STT 465 HW4

Nate Davis

11/20/2017

Question 1

```
library("coda")
```

```
## Warning: package 'coda' was built under R version 3.3.2
```

```
data <- read.table("gout.txt")

colnames(data)=c('sex','race','age','serum_urate','gout')

data$sex=factor(data$sex,levels=c('M','F'))
data$race=as.factor(data$race)

sex <- ifelse(data$sex=='F',1,0) # a dummy variable for female
race <- ifelse(data$race=='W',1,0) # a dummy variable for male

x <- cbind(1, sex, race, data$age)
y <- data$serum_urate
```

Gibbs Sampler Function

```
gibbs_regression <- function(x, y, iter, burn_in, b_prior, df_prior, var_b){

    n <- nrow(x)
    p <- ncol(x)

    betas <- matrix(nrow = iter, ncol = p, 0)
    var_e <- rep(NA, iter)

    s_prior <- var(y) * 0.8 * (df_prior - 2)

    betas[1,] <- 0
    betas[1,1] <- mean(y)
    b <- betas[1,]
    var_e[1] <- var(y)
    resid <- y - betas[1,1]

    for(i in 2:ncol(x)){ x[,i] <- x[,i] - mean(x[,i]) }

    ssx <- colSums(x^2)

    for(i in 2:iter){

        for(j in 1:p){
```

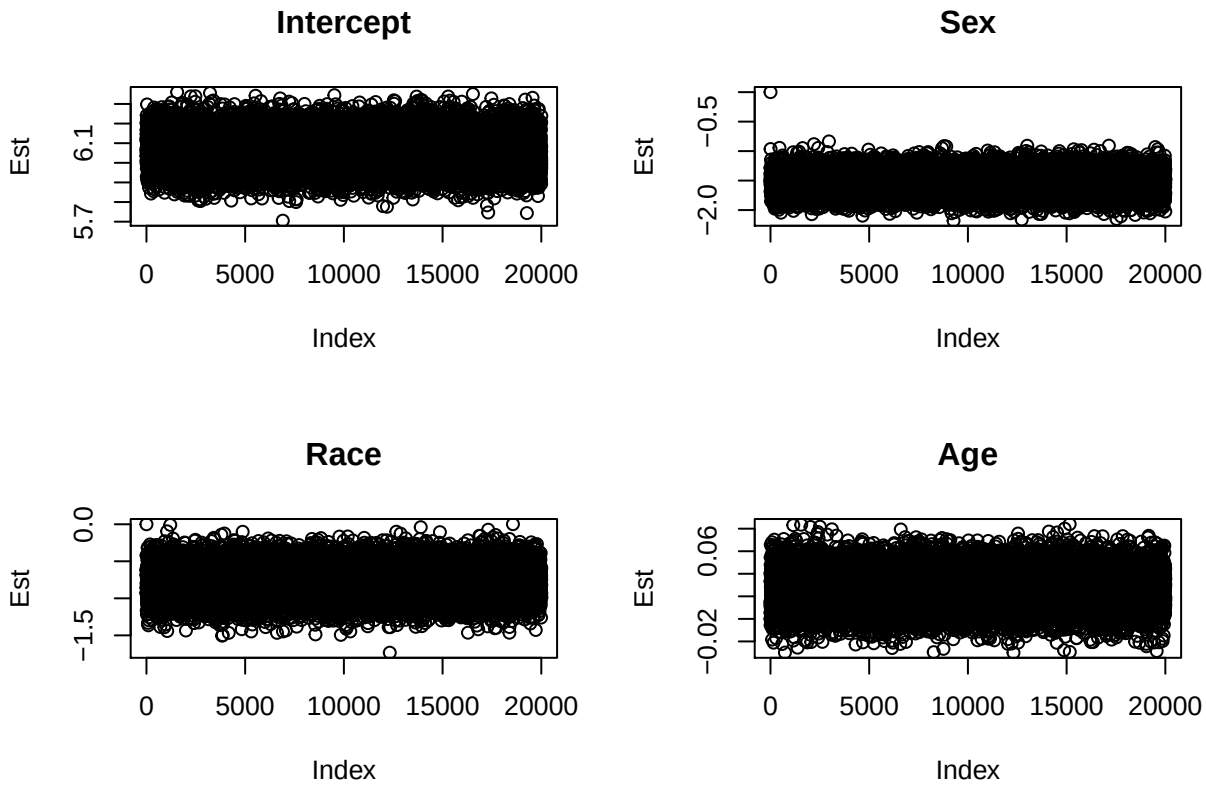
```
            c <- ssx[j] / var_e[i-1] + 1 / var_b
            y_star <- y - x[,-j] %*% b[-j]
            rhs <- sum(x[,j] * y_star) / var_e[i-1] + b_prior / var_b

            cond_mean <- rhs / c
            cond_var <- 1 / c

            b[j] <- rnorm(n = 1, mean = cond_mean, sd = sqrt(cond_var))
            betas[i,j] <- b[j]
        }

        rss <- sum(resid^2)
        df <- n + df_prior
        s <- rss + s_prior

        var_e[i] <- s / rchisq(df = df, n = 1)
    }

    out <- list(effects = betas, var_e = var_e)
    return(out)
}
```

Run model

```
reg <- gibbs_regression(x, y, iter = 20000, b_prior = 0, df_prior = 4, var_b = 10000)

reg_params <- data.frame(reg$effects)

colnames(reg_params) <- c("Intercept", "Sex", "Race", "Age")
```

1.1

```
par(mfrow = c(2, 2))

plot(reg_params$Intercept, xlab = "Index", ylab = "Est", main = "Intercept")
plot(reg_params$Sex, xlab = "Index", ylab = "Est", main = "Sex")
plot(reg_params$Race, xlab = "Index", ylab = "Est", main = "Race")
plot(reg_params$Age, xlab = "Index", ylab = "Est", main = "Age")
```



From here we see that there is little need for burn in (the only parameter with a largely deviant initial value is sex). With our large sample size, it is most likely negligible but we can use a burn-in of 100 to correct:

```
burn_in <- 100

reg_params <- reg_params[-(1:burn_in),]
```

1.2

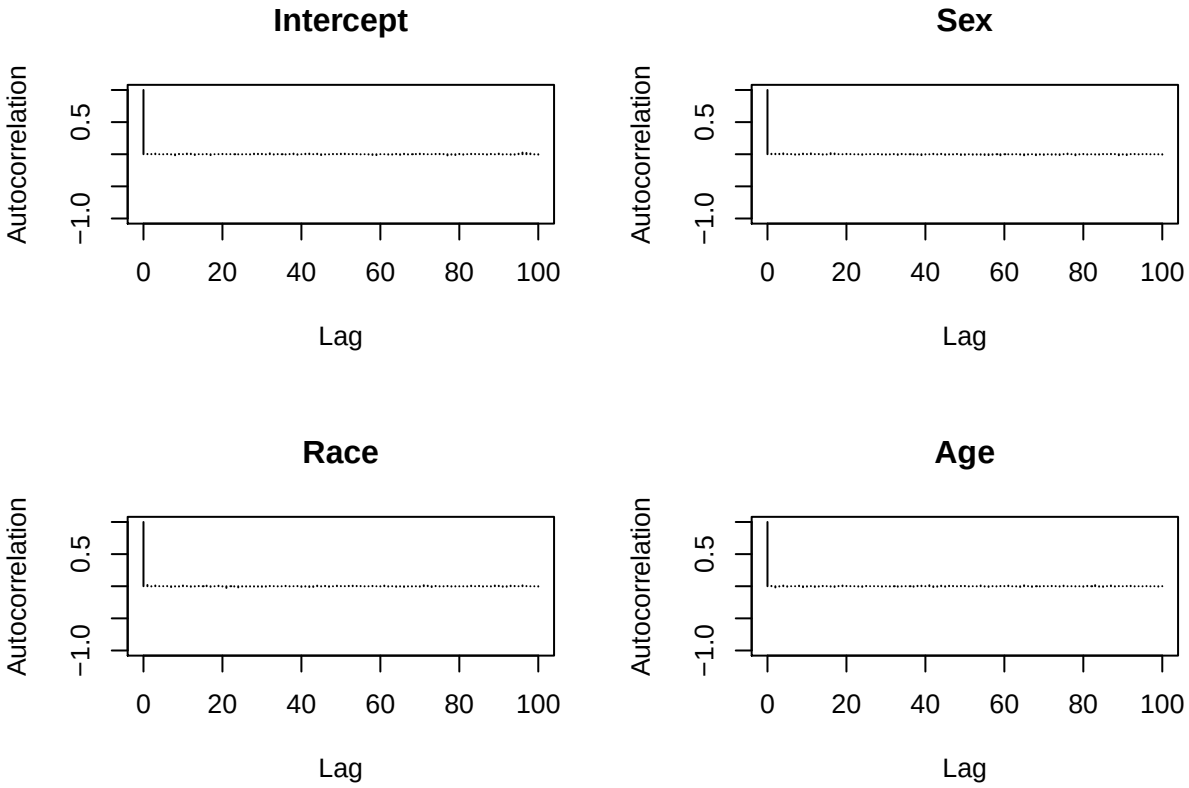
```
reg_mcmc <- mcmc(reg_params)

diag_ <- summary(reg_mcmc)

diag_
```

```
##
## Iterations = 1:19900
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 19900
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##               Mean      SD  Naive SE Time-series SE
## Intercept  6.06291 0.08044 0.0005702      0.0005702
## Sex       -1.52761 0.16385 0.0011615      0.0011615
## Race      -0.78158 0.19355 0.0013720      0.0013990
```

```
## Age        0.02672 0.01492 0.0001058      0.0001044
##
## 2. Quantiles for each variable:
##
##                2.5%     25%      50%      75%    97.5%
## Intercept  5.906889  6.0083  6.06241  6.11686  6.22148
## Sex       -1.847570 -1.6375 -1.52917 -1.41788 -1.20700
## Race      -1.164145 -0.9110 -0.78213 -0.65150 -0.40053
## Age       -0.002664  0.0169  0.02672  0.03675  0.05594
```

1.3

```
autocorr.plot(reg_mcmc, lag.max = 100)
```



1.4

```
effectiveSize(reg_mcmc)
```

```
## Intercept       Sex      Race       Age
##  19900.00  19900.00  19140.40  20418.39
```

1.5

```
hpd <- HPDinterval(reg_mcmc)
```

```r
par(mfrow = c(2, 2))

plot(density(reg_params$Intercept), xlab = "Est", ylab = "Density", main = "Intercept")
abline(v = diag_$statistics[1, 1], lty = 2)
abline(v = hpd[1, 1])
abline(v = hpd[1, 2])

plot(density(reg_params$Sex), xlab = "Est", ylab = "Density", main = "Sex")
abline(v = diag_$statistics[2, 1], lty = 2)
abline(v = hpd[2, 1])
abline(v = hpd[2, 2])

plot(density(reg_params$Race), xlab = "Est", ylab = "Density", main = "Race")
abline(v = diag_$statistics[3, 1], lty = 2)
abline(v = hpd[3, 1])
abline(v = hpd[3, 2])

plot(density(reg_params$Age), xlab = "Est", ylab = "Density", main = "Age")
abline(v = diag_$statistics[4, 1], lty = 2)
abline(v = hpd[4, 1])
abline(v = hpd[4, 2])
```