225 homework 2

Jing Liao

Probelm 1

Provide the posterior distributions of $\sigma 1$ and $\sigma 2$, as well as the regression parameters related to the clinical variables.

Suppose $Y_i \sim Bern(1, q_i)$ represent the patient i will get the stroke or not. The model we built is a logistic regression model with hyperprior.

```
library(R2jags)
```

```
## Loading required package: rjags
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
##
## Attaching package: 'R2jags'
## The following object is masked from 'package:coda':
##
##     traceplot
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.7.1
## - Online documentation and vignettes at mc-stan.org/bayesplot
## - bayesplot theme set to bayesplot::theme_default()
##    * Does _not_ affect other ggplot2 plots
##    * See ?bayesplot_theme_set for details on theme setting
```

```
stroke=read.table("/Users/jing/Desktop/2020 winter/Stat 225/hw2/Stroke.csv",sep=',',header=T)
stroke$Gender<-as.numeric(stroke$Gender)-1
attach(stroke)
n = dim(stroke)[1];
dim2 = dim(stroke)[2]
Y=stroke[,16]
x = as.matrix(stroke[,2:(dim2-1)])
x=scale(x[,2:14])
x = cbind(stroke$Gender, x)
X = cbind(rep(1, n), x)
Xc = X[,1:5]


logistic_model <- "model{

   # Likelihood

   for(i in 1:n){
```

```
    Y[i] ~ dbern(q[i])
    logit(q[i]) <-beta[1]*X[i,1] + beta[2]*X[i,2] +
                   beta[3]*X[i,3] + beta[4]*X[i,4] + beta[5]*X[i,5]+
                   beta[6]*X[i,6] + beta[7]*X[i,7] + beta[8]*X[i,8]+
                   beta[9]*X[i,9] + beta[10]*X[i,10] + beta[11]*X[i,11]+
                   beta[12]*X[i,12] + beta[13]*X[i,13] + beta[14]*X[i,14]+beta[15]*X[i,15]
   }

   #Priors
   beta[1] ~dnorm(0,1/1000)
   for(j in 2:5){
    beta[j] ~ dnorm(0,prec1)
   }
   for(j in 6:15){
    beta[j] ~ dnorm(0,prec2)
   }
  prec1 ~ dgamma(0.001,0.001)
  prec2 ~ dgamma(0.001,0.001)
   sigma.sq1 <- 1/prec1
   sigma.sq2 <- 1/prec2

  }"
dat<- list(Y=Y,n=n,X=X)
jags.param=c("beta","sigma.sq1",'sigma.sq2')
fit <- jags(data=dat, n.chains=5, inits=NULL,parameters=jags.param, n.iter=3000,
               n.burnin=1000, DIC=TRUE, model.file=textConnection(logistic_model))
```

```
## module glm loaded
```

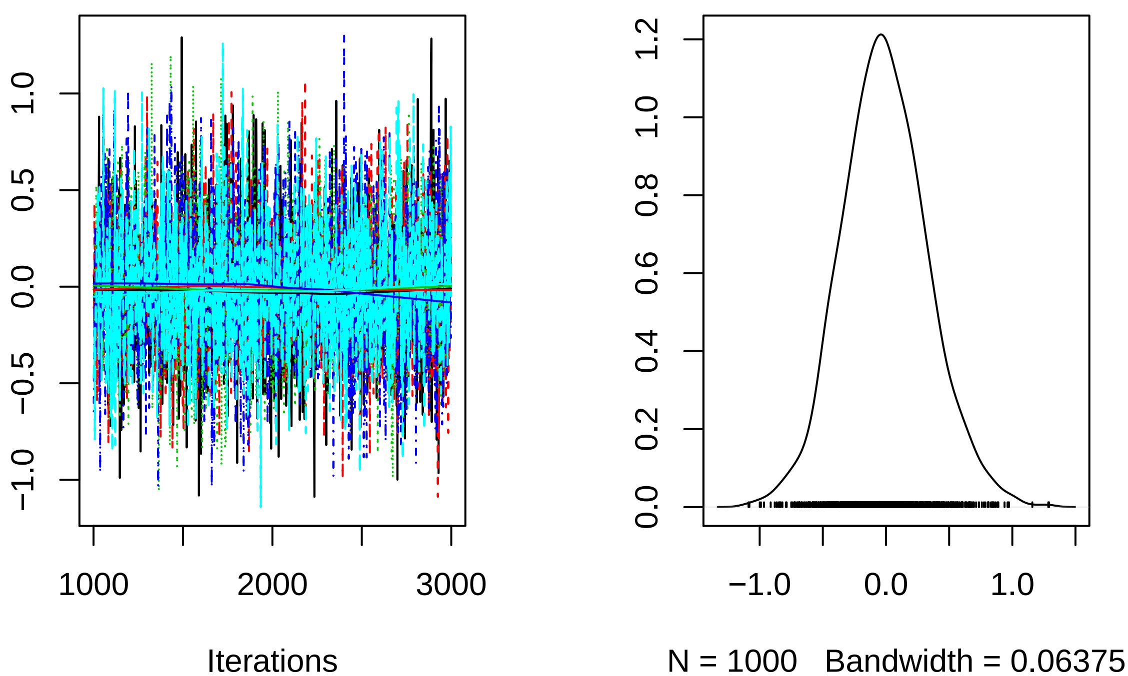
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 100
##    Unobserved stochastic nodes: 17
##    Total graph size: 2878
##
## Initializing model
```

```
print(fit, intervals=c(0.025, 0.975))
```

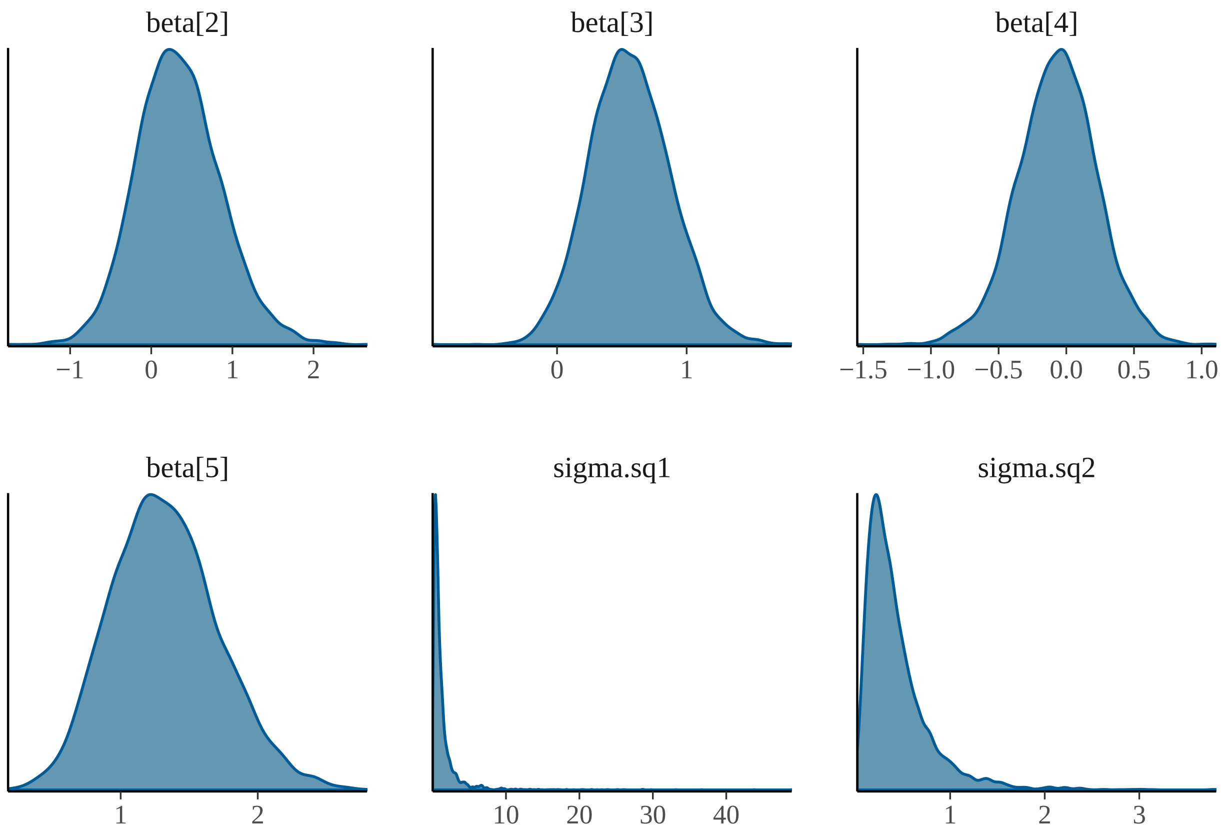
```
## Inference for Bugs model at "4", fit using jags,
##  5 chains, each with 3000 iterations (first 1000 discarded), n.thin = 2
##  n.sims = 5000 iterations saved
##          mu.vect sd.vect   2.5%  97.5%  Rhat n.eff
## beta[1]    -0.761   0.414 -1.647 -0.025 1.001  5000
## beta[2]     0.343   0.520 -0.608  1.449 1.002  2900
## beta[3]     0.564   0.313 -0.020  1.187 1.002  1600
## beta[4]    -0.067   0.294 -0.667  0.509 1.001  4000
## beta[5]     1.325   0.392  0.632  2.149 1.003  1100
## beta[6]    -0.067   0.344 -0.743  0.600 1.002  2800
## beta[7]     0.259   0.380 -0.432  1.065 1.001  5000
## beta[8]    -0.229   0.340 -0.904  0.442 1.002  3000
## beta[9]    -0.051   0.339 -0.723  0.620 1.001  5000
## beta[10]   -0.007   0.340 -0.660  0.695 1.001  5000
```

```
## beta[11]   -1.053   0.438 -2.001 -0.289 1.005   730
## beta[12]   -0.151   0.334 -0.797  0.509 1.002  2800
## beta[13]    0.054   0.369 -0.641  0.811 1.001  5000
## beta[14]   -0.178   0.315 -0.822  0.437 1.001  5000
## beta[15]    0.977   0.350  0.357  1.726 1.001  3100
## sigma.sq1   1.379   2.471  0.129  6.598 1.001  3800
## sigma.sq2   0.438   0.355  0.076  1.395 1.003  1500
## deviance   84.607   5.568 75.743 97.241 1.001  3300
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 15.5 and DIC = 100.1
## DIC is an estimate of expected predictive error (lower deviance is better).
```

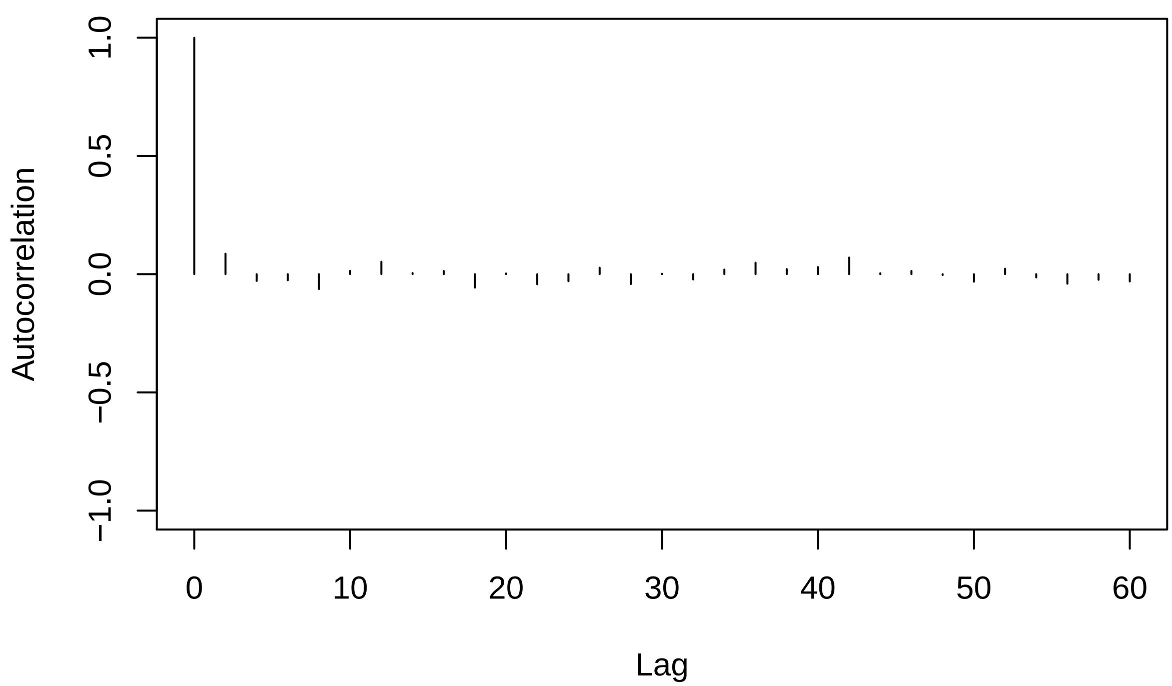
```r
library(bayesplot)
fit.mcmc <- as.mcmc(fit)
plot(fit.mcmc[,2])
```

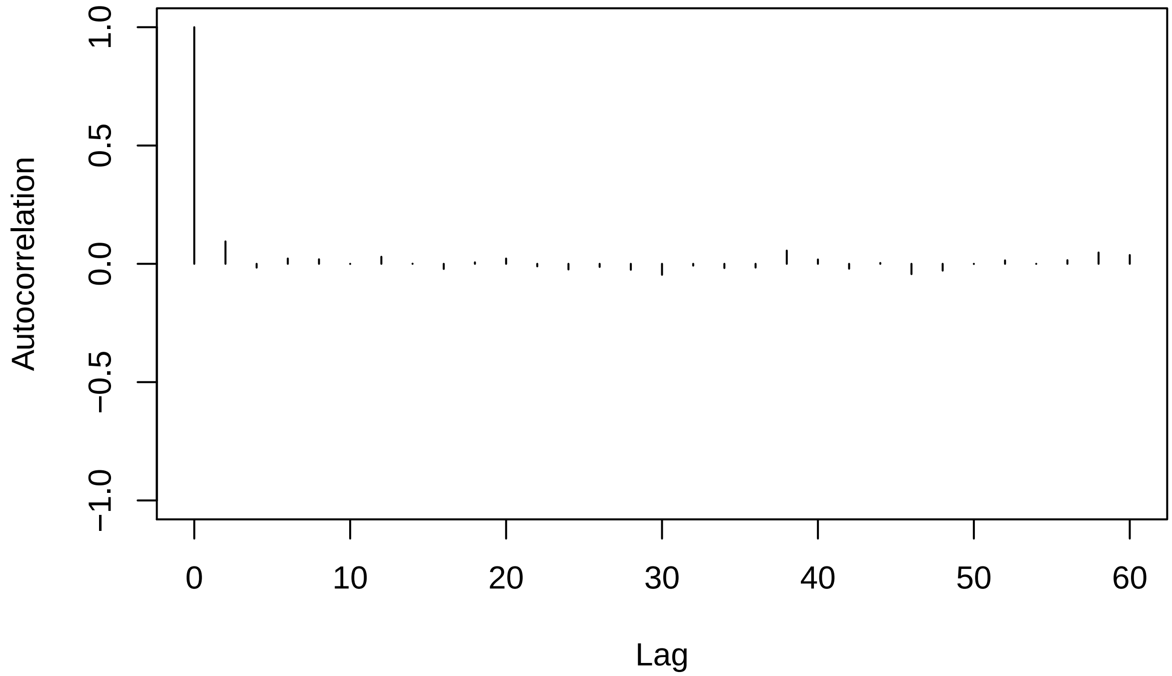


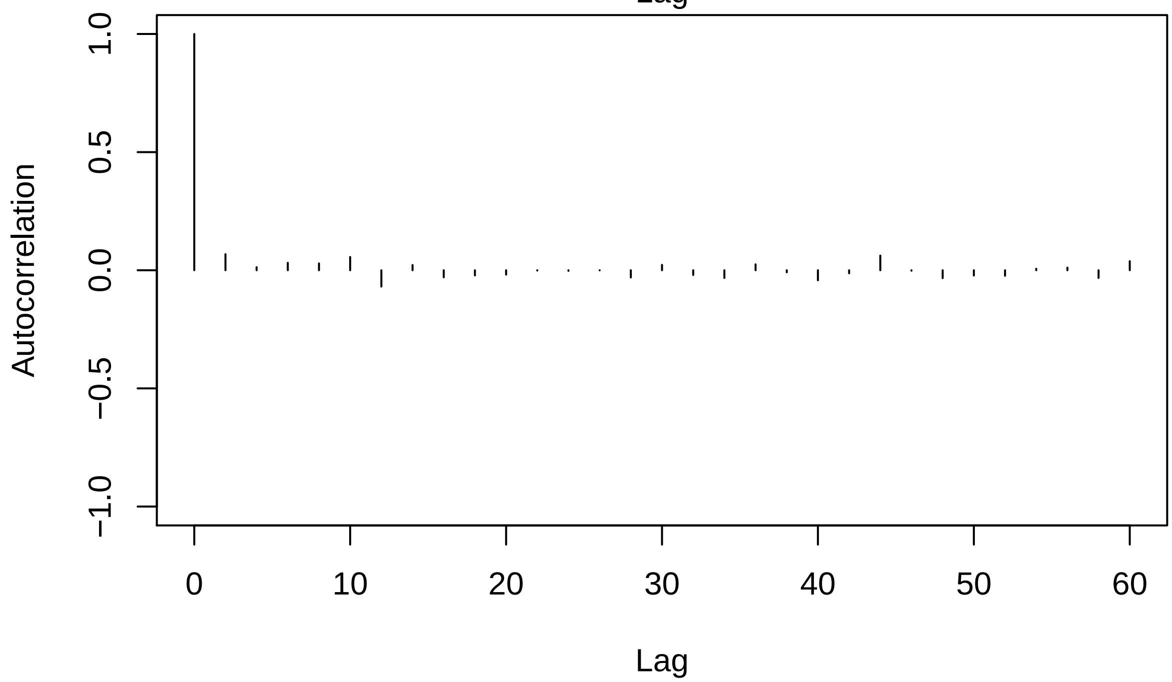

```r
mcmc_dens(fit.mcmc, pars = c('beta[2]','beta[3]','beta[4]','beta[5]',"sigma.sq1", "sigma.sq2"))
```

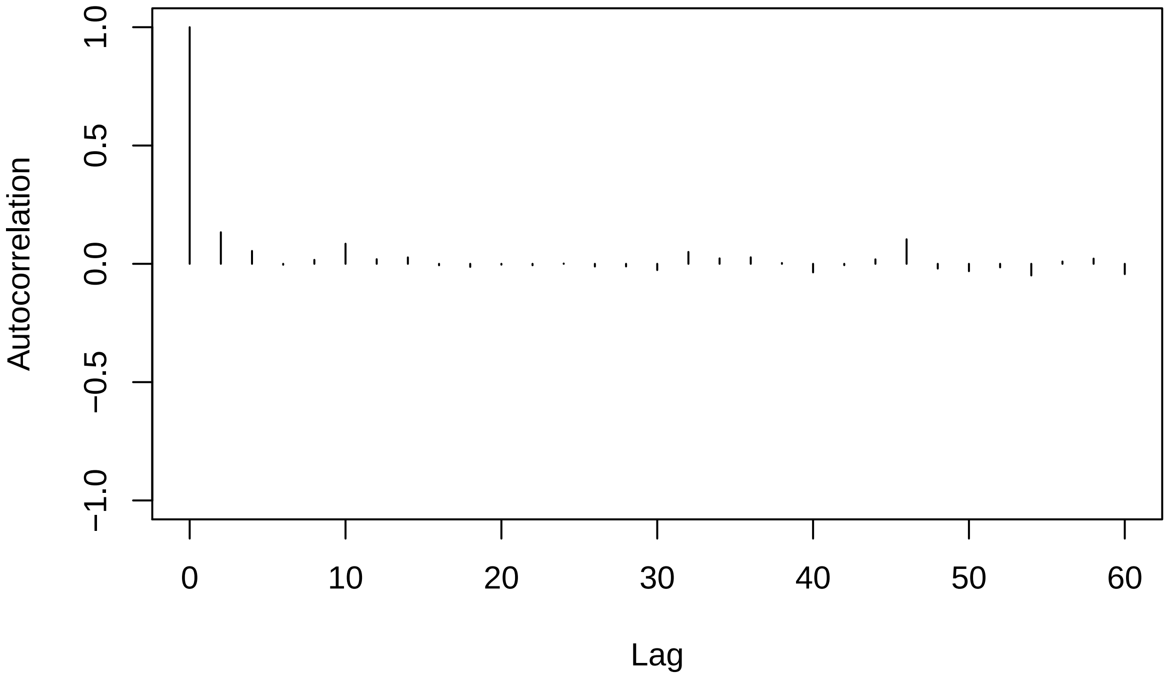


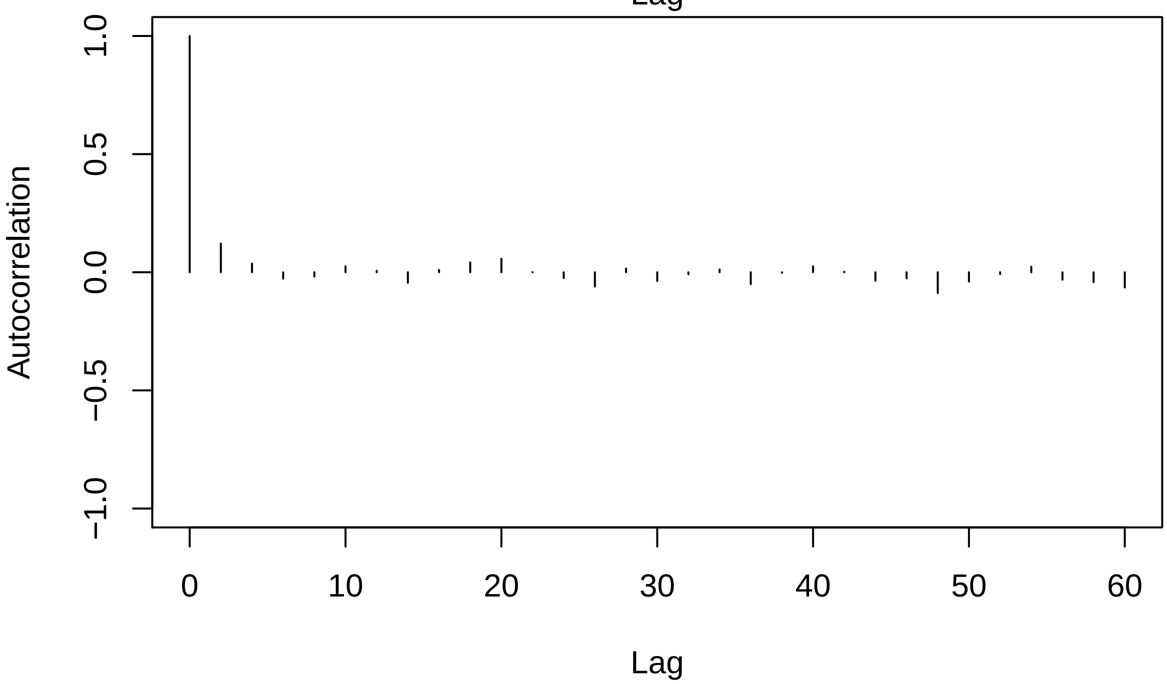

```
autocorr.plot(fit.mcmc[,2])
```




Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
10
20
30
40
50
60
Lag


Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
10
20
30
40
50
60
Lag






```
fit$BUGSoutput$DIC
```

```
## [1] 100.1043
```

from the autocorr.plot, it donsen't show long range dependence, which means our model reach stable.

Probelm 2

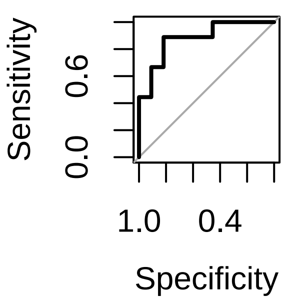
Using a 5-fold cross-validation and DIC, evaluate the performance of your model and compare it to a simpler model that uses the clinical variables only (i.e., excluding the EEG variables).

First, I seperate the whole stroke data set into five folds,then use each fold as test set and the other four folds as train set. Then we use the posterior coeffiencents we got to compute the predictive distribution and compuete the mean of predictive distribution as the estimataed probibility of patient i whether get stroke or not.

```
library(modelr)
#library(ROOC)
library(pROC)
strokenew<-cbind(X,Y)
set.seed(523)
#Randomly shuffle the data
strokecv<-stroke[sample(nrow(stroke)),]
#Create 5 equally size folds
folds <- cut(seq(1,nrow(strokecv)),breaks=5,labels=FALSE)
#Perform 5 fold cross validation

auc1 <- matrix(nrow=5,ncol=1)
DIC1 <- matrix(nrow=5,ncol=1)
  #Segement your data by fold using the which() function

#5 fold CV for full model
for (l in 1:5) {


  testIndexes <- which(folds==l,arr.ind=TRUE)
  testData <- strokenew[testIndexes, ]
  trainData <- strokenew[-testIndexes, ]
  Xr<-trainData[,1:15]
  Yr<-trainData[,16]
  Xt<-testData[,1:15]
  Yt<-testData[,16]

  logistic_model <- "model{

   # Likelihood

   for(i in 1:n){
    Y[i] ~ dbern(q[i])
    logit(q[i]) <-beta[1]*Xr[i,1] + beta[2]*Xr[i,2] +
                   beta[3]*Xr[i,3] + beta[4]*Xr[i,4] + beta[5]*Xr[i,5]+
                   beta[6]*Xr[i,6] + beta[7]*Xr[i,7] + beta[8]*Xr[i,8]+
                   beta[9]*Xr[i,9] + beta[10]*Xr[i,10] + beta[11]*Xr[i,11]+
                   beta[12]*Xr[i,12] + beta[13]*Xr[i,13] + beta[14]*Xr[i,14]+beta[15]*Xr[i,15]
   }

   #Priors
   beta[1] ~dnorm(0,1/1000)
   for(j in 2:5){
    beta[j] ~ dnorm(0,prec1)
   }
   for(j in 6:15){
    beta[j] ~ dnorm(0,prec2)
   }
  prec1 ~ dgamma(0.001,0.001)
  prec2 ~ dgamma(0.001,0.001)
```

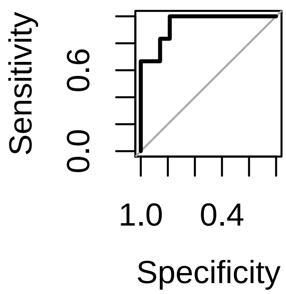
```
   sigma.sq1 <- 1/prec1
   sigma.sq2 <- 1/prec2

   #prediction
 for(k in 1:K) {

  Phat[k] <- 1/(1+exp(-(beta[1]*Xt[k,1] + beta[2]*Xt[k,2] +
    beta[3]*Xt[k,3] + beta[4]*Xt[k,4] + beta[5]*Xt[k,5]+
    beta[6]*Xt[k,6] + beta[7]*Xt[k,7] + beta[8]*Xt[k,8]+
    beta[9]*Xt[k,9] + beta[10]*Xt[k,10] + beta[11]*Xt[k,11]+
    beta[12]*Xt[k,12] + beta[13]*Xt[k,13] + beta[14]*Xt[k,14]+beta[15]*Xt[k,15])))

  }


  }"
dat<- list(Y=Yr,n=80,Xr=Xr,Xt=Xt,K=20)
jags.param=c("beta","Phat")
fit <- jags(data=dat, n.chains=5, inits=NULL,parameters=jags.param, n.iter=3000,
            n.burnin=1000, DIC=TRUE, model.file=textConnection(logistic_model))


ptest<-fit$BUGSoutput$mean$Phat
auc1[l,]<-auc(Yt, ptest)
print(roc(Yt, ptest,plot = T,levels=c("0", "1"), direction="<"))
DIC1[l,]<-fit$BUGSoutput$DIC

}
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 17
##    Total graph size: 2919
##
## Initializing model
```

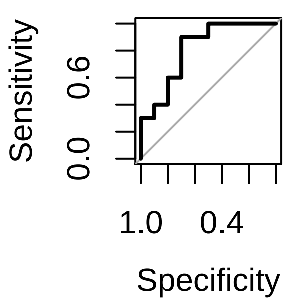



```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 11 controls (Yt 0) < 9 cases (Yt 1).
## Area under the curve: 0.8788
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 17
##    Total graph size: 2919
##
## Initializing model
```

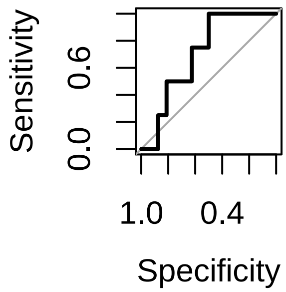



```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 14 controls (Yt 0) < 6 cases (Yt 1).
## Area under the curve: 0.9405
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 17
##    Total graph size: 2919
##
## Initializing model
```

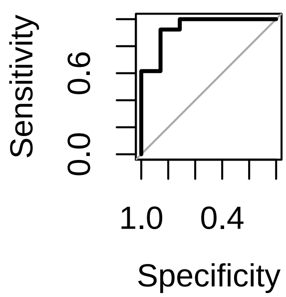



```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 10 controls (Yt 0) < 10 cases (Yt 1).
## Area under the curve: 0.81
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 17
##    Total graph size: 2919
##
## Initializing model
```




```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 16 controls (Yt 0) < 4 cases (Yt 1).
## Area under the curve: 0.7031
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 17
##    Total graph size: 2919
##
## Initializing model
```




```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 7 controls (Yt 0) < 13 cases (Yt 1).
## Area under the curve: 0.9341
```

```r
print(mean(auc1))
```

```
## [1] 0.853291
```

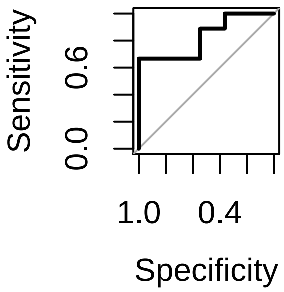
```r
print(mean(DIC1))
```

```
## [1] 83.73847
```

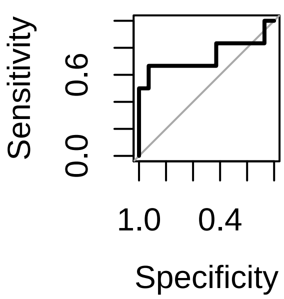
```
# 5 folds cv for simple model
auc2 <- matrix(nrow=5,ncol=1)
DIC2 <- matrix(nrow=5,ncol=1)
for (l in 1:5) {

  testIndexes <- which(folds==l,arr.ind=TRUE)
  testData <- strokenew[testIndexes, ]
  trainData <- strokenew[-testIndexes, ]
  Xr<-trainData[,1:5]
  Yr<-trainData[,16]
  Xt<-testData[,1:5]
  Yt<-testData[,16]

  logistic_model_c <- "model{

   # Likelihood

   for(i in 1:n){
    Y[i] ~ dbern(q[i])
    logit(q[i]) <-beta[1]*Xr[i,1] + beta[2]*Xr[i,2] +
                   beta[3]*Xr[i,3] + beta[4]*Xr[i,4]+beta[5]*Xr[i,5]
   }

   #Priors
   beta[1] ~dnorm(0,1/1000)
   for(j in 2:5){
    beta[j] ~ dnorm(0,prec)
   }
   prec ~ dgamma(0.001,0.001)
    sigma.sq <- 1/prec

   #prediction
 for(k in 1:K) {

  Phat[k] <- 1/(1+exp(-(beta[1]*Xt[k,1] + beta[2]*Xt[k,2] +
    beta[3]*Xt[k,3] + beta[4]*Xt[k,4] + beta[5]*Xt[k,5])))

  }

  }"
  datc<- list(Y=Yr,n=80,Xr=Xr,Xt=Xt,K=20)
  jags.paramc=c("beta","Phat")
  fitc <- jags(data=datc, n.chains=5, inits=NULL,parameters=jags.paramc, n.iter=3000,
             n.burnin=1000, DIC=TRUE, model.file=textConnection(logistic_model_c))
  ptest<-fitc$BUGSoutput$mean$Phat
  auc2[l,]<-auc(Yt, ptest)
  print(roc(Yt, ptest,plot = T,levels=c("0", "1"), direction="<"))
  DIC2[l,]<-fitc$BUGSoutput$DIC
}
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
```

```
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 6
##    Total graph size: 906
##
## Initializing model
```

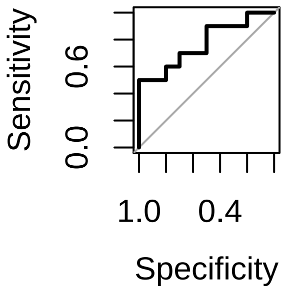



```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 11 controls (Yt 0) < 9 cases (Yt 1).
## Area under the curve: 0.8283
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 6
##    Total graph size: 906
##
## Initializing model
```

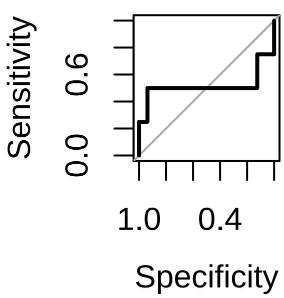



```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 14 controls (Yt 0) < 6 cases (Yt 1).
## Area under the curve: 0.7381
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 6
##    Total graph size: 905
```

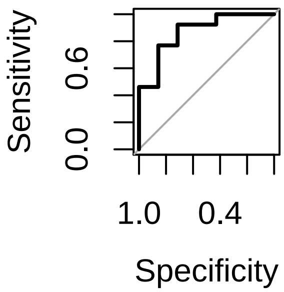
```
##
## Initializing model
```




```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 10 controls (Yt 0) < 10 cases (Yt 1).
## Area under the curve: 0.77
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 6
##    Total graph size: 905
##
## Initializing model
```




```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 16 controls (Yt 0) < 4 cases (Yt 1).
## Area under the curve: 0.5156
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 80
##    Unobserved stochastic nodes: 6
##    Total graph size: 905
##
## Initializing model
```




```
##
## Call:
## roc.default(response = Yt, predictor = ptest, levels = c("0",     "1"), direction = "<", plot = T)
##
## Data: ptest in 7 controls (Yt 0) < 13 cases (Yt 1).
## Area under the curve: 0.8681
```

```
print(mean(auc2))
```

```
## [1] 0.744027
```

```
print(mean(DIC2))
```

```
## [1] 101.7706
```

The mean auc of 5-fold cross validation in my model is 0.85, the mean DIC is 94.21, the mean auc of 5-fold cross validation in simple model is 0.74, the mean DIC is 101.77.