

# Steven Maharaj 695281 Assignment 2, Question 2

Due: Friday 20 September 2019

There are places in this assignment where R code will be required. Therefore set the random seed so assignment is reproducible.

```
set.seed(695281) #Please change random seed to your student id number.
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

## Question Two (20 marks)

In lecture 3, we discussed how a Bayesian framework readily lends itself to combining information from sequential experiments. To demonstrate, consider the following data extracted from the *HealthIron* study.

Serum ferritin levels were measured for two samples of women, one of C282Y homozygotes ( $n = 88$ ) and the other of women with neither of the key mutations (C282Y and H63D) in the HFE gene, so-called HFE 'wildtypes' ( $n = 242$ ). The information available is

- `idnum`: Participant id.
- `homc282y`: Indicator whether individual is Homozygote (1) or Wildtype (0).
- `time`: Time since onset of menopause, measured in years.
- `logsf`: The natural logarithm of the serum ferritin in  $\mu\text{g/L}$ .

The data required to answer this question are `Hiron.csv`, which can be downloaded from LMS.

a) Fit a standard linear regression,

$$E(\text{logsf}) = \beta_0 + \beta_1 \text{time}$$

with responses restricted to those who are homozygote (`homc282y = 1`). This can be done using the `lm` function in R. Report the estimated coefficients  $\hat{\beta}$ , estimated error variance,  $\hat{\sigma}_e^2$  and  $(\mathbf{X}'\mathbf{X})^{-1}$ .

```
# Read the Data
Hiron <- read.csv("Hiron.csv")
HironHomo <- Hiron %>% filter(homc282y==1) %>% select(-idnum)
Hiron <- read.csv("Hiron.csv")
HironWild <- Hiron %>% filter(homc282y==0) %>% select(-idnum)
```

```

#fit linear regression using lm

model <- lm(logsf ~ time,data = HironHomo)
model$coefficients

## (Intercept)          time
## 4.23987253  0.07126085

intercept <-matrix(1,length(HironHomo$time),1)

X <- cbind(intercept,HironHomo$time)
XTX <- crossprod(X)
XTXinv <-solve(XTX)
# (XTX)^-1
XTXinv

##              [,1]      [,2]
## [1,]  0.033734956 -0.0015415009
## [2,] -0.001541501  0.0001062175

#Estimated error variance
sigma(model)^2

## [1] 1.715042

```

- b) Fit a Bayesian regression using a Gibbs sampler to **only the wildtype (homc282y=0) data**. Use the output from your answer in a) to define proper priors for  $\beta, \tau$ . For help, refer to lecture 13. For the Gibbs sampler, run two chains for 10,000 iterations. Discard the first 1000 iterations as burn-in and then remove every second remaining iteration to reduce auto-correlation. When storing results, convert  $\tau$  back to  $\sigma^2$ . When running the Gibbs sampler, incorporate posterior predictive checking, using the test statistic  $T(y, \beta) = \sum_{i=1}^n e_i^2$  and  $T(y^{\text{rep}}, \beta) = \sum_{i=1}^n (e_i^{\text{rep}})^2$ , where  $e_i$  is the predicted residual for observation  $i$  at simulation  $j$  and  $e_i^{\text{rep}}$  is the replicate residual for observation  $i$  at simulation  $j$ . Report posterior means, standard deviations and 95 % central credible intervals for  $\beta_0, \beta_1, \sigma^2$  combining results for the two chains.

- c) Perform convergence checks for the chain obtained in b). Report both graphical summaries and Gelman-Rubin diagnostic results. For the calculation of Gelman-Rubin diagnostics, you will need to install the R package coda. An example of processing chains for calculating Gelman-Rubin diagnostics is given below.

Processing chains for calculation of Gelman-Rubin diagnostics. Imagine you have 4 chains of a multi-parameter problem, and thinning already completed, called par1,par2,par3,par4

```

Step one: Converting the chains into mcmc lists.
library(coda)
par1<-as.mcmc.list(as.mcmc((par1)))
par2<-as.mcmc.list(as.mcmc((par2)))
par3<-as.mcmc.list(as.mcmc((par3)))
par4<-as.mcmc.list(as.mcmc((par4)))

```

Step two: Calculating diagnostics

```

par.all<-c(par1,par2,par3,par4)
gelman.diag(par.all)

```

- d) Fit a standard linear regression,

$$E(\text{logsf}) = \beta_0 + \beta_1 \text{time}$$

to **all the data** using the lm function in R. Report  $\hat{\beta}$ , and associated 95 % confidence intervals. Comparing

these results to the results from b), do you believe that sequential analysis gave the same results as fitting the regression on the full data.

- e) Report the results of posterior predictive checking requested in b). Do you believe the postulated model was plausible. If not, what do you think is a potential flaw in the postulated model.