

Homework 8

Aakash Kothapally, STA-360

5 PM EDT Monday, November 22

```
# libraries
library(MASS)
```

1c (Task 3)

We create a function to help us obtain our posterior predictive distributions.

```
set.seed(42)
sim_count = 5000

predict <- function(y) {
  X = cbind(rep(1, 6), seq(1, 11, by = 2))
  beta_init = c(23, 0)
  sigma2_init = rbind(c(0.25, 0), c(0, 0.1))
  sigma2 = 0.7^2

  beta_matrix = matrix(nrow = sim_count, ncol = length(beta_init))
  sigma_matrix = numeric(sim_count)

  for (i in 1:sim_count) {
    beta = mvrnorm(1, solve(solve(sigma2_init) + (t(X) %*% X) / sigma2) %*%
                     (solve(sigma2_init) %*% beta_init + (t(X) %*% y) / sigma2),
                     solve(solve(sigma2_init) + (t(X) %*% X) / sigma2))
    sum_squares = (t(y) %*% y) - (2 * t(beta) %*% t(X) %*% y) + (t(beta) %*% t(X) %*%
                                                                X %*% beta)
    sigma2 = 1 / rgamma(1, (1 + 6) / 2, (1 * 1/4 + sum_squares) / 2)

    beta_matrix[i,] = beta
    sigma_matrix[i] = sigma2
  }

  return(rnorm(sim_count, beta_matrix %*% c(1, 13), sqrt(sigma_matrix)))
}
```

We can use the function to obtain our posterior predictive distribution for each swimmer. We use the swimming dat file's data to do this, which has data for each swimmer by row. Our posterior predictive distributions are stored in a table-like structure with each swimmer getting their own column with 5000 simulations of the Gibbs sampler. Provided is a small selection of the first few simulations for each of the 4 swimmers.

```
swim = read.table("swim.dat")
swim_pred = apply(swim, MARGIN = 1, predict)
head(swim_pred)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 22.86482 24.20159 22.34956 24.37127
```

```
## [2,] 22.22213 23.75341 22.76882 23.45320
## [3,] 22.16597 23.99481 22.67669 23.35086
## [4,] 22.08643 24.00023 23.05705 23.57172
## [5,] 22.59669 23.59786 23.13615 23.21242
## [6,] 22.34339 23.65125 22.87595 22.30891
```

```
sigma_init = rbind(c(0.25, 0), c(0, 0.1))
sigma_init
```

```
##      [,1] [,2]
## [1,] 0.25 0.0
## [2,] 0.00 0.1
```

1d (Task 4)

We're going to use the posterior predictive distributions' values for each of the simulations to help determine which swimmer is the fastest. Since we are looking at the fastest, we want the swimmer with the smallest time in each run. We can use that to construct the probabilities.

```
fastest_times = apply(swim_pred, MARGIN = 1, which.min)
```

This is the probability that swimmer 1 is the fastest in a given simulation.

```
length(fastest_times[fastest_times==1])/5000
```

```
## [1] 0.6392
```

This is the probability that swimmer 2 is the fastest in a given simulation.

```
length(fastest_times[fastest_times==2])/5000
```

```
## [1] 0.0166
```

This is the probability that swimmer 3 is the fastest in a given simulation.

```
length(fastest_times[fastest_times==3])/5000
```

```
## [1] 0.3152
```

This is the probability that swimmer 4 is the fastest in a given simulation.

```
length(fastest_times[fastest_times==4])/5000
```

```
## [1] 0.029
```

Based on these values, in the majority of simulations, swimmer 1 had the quickest time. We would recommend that the coach picks swimmer 1.