1
+   beta <- rmvnorm(1, M, sigmasq * V)
+   beta.samp[i, ] <- beta
+   # draw sample of sigma^2 ~ IG(a+n/2, b+1/2 inner(y-X\beta))
+   sigmasq <- 1 / rgamma(1, a.post, rate=b.post(t(beta)))
+   sigmasq.samp[i] <- sigmasq
+ }
```

(d) Dashed red lines show the point estimates derived via `lm` above:



(e)

	2.5%	mean	97.5%
β_0	5.87	11.71	17.74
β_1	-0.02	1.00	2.02
β_2	-1.37	-0.34	0.70
σ^2	4.46	5.90	7.78

(f) They don't seem to be overly influential. Both the posterior means and intervals agree pretty well with the point estimates computed via `lm` above.

(g) Missing data can be sampled as another Gibbs step, using the posterior predictive distribution $p(\tilde{y}|\tilde{X}, \beta, \sigma^2) = \mathcal{N}(\tilde{y}|\tilde{X}\beta, \sigma^2 I)$ (where $\tilde{X}$ is the model matrix for the missing data). The other parameters (β and σ^2) can be sampled via Gibbs sampling as above.