ST540 HW9

1. Let $\{X_i\}_{i=1}^{n_1}$, $\{Y_i\}_{i=1}^{n_2}$ denote the responses using placebo and treatment respectively. Under the assumption of equal variance, the likelihood is:

$$\begin{aligned} X_i &\sim \mathrm{N}(\mu, \sigma^2) \\ Y_i &\sim \mathrm{N}(\mu + \delta, \sigma^2) \end{aligned} \tag{1}$$

The Jeffrey's prior on (μ, δ, σ^2) is:

$$\pi(\mu, \delta, \sigma^2) \sim (\frac{1}{\sigma^2})^2 \tag{2}$$

It can be shown(see details in the text book) that the marginal posterior of δ is:

$$\delta | \boldsymbol{X}, \boldsymbol{Y} \sim t_{n_1+n_2}(\bar{Y} - \bar{X}, \widehat{\sigma^2}(\frac{1}{n_1} + \frac{1}{n_2})) \tag{3}$$

where

$$\widehat{\sigma^2} = \frac{1}{n_1 + n_2}(n_1 s_X^2 + n_2 s_Y^2) \tag{4}$$

In this question, we could conduct a Bayesian two-sample t-test:

$$H_0 : \delta = 0 \quad H_a : \delta \neq 0 \tag{5}$$

Using the given data, the marginal posterior of δ is (code attached):

$$\delta | \boldsymbol{X}, \boldsymbol{Y} \sim t_{12}(-2.4, 1.01^2) \tag{6}$$

And the 95% credible interval is:

$$[-2.4 + 1.01 t_{12,0.025}, -2.4 + 1.01 t_{12,0.975}] = [-4.60, -0.19] \tag{7}$$

Since the credible interval doesn't contain 0, we should reject the null hypothesis and conclude that the treatment is effective.

To test if our model is sensitive to the priors, we place proper priors on (μ, δ, σ^2):

$$\begin{aligned} \mu &\sim \mathrm{N}(0, 100^2) \\ \delta &\sim \mathrm{N}(0, 100^2) \\ \sigma^2 &\sim \mathrm{InvGamma}(0.01, 0.01) \end{aligned} \tag{8}$$

The results from JAGS shows:

	2.5%	25%	50%	75%	97.5%
delta	−4.84594	−3.15759	−2.38849	−1.61692	0.08267

This time the credible interval of δ contains 0. So our conclusion is sensitive to our choice of priors.

Note that we can also have a two-sample model with unequal variance: $X_i \sim \mathrm{N}(\mu_X, \sigma_X^2)$, $Y_i \sim \mathrm{N}(\mu_Y, \sigma_Y^2)$. The corresponding posteriors of μ_X and μ_Y are $\mu_X | \boldsymbol{X} \sim t_{n_1}(\bar{X}, s_X^2/n_1)$ and $\mu_Y | \boldsymbol{Y} \sim t_{n_2}(\bar{Y}, s_Y^2/n_2)$.

2. (a) The Bayesian linear model with uninformative Gaussian prior is:

$$\begin{aligned} Y_i &\sim \mathrm{N}(\alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i, \sigma^2) \\ \alpha &\sim \mathrm{N}(0, 100^2) \\ \beta_k &\sim \mathrm{N}(0, 100^2) \quad k = 1, ..., p \\ \sigma^2 &\sim \mathrm{InvGamma}(0.01, 0.01) \end{aligned} \tag{9}$$

where $\{(Y_i, \boldsymbol{x}_i)\}_{i=1}^N$ are the data, α is the intercept and $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)$ are the coefficients.
It's recommended to standardize the covariates except for the categorical ones, which often leads to better model performance. JAGS was used to run 4 chains with each consisting of 150000 iterations. The first 100000 iterations of each chain were discarded as burn-in and thus we retained a total of 200000 posterior samples for each parameter without any thinning.

	2.5%	25%	50%	75%	97.5%
chas	0.9912	2.1049	2.68721	3.2683	4.3691
crim	−1.4847	−1.1209	−0.92951	−0.7391	−0.3721
zn	0.4545	0.8686	1.08436	1.3000	1.7153
indus	−0.6871	−0.1427	0.14235	0.4265	0.9732
nox	−2.9274	−2.3578	−2.05873	−1.7588	−1.1871
rm	2.0966	2.4776	2.67643	2.8732	3.2543
age	−0.7115	−0.2306	0.02018	0.2706	0.7466
dis	−3.9306	−3.3948	−3.10923	−2.8251	−2.2779
rad	1.5324	2.2779	2.66563	3.0596	3.8137
tax	−3.3307	−2.5111	−2.08087	−1.6552	−0.8429
ptratio	−2.6198	−2.2544	−2.06350	−1.8721	−1.5047
black	0.3672	0.6839	0.84987	1.0158	1.3346
lstat	−4.4603	−3.9919	−3.74764	−3.5055	−3.0353

We can see that the 95% Bayesian credible interval for the coefficients of "indus" and "age" both contain 0, which means that they are not statistically significant in this model.

Effective sample size and GR statistics were computed for convergence diagnostic. The coefficient for "tax" has the smallest ESS of 13500, and GR statistics for all coefficients are below 1.0001. So, we can be conclude that the chains have converged.

(b) For frequentist linear regression without regularization(e.g. L1/L2 penalty), it's not necessary to standardize the covariates. The results of lm() function are shown below:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

The adjusted R^2 for this model is 0.7338, which indicates that the model fits the data quite well. We also note that the P-vale for "indus" and "age" is quite large, which is in agreement with the results from (a).

(c) We have the following model for Bayesian linear regression with double exponential priors:

$$\begin{aligned} Y_i &\sim \mathrm{N}(\alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i, \sigma^2) \\ \alpha &\sim \mathrm{N}(0, 10^2) \\ \beta_k &\sim \mathrm{dexp}(0, b) \quad k = 1, ..., p \\ \sigma^2 &\sim \mathrm{InvGamma}(0.01, 0.01) \\ b &\sim \mathrm{InvGamma}(0.01, 0.01) \end{aligned} \tag{10}$$

where the pdf of double exponential distribution is $f(y|\mu, b) = \frac{1}{2b}\exp(-\frac{|y-\mu|}{b})$

	2.5%	25%	50%	75%	97.5%
chas	0.6886	1.7998	2.386143	2.9728	4.0718
crim	−1.4141	−1.0511	−0.859417	−0.6692	−0.3030
zn	0.3449	0.7596	0.975955	1.1906	1.6070
indus	−0.7551	−0.2382	0.009990	0.2638	0.7861
nox	−2.7483	−2.1850	−1.889645	−1.5939	−1.0279
rm	2.1363	2.5164	2.715007	2.9134	3.2928
age	−0.6879	−0.2315	−0.006154	0.2193	0.6779
dis	−3.7865	−3.2473	−2.965126	−2.6832	−2.1390
rad	1.1211	1.8690	2.261682	2.6502	3.3909
tax	−2.9340	−2.1362	−1.716032	−1.2902	−0.4804
ptratio	−2.5807	−2.2164	−2.025399	−1.8341	−1.4707
black	0.3513	0.6660	0.831696	0.9971	1.3120
lstat	−4.4476	−3.9802	−3.737038	−3.4937	−3.0230

Compared to the results obtained in (a), we find that the posterior quantiles of most of the covariates have shrunken towards zero to different degrees.

We also find that the posterior medians of the coefficients for "indus" and "age" are extremely close to 0, while they are 0.14 and 0.02 in (a). This shows that Bayesian Lasso regression encourages sparsity and can be used for feature selection to filter out the "unimportant" features.

(d) A Bayesian ridge regression model was trained on the first 500 observation:

$$\begin{aligned} Y_i &\sim \mathrm{N}(\alpha + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i, \sigma^2) \\ \alpha &\sim \mathrm{N}(0, 10^2) \\ \beta_k &\sim \mathrm{N}(0, \frac{\sigma^2}{\lambda}) \quad k = 1, ..., p \\ \sigma^2 &\sim \mathrm{InvGamma}(0.01, 0.01) \\ \lambda &\sim \mathrm{Gamma}(0.01, 0.01) \end{aligned} \tag{11}$$

We denote the posterior samples as $\{(\alpha_j, \boldsymbol{\beta}_j, \sigma_j^2)\}_{j=1}^S$, where S is the number of samples. For a test observation $(Y^t, \boldsymbol{x}^t)$, the posterior predictions of $\boldsymbol{x}^t$ is $\{Y_j^t\}_{j=1}^S$, where $Y_j^t \sim \mathrm{N}(\alpha_j + \boldsymbol{\beta}_j^{\mathrm{T}}\boldsymbol{x}_t, \sigma_j^2)$.

To measure the performance of the model, we plot a histogram using $\{Y_j^t\}_{j=1}^S$ and compare it with the real response Y^t. As shown in Figure 1, our model performs pretty well on the test observations except for the last one.

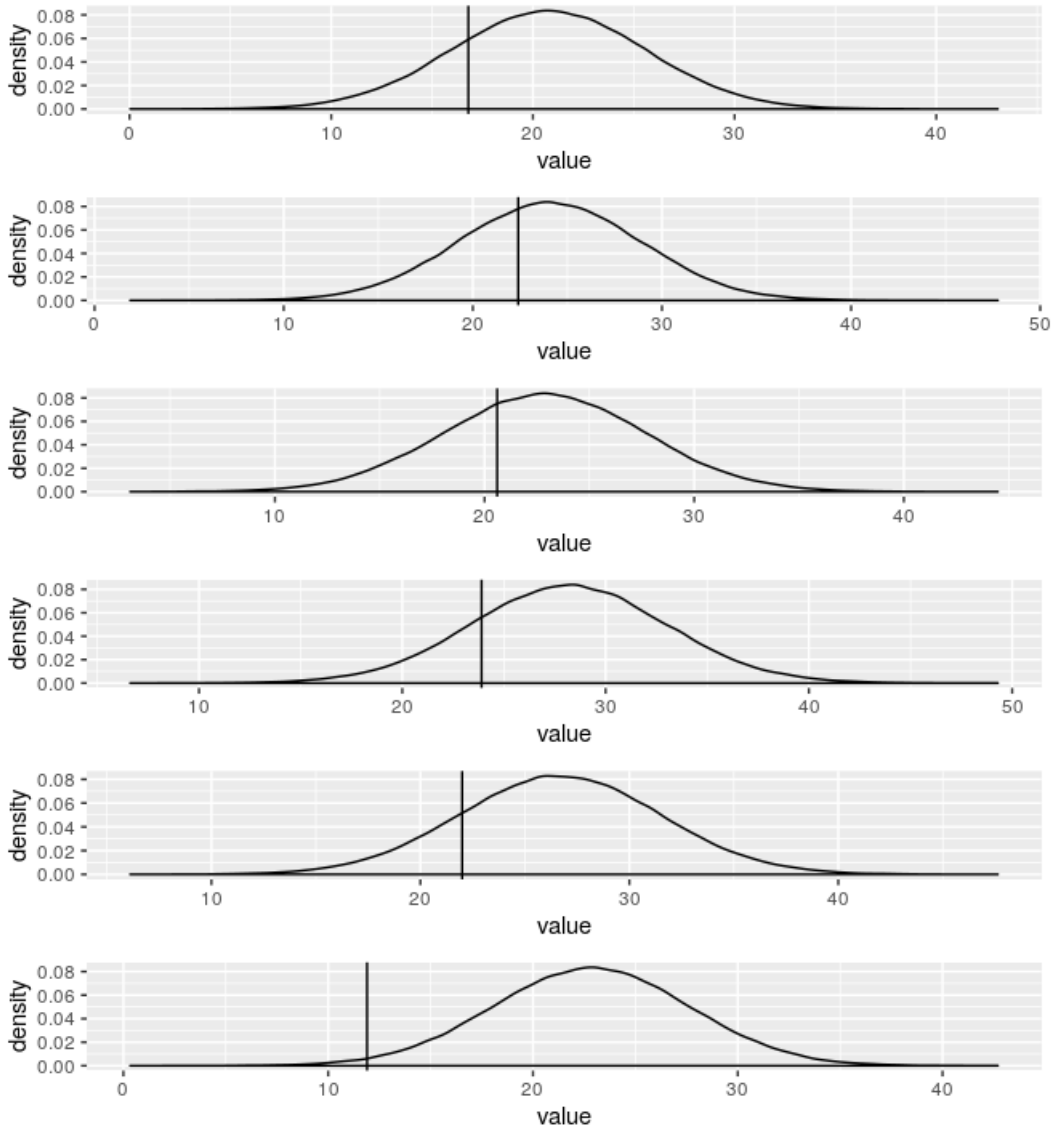

Figure 1: posterior prediction distribution of the last 6 observations

```
#################################################
#              Ch4.1
#################################################
#placebo
X = c(2, -3.1, -1.0, 0.2, 0.3, 0.4)
X_bar = mean(X)
s_X = mean((X-X_bar)^2)
n1 = length(X)

#treatment
Y = c(-3.5, -1.6, -4.6, -0.9, -5.1, 0.1)
Y_bar = mean(Y)
s_Y = mean((Y-Y_bar)^2)
n2 = length(Y)

#posterior of delta
df = n1 + n2
#location parameter
delta_mu = Y_bar - X_bar
#scale parameter
s2 = (n1 * s_X + n2 * s_Y)/(n1 + n2)
```

```
delta_scale = sqrt(s2 * (1/n1 + 1/n2))

#95% credible interval
delta_cred_int = delta_mu + delta_scale * qt(c(0.025, 0.975), df=df)



# sensitivity
model_string_1 <- textConnection("model{
                                 # Likelihood
                                 for(i in 1:n){
                                   X[i] ~ dnorm(mu, inv.var)
                                   Y[i] ~ dnorm(mu + delta, inv.var)
                                 }
                                 # Priors
                                 mu ~ dnorm(0,0.0001)
                                 delta ~ dnorm(0, 0.0001)
                                 inv.var ~ dgamma(0.01,0.01)
                                 }")

model_1 = jags.model(model_string_1, data=list(X=X,Y=Y, n=n1), n.chains=4)
update(model_1, 50000, progress.bar="none")
samples_1 = coda.samples(model_1, variable.names=c('delta'), n.iter=20000)

plot(samples_1)
summary(samples_1)


##############################################
#          Ch4.2
##############################################
#(a)
library(MASS)
library(coda)
library(rjags)

# scale the covariates except for the categorical one "chas"
X  = scale(Boston[,-c(4, 14)])
X = cbind(chas=Boston$chas, X)
covnames = c("chas","crim","zn","indus","nox","rm","age","dis",
             "rad","tax","ptratio","black","lstat")
Y = Boston[,14]
n = length(Y)
p = ncol(X)

# Define model_string
model_string_2a <- textConnection("model{
                                  # Likelihood
                                  for(i in 1:n){
                                  Y[i] ~ dnorm(mu[i], inv.var)
                                  mu[i] <- alpha + inprod(X[i,],beta[])
                                  }
                                  # Priors
                                  for(j in 1:p){
                                  beta[j] ~ dnorm(0,0.0001)
                                  }
                                  alpha ~ dnorm(0,0.0001)
                                  inv.var ~ dgamma(0.01, 0.01)
                                  }")

params = c("beta")
```

```
n_burnin = 100000
n_iter = 50000

model_2a = jags.model(model_string_2a, data=list(Y=Y,X=X,n=n,p=p), n.chains=4, n.adapt=500
update(model_2a, n_burnin)
samples_2a =  coda.samples(model_2a, variable.names=params, n.iter=n_iter)

#Display the posterior summaries of all regression coefficients
summary_2a = summary(samples_2a)
rownames(summary_2a$statistics) = covnames
rownames(summary_2a$quantiles) = covnames
# Diagnostics
ess_2a = effectiveSize(samples_2a)
names(ess_2a) = covnames
gr_2a = gelman.diag(samples_2a)
rownames(gr_2a$psrf) = covnames

#(b)
Y_2b = Boston[,14]
X_2b = as.matrix(Boston[,1:13])
model_2b = lm(Y_2b ~ X_2b)
summary(model_2b)

#(c)
model_string_2c <- textConnection("model{
                            # Likelihood
                            for(i in 1:n){
                              Y[i] ~ dnorm(mu[i], inv.var)
                              mu[i] <- alpha + inprod(X[i,], beta[])
                            }
                            # Priors
                            for(j in 1:p){
                              beta[j] ~ ddexp(0, inv.var.b)
                            }
                            alpha ~ dnorm(0, 0.01)
                            inv.var ~ dgamma(0.01, 0.01)
                            inv.var.b ~ dgamma(0.01, 0.01)
                            }")


model_2c = jags.model(model_string_2c, data=list(Y=Y, X=X, n=n, p=p),
                          n.chains=4, n.adapt=5000)
update(model_2c, n_burnin)
samples_2c =  coda.samples(model_2c, variable.names=params, n.iter=n_iter)


#Display the posterior summaries of all regression coefficients
summary_2c = summary(samples_2c)
rownames(summary_2c$statistics) = covnames
rownames(summary_2c$quantiles) = covnames
# Diagnostics
ess_2c = effectiveSize(samples_2c)
names(ess_2c) = covnames
gr_2c = gelman.diag(samples_2c)
rownames(gr_2c$psrf) = covnames

#(d)
X_train = X[1:500,]
Y_train = Y[1:500]

X_test = X[501:506,]
```

```
Y_test = Y[501:506]

model_string_2d <- textConnection("model{
                            # Likelihood
                            for(i in 1:n){
                              Y[i] ~ dnorm(mu[i], inv.var)
                              mu[i] <- alpha + inprod(X[i,], beta[])
                            }
                            # Priors
                            for(j in 1:p){
                              beta[j] ~ dnorm(0, inv.var * lambda)
                            }
                            alpha ~ dnorm(0, 0.01)
                            inv.var ~ dgamma(0.01, 0.01)
                            lambda ~ dgamma(0.01, 0.01)
                            }")


params = c("alpha","beta","inv.var")
model_2d = jags.model(model_string_2d, data=list(Y=Y_train, X=X_train, n=500, p=p),
                      n.chains=4, n.adapt=5000)
update(model_2d, n_burnin)
samples_2d =  coda.samples(model_2d, variable.names=params, n.iter=n_iter)

samples_2d_m = as.matrix(samples_2d)
pos_alpha_beta = samples_2d_m[,1:14]
pos_sd = sqrt(1. / samples_2d_m[,15])

X_test = cbind(rep(1,6), X_test) # add intercept term
post_mean = pos_alpha_beta %*% t(X_test)

pos_pred = matrix(nrow=200000, ncol=6)
for(i in 1:200000){
  for (j in 1:6){
    pos_pred[i,j] = rnorm(1, mean=post_mean[i,j], sd=pos_sd[i])
  }
}

# plot histograms
library(ggplot2)
library(reshape2)
pos_pred = melt(as.data.frame(pos_pred))

den_plots = list()
for(i in 1:6){
  plot_data = subset(pos_pred, variable == paste("V", i, sep = ''))
  den_plot = ggplot(plot_data, aes(x=value)) + geom_density() + geom_vline(xintercept=Y_te
  den_plots[[i]] = den_plot
}
```