Homework 5

Ryan Hastings
Sta 360: Bayesian and Modern Statistics
Duke University

August 3, 2020

Question 1: Swimming Data

Recall the problem from class on swimming times. Download the data here: http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/swim.dat.

```
Y <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/swim.dat")
colnames(Y) <- paste0("W", seq(2, 12, 2))
Y <- t(Y)
```

The file contains data on the amount of time in seconds it takes each of 4 high school swimmers to swim 50 yards. There are 6 times for each student, taken every two weeks. That is, each swimmer has six measurements at $W = 2, 4, 6, 8, 10, 12$ weeks. Each row corresponds to a swimmer and a higher column index indicates a later date. Assume again that the model for each swimmer is

$$T_i = \beta_0 + \beta_1(W_i - \bar{W}) + \epsilon_i$$

where T_i represents the swimming times and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

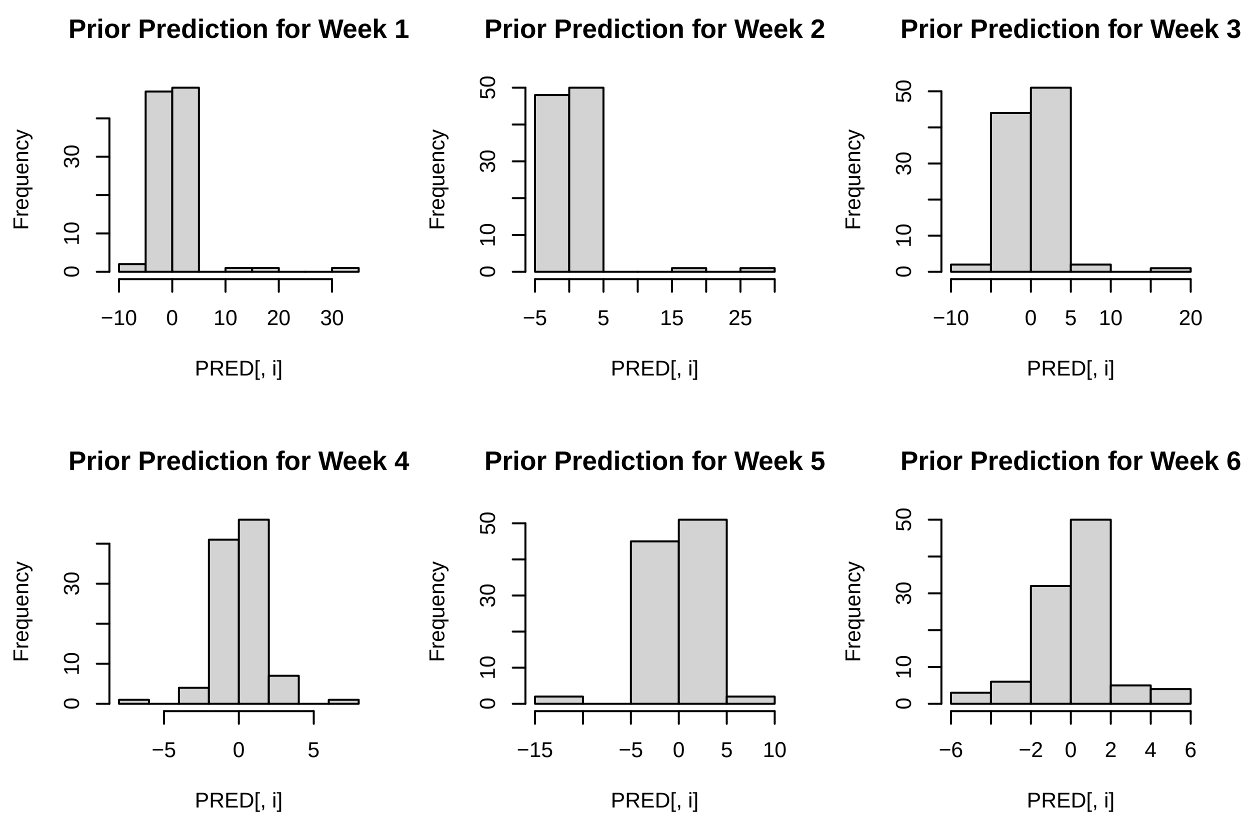
Part A

Using the g-prior with $g = n = 6$, generate samples/realizations from the prior predictive distribution for a single swimmer over the 12 weeks ($W = 2, 4, 6, 8, 10, 12$) and create a density plot of the predictive draws (one for each W). Are the values plausible?

$$T_i = \beta_0 + \beta_1(W_i - \bar{W}) + \epsilon_i$$
$$T \sim \mathcal{N}_n(X\beta, \sigma^2 \mathbb{I}_n)$$
$$\pi(\beta|\sigma^2) = \mathcal{N}_p(\mu_0 = 0, \Sigma_0 = g\sigma^2[X^T X]^{-1})$$
$$\pi(\sigma^2) = \mathcal{IG}(\nu_0/2, \nu_0\sigma_0^2/2)$$

```
# Data summaries
n_swimmers <- ncol(Y)
n <- nrow(Y)
W <- seq(2,12,length.out=n)
X <- cbind(rep(1,n),(W-mean(W)))
p <- ncol(X)
```

```r
# Set hyperparameters
nu0 <- 1; sigma20 <- 0.1; g <- 6

PRED <- NULL
set.seed(2291)

for (i in 1:100) {
  # Draw sigma2
  sigma2 <- 1/rgamma(1, nu0/2, nu0 * sigma20/2)

  # Draw beta
  beta <- rmvnorm(1, c(0,0), g * sigma2 * solve(t(X) %*% X))

  # Generate data using beta, sigma2
  pred <- rmvnorm(1, X %*% t(beta), sigma2 * diag(n))

  PRED <- rbind(PRED, c(pred))
}

par(mfrow = c(2, 3))
for (i in 1:6) {
  hist(PRED[, i], main = paste0("Prior Prediction for Week ", i))
}
```



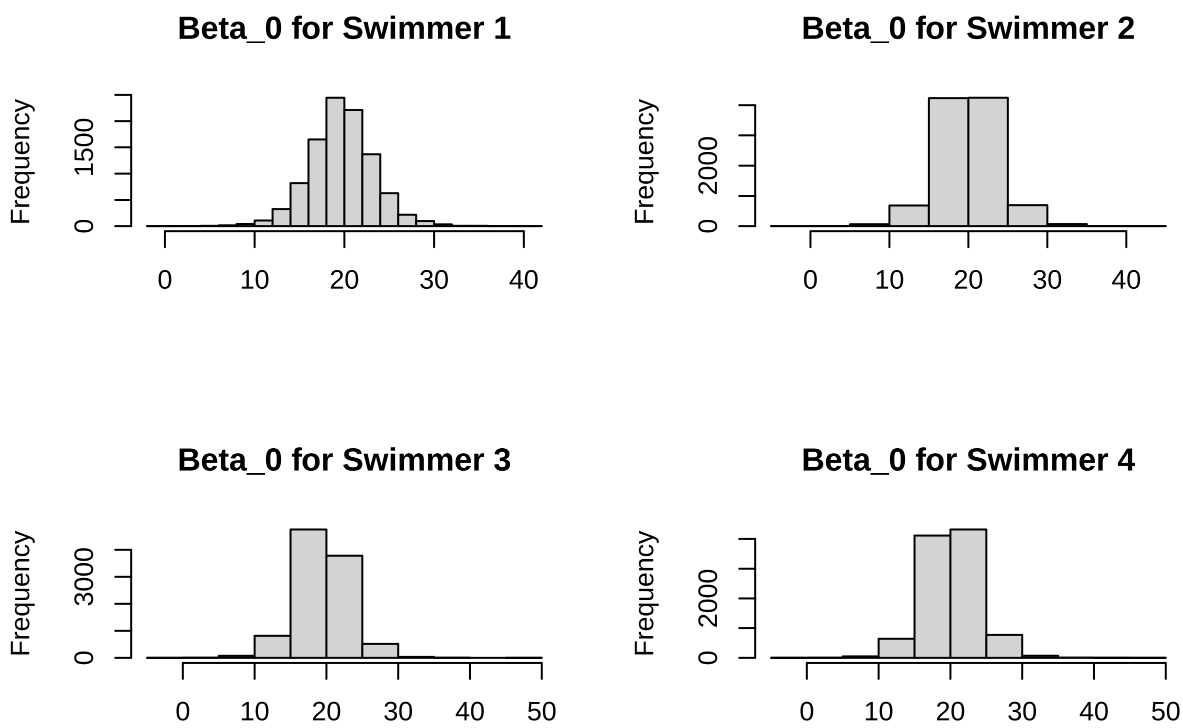
These values are not plausible because you cannot have a negative race time.

Part B

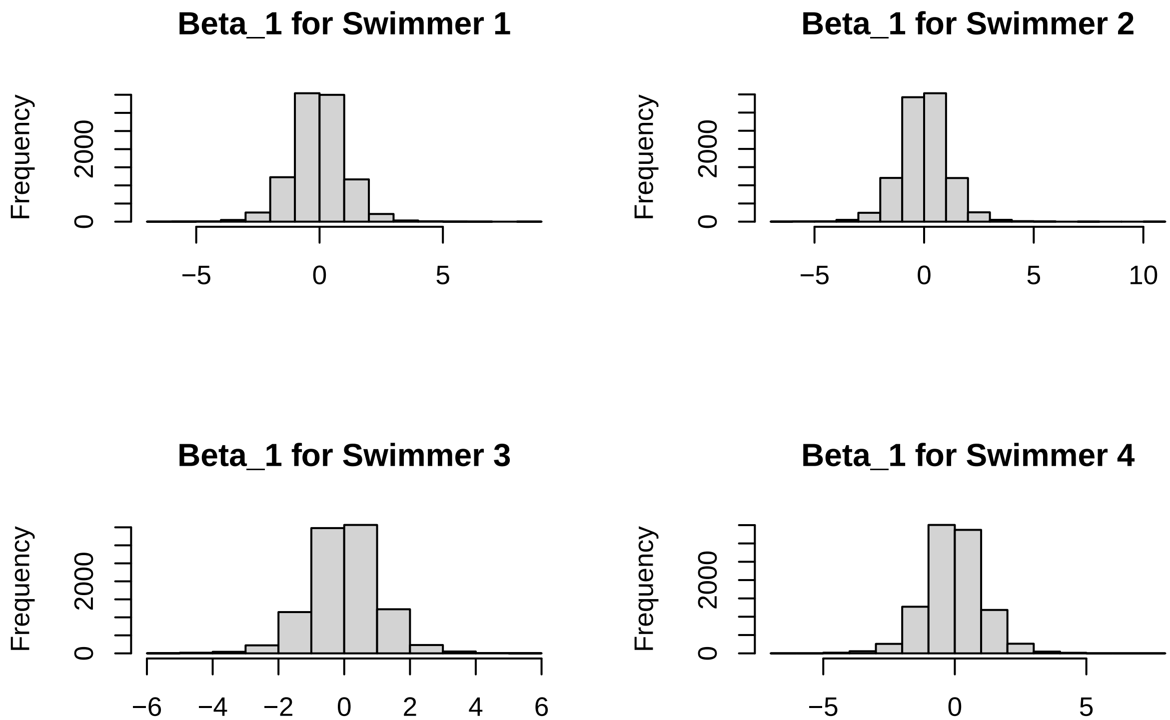
Using the data, and the g-prior with $g = n = 6$ for each swimmer, give the posterior distributions of β_0, β_1 and σ^2 for each swimmer.

```
# Data summaries
n_swimmers <- ncol(Y)
n <- nrow(Y)
W <- seq(2,12,length.out=n)
X <- cbind(rep(1,n),(W-mean(W)))
p <- ncol(X)

# Set hyperparameters
nu0 <- 1; sigma20 <- 0.1; g <- 6

#Initial values for Gibbs sampler
beta <- matrix(c(23,0),nrow=p,ncol=n_swimmers)
sigma_sq <- rep(1, n_swimmers)

n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

#Set null matrices to save samples
BETA <- array(0,c(n_swimmers,n_iter,p))
SIGMA_SQ <- matrix(0,n_swimmers,n_iter)

for(s in 1:(n_iter+burn_in)){
  for(j in 1:n_swimmers){
    #update the sigma_sq
    nu_n <- nu0 + n
    SSRg <- t(Y[,j]) %*%
      (diag(1, nrow = n) - (g/(g+1)) * X %*% solve(t(X)%*%X) %*% t(X)) %*%
      Y[,j]
    nu_n_sigma_n_sq <- nu0 * sigma20 + SSRg
    sigma_sq[j] <- 1/rgamma(1,(nu_n/2),(nu_n_sigma_n_sq/2))

    #update beta
    Sigma_n <- g/(g+1) * sigma_sq[j] * solve(t(X)%*%X)
    mu_n <- Sigma_n %*% ((t(X)%*%Y[,j])/sigma_sq[j])
    beta[,j] <- rmvnorm(1,mu_n,Sigma_n)

    #save results only past burn-in
    if(s > burn_in){
      BETA[j,(s-burn_in),] <- beta[,j]
      SIGMA_SQ[j,(s-burn_in)] <- sigma_sq[j]
    }
  }
}
```

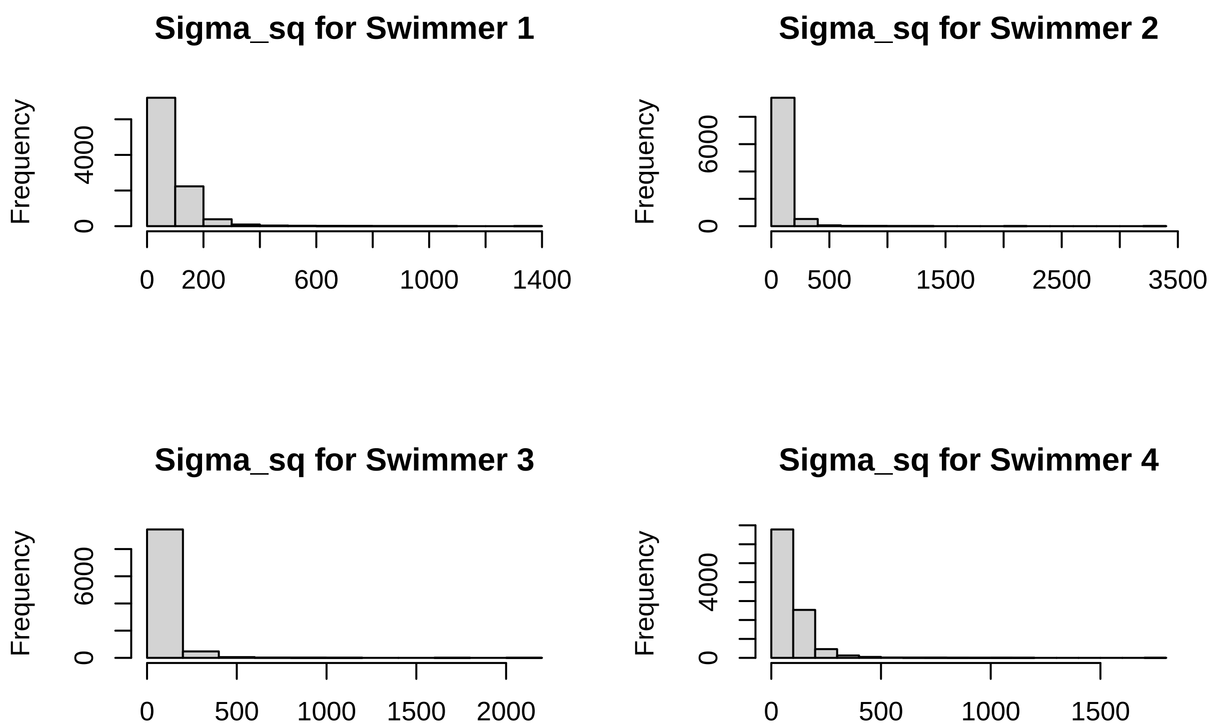
```
par(mfrow = c(2, 2))
for (i in 1:4) {
  hist(BETA[i,,1], main = paste0("Beta_0 for Swimmer ", i), xlab = NA)
}
```




```
for (i in 1:4) {
  hist(BETA[i,,2], main = paste0("Beta_1 for Swimmer ", i), xlab = NA)
}
```




```
for (i in 1:4) {
  hist(SIGMA_SQ[i,], main = paste0("Sigma_sq for Swimmer ", i), xlab = NA)
}
```




```r
beta_postmean <- t(apply(BETA,c(1,3),mean))
colnames(beta_postmean) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
rownames(beta_postmean) <- c("beta_0","beta_1")
beta_postmean
```

```
##         Swimmer 1   Swimmer 2   Swimmer 3   Swimmer 4
## beta_0 19.6900204 20.02837926 19.47313089 20.21197624
## beta_1 -0.0235418  0.01392135  0.01529384 -0.02412876
```

```r
beta_postCI <- apply(BETA,c(1,3),function(x) quantile(x,probs=c(0.025,0.975)))
colnames(beta_postCI) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_postCI[,,1]; beta_postCI[,,2]
```

```
##       Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## 2.5%   12.59796  12.63536  12.44540  12.91288
## 97.5%  26.90785  27.33635  26.51509  27.68794
```

```
##       Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## 2.5%  -2.137704 -2.138734 -2.082402 -2.206636
## 97.5%  2.010768  2.182270  2.095901  2.179629
```

```r
sigma_postmean <- t(apply(SIGMA_SQ,c(1),mean))
colnames(sigma_postmean) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
rownames(sigma_postmean) <- c("sigma_sq")
sigma_postmean
```

```
##          Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## sigma_sq  89.81342  93.83616  90.27378  96.80466
```

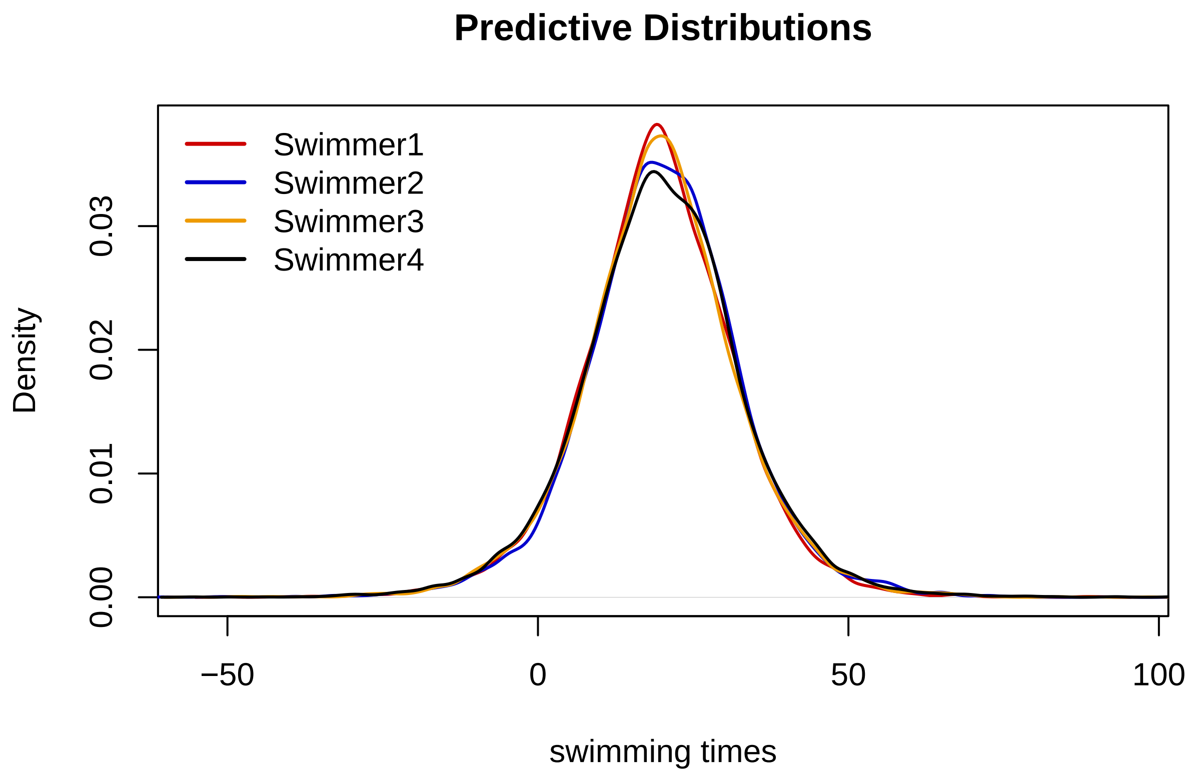
```
sigma_postCI <- apply(SIGMA_SQ,c(1),function(x) quantile(x,probs=c(0.025,0.975)))
colnames(sigma_postCI) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
sigma_postCI
```

```
##       Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## 2.5%   28.16082  29.10394  27.44416  29.62718
## 97.5% 265.45645 271.90046 266.23170 285.08293
```

Part C

For each swimmer j, plot their posterior predictive distributions for a future time $T^{\star}$ two weeks after the last recorded observation (overlay the 4 densities in a single plot).

```
x_new <- matrix(c(1,(14-mean(W))),ncol=1)
post_pred <- matrix(0,nrow=n_iter,ncol=n_swimmers)
for(j in 1:n_swimmers){
  post_pred[,j] <- rnorm(n_iter,BETA[j,,]%*%x_new,sqrt(SIGMA_SQ[j,]))
}
colnames(post_pred) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
plot(density(post_pred[,"Swimmer 1"]),col="red3",lwd=1.5,
     main="Predictive Distributions",xlab="swimming times")
legend("topleft",2,c("Swimmer1","Swimmer2","Swimmer3","Swimmer4"),
       col=c("red3","blue3","orange2","black"),lwd=2,bty="n")
lines(density(post_pred[,"Swimmer 2"]),col="blue3",lwd=1.5)
lines(density(post_pred[,"Swimmer 3"]),col="orange2",lwd=1.5)
lines(density(post_pred[,"Swimmer 4"]),lwd=1.5)
```



Part D

The coach of the team has to recommend which of the swimmers to compete in a swim meet in two weeks time. Using draws from the predictive distributions, compute $P(Y_j^\star = max(Y_1^\star, Y_2^\star, Y_3^\star, Y_4^\star))$ for each swimmer j, and based on this make a recommendation to the coach.

```
post_pred_min <- as.data.frame(apply(post_pred,1,function(x) which(x==min(x))))
colnames(post_pred_min) <- "Swimmers"
post_pred_min$Swimmers <- as.factor(post_pred_min$Swimmers)
levels(post_pred_min$Swimmers) <- c("Swimmer 1","Swimmer 2","Swimmer 3",
                                    "Swimmer 4")
table(post_pred_min$Swimmers)/n_iter
```

```
##
## Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
##    0.2601    0.2377    0.2511    0.2511
```

I would recommend Swimmer 1, as this swimmer has the highest probability of recording the lowest time.

Question 2: Hoff 9.2

Model selection: As described in Example 6 of Chapter 7, The file `azdiabetes.dat` contains data on health-related variables of a population of 532 women. In this exercise we will be modeling the conditional distribution of glucose level (`glu`) as a linear combination of the other variables, excluding the variable `diabetes`.

```
azdiabetes <- read.table(
  "http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/azdiabetes.dat", header = T)

Y <- azdiabetes[, 2]
predictors <- azdiabetes[, -c(2,8)]
```

Part A

Fit a regression model using the g-prior with $g = n, \nu_0 = 2$ and $\sigma_0^2 = 1$. Obtain posterior confidence intervals for all of the parameters.

```
###### g-Prior: with g=n using full model
# Data summaries
n <- length(Y)
X <- as.matrix(cbind(intercept = rep(1, n), predictors))
p <- ncol(X)
g <- n

# OLS estimates
beta_ols <- solve(t(X)%*%X)%*%t(X)%*%Y
SSR_beta_ols <- (t(Y - (X%*%beta_ols)))%*%(Y - (X%*%beta_ols))
sigma_ols <- SSR_beta_ols/(n-p)

# Hyperparameters for the priors
#sigma_0_sq <- sigma_ols
sigma_0_sq <- 1
nu_0 <- 2

# Set number of iterations
S <- 10000
set.seed(1234)

# Sample sigma_sq
nu_n <- nu_0 + n
Hg <- (g/(g+1))* X%*%solve(t(X)%*%X)%*%t(X)
SSRg <- t(Y)%*%(diag(1,nrow=n) - Hg)%*%Y
nu_n_sigma_n_sq <- nu_0*sigma_0_sq + SSRg
sigma_sq <- 1/rgamma(S,(nu_n/2),(nu_n_sigma_n_sq/2))

# Sample beta
mu_n <- g*beta_ols/(g+1)
beta <- matrix(nrow=S,ncol=p)
for(s in 1:S){
  Sigma_n <- g*sigma_sq[s]*solve(t(X)%*%X)/(g+1)
  beta[s,] <- rmvnorm(1,mu_n,Sigma_n)
}
# posterior summaries
colnames(beta) <- colnames(X)
```

```r
mean_beta <- apply(beta,2,mean)
round(mean_beta,4)
```

```
## intercept     npreg        bp      skin       bmi       ped       age
##   52.1295   -0.6577    0.2075    0.1941    0.6370   10.5636    0.7664
```

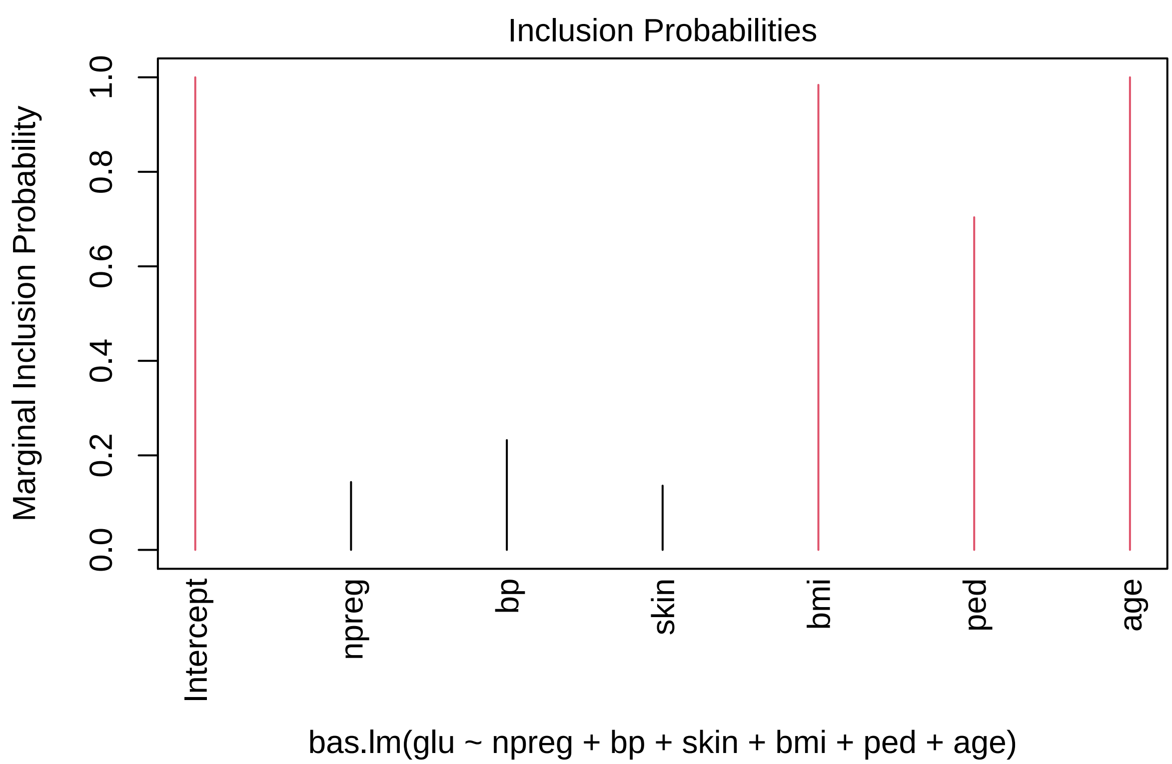
```r
# Confidence intervals
CI_beta <- apply(beta,2,function(x) quantile(x,probs=c(0.025,0.975)))
round(CI_beta, 4)
```

```
##       intercept   npreg      bp    skin    bmi     ped    age
## 2.5%    34.9064 -1.6455 -0.0172 -0.1142 0.1516  3.3094 0.4595
## 97.5%   69.4243  0.3111  0.4332  0.5076 1.1352 17.5851 1.0830
```

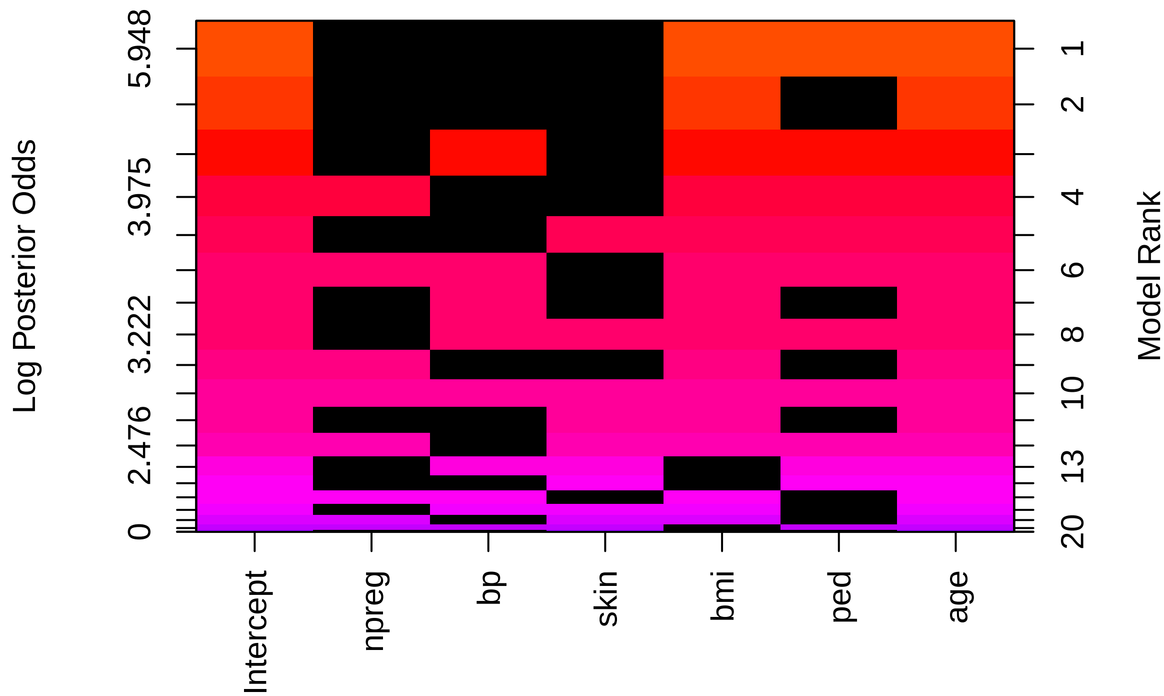
Part B

Perform the model selection and averaging procedure described in Section 9.3. Obtain $Pr(\beta_j \neq 0|y)$, as well as posterior confidence intervals for all of the parameters. Compare to the results in part a).

```r
######## Bayesian Model Selection and Averaging
#library(BAS)
Data_bas <- bas.lm(glu~npreg+bp+skin+bmi+ped+age, data=azdiabetes,
                   prior="g-prior", alpha=n,
                   n.models=2^p, initprobs="Uniform")
plot(Data_bas,which=4)
```



```r
image(Data_bas)
```




```r
summary(Data_bas)
```

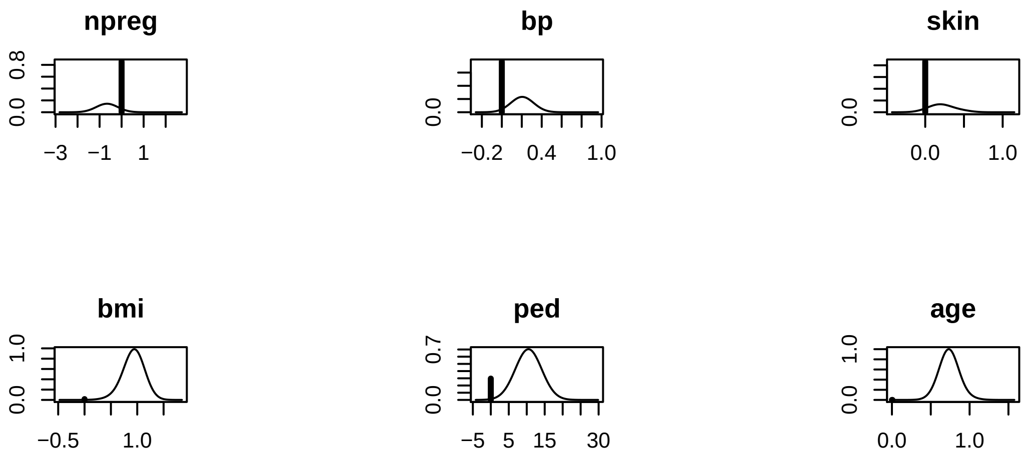
```
##           P(B != 0 | Y)  model 1    model 2    model 3    model 4     model 5
## Intercept     1.0000000  1.00000  1.0000000  1.0000000  1.0000000  1.00000000
## npreg         0.1436174  0.00000  0.0000000  0.0000000  1.0000000  0.00000000
## bp            0.2323307  0.00000  0.0000000  1.0000000  0.0000000  0.00000000
## skin          0.1359460  0.00000  0.0000000  0.0000000  0.0000000  1.00000000
## bmi           0.9839963  1.00000  1.0000000  1.0000000  1.0000000  1.00000000
## ped           0.7038751  1.00000  0.0000000  1.0000000  1.0000000  1.00000000
## age           0.9999919  1.00000  1.0000000  1.0000000  1.0000000  1.00000000
## BF                   NA  1.00000  0.4255451  0.2233412  0.1042567  0.08504614
## PostProbs            NA  0.39100  0.2218000  0.1164000  0.0543000  0.04430000
## R2                   NA  0.14240  0.1294000  0.1477000  0.1453000  0.14460000
## dim                  NA  4.00000  3.0000000  5.0000000  5.0000000  5.00000000
## logmarg              NA 31.29705 30.4426661 29.7979959 29.0361510 28.83248913
```

```r
model_coef <- coef(Data_bas)
confint(model_coef)
```

```
##                     2.5%        97.5%          beta
## Intercept   1.185165e+02 1.234538e+02 121.03007519
## npreg      -9.866018e-01 5.271129e-04  -0.09522102
```

```
## bp        -1.836332e-04 3.106377e-01   0.04757862
## skin      -5.994551e-03 3.197523e-01   0.02997921
## bmi        4.488014e-01 1.377093e+00   0.90861759
## ped        0.000000e+00 1.593012e+01   7.38061463
## age        4.606268e-01 1.002882e+00   0.73746007
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

```r
par(mfrow=c(3,3))
plot(coef(Data_bas), subset=2:7)
```



The intercept is quite different, likely due to differences in mean-centering. The coefficients on `npreg`, `bp`, `skin`, and `ped` are smaller in magnitude in the model averaging procedure, while that on `age` is about the same and that on `bmi` is larger.

Question 3: Metropolis Hastings

Consider the following sampling model:

$$y_1, \ldots, y_n | \theta_1, \theta_2 \sim p(y|\theta_1, \theta_2),$$

with the priors on θ_1 and θ_2 set as $\pi_1(\theta_1)$ and $\pi_2(\theta_2)$ respectively, where $\theta_1, \theta_2 \in R$.

Suppose we are interested in generating random samples from the posterior distribution $\pi_(\theta_1, \theta_2|y_1, \ldots, y_n)$. For each of the following proposal distributions, write down the acceptance ratio for using Metropolis-Hastings to generate the samples we desire. Make sure to simplify the ratios as much as possible for each proposal! Also, comment on whether or not the proposals are intuitive.

In each case, you only need to spend time working through the acceptance ratio for one of the two parameters. The other one should become obvious once you've completed the first.

For all parts,

$$\begin{aligned} r &= \frac{\pi(\theta_1^\star, \theta_2^{(s)}|y)}{\pi(\theta_1^{(s)}, \theta_2^{(s)}|y)} * \frac{g_{\theta_1}[\theta_1^{(s)}|\theta_1^\star, \theta_2^{(s)}]}{g_{\theta_1}[\theta_1^\star|\theta_1^{(s)}, \theta_2^{(s)}]} \\ &= \frac{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star) * \pi(\theta_2^{(s)})}{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)}) * \pi(\theta_2^{(s)})} * \frac{g_{\theta_1}[\theta_1^{(s)}|\theta_1^\star, \theta_2^{(s)}]}{g_{\theta_1}[\theta_1^\star|\theta_1^{(s)}, \theta_2^{(s)}]} \end{aligned}$$

Part A

Full conditionals

$$\begin{aligned} g_{\theta_1}[\theta_1^\star|\theta_1^{(s)}, \theta_2^{(s)}] &= p(\theta_1^\star|y_1, \ldots, y_n, \theta_2^{(s)}); \\ g_{\theta_2}[\theta_2^\star|\theta_1^{(s)}, \theta_2^{(s)}] &= p(\theta_2^\star|y_1, \ldots, y_n, \theta_1^{(s)}). \end{aligned}$$

$$\begin{aligned} p(\theta_1^\star|y_1, \ldots, y_n, \theta_2^{(s)}) &\propto p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star) \\ r_{\theta_1} &= \frac{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star) * \pi(\theta_2^{(s)})}{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)}) * \pi(\theta_2^{(s)})} * \frac{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)})}{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star)} \\ &= \frac{\pi(\theta_2^{(s)})}{\pi(\theta_2^{(s)})} \\ &= 1 \\ r_{\theta_2} &= 1 \end{aligned}$$

Note that the normalizing constants for the full conditionals of $\theta_1^\star$ and $theta_1^{(s)}$ are equal and will thus cancel.

Part B

Priors

$$\begin{aligned} g_{\theta_1}[\theta_1^\star|\theta_1^{(s)}, \theta_2^{(s)}] &= \pi_1(\theta_1^\star); \\ g_{\theta_2}[\theta_2^\star|\theta_1^{(s)}, \theta_2^{(s)}] &= \pi_2(\theta_2^\star). \end{aligned}$$

$$
\begin{aligned}
r_{\theta_1} &= \frac{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star) * \pi(\theta_2^{(s)})}{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)}) * \pi(\theta_2^{(s)})} * \frac{\pi(\theta_1^s)}{\pi_1(\theta_1^\star)} \\
&= \frac{p(y|\theta_1^\star, \theta_2^{(s)})}{p(y|\theta_1^{(s)}, \theta_2^{(s)})} \\
r_{\theta_2} &= \frac{p(y|\theta_2^\star, \theta_1^{(s)})}{p(y|\theta_2^{(s)}, \theta_1^{(s)})}
\end{aligned}
$$

Part C

Random Walk

$$
\begin{aligned}
g_{\theta_1}[\theta_1^\star|\theta_1^{(s)}, \theta_2^{(s)}] = \mathcal{N}(\theta_1^{(s)}, \delta^2); \\
g_{\theta_2}[\theta_2^\star|\theta_1^{(s)}, \theta_2^{(s)}] = \mathcal{N}(\theta_2^{(s)}, \delta^2).
\end{aligned}
$$

$$
\begin{aligned}
r_{\theta_1} &= \frac{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star) * \pi(\theta_2^{(s)})}{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)}) * \pi(\theta_2^{(s)})} * \frac{g_{\theta_1}[\theta_1^{(s)}|\theta_1^\star, \theta_2^{(s)}]}{g_{\theta_1}[\theta_1^\star|\theta_1^{(s)}, \theta_2^{(s)}]} \\
&= \frac{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star) * \pi(\theta_2^{(s)})}{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)}) * \pi(\theta_2^{(s)})} * \frac{\exp -1/(2\delta^2) * (\theta_1^\star - \theta_1^{(s)})^2}{\exp -1/(2\delta^2) * (\theta_1^{(s)} - \theta_1^\star)^2} \\
&= \frac{p(y|\theta_1^\star, \theta_2^{(s)}) * \pi(\theta_1^\star)}{p(y|\theta_1^{(s)}, \theta_2^{(s)}) * \pi(\theta_1^{(s)})}
\end{aligned}
$$

$$
r_{\theta_2} = \frac{p(y|\theta_2^\star, \theta_1^{(s)}) * \pi(\theta_2^\star)}{p(y|\theta_2^{(s)}, \theta_1^{(s)}) * \pi(\theta_2^{(s)})}
$$