# Homework 8, STA 360

Austin Kao

## Lab Component

```
### Call necessary R libraries
library(mvtnorm)
### Load in the data
dataset <- read.table("data/swim.dat", header = FALSE)
datamat <- data.matrix(dataset)
```

c.

```
### Based on code provided in Olivier's lab
### Set seed for reproducibility
set.seed(123)

### Initialize parameters
beta0 = c(23, 0)
dim(beta0) <- c(2,1)
a = 0.1
b = 0.1
Sigma0 = diag(c(5,2))
n.iter = 5000
X_i = c(1,1,1,1,1,1,1,1,3,5,7,9,11)
dim(X_i) <- c(6,2)

gibbs_sampler <- function(i) {
  ### Initialize empty arrays for Gibbs sampling
  beta = array(dim = c(2, n.iter))
  sigma2 = array(dim = c(n.iter))
  beta[,1] = beta0
  sigma2[1] = 0.1
  Y_i <- datamat[i,]
  dim(Y_i) <- c(6,1)

  ### Define functions to compute parameters for Beta
  ### condition distribution
  find_beta_params <- function(sigma2, i) {
    variance = solve(solve(Sigma0) + sigma2 * t(X_i) %*% X_i)
    mean = variance %*% (solve(Sigma0) %*% beta0 + sigma2 * t(X_i) %*% Y_i)
    return (data.frame(mean, variance))
  }

  ### Do sampling
```

```r
  for (s in 2:n.iter) {
    params = find_beta_params(sigma2[s-1], i)
    mean <- data.matrix(params[1])
    variance <- data.matrix(params[2:3])
    beta[,s] = rmvnorm(1, mean, variance)
    sigma2[s] = rgamma(1, a+3, b+(t(Y_i - X_i %*% beta[,s])%*%(Y_i - X_i %*% beta[,s]))/2)
  }

  return (data.frame(t(beta), sigma2))
}
# Construct a list of posterior samples for each swimmer
Res = list()
for (i in 1:4) {
  samples <- gibbs_sampler(i)
  Res[[i]] = samples
}

# Following code provided by Prof. Steorts
# This function samples from the posterior predictive distribution for
# a swimmer who has posterior samples in the matrix draws
# n.samps is number of samples, burn is burnin

postPredSampler <- function(n.samps, draws, burn = 1000){
# take my draws and remove the burn-in
draws <- draws[burn:nrow(draws),]
# now sample
rows <- sample(nrow(draws), n.samps)
# n.samps is the number of samples to draw
# draws is where you grab these from
# Utilize 13 here as we want to predict the 13th week given that
# we have the previous 12 weeks
y.draws <- rnorm(n.samps, mean = draws[rows,1] + 13*draws[rows,2], sd = sqrt(1/draws[rows,3]))
return(y.draws)
}

# draw 1000 samples
n.samps <- 1000
post.pred.draws <- sapply(Res, function(x) {postPredSampler(n.samps, x)})
tail(post.pred.draws)
```

```
##              [,1]     [,2]     [,3]     [,4]
##  [995,] 23.00950 23.88783 22.69619 23.37417
##  [996,] 22.42388 23.16576 22.64891 23.55420
##  [997,] 22.89453 23.46879 22.86452 23.57479
##  [998,] 21.79182 23.75823 22.93335 23.58786
##  [999,] 22.60869 23.30031 22.45205 23.11634
## [1000,] 22.54136 23.21610 22.67729 22.44124
```

d.

```r
# calculate probability that swimmer is slowest at next meet
maxes <- apply(post.pred.draws, 1, which.max)
(swimmers <- paste0("Swimmer ", 1:4))
```

```
## [1] "Swimmer 1" "Swimmer 2" "Swimmer 3" "Swimmer 4"
```

```r
(max.probs <- table(maxes)/n.samps)
```

```
## maxes
##     1     2     3     4
## 0.024 0.608 0.055 0.313
```

```r
names(max.probs) <- swimmers
```

Based on these probabilities, it is best to recommend Swimmer 1. This is because swimmer 1 is the least likely to be the slowest at the next meet between all four swimmers. In other words, swimmer 1 swims the most consistently fast, and so is the least likely to get a slow time at the next meet. The swimming team is least likely to lose the next meet if the coach picks Swimmer 1.

## Extra Credit

a. Because $\theta$ has a multivariate normal distribution, the covariance matrix $T$ is positive-definite. Since positive-definite matrices are also symmetric, $T = T^T$. By extension, $T^{-1} = (T^{-1})^T$ by the matrix property $(T^{-1})^T = (T^T)^{-1}$.

$$\left((\theta^T T^{-1})(\mu)\right)^T = \mu^T(\theta^T T^{-1})^T = \mu^T \left(T^{-1}\right)^T \left(\theta^T\right)^T = \mu^T T^{-1}\theta$$

b. Because $\mu$ and $\theta$ are $d \times 1$ matrices, and $T$ is a $d \times d$ matrix, $\theta^T T^{-1}\mu$ is a $1 \times 1$ matrix. The transpose of any $1 \times 1$ matrix is the same matrix, so $\left(\theta^T T^{-1}\mu\right)^T = \theta^T T^{-1}\mu$.

$$\begin{aligned}
p(\theta) &\propto e^{-\frac{1}{2}(\theta-\mu)^T T^{-1}(\theta-\mu)} = e^{-\frac{1}{2}(\theta^T-\mu^T)T^{-1}(\theta-\mu)} = e^{-\frac{1}{2}(\theta^T T^{-1}-\mu^T T^{-1})(\theta-\mu)} \\
&= e^{-\frac{1}{2}(\theta^T T^{-1}\theta-\mu^T T^{-1}\theta-\theta^T T^{-1}\mu+\mu^T T^{-1}\mu)} \\
&\propto e^{-\frac{1}{2}(\theta^T T^{-1}\theta-\mu^T T^{-1}\theta-\theta^T T^{-1}\mu)} = e^{-\frac{1}{2}(\theta^T T^{-1}\theta-\left(\theta^T T^{-1}\mu\right)^T-\theta^T T^{-1}\mu)} = e^{-\frac{1}{2}(\theta^T T^{-1}\theta-\theta^T T^{-1}\mu-\theta^T T^{-1}\mu)} \\
p(\theta) &\propto e^{-\frac{1}{2}(\theta^T T^{-1}\theta-2\theta^T T^{-1}\mu)}
\end{aligned}$$

c.

$$\begin{aligned}
p(y \mid \theta, \Sigma) &\propto \prod_{i=1}^n e^{-\frac{1}{2}(y_i-\theta)^T\Sigma^{-1}(y_i-\theta)} = \prod_{i=1}^n e^{-\frac{1}{2}(y_i^T\Sigma^{-1}y_i-2\theta^T\Sigma^{-1}y_i+\theta^T\Sigma^{-1}\theta))} \\
p(\theta \mid y, \Sigma) &\propto p(y \mid \theta, \Sigma)p(\theta) \\
p(\theta \mid y, \Sigma) &\propto \prod_{i=1}^n e^{-\frac{1}{2}(y_i^T\Sigma^{-1}y_i-2\theta^T\Sigma^{-1}y_i+\theta^T\Sigma^{-1}\theta))} e^{-\frac{1}{2}(\theta^T T^{-1}\theta-2\theta^T T^{-1}\mu)} \\
p(\theta \mid y, \Sigma) &\propto \prod_{i=1}^n e^{-\frac{1}{2}(-2\theta^T\Sigma^{-1}y_i+\theta^T\Sigma^{-1}\theta))} e^{-\frac{1}{2}(\theta^T T^{-1}\theta-2\theta^T T^{-1}\mu)} \\
p(\theta \mid y, \Sigma) &\propto e^{-\frac{1}{2}(\sum_{i=1}^n\left(-2\theta^T\Sigma^{-1}y_i\right)+n\theta^T\Sigma^{-1}\theta)} e^{-\frac{1}{2}(\theta^T T^{-1}\theta-2\theta^T T^{-1}\mu)} \\
\bar{y} &= \frac{1}{n}\sum_{i=1}^n y_i \\
p(\theta \mid y, \Sigma) &\propto e^{-\frac{1}{2}(\left(-2\theta^T\Sigma^{-1}n\bar{y}\right)+n\theta^T\Sigma^{-1}\theta)} e^{-\frac{1}{2}(\theta^T T^{-1}\theta-2\theta^T T^{-1}\mu)} \\
p(\theta \mid y, \Sigma) &\propto e^{-\frac{1}{2}(\theta^T\left(n\Sigma^{-1}+T^{-1}\right)\theta-2\theta^T\left(\Sigma^{-1}n\bar{y}+T^{-1}\mu\right))} \\
T^* &= \left(n\Sigma^{-1}+T^{-1}\right)^{-1} \\
\mu^* &= T^*\left(\Sigma^{-1}n\bar{y}+T^{-1}\mu\right)
\end{aligned}$$

d.

$$\operatorname{tr}(\psi\Sigma^{-1}) + \sum_{i=1}^{n}(y_i-\theta)^T\Sigma^{-1}(y_i-\theta) = \operatorname{tr}(\psi\Sigma^{-1}) + \operatorname{tr}\left\{\sum_{i=1}^{n}(y_i-\theta)^T\Sigma^{-1}(y_i-\theta)\right\} =$$

$$\operatorname{tr}(\psi\Sigma^{-1}) + \operatorname{tr}\left\{\sum_{i=1}^{n}(y_i-\theta)(y_i-\theta)^T\Sigma^{-1}\right\} = \operatorname{tr}\left\{\psi\Sigma^{-1} + \sum_{i=1}^{n}(y_i-\theta)(y_i-\theta)^T\Sigma^{-1}\right\} =$$

$$\operatorname{tr}\left\{\left(\psi + \sum_{i=1}^{n}(y_i-\theta)(y_i-\theta)^T\right)\Sigma^{-1}\right\}$$

e.

$$p(y \mid \theta, \Sigma) \propto \prod_{i=1}^{n}\det(\Sigma)^{-\frac{1}{2}}e^{-\frac{1}{2}(y_i-\theta)^T\Sigma^{-1}(y_i-\theta)} = \det(\Sigma)^{-\frac{n}{2}}e^{-\frac{1}{2}\sum_{i=1}^{n}\left((y_i-\theta)^T\Sigma^{-1}(y_i-\theta)\right)}$$

$$p(\Sigma \mid y, \theta) \propto p(y \mid \theta, \Sigma)p(\Sigma)$$

$$p(\theta \mid y, \Sigma) \propto \det(\Sigma)^{-\frac{n}{2}}e^{-\frac{1}{2}\sum_{i=1}^{n}\left((y_i-\theta)^T\Sigma^{-1}(y_i-\theta)\right)}\det(\Sigma)^{-\frac{\nu+d+1}{2}}e^{-\frac{1}{2}(\operatorname{tr}(\psi\Sigma^{-1}))}$$

$$p(\theta \mid y, \Sigma) \propto \det(\Sigma)^{-\frac{n+\nu+d+1}{2}}e^{-\frac{1}{2}\left(\operatorname{tr}(\psi\Sigma^{-1}+\sum_{i=1}^{n}\left((y_i-\theta)^T\Sigma^{-1}(y_i-\theta)\right)\right)}$$

$$p(\theta \mid y, \Sigma) \propto \det(\Sigma)^{-\frac{n+\nu+d+1}{2}}e^{-\frac{1}{2}\left(\operatorname{tr}\left\{\left(\psi+\sum_{i=1}^{n}(y_i-\theta)(y_i-\theta)^T\right)\Sigma^{-1}\right\}\right)}$$

$$\nu^* = n + \nu$$

$$\psi^* = \left(\psi + \sum_{i=1}^{n}(y_i-\theta)(y_i-\theta)^T\right)^{-1}$$