Bayesian learning: Assignment 3

Måns Magnusson Josef Wilzén

November 2012

Question 1)

Question 1a)

The gibbs sampler were implemented in `R version 2.15.1 (2012-06-22)` as a function. The function implemented coded in `R` looks as follows.

```
gibbS <- function(y, prior, start.val = c(0, 1), sim = 1000) {
    # Parameter matrix c('mu','sigma')
    pardraws <- matrix(0, nrow = sim, ncol = 2)
    # Starting values
    pardraws[1, ] <- start.val
    # Calculating values that will be used many times
    small_n <- length(y)
    ymean <- mean(y)
    nu_n <- small_n + prior[3]

    # The Gibbs sampler
    for (i in 2:sim) {
        # Calculate tau_n and mu_n
        tau_n2 <- 1/((1/prior[2]^2) + (small_n/pardraws[i - 1, 2]^2))
        mu_n <- ((1/prior[2]^2) * prior[1] + (small_n/pardraws[i -
            1, 2]^2) * ymean) * tau_n2
        # Draw one sample of mu
        pardraws[i, 1] <- rnorm(1, mu_n, sqrt(tau_n2))
        # Calculate sig_n
        sig_n <- (prior[3] * prior[4]^2 + sum((y - pardraws[i, 1])^2))/nu_n
        # Draw one sample of sig_n
        pardraws[i, 2] <- sqrt(rinvchisq(1, nu_n, sig_n))
    }
    pardraws <- as.data.frame(pardraws)
    return(pardraws)
}
```

Question 1b)

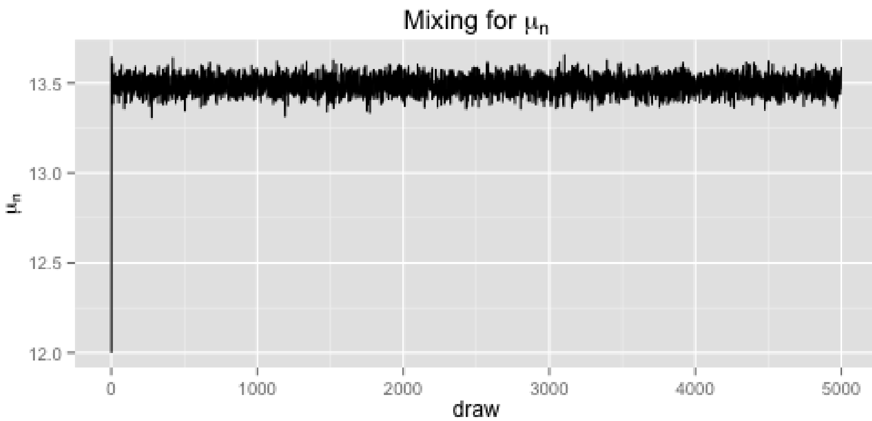
The dataset `CanadianWages.dat` were loaded into R as the dataset `data`. To draw samples from the posterior distribution. In the model the prior values were set to $\mu_0 = 12$, $\tau_0 = 1000$, $\nu_0 = 1$ and $\sigma_0 = 2$. Since these values are the prior beliefs the starting values is set to the same prior beliefs of μ_0 and σ_0.

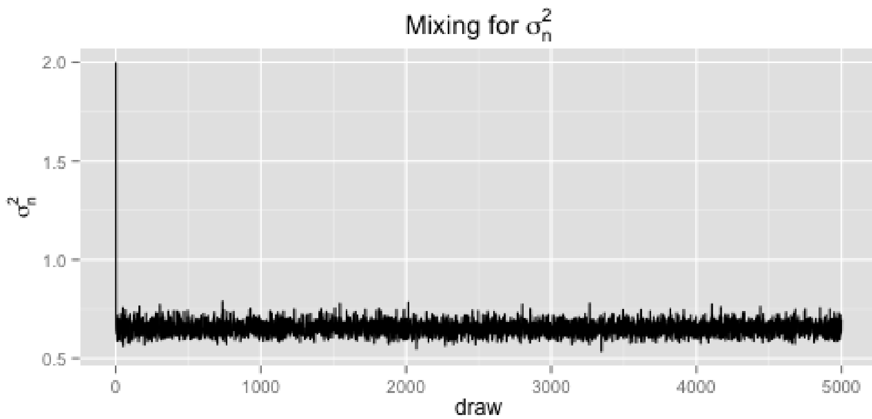
```
# Parameters for the gibbS function
sim <- 5000  # Number of simulations
y <- data[, 1]
start.val <- c(prior[1], prior[4])
```

Based on these prior values as starting values (`12, 2`) the `gibbS` function is used by drawing `5000` draws from the posterior.

```
res1 <- gibbS(y = y, prior = prior, start.val = start.val, sim = sim)
```

The mixing behaviour of μ_n and σ_n can be seen in the following two figures.





It is apparent that the two paramater distributions starts to mix almost directly based on the starting values above.

Question 2

Question 2a)

As a first step the part of the code defining τ is "commented" away som these paramaters can be chosen "outside" the code. The output from the file was also changed to enable printing `Rmd`-tables. The new code is called `MainOptimizeSpamMansJosef.R`. In this question τ is changed to `10` and the code is sourced as can bee seen in the following code.

```
Probit <- 0
chooseCov <- c(1:16)
```

```r
tau <- 10
source("MainOptimizeSpamMansJosef.R")
```

The modes of the resulting posteriors together with the posterior standard deviation can be seen in the following table.

	mode	sd.post
our	0.4297	0.0729
over	1.3007	0.2372
remove	3.2207	0.3556
internet	0.9720	0.1715
free	1.1727	0.1335
hpl	-0.8660	0.3660
X.	0.5600	0.0942
X..1	7.4692	0.6458
CapRunMax	0.0125	0.0018
CapRunTotal	0.0006	0.0001
const	-1.4570	0.0853
hp	-2.3793	0.3383
george	-5.4922	0.9963
X1999	-0.5300	0.2010
re	-0.7355	0.1388
edu	-2.0314	0.2970

Table 1: Posterior modes and sd

Question 2b)

The probit model were implemented in `R` the following way as function called `gibbSprobit`.

```r
gibbSprobit <- function(y, X, tau, v0 = 1, sigma0 = 1, beta0 = NULL,
    sim = 1000) {

    # Parameter matrix c('beta.vec')
    p.no <- dim(X)[2]
```

```
    pardraws <- matrix(0, nrow = sim, ncol = p.no)

    # Starting values (given by linear regression)
    pardraws[1, ] <- t(solve(t(X) %*% X) %*% t(X) %*% y)

    # Defining prior values
    omega0 <- tau^2 * diag(p.no)
    if (is.null(beta0))
        beta0 <- numeric(p.no)

    for (i in 2:sim) {
        # u_i given beta ,y i<-2 Since the u vector is not of further
        # intrest, these values is not saved Draw sample of u.vec
        u.vec <- rtnorm(n = length(y), mean = X %*% pardraws[i - 1,
            ], sd = 1, lower = ifelse(y == 1, 0, -Inf), upper = ifelse(y ==
            0, 0, Inf))
        # beta given u,y Calculate hyperparameters
        betahat <- solve(t(X) %*% X) %*% t(X) %*% u.vec
        betaN <- solve((t(X) %*% X + omega0)) %*% (t(X) %*% X %*%
            betahat + omega0 %*% beta0)
        omegaN <- (t(X) %*% X + omega0)
        vN <- v0 + dim(X)[1]
        sigmaN <- (v0 * sigma0^2 + (t(u.vec) %*% u.vec + t(beta0) %*%
            omega0 %*% beta0 - t(betaN) %*% omegaN %*% betaN))/vN

        # Draw sample of \beta
        pardraws[i, ] <- mvrnorm(1, mu = betaN, Sigma = as.numeric(sigmaN^2) *
            solve(omegaN))
    }
    pardraws <- as.data.frame(pardraws)
    colnames(pardraws) <- colnames(X)
    return(pardraws)
}
```

Question 2c)

Based on the function above `5000` draws from the posterior were drawn. This code was not as effective as the earlier code and of this reason it took much longer to simulate from the psoterior.

```
res2 <- gibbSprobit(y, X, tau, v0 = 1, sigma0 = 1, beta0 = NULL, sim = sim)
```

The mixing behaviour of the β were studied using graphical methods. As we can see the mixing behaviour (in the appendix) of some of the β do not suggest that the draws from the posterior is independent. This is especially true for β_9 , β_{12} and β_{13} . Of this reason it is hard to be able to draw conclusions f sample when it comes to the mode and sd of the mode of the distribution.

Question 2d)

	mode.logit	mean.probit	sd.logit
our	0.4297	0.2519	0.0729
over	1.3007	0.3787	0.2372
remove	3.2207	0.7329	0.3556
internet	0.9720	0.3914	0.1715
free	1.1727	0.5262	0.1335
hpl	-0.8660	-0.4144	0.3660
X.	0.5600	0.1845	0.0942
X..1	7.4692	0.7149	0.6458
CapRunMax	0.0125	0.0045	0.0018
CapRunTotal	0.0006	0.0003	0.0001
const	-1.4570	-0.5391	0.0853
hp	-2.3793	-0.6979	0.3383
george	-5.4922	-0.6841	0.9963
X1999	-0.5300	-0.2679	0.2010
re	-0.7355	-0.3031	0.1388
edu	-2.0314	-0.5037	0.2970

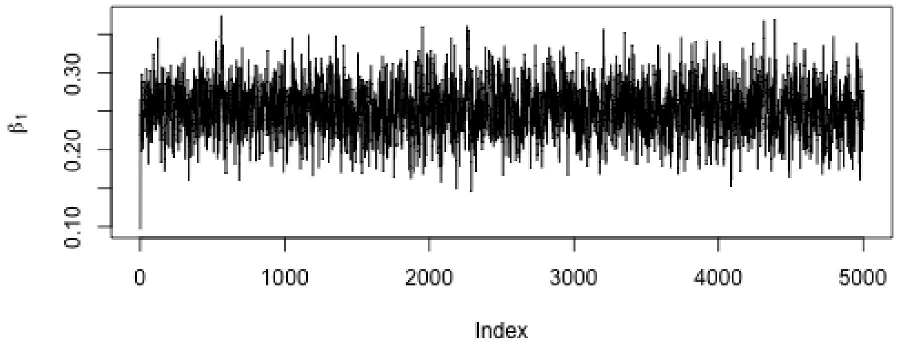
Table 2: Posterior modes/mean and sd for logit and probit model (continued below)

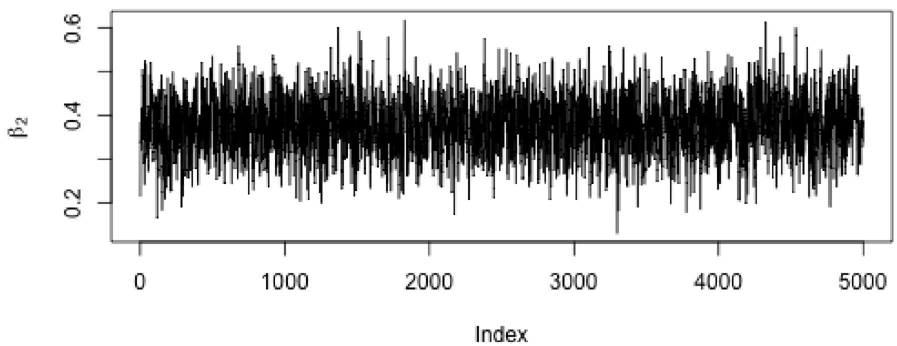
	sd.probit
our	0.0324
over	0.0647
remove	0.0650
internet	0.0550
free	0.0487
hpl	0.0782
X.	0.0234
X..1	0.0858
CapRunMax	0.0006
CapRunTotal	0.0001
const	0.0374
hp	0.0667
george	0.0907
X1999	0.0640
re	0.0476
edu	0.0674

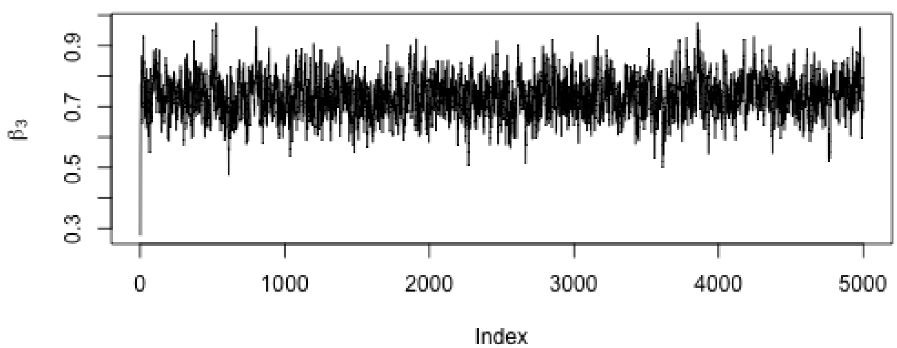
As we can see there is quite a difference between the β:s in the logitic model and in the probit model. This is of course expected since the parametrisation of the model is different and the intepretation of the different β:s is different as well.

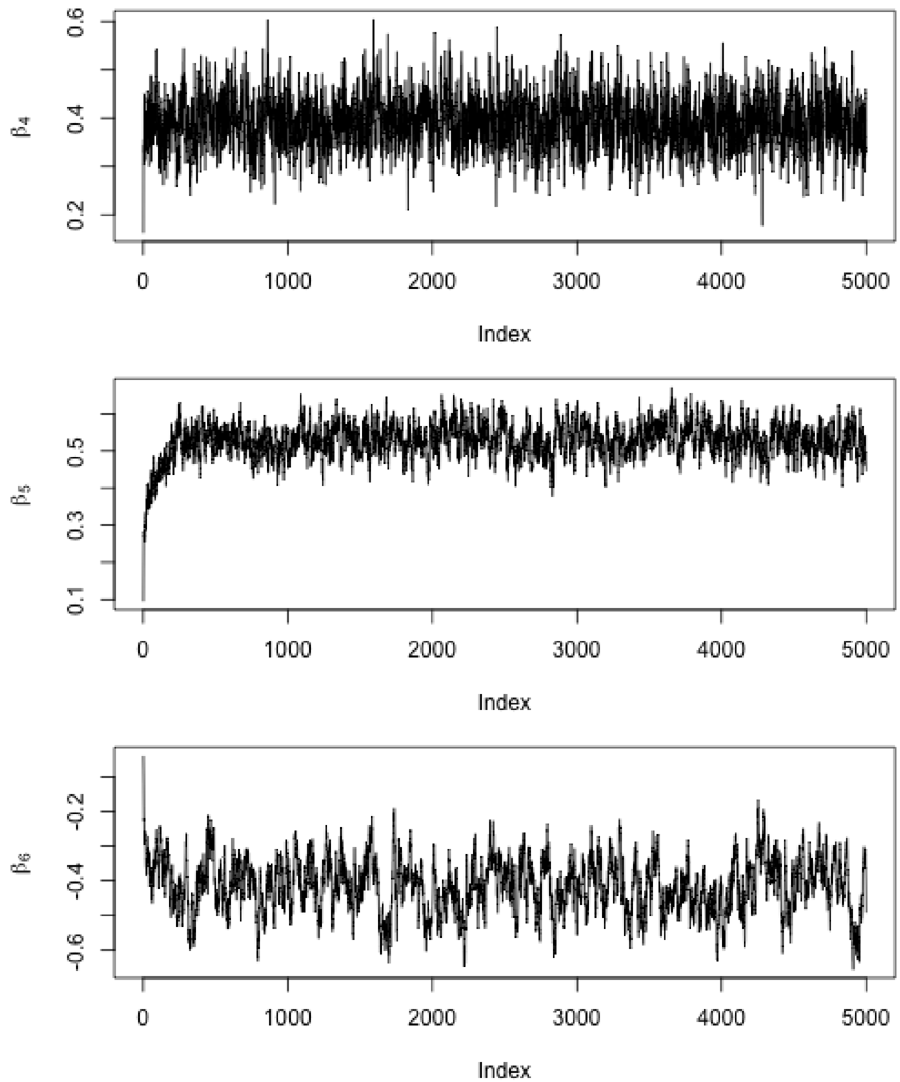
Appendix

Mixing behaviour for different $beta_1$ to $beta_1 6$ in the probit model.

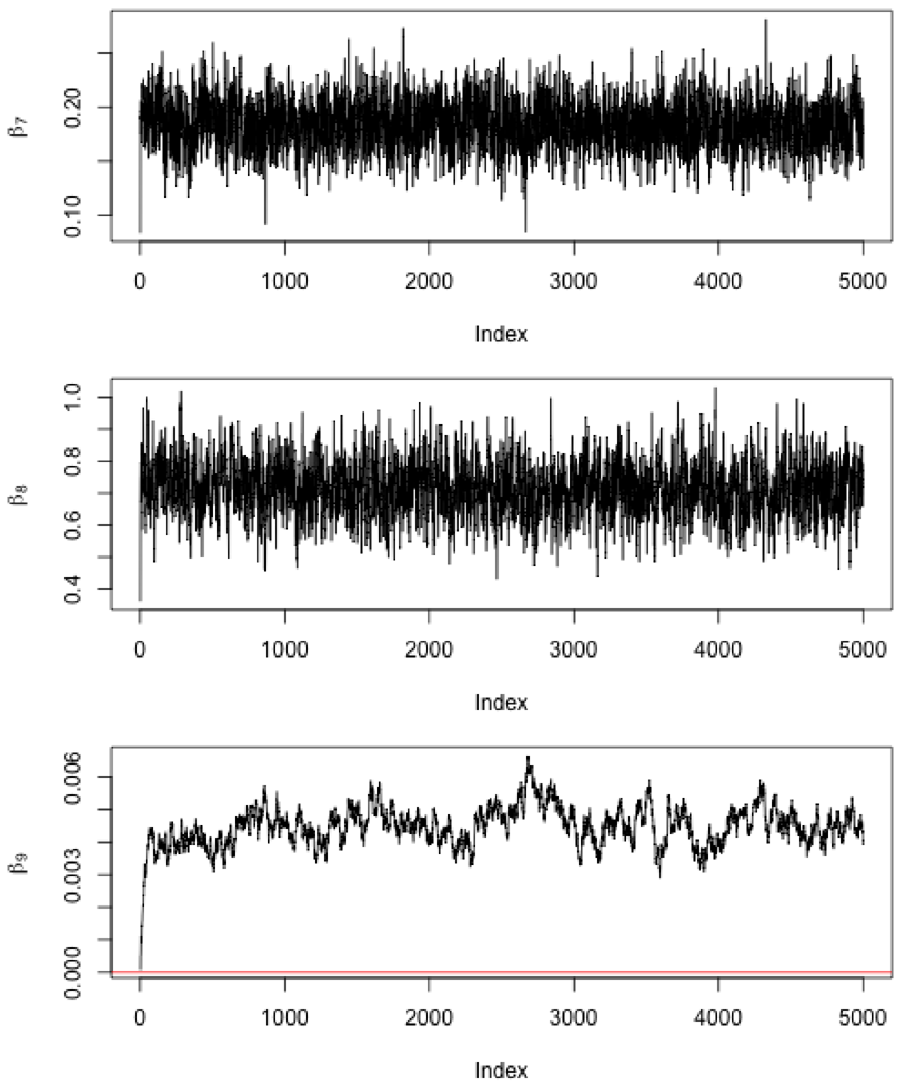




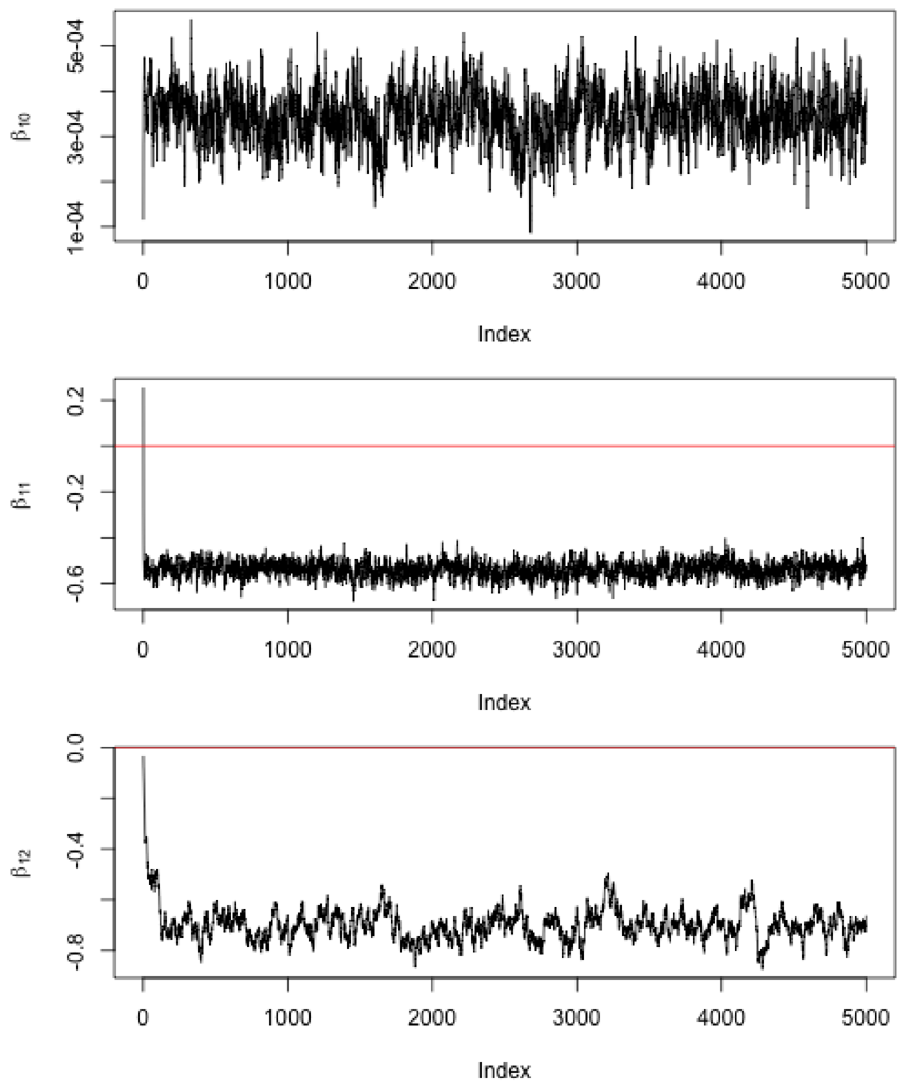


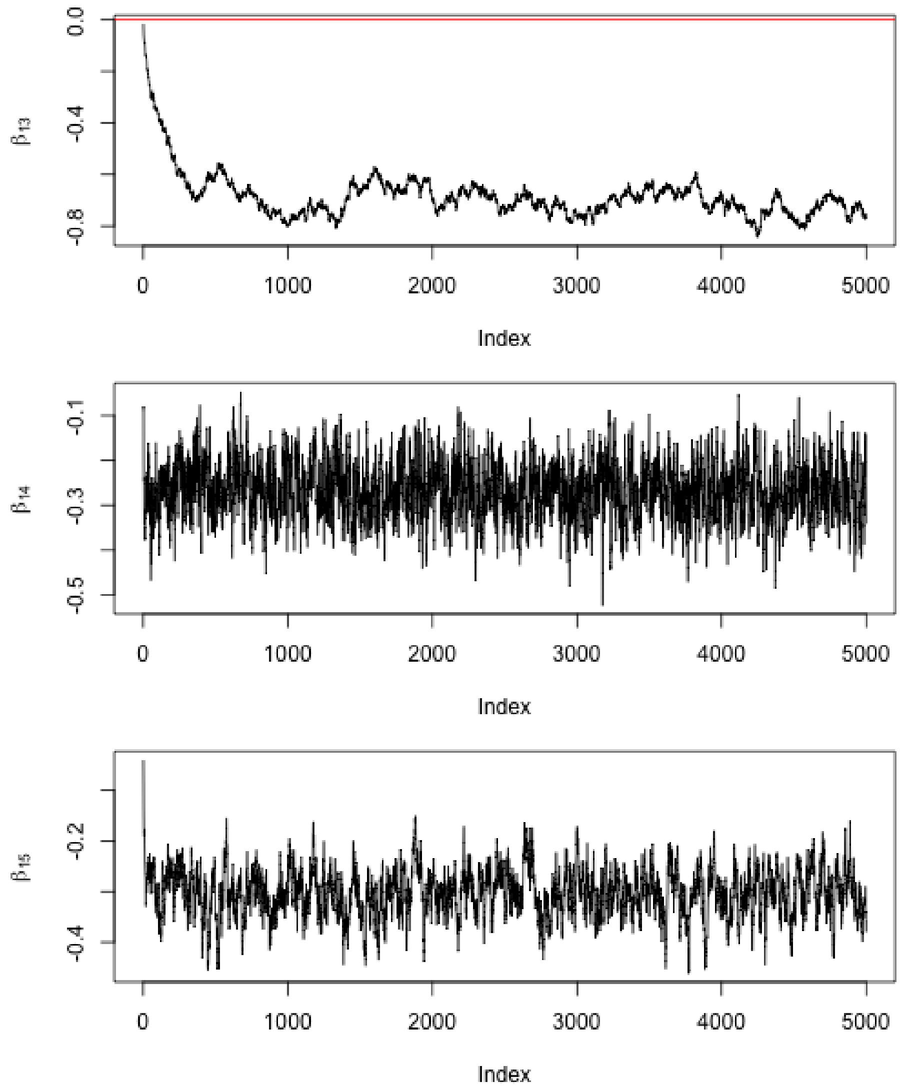


β4
0.2
0.4
0.6
0
1000
2000
3000
4000
5000
Index
β5
0.1
0.3
0.5
0
1000
2000
3000
4000
5000
Index
β6
-0.6
-0.4
-0.2
0
1000
2000
3000
4000
5000
Index

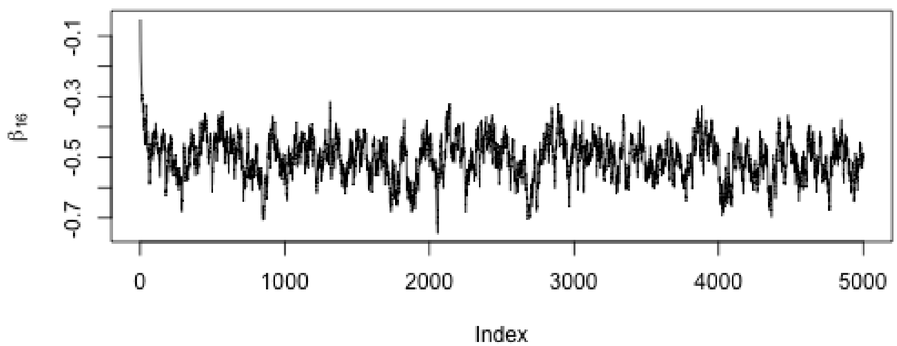


β_7
0.10
0.20
0
1000
2000
3000
4000
5000
Index
β_8
0.4
0.6
0.8
1.0
0
1000
2000
3000
4000
5000
Index
β_9
0.000
0.003
0.006
0
1000
2000
3000
4000
5000
Index


β_{10}
5e-04
3e-04
1e-04
0
1000
2000
3000
4000
5000
Index
β_{11}
0.2
-0.2
-0.6
0
1000
2000
3000
4000
5000
Index
β_{12}
0.0
-0.4
-0.8
0
1000
2000
3000
4000
5000
Index



β_{13}
0.0
-0.4
-0.8
0
1000
2000
3000
4000
5000
Index
β_{14}
-0.1
-0.3
-0.5
0
1000
2000
3000
4000
5000
Index
β_{15}
-0.2
-0.4
0
1000
2000
3000
4000
5000
Index



-0.1
-0.3
-0.5
-0.7
β_{16}
0
1000
2000
3000
4000
5000
Index