Project on Bayesian analysis on heart-diease data

2023-11-02

Introduction

The dataset is a heart disease dataset from the UC Irvine Machine Learning Repository1, one of the most popular datasets available. I chose this dataset primarily because of its prominent position on the website, which piqued my interest. Furthermore, the dataset exhibits a small number of instances and a moderate volume of variables, making it an ideal candidate for Bayesian analysis with MCMC methods, which may not take too long to converge.

Additionally, one response variable in the dataset, named "heart-disease," can be interpreted as an ordinal variable, providing severity of the disease. This makes it a valuable dataset for exploring Bayesian analysis in multiple categories classification, which is crucial in many medical applications.

The primary objectives of my project are to identify the appropriate model and significant factors for this dataset.

The original dataset is more extensive, comprising 76 attributes and numerous missing data points. However, there is a shorter version of the dataset with 14 attributes, commonly used in machine learning. I chose the brief version for its convenience, as it requires less data processing and generation.

The dataset I am working with includes 14 variables: 8 symbolic and 6 numeric. These variables encompass age, sex, chest pain type (angina, abnang, notang, asympt), Trestbps (resting blood pressure), cholesterol, fasting blood sugar (< 120, true or false), resting ECG (norm, abn, hyper), max heart rate, exercise-induced angina (true or false), old peak, slope (up, flat, down), number of vessels colored (with no detailed description), and thal (norm, fixed, revers). Finally, the classes are either "healthy" (buff) or "with heart disease" (sick).

The dataset contains 303 instances, with only 7 instances having missing data. Since the amount of missing data is relatively small, we can either gernate missing values or remove the instances containing missing data1.

Methodology

Data processing

Standardize the numerical data.Since the dataset seems to have some ordinal covariates , I transformed them into a numerical covariates to implement the following model without standardizing them. ## Logstic regression model First let us see how a fundamental model,logistic regression model, performs on a general classification question whether someone is healhy or not. The logistic model can be described as below: The probability of being sick or healthy depends on the other 13 variables and it can be given by

$$P(y = 1|\theta) = \frac{exp(\theta)}{1 + exp(\theta)} = \frac{1}{1 + exp(-\theta)} \tag{1}$$

where θ is given by

$$\theta = X(\beta\gamma) \tag{2}$$

where

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$$

$$\beta\gamma = \begin{bmatrix} \beta_0 \\ \beta_1\gamma_1 \\ \beta_2\gamma_2 \\ \vdots \\ \beta_p\gamma_p \end{bmatrix}$$

where $\beta_i, (0 \leq i \leq p)$ has a prior Normal distribution

$$\beta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

and $\gamma_i, (1 \leq i \leq p)$ follows a prior Bernoulli distribution

$$\gamma_i \sim \text{Bernoulli}(p_i)$$

the prior distrubution of β can be weakly informative with large spread σ^2 if we have some good piror information.Furthermore,$\mu = 0$ is usually applied in many case which may means we assume that weak prior information and correlation as it is also a good beginning to train a mcmc process. γ is used to select model covariates so we may assume $p = 0.5$.

In this case,I assume that $\sigma_0 = 2$ and other $sigma_i = 1$.

Now we can derive the post distribution which is always proportionate to joint distribution and if we want to implement a Metropolis Hasting Algorithm , we usually use ratio of the distribution which mean the proportionality coefficient does not influence the results.So we just need to derive our joint distribution density .

$$\begin{aligned} P(Y, \theta, X, \beta, \gamma) &= P(Y|\theta, X, \beta, \gamma)P(\theta|X, \beta, \gamma)P(\beta)P(\gamma) \\ &= P(Y|\theta, X, \beta, \gamma)P(\beta)P(\gamma) \\ &= \prod_{i=1}^{n} P(y_i|X_i = [1, x_{i,1}, x_{i,2}, \ldots, x_{i,p}]; \beta)P(\beta)P(\gamma) \\ &\propto \prod_{i=1}^{n} \left(\frac{1}{1+\exp(-X_i\beta)}\right)^{y_i} \left(1 - \frac{1}{1+\exp(-X_i\beta)}\right)^{1-y_i} \prod_{j=0}^{p} \text{Norm}(\beta_j|0, \sigma_j^2) \end{aligned}$$

so we can derive our log form of density function or log likelihood function given by:

$$\mathcal{L}(\beta, \gamma, \theta; Y, X) = \sum_{i=1}^{n} \left[y_i \log\left(\frac{1}{1+\exp(-X_i\beta)}\right) + (1-y_i)\log\left(1 - \frac{1}{1+\exp(-X_i\beta)}\right)\right] + \sum_{j=0}^{p} \log(\text{Norm}(\beta_j|0, \sigma_j^2)) \tag{3}$$

the acceptance ration r of Metropolis Hasting Algorithm could be derived. The general form of $log(r)$ is when every time we just consider updating a cluster of parameters denote α as eitherβ or γ.

$$\log(r) = \mathcal{L}(\alpha^*) - \mathcal{L}(\alpha^{(s)}) + \log[J(\alpha^{(s)}|\alpha^*)] - \log[J(\alpha^*|\alpha^{(s)})] \tag{4}$$

To implement a MCMC method quickly and easily, we will use both Gibbs sampler and Metropolis Hasting Algorithm.

We can first update γ.We can either update the cluster of whole γ or update each γ_i the poster distribution of γ is a multinomial distribution which is a little bit complex while γ_i is can be easily dirived by (3) and (4)

since it follows a Bernoulli distribution only be either 1 or 0 .Furthermore ,the ratio r can be interpreted as odd ratio. So postier distribution is:

$$P(\gamma_i = 1) = logitic[log(odds)] = logistic[log(r)]$$

Then we need to update the whole cluster of β based on a multivariate normal distribution

$$J(\beta^*|\beta^{(s)}) = MVNorm(\beta^*|\beta^{(s)}, \nu\Sigma)$$

(in numerical case I ues $\nu = 0.08$ and Σ generated from glm model coefficients matrix which may suggest a high informative) since multivariate normal distribution is symmetric ,so the log of ratio can be reduced as:

$$log(r) = \mathcal{L}(\beta^*; \gamma^{(s+1)}, \theta, Y, X) - \mathcal{L}(\beta^{(s)}; \gamma^{(s+1)}, \theta, Y, X)$$

Then we need to sample a random variable following $rv \sim U(0, 1)$ and test $rv < log(r)$ .If true,we update $\beta^{(s+1)}$ as β^*.If false, remain $\beta^{(s+1)}$ as $\beta^{(s)}$. Finally ,we iterate the above updating processes for more than thousand times.

Ordinal regression model

Let's then take a look at the ordinal regression model.Ordinal regression model need ordinal response variable which have q levels.The model is displayed below:

$$\frac{P(y \leq k|\theta_1, \theta_2, \ldots, \theta_{q-1})}{P(y > k|\theta_1, \theta_2, \ldots, \theta_{q-1})} = \exp(\theta_k) \tag{5}$$

$$\theta_k = X(\beta_k\gamma_k) \tag{6}$$

where

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$$

$$\beta_k\gamma_k = \begin{bmatrix} \beta_{0,k} \\ -\beta_1\gamma_1 \\ -\beta_2\gamma_2 \\ \vdots \\ -\beta_p\gamma_p \end{bmatrix}$$

In this model,although the vector $\beta_k\gamma_k$ is different , $\gamma_i(1 \leq i \leq p)$ follows the same prior Bernoulli distribution

$$\gamma_i \sim \text{Bernoulli}(p_i)$$

(in the numerical case I assume all $p_i = 0.5$) and meanwhile $\beta_i(1 \leq i \leq p)$ has a prior Normal distribution

$$\beta_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \text{ for i=1,2...,p}$$

(in the numerical case I assume all $\mu_i = 0$ and all $\sigma_i = 1$) For $\beta_{0,k}(0 \leq k \leq q-1)$,it is a little bit complicated

$$\beta_{0,1} \sim \mathcal{N}(\mu_{0,1}, \sigma_{0,1}^2)$$

$$P(\beta_{0,k}|\beta_{0,k-1}) = \text{Norm}(\beta_{0,k}|\beta_{0,k} > \beta_{0,k-1}; \mu_{0,k}, \sigma_{0,k}^2) \text{ for k=2,..q-1}$$

(In the numerical case , I assume all $\mu_{0,k} = 0$ while all $\sigma_{0,k} = 20$ which is a really weak piror) we can view $\beta_{0,k}$ as individual independent variable first but have constrain

$$\beta_{0,k} > \beta_{0,k-1} \text{for k=1,2...q-1} \tag{7}$$

to satisfy the ordinal regression model.So the prior distribution of the model is a truncated distribution and in this case we assume it follows a truncated normal distribution.

$$\begin{aligned} P(Y,\theta,X,\beta,\gamma) &= P(Y|\theta,X,\beta,\gamma)P(\theta|X,\beta,\gamma)P(\beta)P(\gamma) \\ &= P(Y|\theta,X,\beta,\gamma)P(\beta)P(\gamma) \\ &\propto \prod_{i=1}^{n} P(y_i|X_i=[1,x_{i,1},x_{i,2},\ldots,x_{i,p}];\beta)\prod_{j=1}^{p}P(\beta_j)\prod_{k=1}^{q-1}P(\beta_{0,k}) \end{aligned}$$

so we can derive our log form of density function or log likelihood function given by:

$$\mathcal{L}(\beta,\gamma,\theta;Y,X)=\sum_{i=1}^{n}[P(y_i|X_i=[1,x_{i,1},x_{i,2},\ldots,x_{i,p}];\beta)]+\sum_{j=0}^{p}log[\text{Norm}(\beta_j|0,\sigma_j^2)]+\sum_{k=2}^{q-1}log[\text{Norm}(\beta_{0,k}|\beta_{0,k}>\beta_{0,k-1};0,\sigma_j^2)] \tag{8}$$

where we make $\beta_{0,0}$ as β_0 to make the whole equation simple.

The last part of

$$\mathcal{L}(\beta,\gamma,\theta;Y,X)$$

which is

$$\sum_{k=2}^{q-1}log[\text{Norm}(\beta_{0,k}|0,\sigma_j^2,\beta_{0,k}>\beta_{0,k-1})]$$

In R we can use a package called "truncnorm" in R to caculate each term in the sum. The updating process is a little bit different from the process in logistic regression. The first step and second step is mainly the same as what we do in the logistic regression model. However, in the ordinal regression we need to do a further step which is to update the different intercepts corresponding to the different ordinaldata. In this step,we need to update the $\beta_{0,k}$ one by one ,because the model has a constrain (7).So the distribution of $\beta_{0,k}$ depends on the value of $\beta_{0,k-1}$ which is also a Markvo chain. For the first element $\beta_{0,0}$, it is a start point do not depend on other $\beta 0,k$, for $k\neq 0$.Although it follows a normal distribution.we can also use truncated normal distribution to sample by setting the lower boundary to be negative Infinite .It was sample by

$$J(\beta_{0,1}^*|\beta_{0,1}^{(s)})=Norm(\beta_{0,1}^*|\beta_{0,1}^{(s)},\sigma_{0,1}^2)=Norm(\beta_{0,1}^*|\beta_{0,1}^*>-\infty;\beta_{0,1}^{(s)},\nu\sigma_{0,1}^2)$$

In R , input could be :"rtruncnorm(a=-Inf,b=Null,mean=beta_s,sd=sigma" while the other "dtruncnorm(x=,a=,b=,mean=,sd=)" may also be useful in the futher steps.And then we can sample the following $\beta 0,k$ one by one using distribution

$$J(\beta_{0,1}^*|\beta_{0,1}^{(s)})=Norm(\beta_{0,1}^*|\beta_{k,1}^*>\beta_{0,k-1}^{(s)};\beta_{0,1}^{(s)},\nu\sigma_{0,1}^2)$$

Then it comes to check whether to accept it or not.According to the logistic regression steps and (4) ,since $J(\beta_{0,1}^*|\beta_{0,1}^{(s)})=J(\beta_{0,1}^{(s)}|\beta_{0,1}^*)$ so the log of ratio keep similar form:

$$log(r)=\mathcal{L}(\beta_{0,k}^*;\beta_{-(0,k)}^{(s)}\gamma^{(s+1)},Y,X)-\mathcal{L}(\beta_{0,k}^{(s)};\beta_{-(0,k)}^*,\gamma^{(s+1)},Y,X)$$

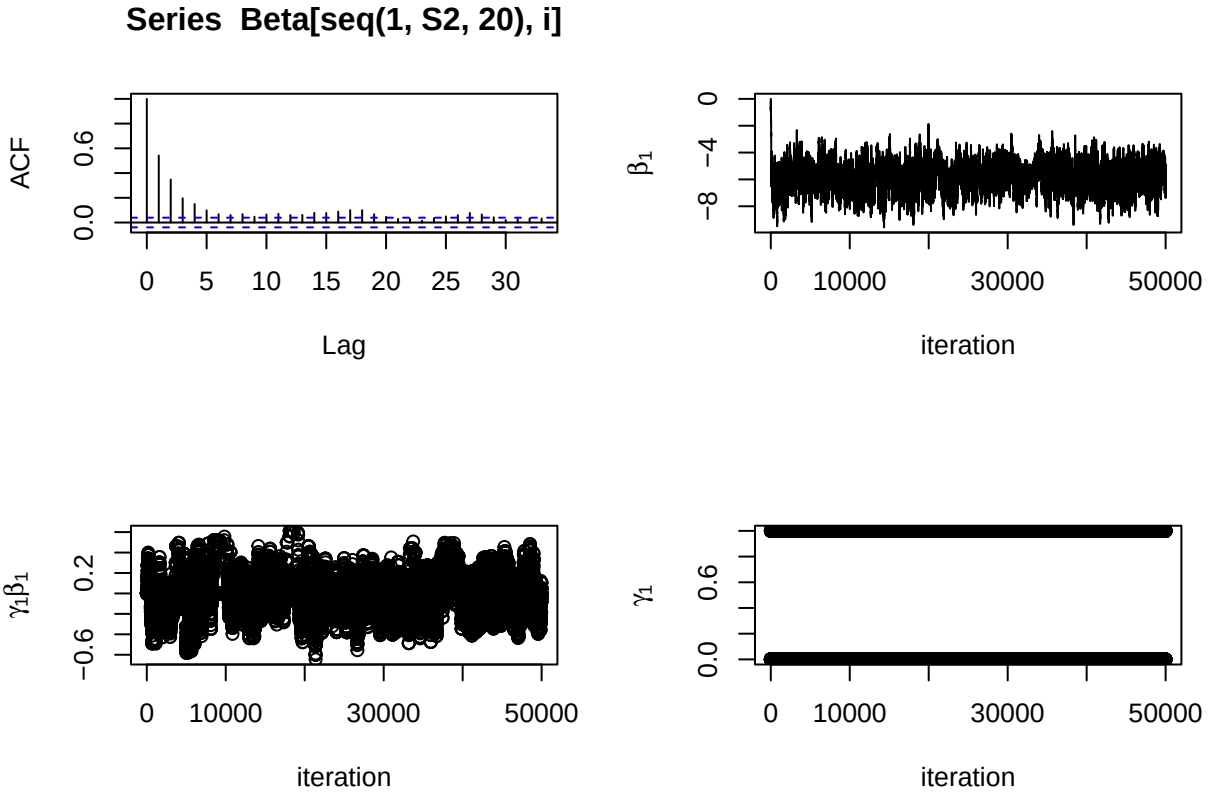
Then we just need to run the above process for thousands of iterations. ## How numerical assumptions was set Some distribution parameters numerical assumption may be just a trial and maybe no more informative.To some degree, it may just control the model to make it not fit too well which may cause overfitting. Meanwhile, some parameters we usually call them hyper parameters are chosen under some practices to make the accepted rate look well (between 20% to 50%) eg.ν

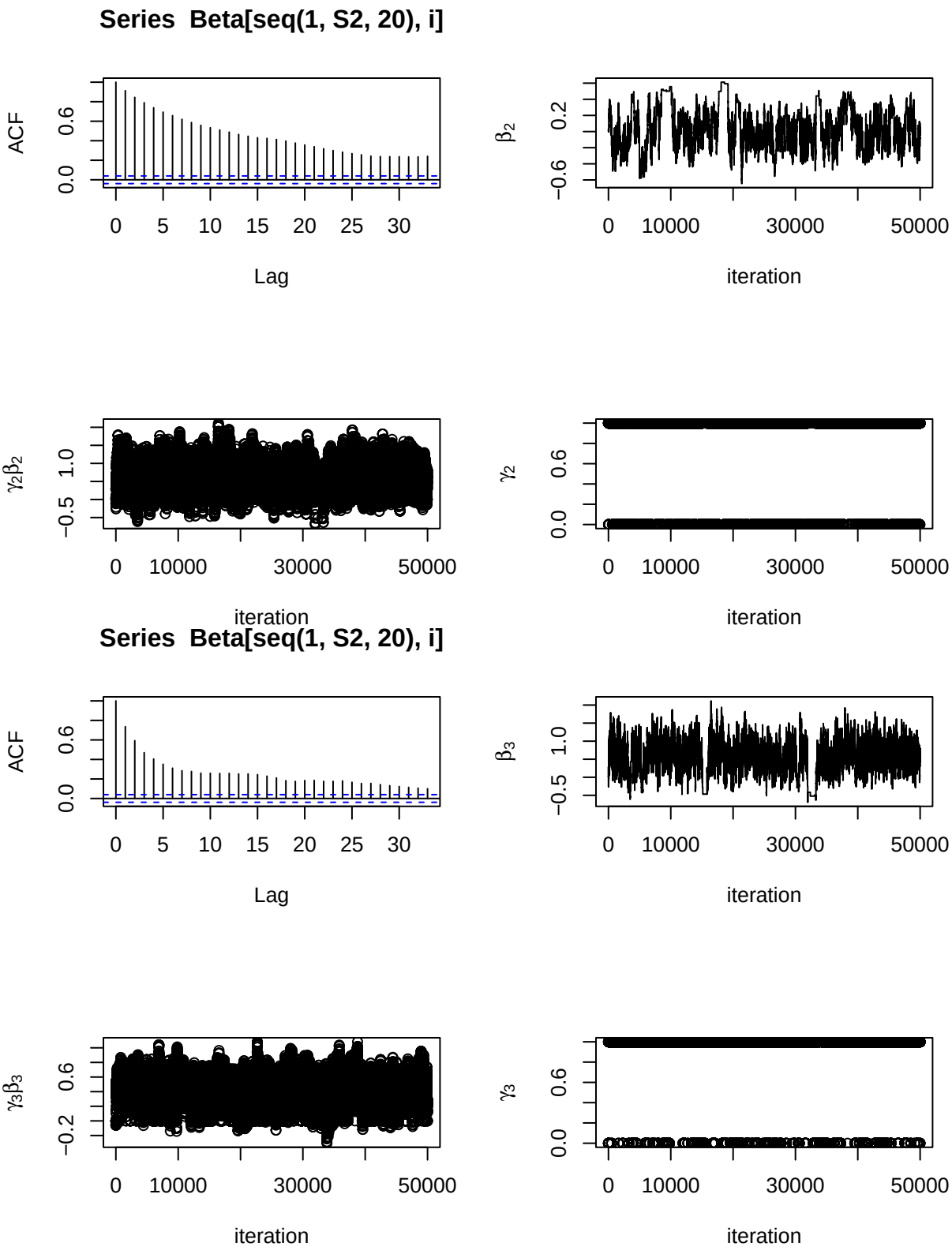
Results

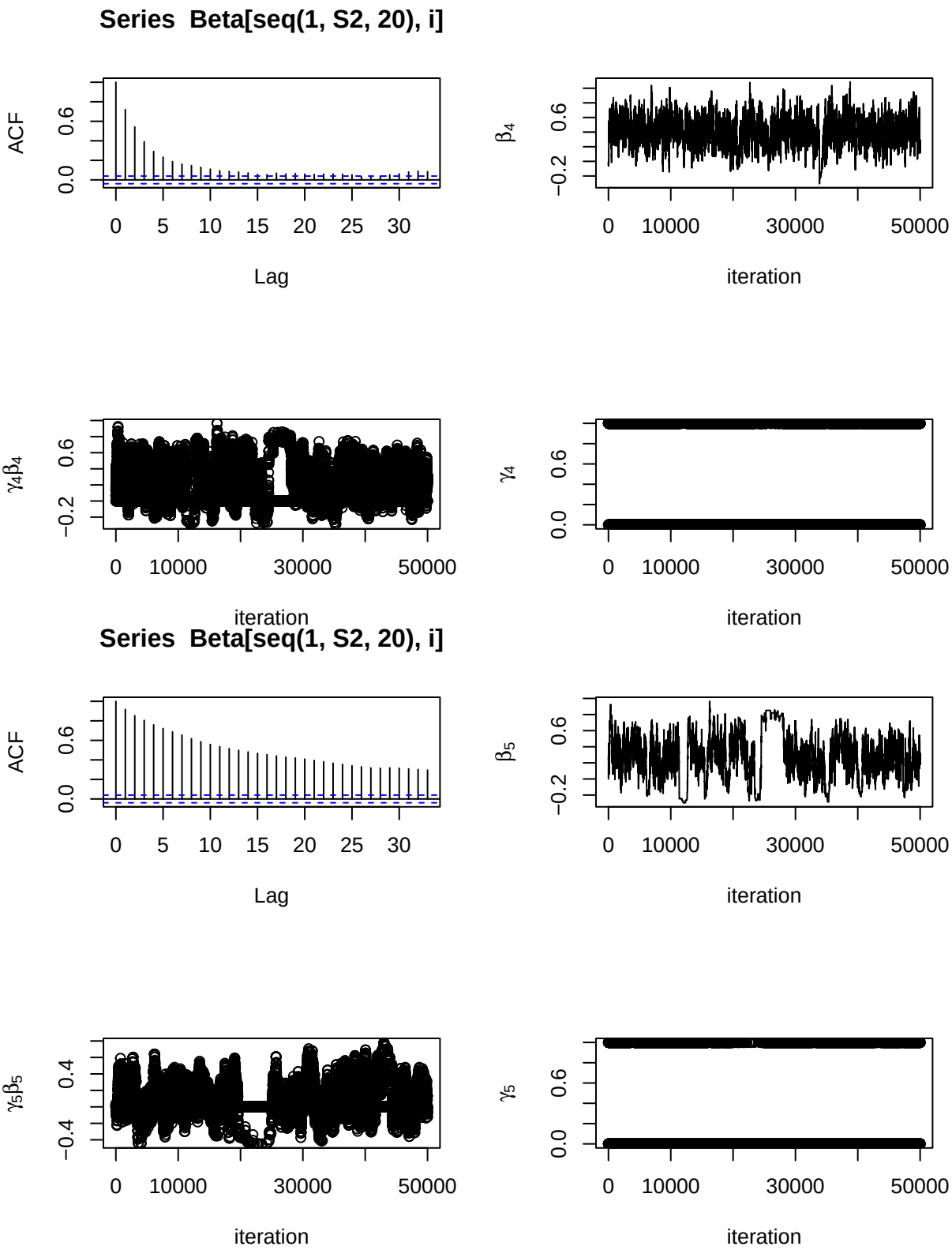
Logsitic model on herat-diease (healthy/sick)

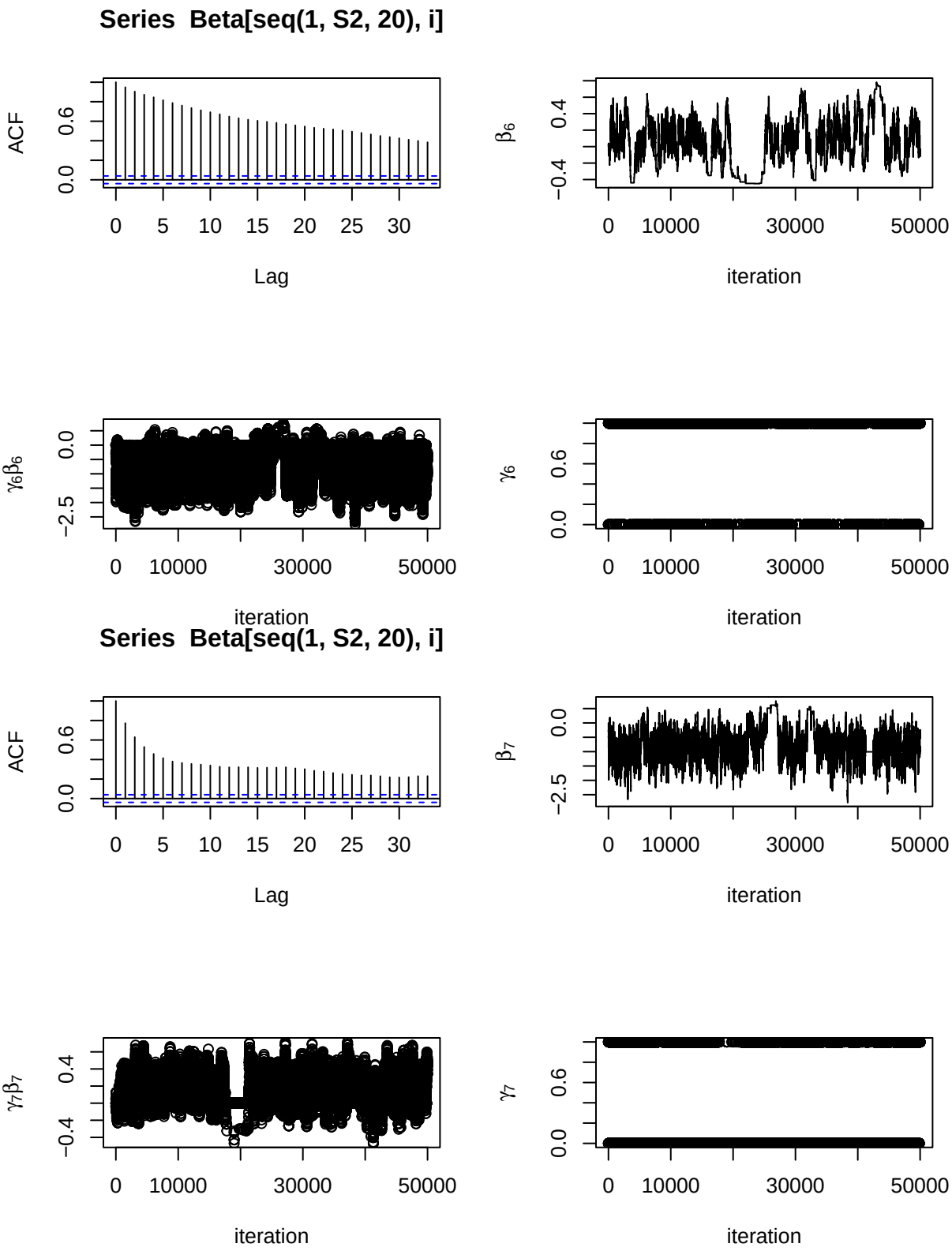
The plots below display the mcmc process .The acf plot perform not well when we record every updating coefficients. After applying choosing record one in 20 updating coefficients,those plots with high poster probability in γ_i perform really well which seems to imply a stationary process of Markov chain. The pattern may suggest the chain convergence .The γ performance seems to suggest chest pain type,max heart rate,oldpeak,number of vessels colored and thal are important factor.When it comes to apply a Bayesian Interval of 90%, the Interval without zero suggests that resting blood pres ,slope and thal are significant factors.

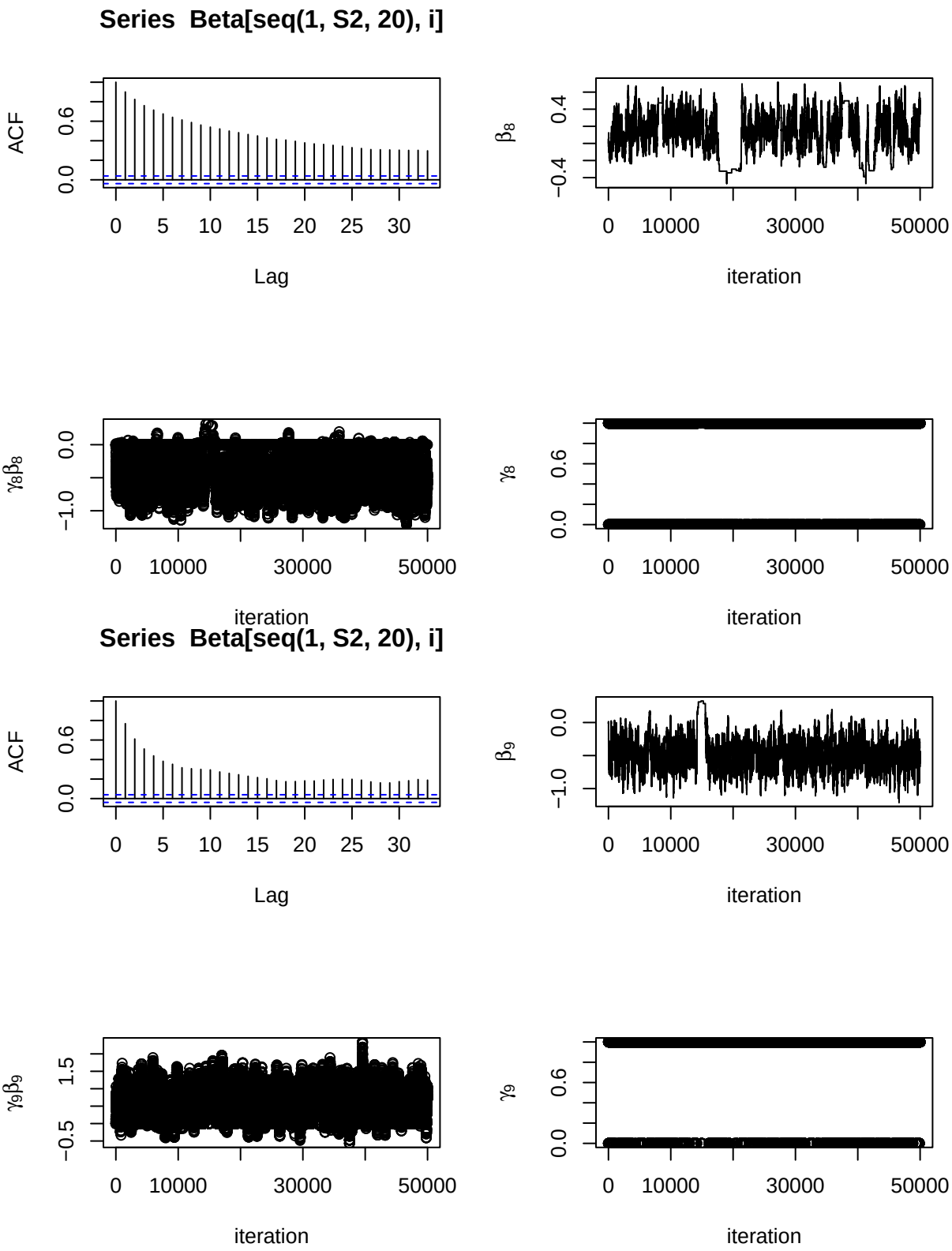
Furthermore, if the model coefficients is given by mean of $\gamma \cdot \beta$.Then we compared the MCMC model with the glm model.It shows that MCMC(0.861 while glm is 0.854) got a little bit higher accuracy but in this test data are not split into training and testing sets.

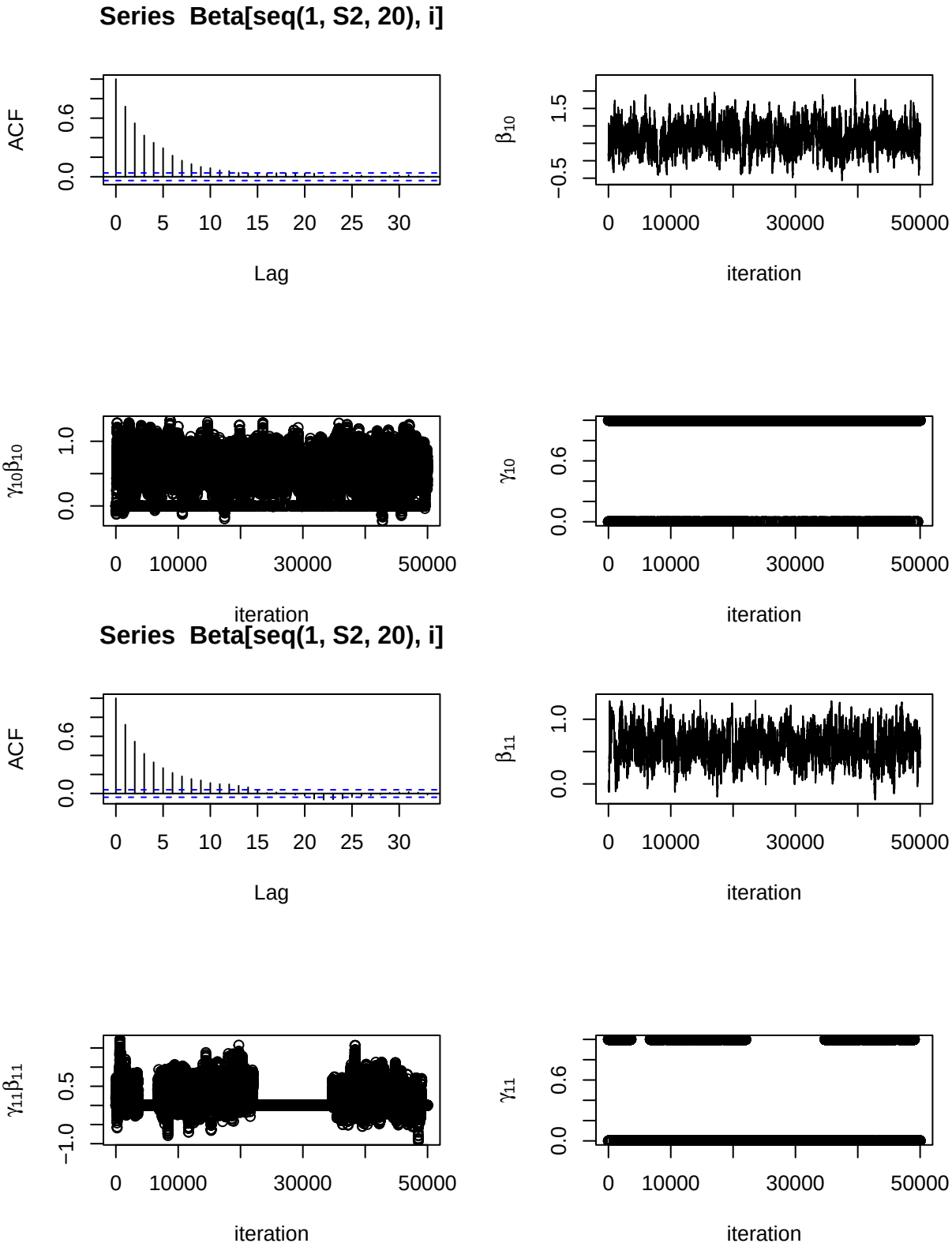


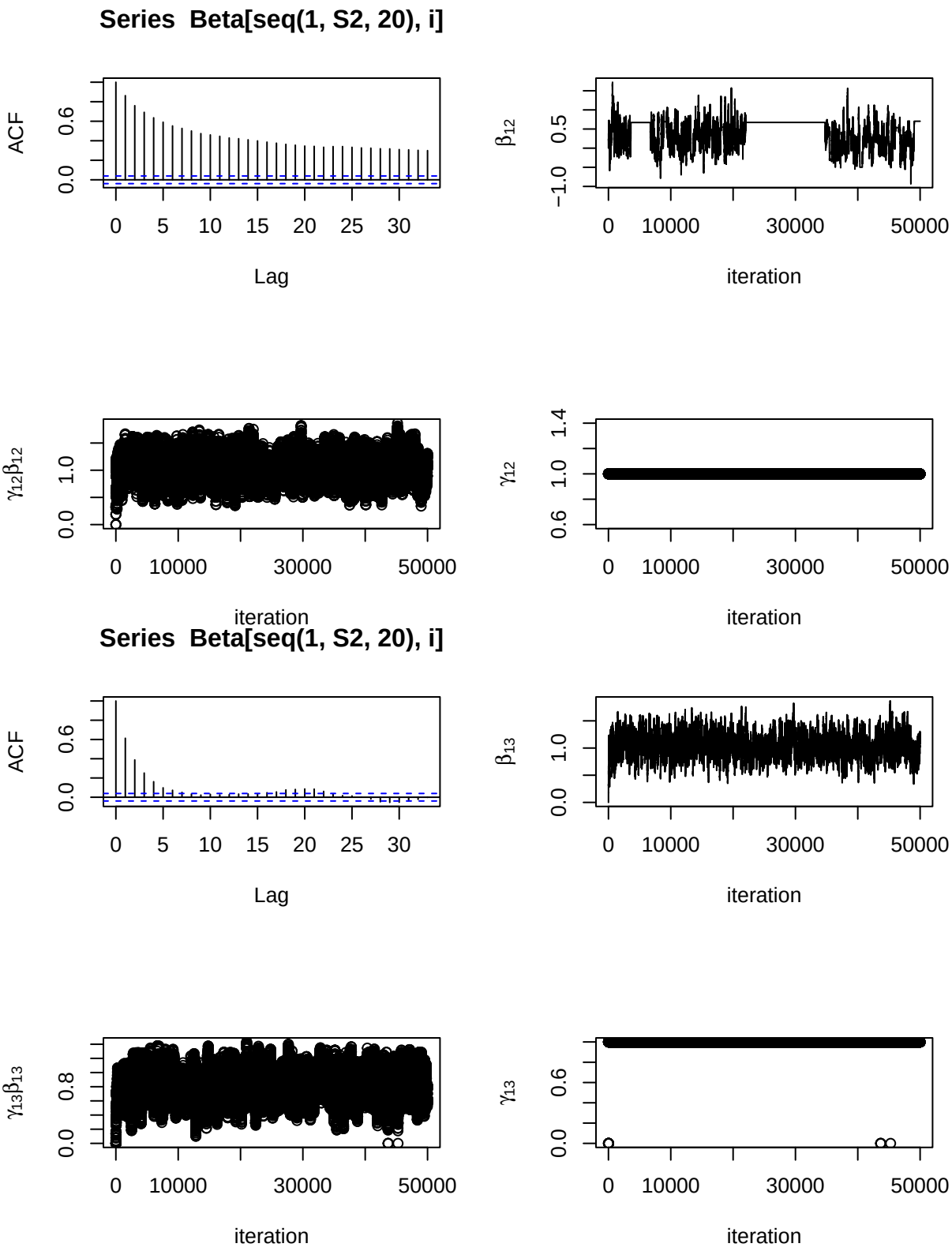

Series Beta[seq(1, S2, 20), i]
ACF
Lag
β2
iteration
γ2β2
iteration
γ2
iteration
Series Beta[seq(1, S2, 20), i]
ACF
Lag
β3
iteration
γ3β3
iteration
γ3
iteration


Series Beta[seq(1, S2, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β4
−0.2
0.6
0 10000 30000 50000
iteration
γ4β4
−0.2
0.6
0 10000 30000 50000
iteration
γ4
0.0
0.6
0 10000 30000 50000
iteration
Series Beta[seq(1, S2, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β5
−0.2
0.6
0 10000 30000 50000
iteration
γ5β5
−0.4
0.4
0 10000 30000 50000
iteration
γ5
0.0
0.6
0 10000 30000 50000
iteration


Series Beta[seq(1, S2, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β6
−0.4
0.4
0 10000 30000 50000
iteration
γ6β6
−2.5
0.0
0 10000 30000 50000
iteration
γ6
0.0
0.6
0 10000 30000 50000
iteration
Series Beta[seq(1, S2, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β7
−2.5
0.0
0 10000 30000 50000
iteration
γ7β7
−0.4
0.4
0 10000 30000 50000
iteration
γ7
0.0
0.6
0 10000 30000 50000
iteration


Series Beta[seq(1, S2, 20), i]
ACF
Lag
β8
iteration
γ8β8
iteration
γ8
iteration
Series Beta[seq(1, S2, 20), i]
ACF
Lag
β9
iteration
γ9β9
iteration
γ9
iteration


Series Beta[seq(1, S2, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β10
−0.5
1.5
0 10000 30000 50000
iteration
γ10β10
0.0
1.0
0 10000 30000 50000
iteration
γ10
0.0
0.6
0 10000 30000 50000
iteration
Series Beta[seq(1, S2, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β11
0.0
1.0
0 10000 30000 50000
iteration
γ11β11
−1.0
0.5
0 10000 30000 50000
iteration
γ11
0.0
0.6
0 10000 30000 50000
iteration

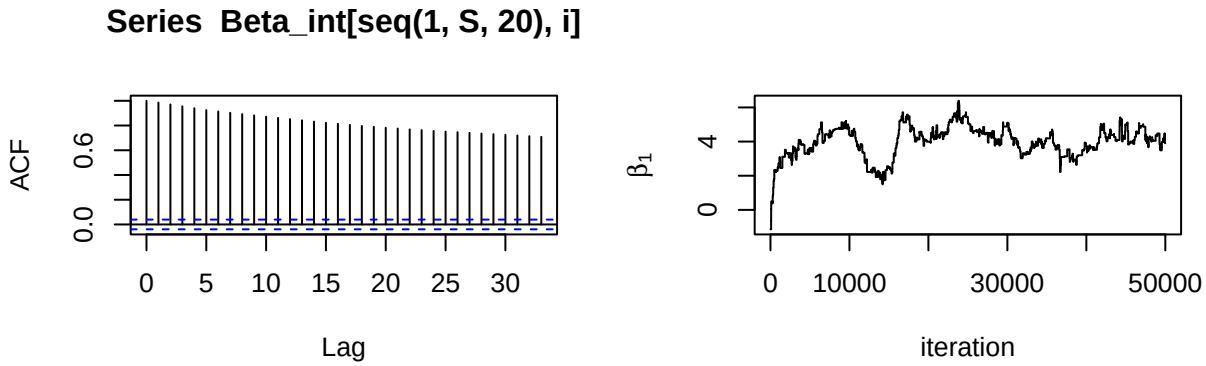


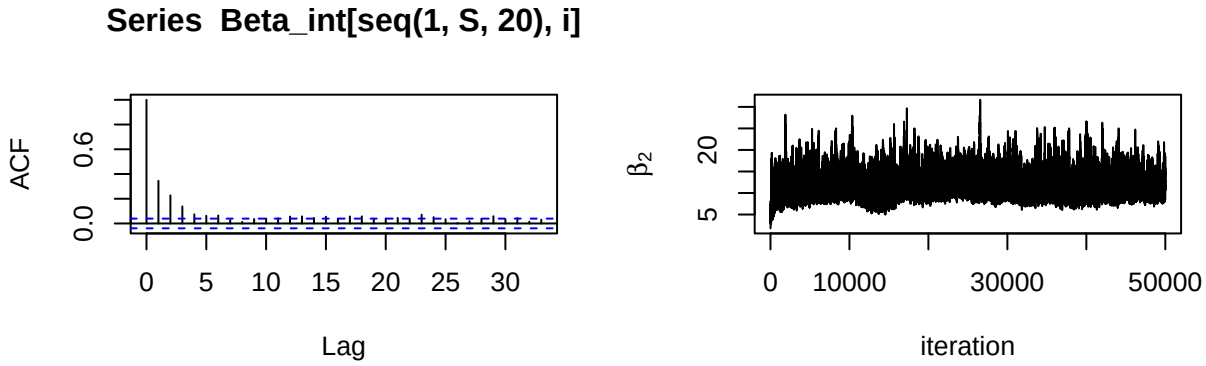

```
## [1] 0.8614865
```

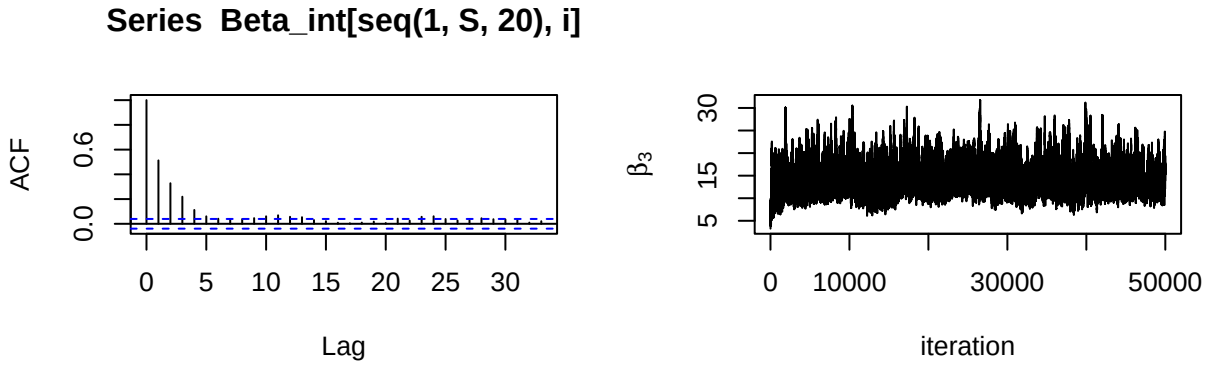
```
## [1] 0.8547297
```

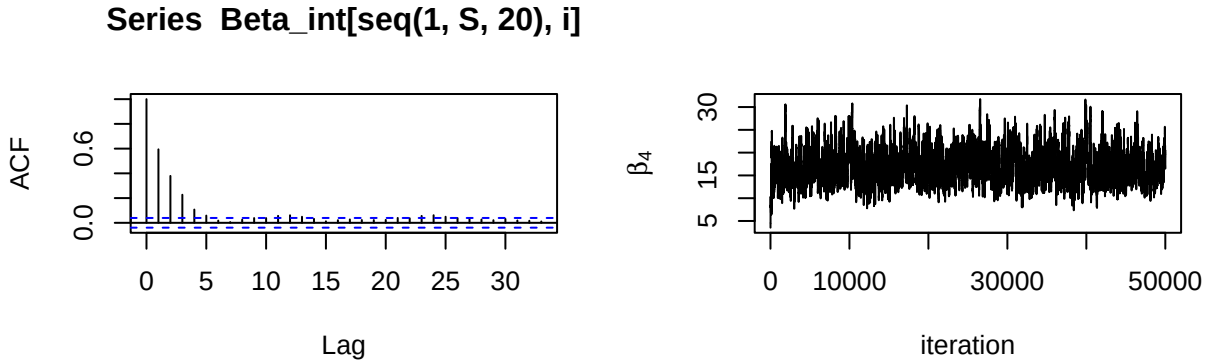
ordinal model on herat-diease (healthy/sick1,2,3,4)

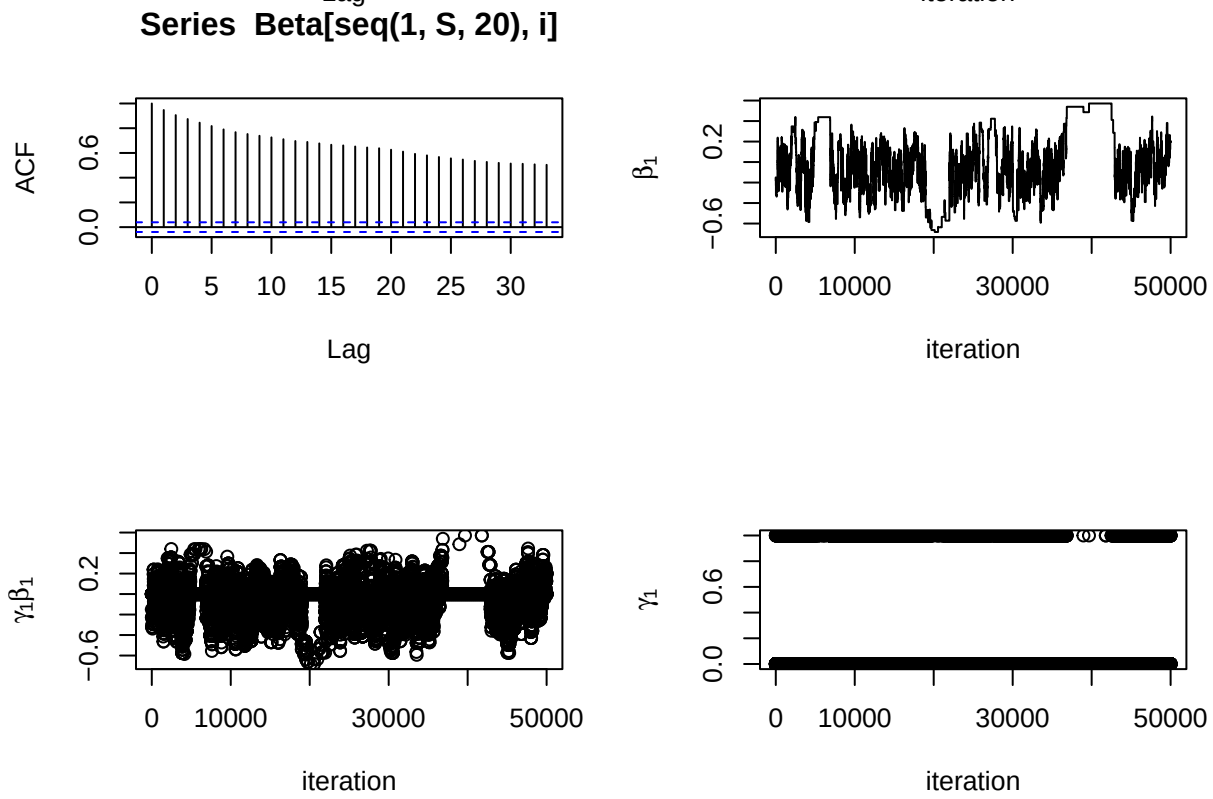
When it comes to analyze more on sick type(S1,S2,S3,S4) which probably mean the degree of sickness.The acf plot also perform much better after select 1 generated data in 20 updating coefficients while this time less $gamma_i$ have high probabilty equal to one.This acf may also imply the converge of Markov chains.In this model, *gamma* seems to support on less factors, however,number of vessels colored and thal are still important factors.When we check the Bayesian Interval ,it keep the same inference as the logistic model do. FInally, it is also interesting to see applying MCMC model to check agian have 0.71 accurancy higher than glm in R which is 0.65.

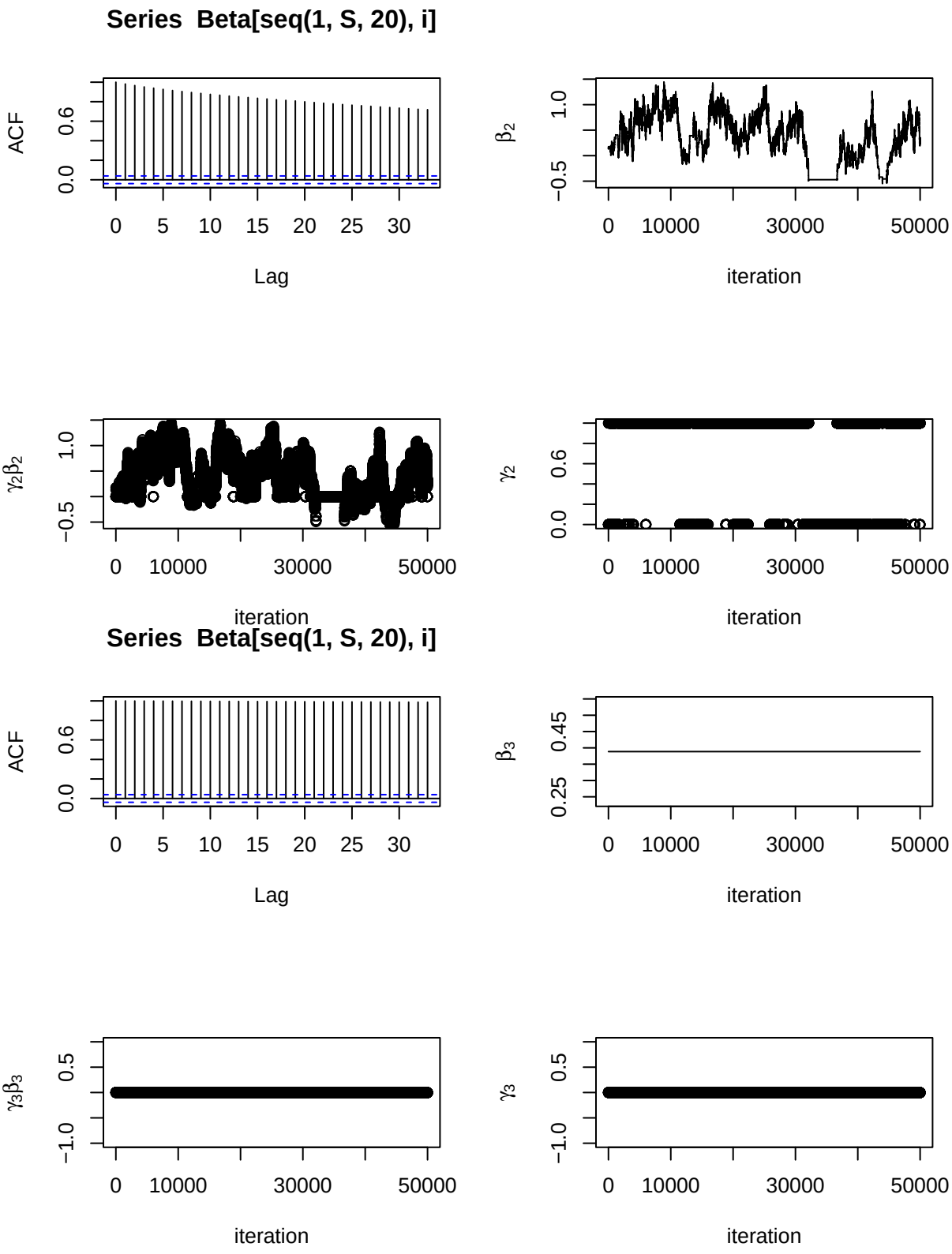


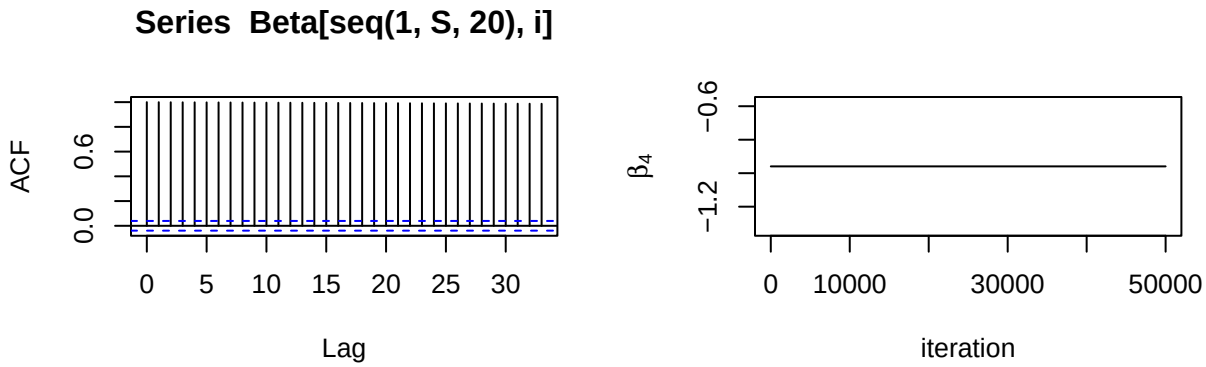



Series Beta_int[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β3
5 15 30
0 10000 30000 50000
iteration

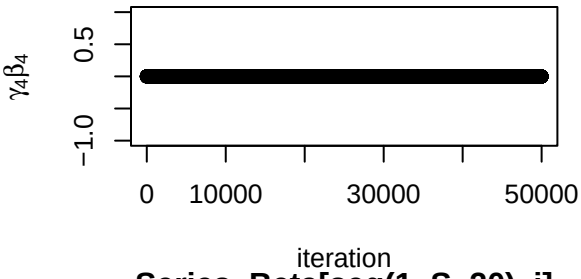

Series Beta_int[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β4
5 15 30
0 10000 30000 50000
iteration

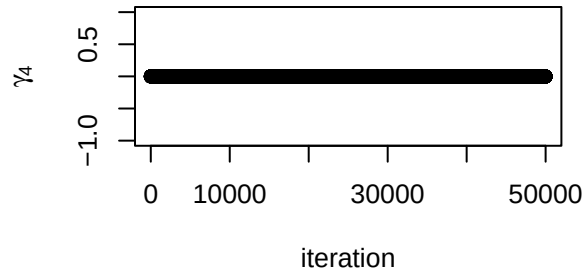

Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β1
−0.6 0.2
0 10000 30000 50000
iteration
γ1β1
−0.6 0.2
0 10000 30000 50000
iteration
γ1
0.0 0.6
0 10000 30000 50000
iteration

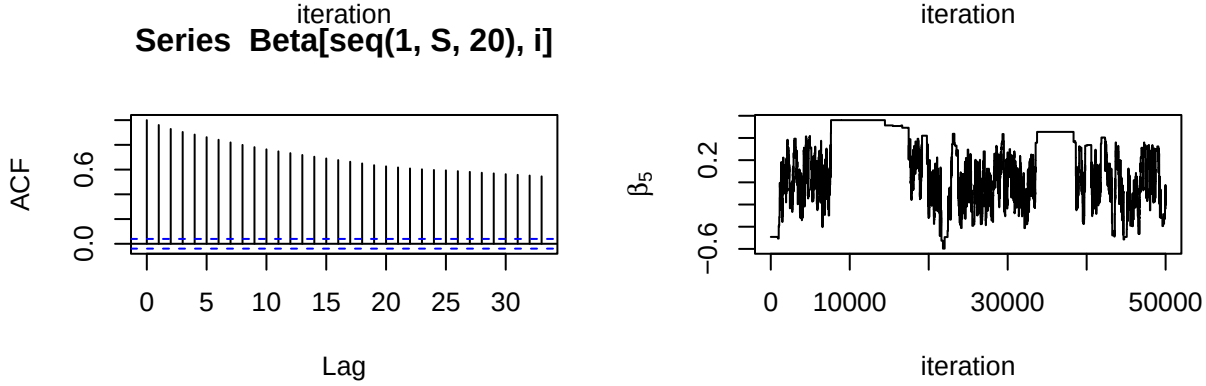

Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β2
−0.5
1.0
0 10000 30000 50000
iteration
γ2β2
−0.5
1.0
0 10000 30000 50000
iteration
γ2
0.0
0.6
0 10000 30000 50000
iteration
Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β3
0.25
0.45
0 10000 30000 50000
iteration
γ3β3
−1.0
0.5
0 10000 30000 50000
iteration
γ3
−1.0
0.5
0 10000 30000 50000
iteration


Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0
5
10
15
20
25
30
Lag

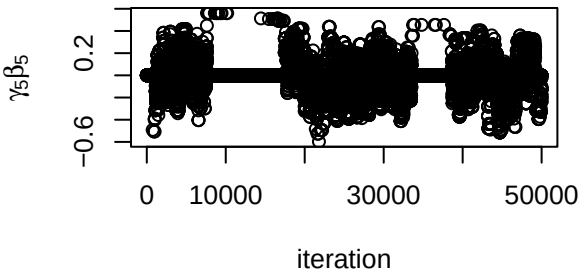
β_4
−1.2
−0.6
0
10000
30000
50000
iteration

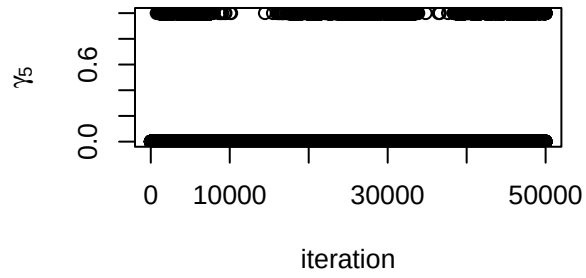

$\gamma_4\beta_4$
−1.0
0.5
0
10000
30000
50000
iteration

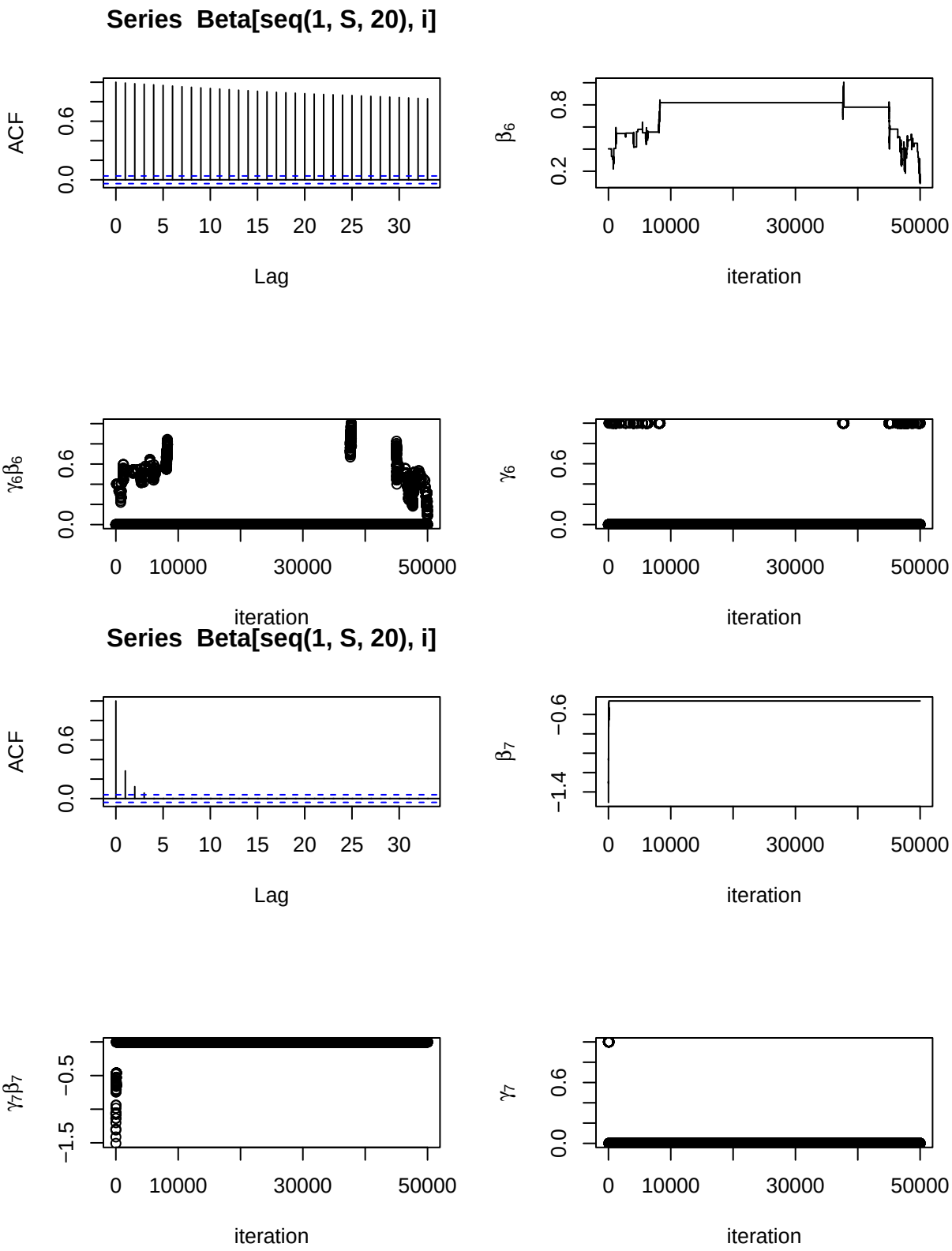

γ_4
−1.0
0.5
0
10000
30000
50000
iteration

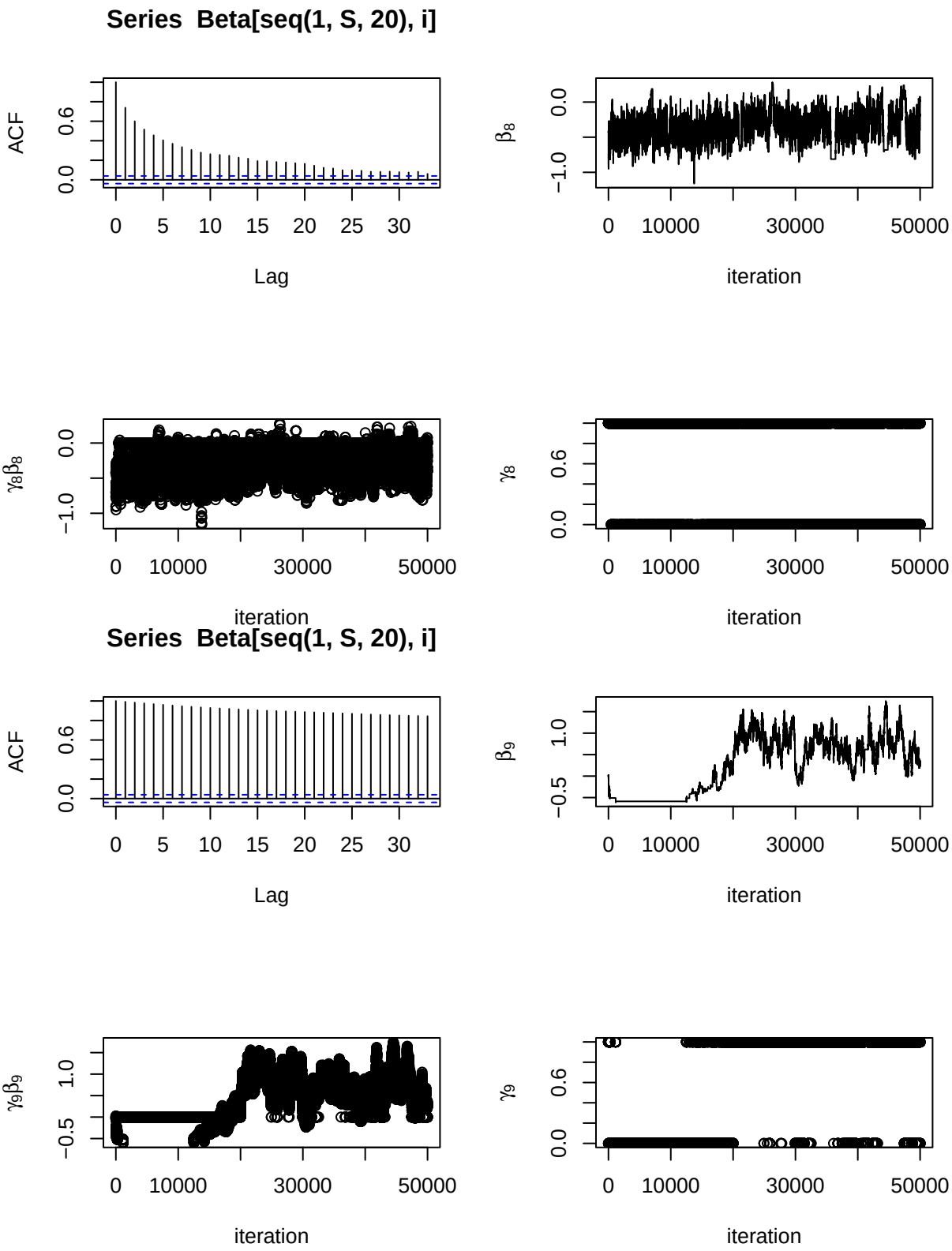

Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0
5
10
15
20
25
30
Lag

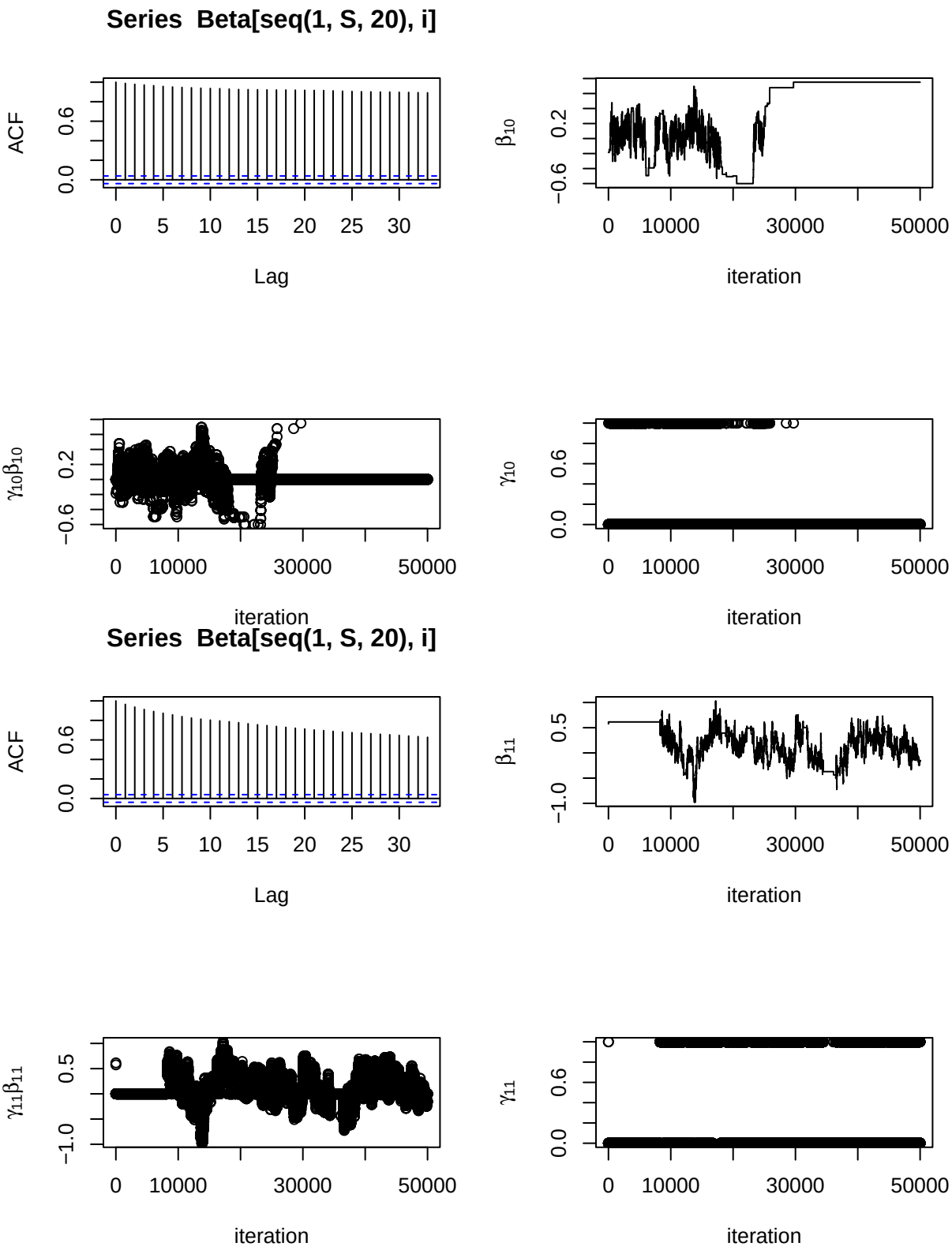
β_5
−0.6
0.2
0
10000
30000
50000
iteration

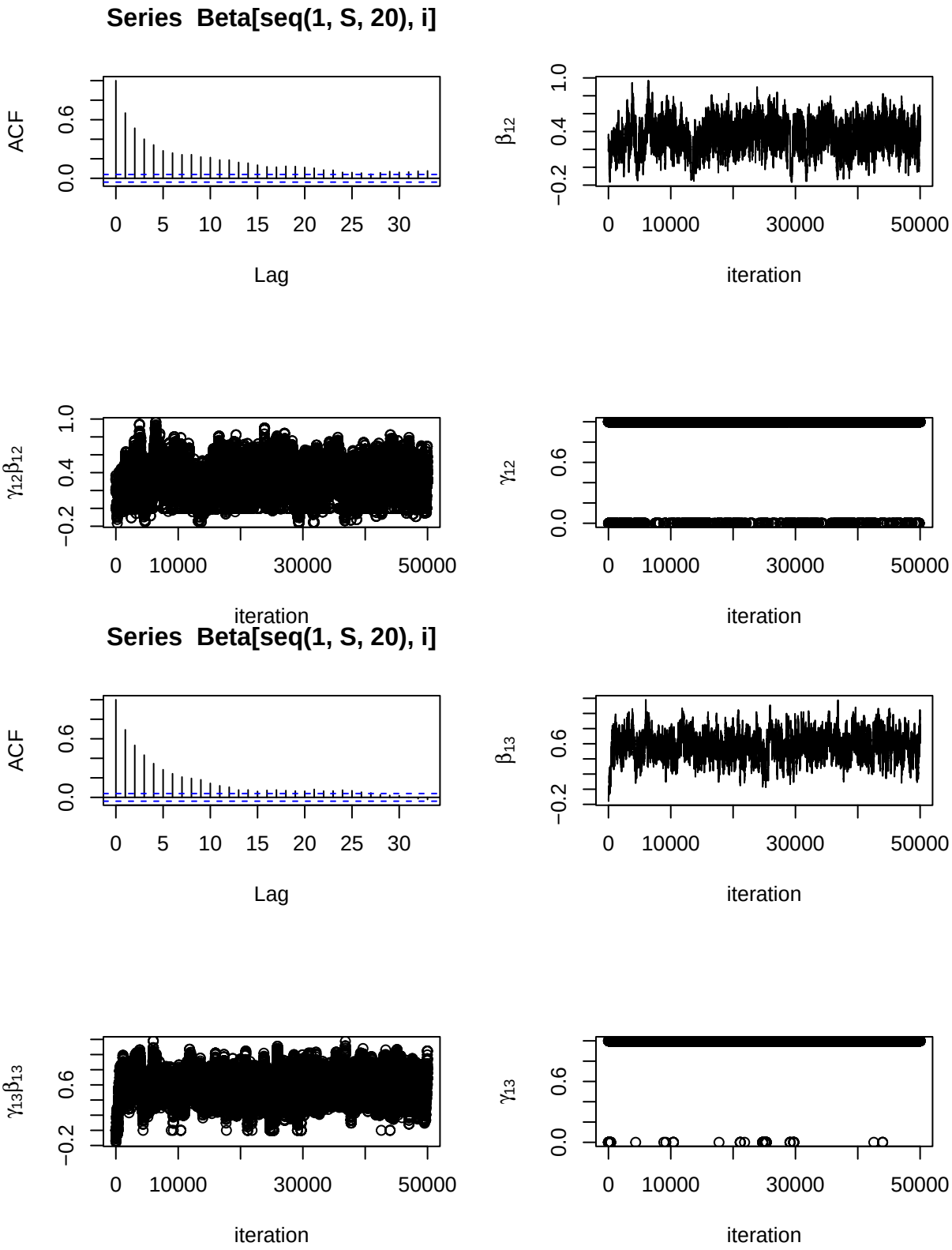

$\gamma_5\beta_5$
−0.6
0.2
0
10000
30000
50000
iteration


γ_5
0.0
0.6
0
10000
30000
50000
iteration


Series Beta[seq(1, S, 20), i]
ACF
Lag
β6
iteration
γ6β6
iteration
γ6
iteration
Series Beta[seq(1, S, 20), i]
ACF
Lag
β7
iteration
γ7β7
iteration
γ7
iteration


Series Beta[seq(1, S, 20), i]
ACF
Lag
β8
iteration
γ8β8
iteration
γ8
iteration
Series Beta[seq(1, S, 20), i]
ACF
Lag
β9
iteration
γ9β9
iteration
γ9
iteration


Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β10
−0.6
0.2
0 10000 30000 50000
iteration
γ10β10
−0.6
0.2
0 10000 30000 50000
iteration
γ10
0.0
0.6
0 10000 30000 50000
iteration
Series Beta[seq(1, S, 20), i]
ACF
0.0
0.6
0 5 10 15 20 25 30
Lag
β11
−1.0
0.5
0 10000 30000 50000
iteration
γ11β11
−1.0
0.5
0 10000 30000 50000
iteration
γ11
0.0
0.6
0 10000 30000 50000
iteration


Series Beta[seq(1, S, 20), i]
ACF
Lag
β12
iteration
γ12β12
iteration
γ12
iteration
Series Beta[seq(1, S, 20), i]
ACF
Lag
β13
iteration
γ13β13
iteration
γ13
iteration

Conclusion

The MCMC method performs well on both models to a certain extent, although it might require more time for training with larger datasets. MCMC provides a direct and straightforward approach to analyzing data, identifying important factors, and fitting models. In this case of heart disease, we obtained the same important factors from both models: the number of vessels colored and thal. Additionally, we identified three significant factors: resting blood pressure, slope, and thal. This suggests that thal likely plays a pivotal role in assessing one's heart health condition.

Beides, we can derive our models after training by simply taking the mean of the coefficients after the burn-in phase. It seems to give better results compared to the glm may be attributed to some random effects. It's worth exploring if the model's performance can be enhanced by adjusting parameters, such as using different seeds.

There are still many areas for further extension and discussion. I'm interested in testing the performance of various models, including multi-categorical regression models. Additionally, adding more attributes to the dataset to test on a larger scale model might give a better model.

It's important to note that the ordinal model took longer to train due to the larger number of coefficients.

Lastly, it's essential to consider computational techniques carefully when writing MCMC code to prevent computation errors, especially when working with vectorized operations.