Missing data analysis of 2016 Boston marathon data

Chapter 6: Case studies using hierarchical modelling

Let Y_i be the speed in mile 26 for runner i and X_{ij} be the speed for runner i in file j. We fit the linear regression model

$$Y_i = \alpha + \sum_{j=1}^{25} X_{ij}\beta_j + \epsilon_i$$

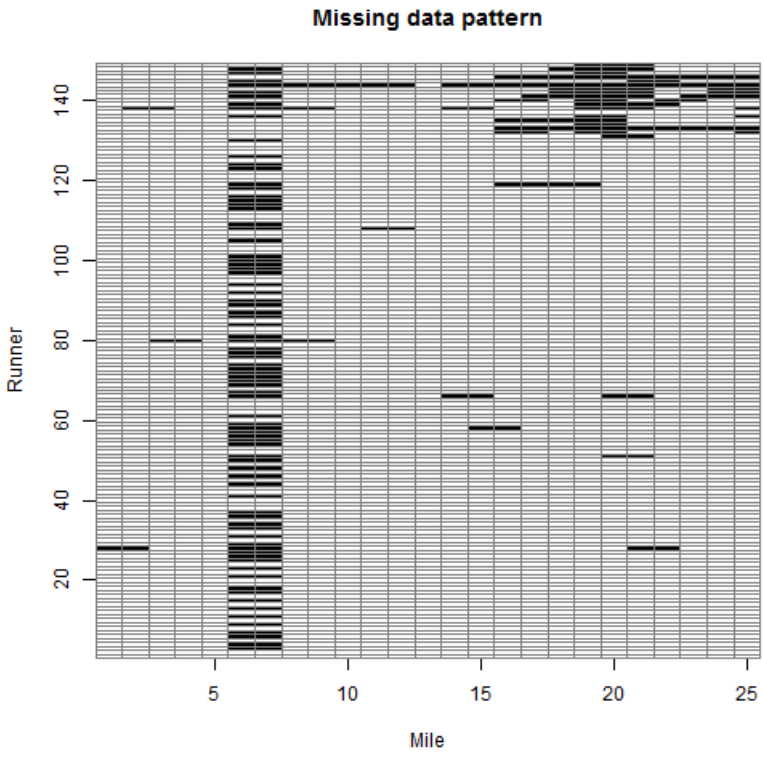
where $\epsilon_i \sim \text{Normal}(0, \sigma^2)$. In these data there are missing X_{ij}. In our Bayesian analysis we specify a first-order autoregressive prior for the missing (standardized) covariates

$$X_{i1} \sim \text{Normal}(0, \sigma_1^2) \text{ and } X_{ij+1} \,|\, X_{ij} \sim \text{Normal}\left(\rho X_{ij}, \sigma_2^2\right).$$

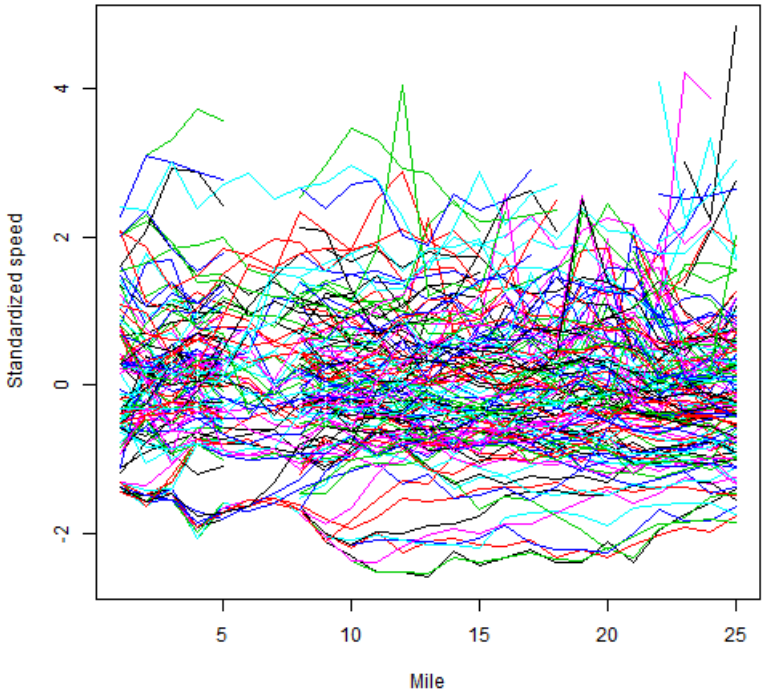
The remaining hyperparameters have uninformative priors.

Load and plot the data

```
load("BostonMarathon2016.Rdata")

Y  <- Marathon_female$SPEED_mile[,26]
X  <- Marathon_female$SPEED_mile[,-26]
X  <- scale(X)
n  <- length(Y)
p  <- ncol(X)

image(1:p,1:n,is.na(t(X)),col=0:1,
      xlab="Mile",ylab="Runner",
      main="Missing data pattern")
for(j in 0:p){
  abline(v=j+0.5,col=gray(0.5))
}
for(j in 0:n){
  abline(j+0.5,0,col=gray(0.5))
}
```



```
matplot(t(X),type="l",lty=1,xlab="Mile",ylab="Standardized speed")
```



Define the model in JAGS

```
library(rjags)
```

```
## Loading required package: coda
```

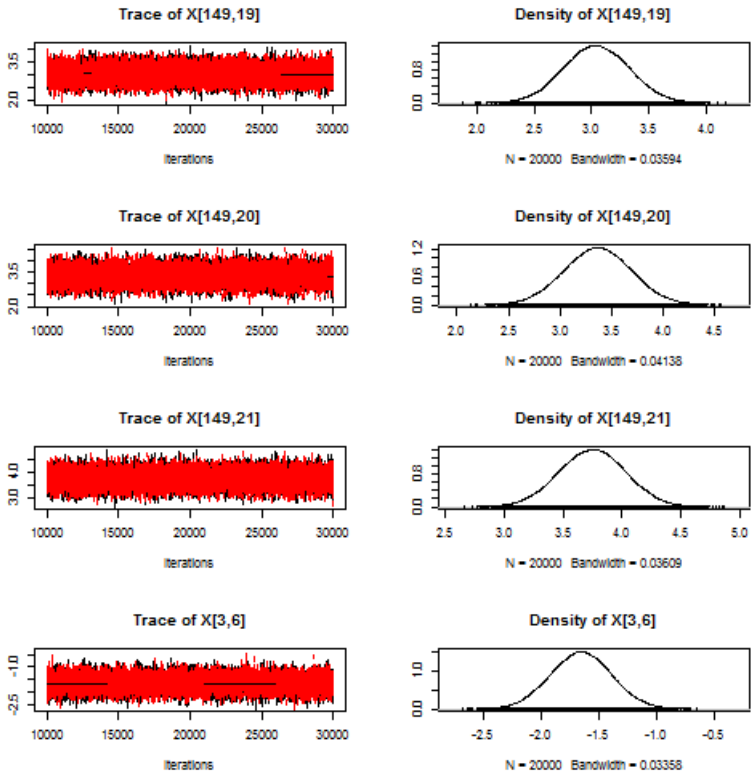
```
## Linked to JAGS 4.2.0
```

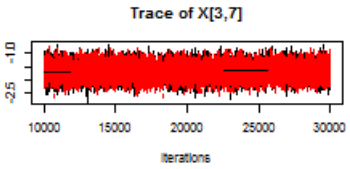
```
## Loaded modules: basemod,bugs
```

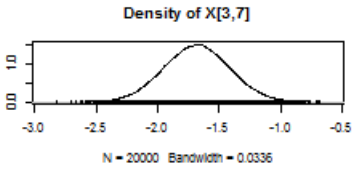
```
model_string <- textConnection("model{

  # Likelihood
   for(i in 1:n){
     Y[i] ~ dnorm(alpha + inprod(X[i,],beta[]),taue)
   }

  # Missing data model
   for(i in 1:n){
     X[i,1]  ~ dnorm(0,tau1)
     for(j in 2:p){
       X[i,j] ~ dnorm(rho*X[i,j-1],tau2)
     }
   }

  # Priors
   alpha ~ dnorm(0,0.01)
   for(j in 1:p){
     beta[j] ~ dnorm(0,0.01)
   }
   taue  ~  dgamma(0.1, 0.1)
   tau1  ~  dgamma(0.1, 0.1)
   tau2  ~  dgamma(0.1, 0.1)
   rho   ~  dnorm(0, 0.01)
}")
```

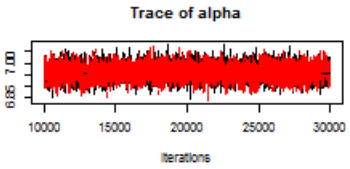
Fit the model

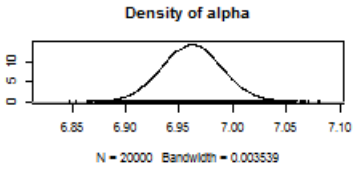
```
inits <- list(rho=rnorm(1))
data  <- list(Y=Y,X=X,n=n,p=p)
model <- jags.model(model_string,data = data, quiet=TRUE,inits=inits, n.chains=2)

update(model, 10000, progress.bar="none")
params  <- c("alpha","beta","X[3,6]","X[3,7]",
             "X[149,19]","X[149,20]","X[149,21]","rho")
samples <- coda.samples(model,
           variable.names=params,
           n.iter=20000, progress.bar="none")

plot(samples)
```

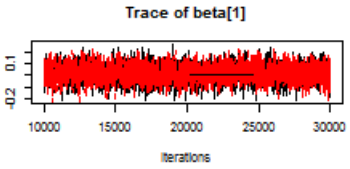


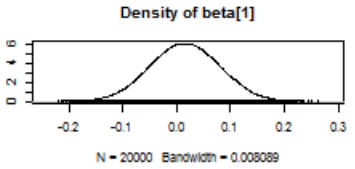

Trace of X[3,7]
-1.0
-2.5
10000
15000
20000
25000
30000
Iterations

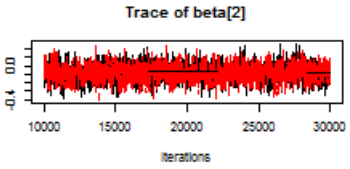

Density of X[3,7]
1.0
0.0
-3.0
-2.5
-2.0
-1.5
-1.0
-0.5
N = 20000 Bandwidth = 0.0336

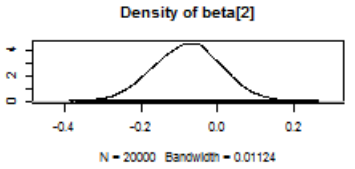

Trace of alpha
7.00
6.85
10000
15000
20000
25000
30000
Iterations

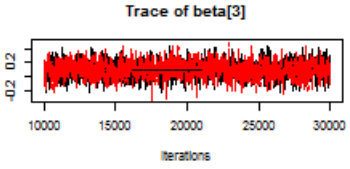

Density of alpha
10
5
0
6.85
6.90
6.95
7.00
7.05
7.10
N = 20000 Bandwidth = 0.003539

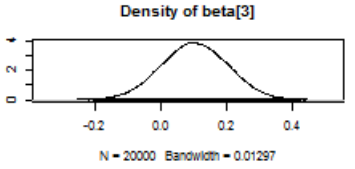

Trace of beta[1]
0.1
-0.2
10000
15000
20000
25000
30000
Iterations

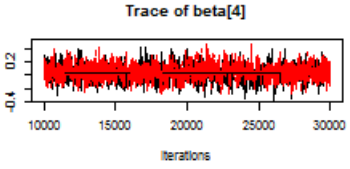

Density of beta[1]
6
4
2
0
-0.2
-0.1
0.0
0.1
0.2
0.3
N = 20000 Bandwidth = 0.008089

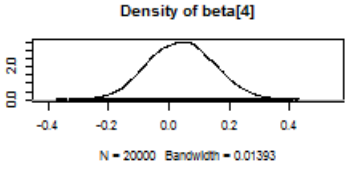

Trace of beta[2]
0.0
-0.4
10000
15000
20000
25000
30000
Iterations

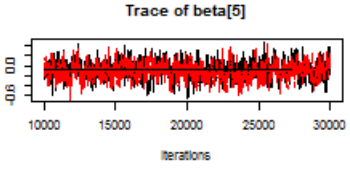

Density of beta[2]
4
2
0
-0.4
-0.2
0.0
0.2
N = 20000 Bandwidth = 0.01124

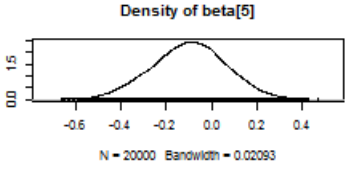

Trace of beta[3]
0.2
-0.2
10000
15000
20000
25000
30000
Iterations

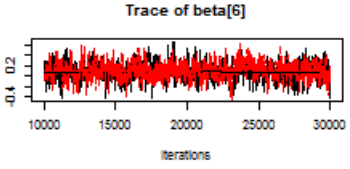

Density of beta[3]
4
2
0
-0.2
0.0
0.2
0.4
N = 20000 Bandwidth = 0.01297

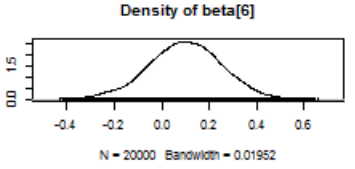

Trace of beta[4]
0.2
-0.4
10000
15000
20000
25000
30000
Iterations

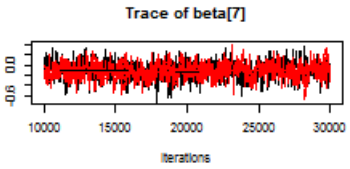

Density of beta[4]
2.0
0.0
-0.4
-0.2
0.0
0.2
0.4
N = 20000 Bandwidth = 0.01393

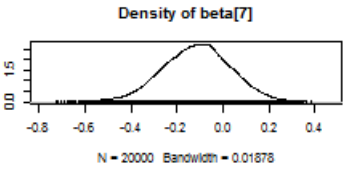

Trace of beta[5]
0.0
-0.6
10000
15000
20000
25000
30000
Iterations

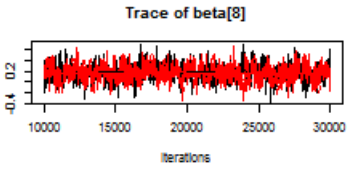

Density of beta[5]
1.5
0.0
-0.6
-0.4
-0.2
0.0
0.2
0.4
N = 20000 Bandwidth = 0.02093

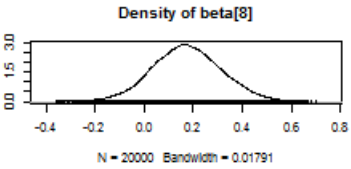

Trace of beta[6]
0.2
-0.4
10000
15000
20000
25000
30000
Iterations

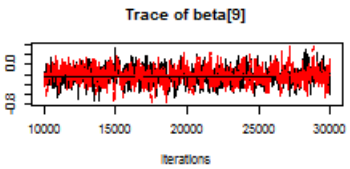

Density of beta[6]
1.5
0.0
-0.4
-0.2
0.0
0.2
0.4
0.6
N = 20000 Bandwidth = 0.01952

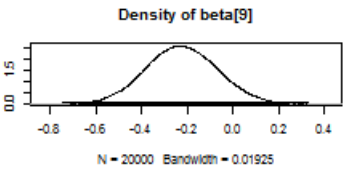

Trace of beta[7]
Iterations

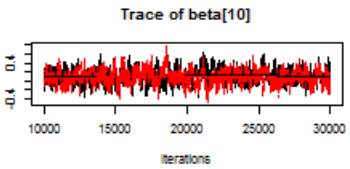

Density of beta[7]
N = 20000 Bandwidth = 0.01878

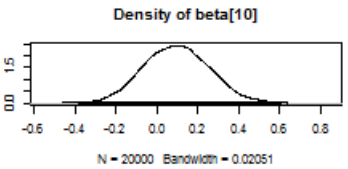

Trace of beta[8]
Iterations

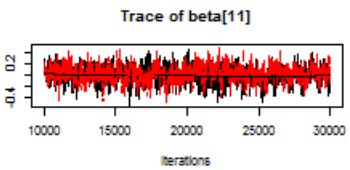

Density of beta[8]
N = 20000 Bandwidth = 0.01791

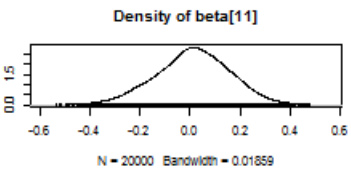

Trace of beta[9]
Iterations

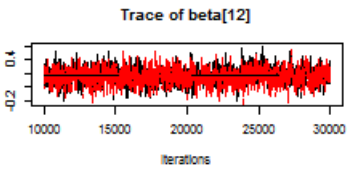

Density of beta[9]
N = 20000 Bandwidth = 0.01925

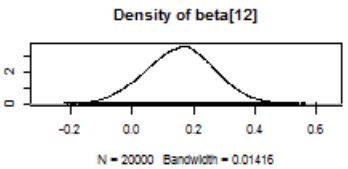

Trace of beta[10]
Iterations

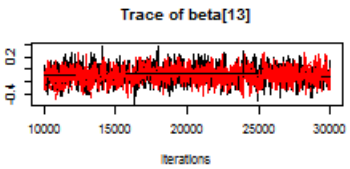

Density of beta[10]
N = 20000 Bandwidth = 0.02051

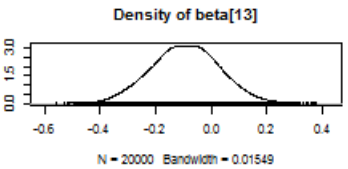

Trace of beta[11]
Iterations

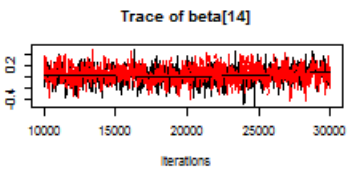

Density of beta[11]
N = 20000 Bandwidth = 0.01859

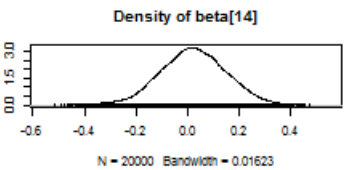

Trace of beta[12]
Iterations

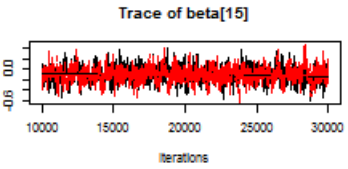

Density of beta[12]
N = 20000 Bandwidth = 0.01416

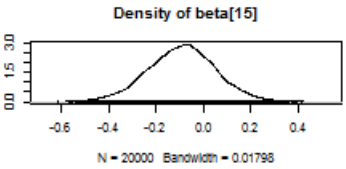

Trace of beta[13]
Iterations

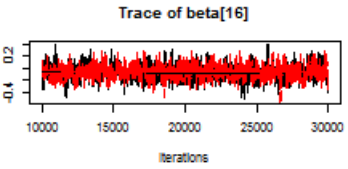

Density of beta[13]
N = 20000 Bandwidth = 0.01549

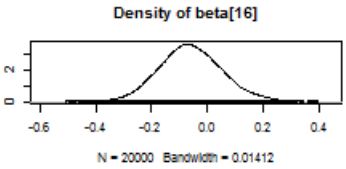

Trace of beta[14]
Iterations

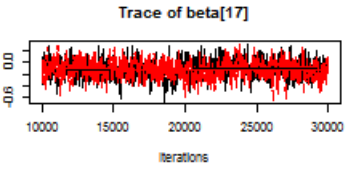

Density of beta[14]
N = 20000 Bandwidth = 0.01623

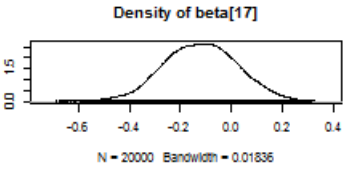

Trace of beta[15]
Iterations

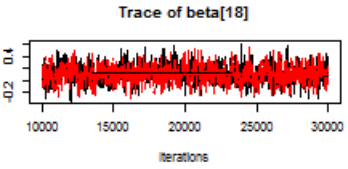

Density of beta[15]
N = 20000 Bandwidth = 0.01798

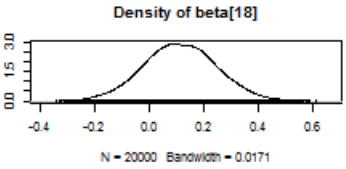

Trace of beta[16]
Iterations

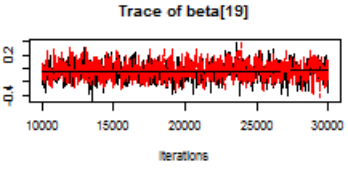

Density of beta[16]
N = 20000 Bandwidth = 0.01412

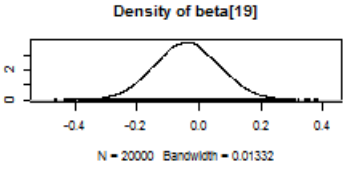

Trace of beta[17]
Iterations

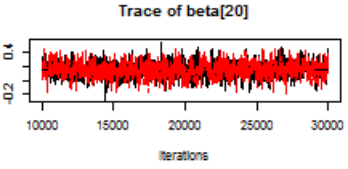

Density of beta[17]
N = 20000 Bandwidth = 0.01836

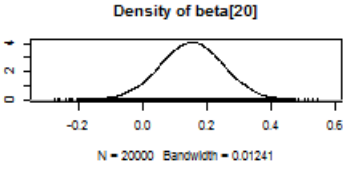

Trace of beta[18]
Iterations

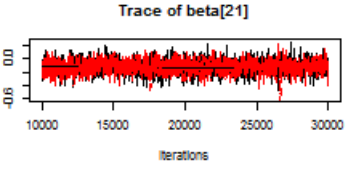

Density of beta[18]
N = 20000 Bandwidth = 0.0171

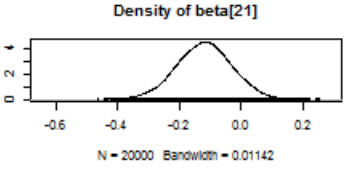

Trace of beta[19]
Iterations

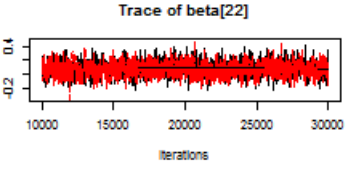

Density of beta[19]
N = 20000 Bandwidth = 0.01332

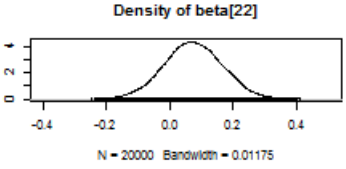

Trace of beta[20]
Iterations

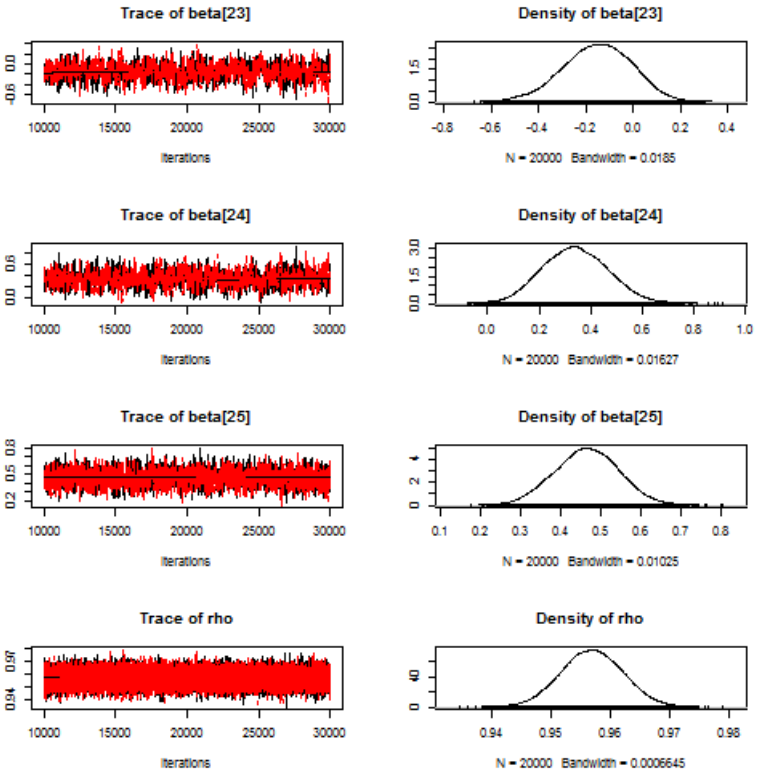

Density of beta[20]
N = 20000 Bandwidth = 0.01241


Trace of beta[21]
Iterations


Density of beta[21]
N = 20000 Bandwidth = 0.01142

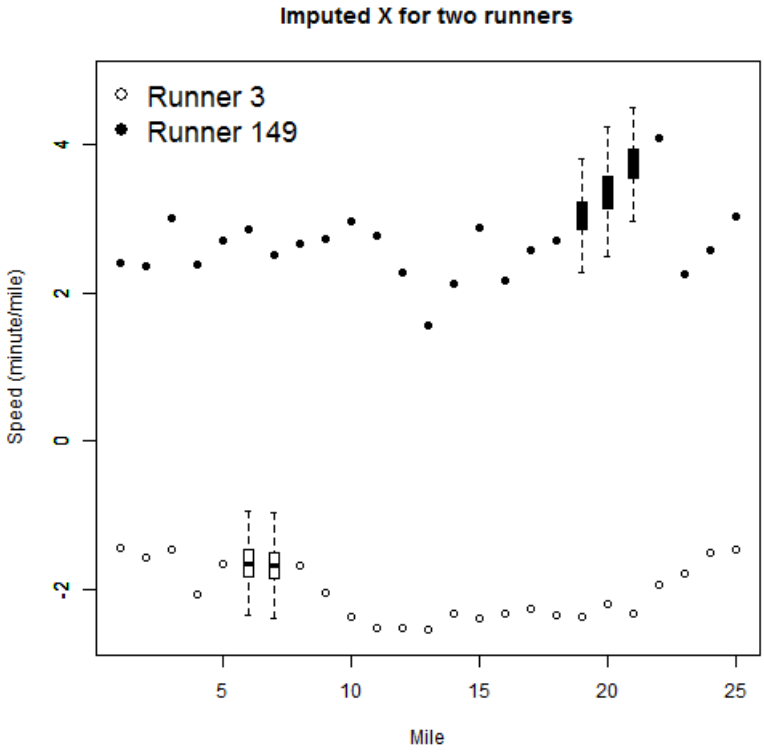

Trace of beta[22]
Iterations


Density of beta[22]
N = 20000 Bandwidth = 0.01175



Plot the imputed covariates for two runners

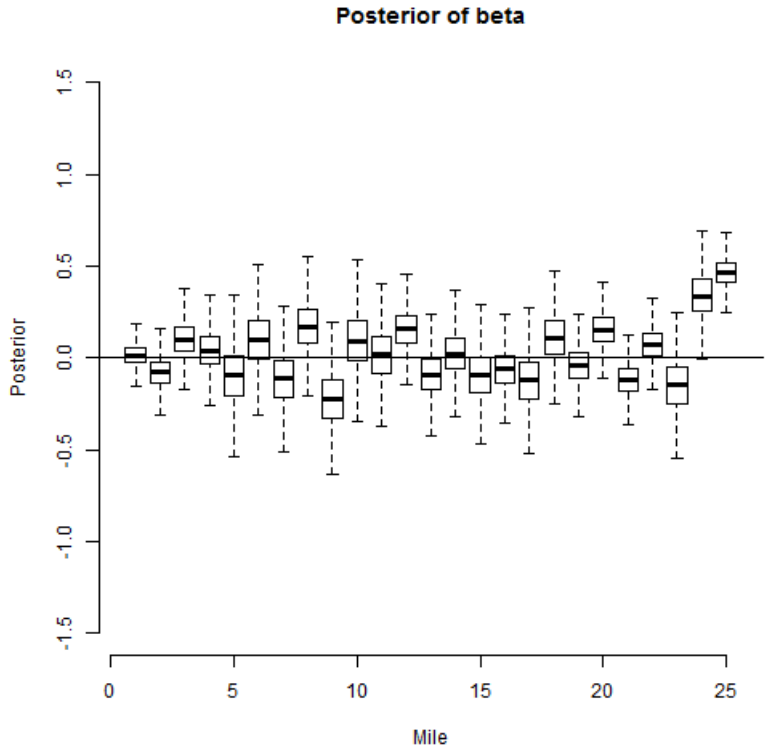
```
X_miss <- rbind(samples[[1]],samples[[2]])[,1:5]
r      <- range(na.omit(as.vector(X)))

# Runner 3
plot(X[3,],ylim=r,
           xlab="Mile",ylab="Speed (minute/mile)",
           main="Imputed X for two runners")
boxplot(X_miss[,4],at=6,outline=FALSE,add=TRUE)
boxplot(X_miss[,5],at=7,outline=FALSE,add=TRUE)

# Runner 149
points(X[149,],pch=19)
boxplot(X_miss[,1],at=19,col=1,outline=FALSE,add=TRUE)
boxplot(X_miss[,2],at=20,col=1,outline=FALSE,add=TRUE)
boxplot(X_miss[,3],at=21,col=1,outline=FALSE,add=TRUE)

legend("topleft",c("Runner 3","Runner 149"),pch=c(1,19),bty="n",cex=1.5)
```



Summary: The posterior of the missing X_{ij} is similar to adjacent speeds.

Plot the posterior of beta

```
beta <- rbind(samples[[1]],samples[[2]])[,1:p+6]

boxplot(beta,xlab="Mile",ylab="Posterior",
        main="Posterior of beta",
        ylim=1.5*c(-1,1),outline=FALSE,axes=FALSE)
axis(1)
axis(2)

abline(0,0)
```



Summary: Only the speed for miles 24 and 25 are significant predictors of the speed in the final mile.

Loading [MathJax]/jax/output/HTML-CSS/jax.js