MA 578 HW6 Solutions

We will be using the following packages and definitions:

```
library(rstan)
library(bayesplot)
library(beanplot)
source("bslm.R")

iter <- 2000
warmup <- floor(iter / 2)
nsamples <- iter - warmup
nchains <- 4
```

1 (BDA 14.1)

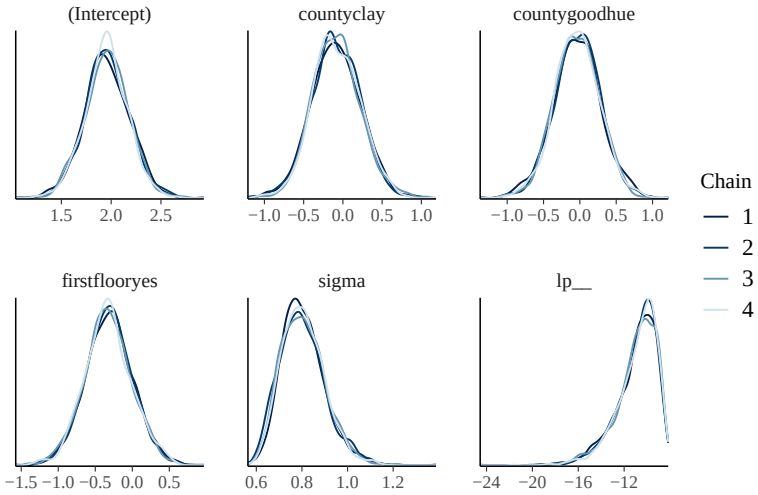
(a)

Assuming noninformative priors:

```
radon <- read.csv("data/radon.csv", comment = "#")
y <- log(radon$radon)
sims <- bslm_sample(y, model.matrix(~ county + firstfloor, data = radon),
                    chains = nchains, iter = iter, warmup = warmup)
monitor(sims)
```

```
Inference for the input samples (4 chains: each with iter = 1000; warmup = 500):

                 Q5   Q50  Q95  Mean  SD  Rhat Bulk_ESS Tail_ESS
(Intercept)     1.6   2.0  2.3   2.0 0.2     1     2137     1948
countyclay     -0.6  -0.1  0.4  -0.1 0.3     1     2057     2016
countygoodhue  -0.6   0.0  0.5   0.0 0.3     1     2118     1938
firstflooryes  -0.8  -0.3  0.2  -0.3 0.3     1     1895     1957
sigma           0.7   0.8  1.0   0.8 0.1     1     1705     1636
lp__          -14.1 -10.5 -8.8 -10.9 1.7     1     1639     1678

For each parameter, Bulk_ESS and Tail_ESS are crude measures of
effective sample size for bulk and tail quantities respectively (an ESS > 100
per chain is considered good), and Rhat is the potential scale reduction
factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

```
mcmc_dens_overlay(sims)
```

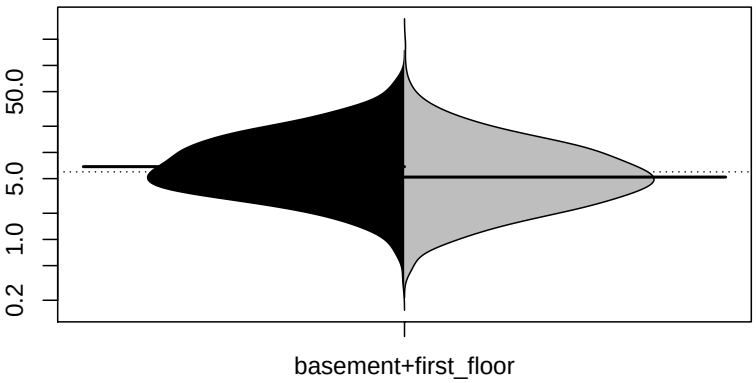


The contrast coefficients for the two other counties, Clay and Good Hue, have high posterior probability of being close to zero, so there doesn't seem to be a significant difference in radon levels across counties. Measurements taken in the first floor are slightly lower, about $e^{-0.3} \approx 0.74$ times lower than basement measurements. On average, basement radon levels are $e^2 \approx 7.4$ picoCuries/L.

(b)

As expected, the two posterior predictive distributions are very similar:

```
ypred_ffno <- exp(rnorm(nsamples, sims[, 1, 1], sims[, 1, 5]))
ypred_ffyes <- exp(rnorm(nsamples, sims[, 1, 1] + sims[, 1, 4], sims[, 1, 5]))
beanplot(ypred ~ firstfloor, side = "both",
         data = data.frame(ypred = c(ypred_ffno, ypred_ffyes),
                           firstfloor = c(rep("basement", nsamples),
                                          rep("first_floor", nsamples))),
         col = list("black", "gray"), what = c(1, 1, 1, 0), log = "y")
```




```
quantile(ypred_ffno, c(.025, .975))
```

```
     2.5%     97.5%
 1.396387 31.996946
```

```
quantile(ypred_ffyes, c(.025, .975))
```

```
      2.5%      97.5%
 0.9548429 28.1956678
```

2 (BDA 14.11–14.13)

(11.a)

The model states the conditional likelihood

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \,\Big|\, u_i, a, b, \Sigma \overset{\text{ind}}{\sim} N\left(\begin{bmatrix} 0 \\ a \end{bmatrix} + \begin{bmatrix} 1 \\ b \end{bmatrix} u_i, \Sigma \right), \quad i = 1, \dots, n.$$

But marginalizing out $u_i \overset{\text{iid}}{\sim} N(\mu, \tau^2)$ we have

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \,\Big|\, a, b, \Sigma \overset{\text{ind}}{\sim} N\left(\begin{bmatrix} 0 \\ a \end{bmatrix} + \begin{bmatrix} 1 \\ b \end{bmatrix} \mu, \Sigma + \tau^2 \begin{bmatrix} 1 \\ b \end{bmatrix} \begin{bmatrix} 1 \\ b \end{bmatrix}^\top \right).$$

(11.b)

Since a and b are ultimately coefficients in a linear regression, it should be fine to set a flat prior on them. The main concern lies in the slope coefficient b since it is also involved in the variance, but Σ should be able to control the posterior and keep it proper.

(12)

From now on, for more generality, let us assume that $\mathbf{x}_i$ has p covariates, and so $\mathbb{E}[y_i \,|\, a, \mathbf{b}, \mathbf{u}_i] = a + \mathbf{b}^\top \mathbf{u}_i$, and that $\Sigma = \sigma^2(I_p \oplus K) = \sigma^2 \text{Diag}\{1, \dots, 1, K\}$ is a diagonal matrix. Moreover, let us set non-informative priors on each $\mathbf{u}_i$, also of dimension p, $\mathbb{P}(\mathbf{u}) \propto 1$, and σ^2, $\mathbb{P}(\sigma) \propto 1/\sigma$.

We can use Stan to sample from this model,

```
data {
  int<lower=0> n; // #observations
  int<lower=0> p; // #predictors
  real<lower=0> K; // variance scale
  matrix[n, p] X;
  vector[n] y;
}
parameters {
  real a;
  vector[p] b;
  matrix[n, p] U;
  real<lower=0> sigma;
}
model {
  for (i in 1:n)
    for (j in 1:p)
      X[i, j] ~ normal(U[i, j], sigma);
  y ~ normal(a + U * b, sqrt(K) * sigma);
  target += -log(sigma);
}
```

or just resort to Gibbs sampling. In this case, conditional on $\mathbf{u}$ we can sample a and $\mathbf{b}$ from a usual linear regression of y on $\mathbf{u}$ since $y_i \,|\, \mathbf{u}_i, \sigma^2 \overset{\text{ind}}{\sim} N(a + \mathbf{b}^\top \mathbf{u}_i, \sigma^2 K)$. The conditional posterior on σ^2 needs to account for the variance in $\mathbf{x}$,

$$\mathbb{P}(\sigma^2 \,|\, a, \mathbf{b}, \mathbf{u}, y, \mathbf{x}) \propto \frac{1}{\sigma^2} \prod_{i=1}^{n} (\sigma^2)^{-p/2} \exp\left\{ -\frac{1}{2\sigma^2} (\mathbf{x}_i - \mathbf{u}_i)^\top (\mathbf{x}_i - \mathbf{u}_i) \right\} (\sigma^2)^{-1/2} \exp\left\{ -\frac{1}{2K\sigma^2} (y_i - a - \mathbf{b}^\top \mathbf{u}_i)^2 \right\}$$

that is,

$$\sigma^2 \,|\, a, \mathbf{b}, \mathbf{u}, y, \mathbf{x} \sim \text{Inv-}\chi^2\left(n(p+1), \frac{\sum_{i=1}^n (\mathbf{x}_i - \mathbf{u}_i)^\top (\mathbf{x}_i - \mathbf{u}_i) + (y_i - a - \mathbf{b}^\top \mathbf{u}_i)^2/K}{n(p+1)}\right).$$

Finally, since for $i = 1, \ldots, n$

$$\begin{bmatrix} \mathbf{x}_i \\ y_i \end{bmatrix} \,\Bigg|\, a, \mathbf{b}, \sigma^2, \mathbf{u}_i \sim N\left(\begin{bmatrix} 0 \\ a \end{bmatrix} + \begin{bmatrix} I_p \\ \mathbf{b}^\top \end{bmatrix} \mathbf{u}_i, \sigma^2 \begin{bmatrix} I_p & 0 \\ 0 & K \end{bmatrix} \right),$$

that is,

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{x}_i \\ (y_i - a)/\sqrt{K} \end{bmatrix} \,\Bigg|\, \mathbf{b}, \sigma^2, \mathbf{u}_i \sim N\Bigg(\underbrace{\begin{bmatrix} I_p \\ \mathbf{b}^\top/\sqrt{K} \end{bmatrix}}_{X_u} \mathbf{u}_i, \sigma^2 I_{p+1} \Bigg),$$

we just need to regress $\mathbf{z}_i$ on X_u to sample from $\mathbf{u}_i$. Putting all together, we have our sampler:

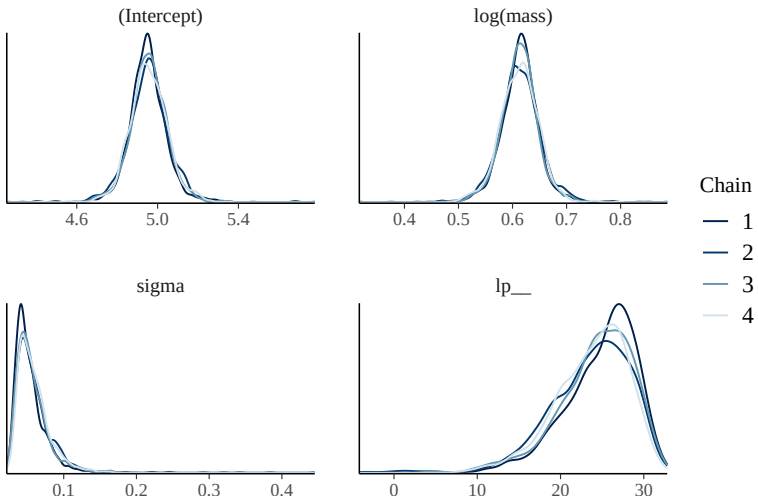
```
# [x_i', y_i]' ~ N([u_i', a + b' * u_i]', sigma2 * (I_p (+) K))
# Non-informative priors for u_i, a, b, and sigma2
sample_error_in_vars <- function (y, X, K = 1,
                                  iter = 2000, warmup = floor(iter / 2), nchains = 4) {
  n <- nrow(X); p <- ncol(X)
  sqrt_K <- sqrt(K)
  params <- c("(Intercept)", colnames(X), "sigma", "lp__")
  sims <- mcmc_array(iter - warmup, nchains, params)
  for (chain in 1:nchains) {
    bl <- lm(y ~ X); gamma <- coef(bl)
    alpha <- gamma[1]; beta <- gamma[-1]
    U <- X
    for (it in 1:iter) {
      # [ sigma | alpha, beta, U, X, Y ]
      sigma <- sqrt(rinvchisq(1, n * (p + 1),
                              sum(apply(X - U, 1, crossprod)) +
                                sum((y - alpha - U %*% beta) ^ 2) / K))
      # [ U | alpha, beta, sigma, X, Y ]
      Xu <- rbind(diag(p), beta / sqrt_K)
      Z <- cbind(X, (y - alpha) / sqrt_K)
      for (i in 1:n) U[i,] <- bslm_sample1(Z[i,], Xu, sigma)
      # [ alpha, beta | sigma, U, X, Y ]
      gamma <- bslm_sample1(y, cbind(1, U), sqrt_K * sigma)
      alpha <- gamma[1]; beta <- gamma[-1]

      target <- sum(dnorm(X, U, sigma, log = TRUE)) +
        sum(dnorm(y, alpha + U %*% beta, sqrt_K * sigma, log = TRUE)) -
        log(sigma)
      if (it > warmup)
        sims[it - warmup, chain, ] <- c(gamma, sigma, target)
    }
  }
  sims
}
```

(12.a-b)

```
dogs <- read.csv("data/dogs.csv", comment = "#")
y <- log(dogs$rate)
X <- model.matrix(~ log(mass) - 1, data = dogs)
sims <- sample_error_in_vars(y, X, K = 1)
monitor(sims)
```

```
Inference for the input samples (4 chains: each with iter = 1000; warmup = 500):

             Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
(Intercept)  4.8  5.0  5.1  5.0 0.1  1.00     1184     1068
log(mass)    0.5  0.6  0.7  0.6 0.0  1.00     1182     1001
sigma        0.0  0.1  0.1  0.1 0.0  1.02      247      342
lp__        14.7 24.6 29.8 23.7 4.8  1.02      246      349

For each parameter, Bulk_ESS and Tail_ESS are crude measures of
effective sample size for bulk and tail quantities respectively (an ESS > 100
per chain is considered good), and Rhat is the potential scale reduction
factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

```
mcmc_dens_overlay(sims)
```



In the original scale we have good evidence of a positive association between metabolic rate and body mass of dogs, with roughly rate $\propto$ mass$^{0.6}$.

(12.c)

```
sims <- sample_error_in_vars(y, X, K = 2)
monitor(sims)
```

```
Inference for the input samples (4 chains: each with iter = 1000; warmup = 500):

             Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
(Intercept)  4.8  5.0  5.1  5.0 0.1     1     1412     1184
log(mass)    0.6  0.6  0.7  0.6 0.0     1     1368     1205
sigma        0.0  0.0  0.1  0.0 0.0     1      403      376
lp__        18.0 26.8 31.6 26.1 4.4     1      428      465
```

```
For each parameter, Bulk_ESS and Tail_ESS are crude measures of
effective sample size for bulk and tail quantities respectively (an ESS > 100
per chain is considered good), and Rhat is the potential scale reduction
factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

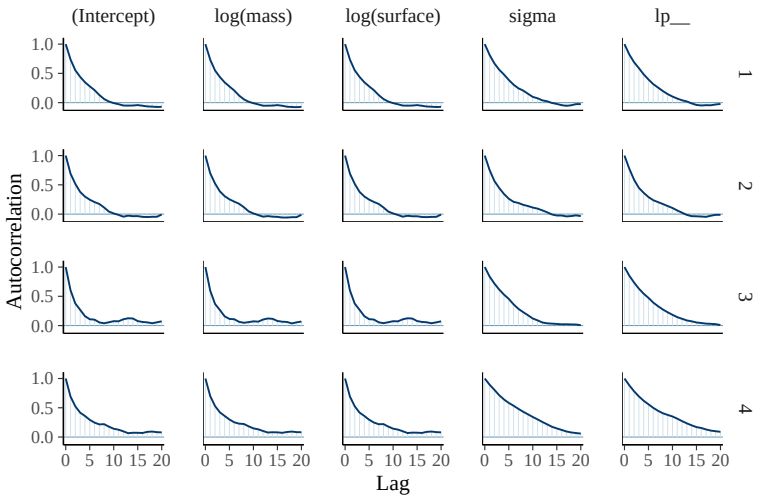
Coefficient estimates have not changed considerably, but the shared scale on measurement errors, σ, has a lower estimate now.

(13)

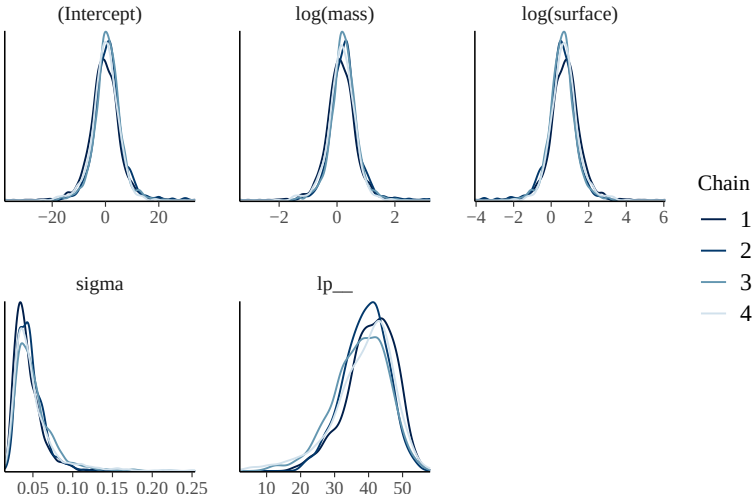
```
X <- model.matrix(~ log(mass) + log(surface) - 1, data = dogs)
sims <- sample_error_in_vars(y, X, K = 1)
monitor(sims)
```

```
Inference for the input samples (4 chains: each with iter = 1000; warmup = 500):

               Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
(Intercept)  -6.9  0.6  9.0  0.7 5.4  1.02      291      460
log(mass)    -0.5  0.2  1.0  0.2 0.5  1.02      293      464
log(surface) -0.6  0.6  1.7  0.6 0.8  1.02      291      454
sigma         0.0  0.0  0.1  0.0 0.0  1.06       94      210
lp__         25.4 39.5 49.2 38.6 7.7  1.06       97      217

For each parameter, Bulk_ESS and Tail_ESS are crude measures of
effective sample size for bulk and tail quantities respectively (an ESS > 100
per chain is considered good), and Rhat is the potential scale reduction
factor on rank normalized split chains (at convergence, Rhat <= 1.05).
```

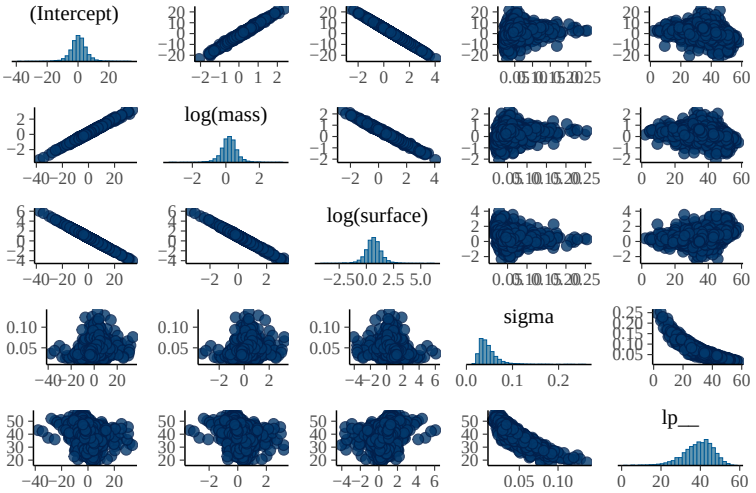
```
mcmc_acf(sims)
```




```
mcmc_dens_overlay(sims)
```




```
mcmc_pairs(sims)
```



The high collinearity between log(mass) and log(surface) inflates the posterior covariance of **b** and makes convergence difficult due to a thinly shaped region of high posterior mass, as evidenced by the higher autocorrelation in the `mcmc_acf` plots. From the `mcmc_pairs` plot we can see that all covariates are highly correlated *a posteriori*. Note that, due to variance inflation, log(mass) and log(surface) don't seem to be relevant in explaining log(rate).

3

(a)

The joint distribution is

$$\mathbb{P}(\beta, \sigma^2, \mathbf{y}) \propto (\sigma^2)^{-\left(\frac{n+p+\nu}{2}+1\right)} \exp\left\{ -\frac{1}{2\sigma^2}\left[\nu\tau^2 + RSS(\beta) + (\beta-\beta_0)^\top \Sigma_0^{-1}(\beta-\beta_0)\right]\right\}.$$

As we've seen in class, with $\beta^* = (X^\top X)^{-1}X^\top \mathbf{y}$,

$$RSS(\beta) = RSS(\beta^*) + (\beta-\beta^*)^\top X^\top X(\beta-\beta^*).$$

Moreover, by the centering trick,

$$(\beta-\beta^*)^\top X^\top X(\beta-\beta^*)+(\beta-\beta_0)^\top\Sigma_0^{-1}(\beta-\beta_0) = (\widehat{\beta}-\beta^*)^\top X^\top X(\widehat{\beta}-\beta^*)+(\widehat{\beta}-\beta_0)^\top\Sigma_0^{-1}(\widehat{\beta}-\beta_0)+(\beta-\widehat{\beta})^\top\Sigma_\beta^{-1}(\beta-\widehat{\beta}),$$

with $\Sigma_\beta^{-1} = X^\top X + \Sigma_0^{-1}$ and $\widehat{\beta} = \Sigma_\beta(X^\top X\beta^* + \Sigma_0^{-1}\beta_0) = \Sigma_\beta(X^\top \mathbf{y} + \Sigma_0^{-1}\beta_0)$. Finally, since $X^\top(\mathbf{y} - X\beta^*) = 0$, $(\widehat{\beta} - \beta^*)^\top X^\top(\mathbf{y} - X\beta^*) = 0$, and so

$$RSS(\beta^*) + (\widehat{\beta} - \beta^*)^\top X^\top X(\widehat{\beta} - \beta^*) = RSS(\widehat{\beta}),$$

and the result follows, that is,

$$RSS(\beta) + (\beta - \beta_0)^\top \Sigma_0^{-1}(\beta - \beta_0) = RSS(\widehat{\beta}) + (\widehat{\beta} - \beta_0)^\top \Sigma_0^{-1}(\widehat{\beta} - \beta_0) + (\beta - \widehat{\beta})^\top \Sigma_\beta^{-1}(\beta - \widehat{\beta}).$$

(b)

We can recognize the density of $\beta \sim N(\widehat{\beta}, \sigma^2\Sigma_\beta)$, $(\sigma^2)^{-p/2}\exp\{-(\beta - \widehat{\beta})^\top \Sigma_\beta^{-1}(\beta - \widehat{\beta})/(2\sigma^2)\}$ in the joint, and so, since it integrates to a constant, we have after marginalizing β,

$$\mathbb{P}(\sigma^2, \mathbf{y}) \propto (\sigma^2)^{-\left(\frac{n+\nu}{2}+1\right)} \exp\left\{ -\frac{1}{2\sigma^2}\Big[\nu\tau^2 + RSS(\widehat{\beta}) + (\widehat{\beta} - \beta_0)^\top \Sigma_0^{-1}(\widehat{\beta} - \beta_0)\Big]\right\},$$

that is, $\sigma^2 \,|\, \mathbf{y} \sim \mathsf{Inv}\text{-}\chi^2(n + \nu, (\nu\tau^2 + RSS(\widehat{\beta}) + (\widehat{\beta} - \beta_0)^\top \Sigma_0^{-1}(\widehat{\beta} - \beta_0))/(n + \nu))$.

(c)

Expanding, we have

$$RSS(\widehat{\beta}) + \widehat{\beta}^\top \Sigma_0^{-1}\widehat{\beta} = \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top X\widehat{\beta} - \widehat{\beta}^\top X^\top \mathbf{y} + \widehat{\beta}^\top(X^\top X + \Sigma_0^{-1})\widehat{\beta},$$

but since $\widehat{\beta} = \Sigma_\beta X^\top \mathbf{y}$ and thus

$$\widehat{\beta}^\top(X^\top X + \Sigma_0^{-1})\widehat{\beta} = \widehat{\beta}^\top \Sigma_\beta^{-1}\widehat{\beta} = \mathbf{y}^\top X\Sigma_\beta\Sigma_\beta^{-1}\Sigma_\beta X^\top \mathbf{y} = \mathbf{y}^\top X\widehat{\beta},$$

we then have

$$RSS(\widehat{\beta}) + \widehat{\beta}^\top \Sigma_0^{-1}\widehat{\beta} = \mathbf{y}^\top\mathbf{y} - \widehat{\beta}X^\top \mathbf{y} = \mathbf{y}^\top(I_n - \underbrace{X\Sigma_\beta X^\top}_{H})\mathbf{y}.$$