

Homework 8, STA 360

Jerry Xin

Total points: 10 (reproducibility) + 10 (Q1) = 20 points. Q2 is worth up to 8 points of extra credit on this assignment to help anyone that has had a difficult time in the course and also to help you prepare for the final exam.

General instructions for homeworks: Please follow the uploading file instructions according to the syllabus. You will give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used. Your code must be completely reproducible and must compile. Syllabus: (<https://github.com/resteorts/modern-bayes/blob/master/syllabus/syllabus-sta602-spring19.pdf>)

Advice: Start early on the homeworks and it is advised that you not wait until the day of. While the professor and the TA's check emails, they will be answered in the order they are received and last minute help will not be given unless we happen to be free.

Commenting code Code should be commented. See the Google style guide for questions regarding commenting or how to write code <https://google.github.io/styleguide/Rguide.xml>. No late homework's will be accepted.

Please look over the homework before lab this week. TA's will answer questions on the homework this week regarding these two problems below. I recommend that you work through them as much as possible before lab this week.

1. Lab component (10 points) Please complete Lab 10, parts c and d which correspond with linear regression, which can be found here: https://github.com/resteorts/modern-bayes/blob/master/labs/10-linear-regression/11-linear-regression_v2.pdf. It is highly recommend that you work through parts (a) and (b) on your own and derive these as these are excellent practice exercises for the exam. You can check your own work on this.
 - c) (5 points) Complete lab 10, part c.
 - d) (5 points) Complete lab 10, part d

```
library(plyr)
library(ggplot2)
library(dplyr)
library(xtable)
library(reshape)
library(tidyverse)
library(ggrepel)
library(MASS)
library(mvtnorm)
```

c) (5 points) Gibbs sampler (Task 3) Code a Gibbs sampler to fit each of the models. For each swimmer i , obtain draws from the posterior predictive distribution for y_i , the time of swimmer i if they were to swim two weeks from the last recorded time.

```
set.seed(123)
Y = t(matrix(c(23.1, 23.2, 22.9, 22.9, 22.8, 22.7,
              23.2, 23.1, 23.4, 23.5, 23.5, 23.4,
              22.7, 22.6, 22.8, 22.8, 22.9, 22.8,
              23.7, 23.6, 23.7, 23.5, 23.5, 23.4), nrow = 6, ncol = 6))
X = cbind(rep(1, 6), c(1, 3, 5, 7, 9, 11))
a = 0.1
b = 0.1

beta_0 = matrix(c(23, 0), nrow = 2)
Sigma0 = matrix(c(5, 0, 0, 2), nrow = 2, ncol = 2, byrow = T)
Sigma0_inv = solve(Sigma0)
S = 3000
beta = matrix(0, nrow = 3000, ncol = 2); beta[1,] = c(0, 0)
tau = rep(0, S); tau[1] = 2
Ystar = rep(0, S)
Ystar[1] = 0

gibbs = function(i){
  Yi = Y[,i]
  for(s in 2:S){
    #update beta
    post_V = solve(tau[s-1]*t(X)%*%X + Sigma0_inv)
    post_mean = solve(tau[s-1]*t(X)%*%X + Sigma0_inv)%*%(Sigma0_inv)%*%beta_0 + tau[s-1]*t(X)%*%Yi
    beta[s,] = rmvnorm(n = 1, mean = post_mean, sigma = post_V)

    #update tau
    SSR = t(Yi - X)%*%beta[s,])%*%(Yi - X)%*%beta[s,])
    tau[s] = rgamma(1, shape = 3 + a, rate = b + SSR/2)

    x = matrix(c(1, 13), nrow = 1)
    Ystar[s] = rnorm(n = 1, mean = x)%*%beta[s,], sd = sqrt(1/tau[s]))
  }
  return(list(post_beta = beta, post_tau = tau, post_pred = Ystar))
}

set.seed(123)
swimmer1_res = gibbs(1)
swimmer2_res = gibbs(2)
swimmer3_res = gibbs(3)
swimmer4_res = gibbs(4)
```

d) (5 points) Posterior Prediction (Task 4) The coach has to decide which swimmer should compete in a meet two weeks from the last recorded time. Using the posterior predictive distributions, compute $\Pr \{y_i = \max(y_1, y_2, y_3, y_4)\}$ for each swimmer i and use these probabilities to make a recommendation to the coach.

```
set.seed(123)
post_pred_all = cbind(swimmer1_res$post_pred, swimmer2_res$post_pred, swimmer3_res$post_pred, swimmer4_res$post_pred)
for(j in 1:4){
  total = 0;
  for(i in 1:S){
    {
      if(post_pred_all[i,j] == max(post_pred_all[i,])){
        total = total + 1
      }
      else{
        total = total
      }
    }
  }
  print(paste("swimmer",j,":", total/S))
}
```

```
## [1] "swimmer 1 : 0.227666666666667"
## [1] "swimmer 2 : 0.211333333333333"
## [1] "swimmer 3 : 0.292666666666667"
## [1] "swimmer 4 : 0.269333333333333"
```

Swimmer 2 has the lowest probability of having the largest recorded swim time, 0.21133 (less than 0.22767, 0.29267, 0.26933). Because Swimmer 2 is the lowest probability, the coach should send Swimmer 2 to compete in the competition because she has the lowest chance to come in last between the 4 swimmers (only a 21.133% chance to come in last).