Models

2025-04-28

```r
library(tidyverse)
library(janitor)
library(rstan)
library(rstanarm)
library(bayesplot)
library(MCMCpack)
library(lme4)
student_data <- read.csv("student-scores.csv");
clean_data <- read.csv("student-scores-clean.csv")
```

$$Y_i|\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \sigma^2 \sim MVN(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i}, \sigma^2)$$

where: $x_{1i}, x_{2i}, x_{3i}, x_{4i}$ are the predictors for observation i

$$\beta|\sigma^2, y \; MVN(\hat{\beta}, \widehat{V})$$

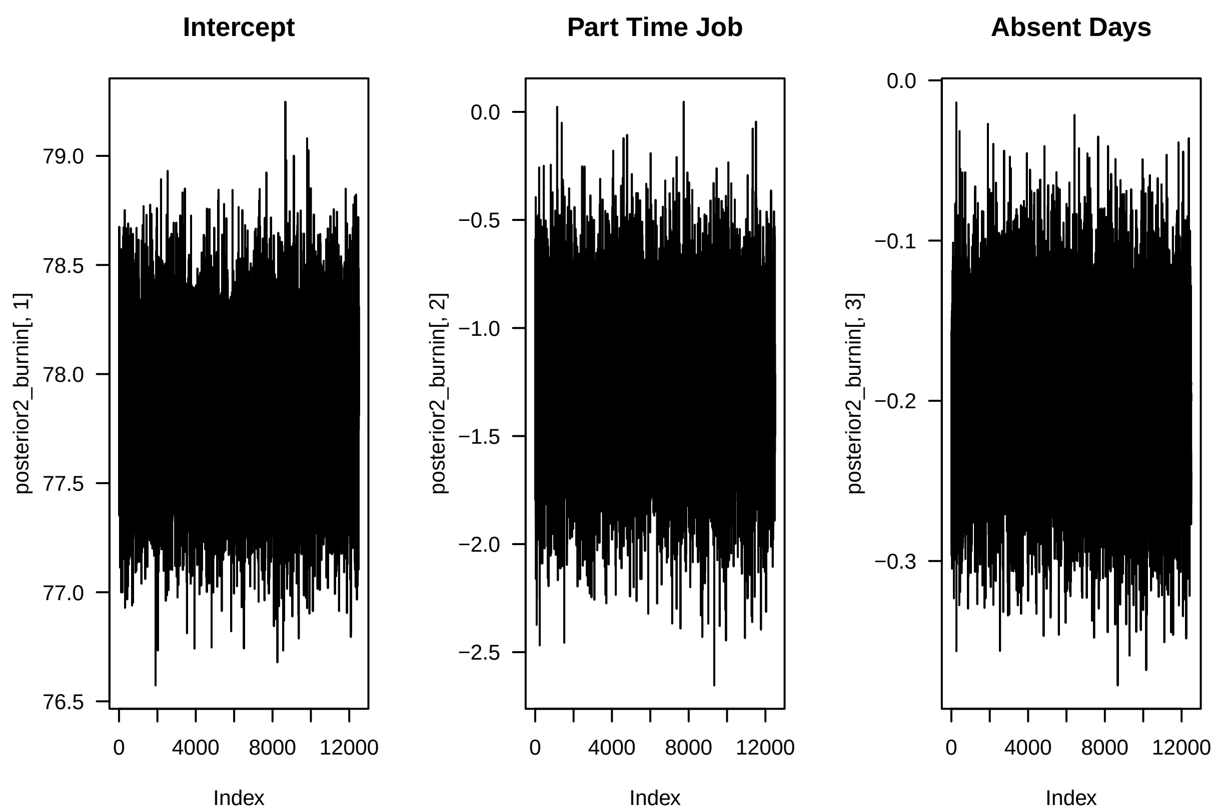
$$\sigma^2|\beta, y \sim InvGamma[a + \frac{n}{2}, b + \frac{1}{2}(y - X\beta)^T(y - X\beta)]$$

```r
#Block Gibbs Sampler
set.seed(4889)
clean_data <- read.csv("student-scores-clean.csv")


set.seed(8451)
y <- clean_data$average_score
x1 <- clean_data$part_time_job
x2 <- clean_data$absence_days
x3 <- clean_data$extracurricular_activities
x4 <- clean_data$weekly_self_study_hours


# Design matrix
X <- cbind(1, x1, x2, x3, x4)
n <- length(y)
p <- ncol(X)

# Hyperparameters
tau2 <- 10000^2
a <- b <- 1
mu0 <- rep(0, p)
```

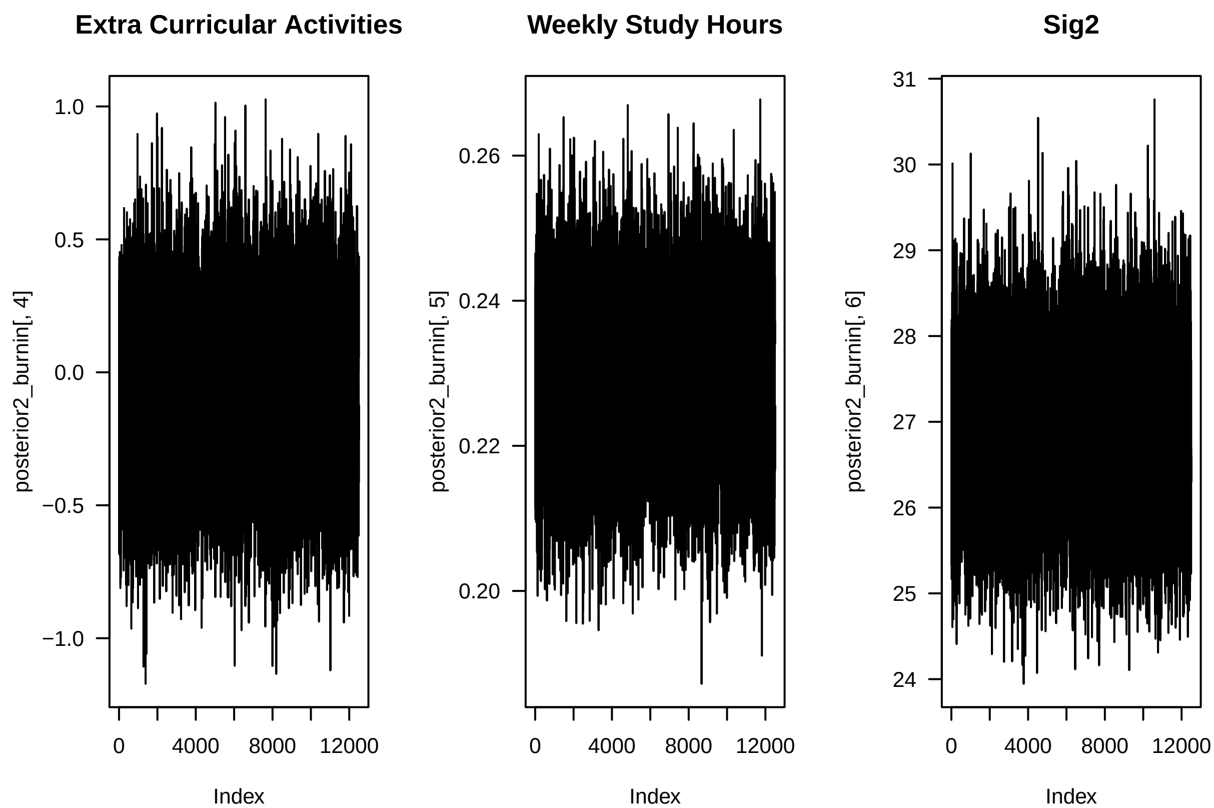
```r
S <- 2.5e4

#place to store data
posterior_beta <- matrix(NA, S, p)
posterior_sig2 <- rep(NA, S)

beta <- rep(0, p)
sig2 <- 1

XX <- t(X) %*% X
Xy <- t(X) %*% y

# block Gibbs sampler
for (s in 1:S) {

  # Update beta0
  v <- solve(XX / sig2 + diag(rep(1/tau2, p)))
  m <- v %*% (Xy / sig2 + mu0 / tau2)
  beta <- m + t(chol(v)) %*% rnorm(p)

  # Update sig2 (variance)
  sig2 <- rinvgamma(1, a + n/2,
                    b + t(y - X %*% beta)%*%(y-X%*%beta) / 2)

  # Store results
  posterior_beta[s, ] <- beta
  posterior_sig2[s] <- sig2
}

posterior2 <- cbind(posterior_beta, posterior_sig2)
colnames(posterior2) <- c("Intercept", "Part Time Job", "Absent Days",
                          "Extra Curricular Activities", "Weekly Study Hours", "Sig2")

#remove burn-in
posterior2_burnin <- posterior2[1:round(s/2),]

head(posterior2_burnin)
```

```
##      Intercept Part Time Job Absent Days Extra Curricular Activities
## [1,]  77.87654     -1.281166  -0.2003770                  -0.1229279
## [2,]  77.90033     -0.589276  -0.2092001                   0.1524083
## [3,]  78.04359     -1.461563  -0.2041925                  -0.2506536
## [4,]  77.76850     -1.559494  -0.2369275                  -0.0571564
## [5,]  78.24938     -1.271030  -0.2965047                  -0.1193520
## [6,]  78.64778     -1.082733  -0.2756962                  -0.6835690
##      Weekly Study Hours     Sig2
## [1,]          0.2255845 27.69401
## [2,]          0.2279868 25.71107
## [3,]          0.2197806 26.16819
## [4,]          0.2424850 27.25814
## [5,]          0.2139665 27.47846
## [6,]          0.2114292 27.02233
```

```
# Block Gibbs Sampler Trace plots
par(mfrow=c(1,3))
plot(posterior2_burnin[,1], type="l", las=1, main="Intercept")
plot(posterior2_burnin[,2], type="l", las=1, main="Part Time Job")
plot(posterior2_burnin[,3], type="l", las=1, main="Absent Days")
```




```
plot(posterior2_burnin[,4], type="l", las=1, main="Extra Curricular Activities")
plot(posterior2_burnin[,5], type="l", las=1, main="Weekly Study Hours")
plot(posterior2_burnin[,6], type="l", las=1, main="Sig2")
```




```r
results <-data.frame(
  mean=colMeans(posterior2_burnin),
  sd=apply(posterior2_burnin,2,sd),
  lower=apply(posterior2_burnin,2,quantile,0.025),
  upper=apply(posterior2_burnin,2,quantile,0.975),
  row.names=colnames(posterior2_burnin))
round(results,2)
```

```
##                               mean   sd lower upper
## Intercept                    77.84 0.31 77.23 78.45
## Part Time Job                -1.27 0.33 -1.91 -0.63
## Absent Days                  -0.20 0.05 -0.29 -0.10
## Extra Curricular Activities -0.09 0.29 -0.65  0.48
## Weekly Study Hours            0.23 0.01  0.21  0.25
## Sig2                         26.86 0.85 25.23 28.58
```

```r
lm(y~x1+x2+x3+x4, data=clean_data)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = clean_data)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4
##    77.84436     -1.27204     -0.19562     -0.08946      0.22935
```

```r
# fit model in rstanarm
grades_lmer <- stan_lmer(average_score ~ part_time_job +
```

```r
                         absence_days + extracurricular_activities +
                         weekly_self_study_hours + (1|gender),
                     data = clean_data)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000171 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.71 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 16.353 seconds (Warm-up)
## Chain 1:                8.287 seconds (Sampling)
## Chain 1:                24.64 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000114 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.14 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 23.281 seconds (Warm-up)
## Chain 2:                39.583 seconds (Sampling)
## Chain 2:                62.864 seconds (Total)
## Chain 2:
```

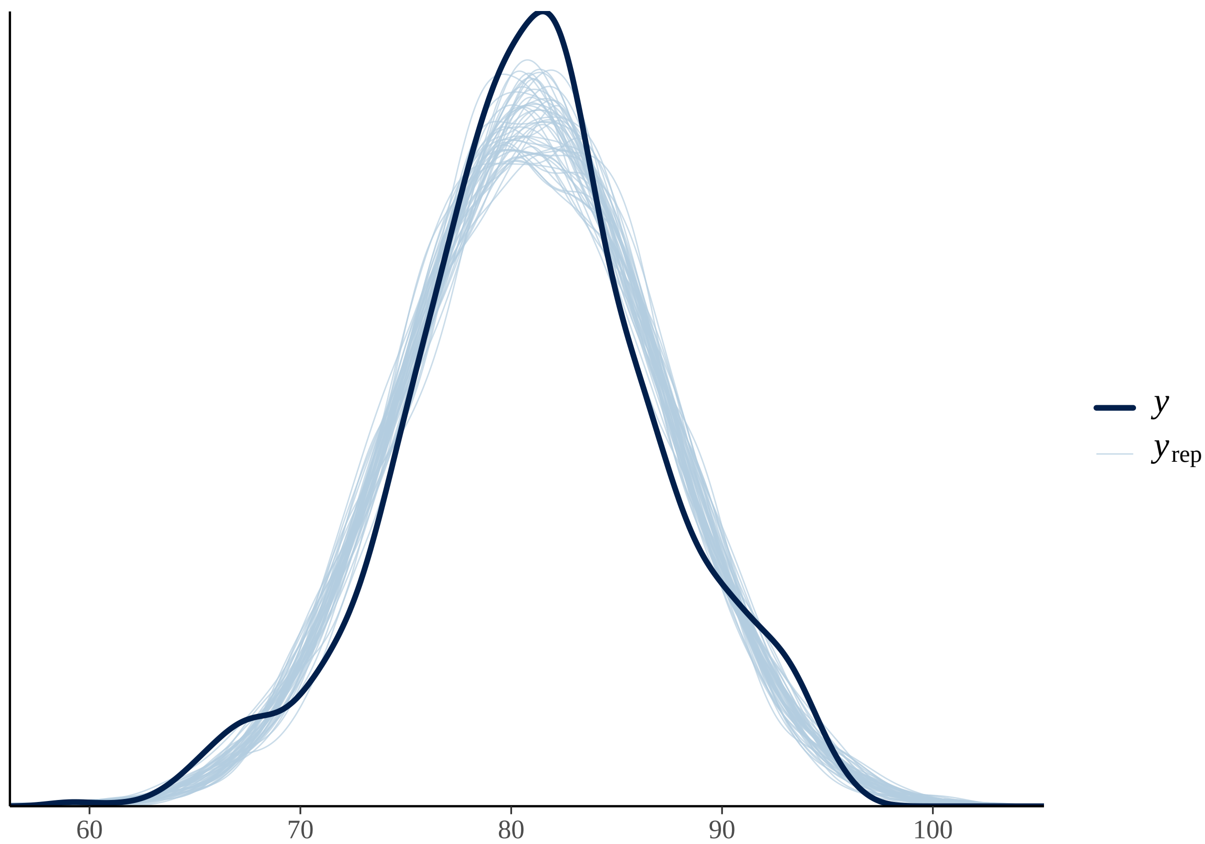
```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000119 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 1.19 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 13.409 seconds (Warm-up)
## Chain 3:                8.992 seconds (Sampling)
## Chain 3:                22.401 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000112 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.12 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 18.612 seconds (Warm-up)
## Chain 4:                9.492 seconds (Sampling)
## Chain 4:                28.104 seconds (Total)
## Chain 4:
```

```r
# show results
summary(grades_lmer, digits = 3)
```

```
##
```

```
## Model Info:
##  function:     stan_lmer
##  family:       gaussian [identity]
##  formula:      average_score ~ part_time_job + absence_days + extracurricular_activities +
##     weekly_self_study_hours + (1 | gender)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 2000
##  groups:       gender (2)
##
## Estimates:
##                                          mean   sd     10%    50%    90%
## (Intercept)                            77.834  1.123 76.765 77.830 78.859
## part_time_job                          -1.280  0.325 -1.695 -1.281 -0.867
## absence_days                           -0.197  0.047 -0.257 -0.198 -0.134
## extracurricular_activities             -0.095  0.292 -0.459 -0.098  0.280
## weekly_self_study_hours                 0.230  0.010  0.217  0.230  0.243
## b[(Intercept) gender:0]                 0.132  1.087 -0.792  0.082  1.172
## b[(Intercept) gender:1]                -0.115  1.085 -1.091 -0.063  0.836
## sigma                                   5.183  0.083  5.077  5.182  5.291
## Sigma[gender:(Intercept),(Intercept)]  3.931 12.664  0.010  0.437  9.653
##
## Fit Diagnostics:
##            mean   sd     10%    50%    90%
## mean_PPD 80.982  0.162 80.777 80.984 81.188
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                       mcse  Rhat  n_eff
## (Intercept)                           0.037 1.010  927
## part_time_job                         0.006 0.999 3053
## absence_days                          0.001 1.000 2920
## extracurricular_activities            0.005 1.000 3281
## weekly_self_study_hours               0.000 1.001 2955
## b[(Intercept) gender:0]               0.036 1.009  926
## b[(Intercept) gender:1]               0.036 1.008  910
## sigma                                 0.002 1.000 2672
## Sigma[gender:(Intercept),(Intercept)] 0.326 1.005 1505
## mean_PPD                              0.003 1.000 3556
## log-posterior                         0.081 1.003 1037
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
pp_check(grades_lmer)
```



60
70
80
90
100
y
y rep