Joint hierarchical Bayesian logistic regression model for time-dependent covariates using PROC MCMC

Contact Information

Elsa Vazquez Arreola
Arizona State University
Maria.Vazquez@asu.edu

Jeffrey R Wilson
Arizona State University
Jeffrey.Wilson@asu.edu

Appendix

To be able to fit the Bayesian logistic regression model, the first step is to determine which variables are time-independent and which are time-dependent. In our example, using the Add Health dataset, the outcome was obesity status (BMI>=30), race (white vs non-white) was a time-independent covariate while depression score (feelingscale) and physical activity levels (acitivityscale) were time-dependent covariates.

For time-dependent covariates, depression score and physical activity level, we tested for valid moment conditions for one, two and three time-period lags. Regression coefficients will only be estimated for time-period lags with valid moments. We did not have to test the moment conditions for time-independent covariates and for cross-sectional measurements of time-dependent covariates (time-dependent covariates measured at the same time as the outcome (BMI>=30)). To test for valid moment conditions, we used the *%partitionedGMM2* macro that can be found at https://github.com/ElsaVazquez29/Bayesian-GMM that tests for valid moments and provides a dataset that contains lagged time-dependent covariates with valid moments as well as outcome, time variable, subject id, time-independent covariates and cross-sectional vectors for time-dependent covariates. The *%partitionedGMM2* macro is a modified version of the *%partitionedGMM* macro by Irimata & Wilson (2018), it uses the same syntax and can be found at https://github.com/kirimata/Partitioned-GMM.

```
%partitionedGMM2(file=add_health, timeVar=wave, outVar=bmi,
 predVarTD= feelingscale activityscale ,
 idVar=id, alpha=0.05, predVarTI=race_, distr=bin,
 optim=NLPCG, MC=LWY);
```

Which produced the following output,

Partitioned GMM

Moment Condition Notes
There are no valid moment conditions for 1 covariate relationship(s).

These covariate relationships will be omitted in the analysis.

Each row of TypeMtx is the shifted type vector for each of the predictors, by individual test

TypeMtxCombo3

	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12	COL13	COL14	COL15	COL16
ROW1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
ROW2	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
ROW3	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
ROW4	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
ROW5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
ROW6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

The %partitionedGMM2 macro produces a dataset call mydata3 that contains the following

Partitioned GMM

Obs	ID	BMI	WAVE	RACE_	FEELINGSCALE_0	ACTIVITYSCALE_0	FEELINGSCALE_1	ACTIVITYSCALE_1	FEELINGSCALE_2	ACTIVITYSCALE_2	ACTIVITYSCALE_3
1	57101310	0	1	1	1.10526	0.37500	0.00000	0.0000	0.00000	0.000	0.000

It was possible to observe that depression score had valid moment conditions only at cross-sectional, across a one time-period lag and across a two time-period lag but not across a three time-period lag, while physical activity level had valid moment conditions across all time-period lags.

We then used PROC TRANSREG to obtain the design matrix that we used in PROC MCMC to calculate sample moments and the optimal weight matrix, as well as the objective quadratic form used to get the quasi-likelihood function.

```
proc transreg data=mydata3 design;
model class(RACE_ /zero=last);
id bmi race_FEELINGSCALE_0  ACTIVITYSCALE_0 FEELINGSCALE_1 ACTIVITYSCALE_1
FEELINGSCALE_2  ACTIVITYSCALE_2 ACTIVITYSCALE_3;
output out=design_bmi;
run;
```

Obs	_TYPE_	_NAME_	Intercept	RACE_0	BMI	RACE_	FEELINGSCALE_0	ACTIVITYSCALE_0	FEELINGSCALE_1	ACTIVITYSCALE_1	FEELINGSCALE_2	ACTIVITYSCALE_2	ACTIVITYSCALE_3
1	SCORE	0	1	0	0	1	1.10526	0.37500	0.00000	0.0000	0.00000	0.000	0.000
2	SCORE	0	1	0	0	1	0.78947	2.75000	1.10526	0.3750	0.00000	0.000	0.000
3	SCORE	0	1	0	0	1	1.60000	1.37500	0.78947	2.7500	1.10526	0.375	0.000
4	SCORE	1	1	0	1	1	1.00000	0.71429	1.60000	1.3750	0.78947	2.750	0.375
5	SCORE	0	1	1	0	0	0.63158	1.62500	0.00000	0.0000	0.00000	0.000	0.000
6	SCORE	0	1	1	0	0	0.78947	3.50000	0.63158	1.6250	0.00000	0.000	0.000
7	SCORE	0	1	1	0	0	1.00000	1.31250	0.78947	3.5000	0.63158	1.625	0.000
8	SCORE	0	1	1	0	0	0.75000	1.21429	1.00000	1.3125	0.78947	3.500	1.625

```
data mcmc_bmi;
set desing_bmi;
drop _TYPE_ _NAME_ RACE_0;
run;
```

Obs	Intercept	BMI	RACE_	FEELINGSCALE_0	ACTIVITYSCALE_0	FEELINGSCALE_1	ACTIVITYSCALE_1	FEELINGSCALE_2	ACTIVITYSCALE_2	ACTIVITYSCALE_3
1	1	0	1	1.10526	0.37500	0.00000	0.0000	0.00000	0.000	0.000
2	1	0	1	0.78947	2.75000	1.10526	0.3750	0.00000	0.000	0.000
3	1	0	1	1.60000	1.37500	0.78947	2.7500	1.10526	0.375	0.000
4	1	1	1	1.00000	0.71429	1.60000	1.3750	0.78947	2.750	0.375
5	1	0	0	0.63158	1.62500	0.00000	0.0000	0.00000	0.000	0.000
6	1	0	0	0.78947	3.50000	0.63158	1.6250	0.00000	0.000	0.000
7	1	0	0	1.00000	1.31250	0.78947	3.5000	0.63158	1.625	0.000
8	1	0	0	0.75000	1.21429	1.00000	1.3125	0.78947	3.500	1.625

We used the mcmc_bmi dataset to fit our Bayesian logistic regression model. The PROC MCMC code we used was the following.

```
/*with data read into arrays*/
proc mcmc data=mcmc_bmi nbi=300000 nmc=100000 thin=10;

/*array with regression coefficients that will be estimated*/
array beta[9]; /*we will estimate a total of 9, intercept, race,
depression, physical activity, depression across lag-1, physical
activity across lag-1, depression across lag-2, physical activity
across lag-2 and physical activity across lag-3*/


/*array with data design matrix*/
array data[10848,9]/nosymbols; /*there are 10848 observations and 9
regression coefficients to be estimated corresponding to variables
mentioned above*/

/*array with outcome */
array y[10848,1]/nosymbols; /*y is the vector of outcomes BMI>=30*/
/*vector where estimated probabilities for BMI>=30 will be stored at
each iterations*/
array yhat[10848,1];

/*array that will store estimated logits at each iteration*/
array logitss[10848,1];

/*array that will store estimated residuals at each iteration*/
array resid[10848,1];

/*array to calculate D matrix*/
array D[9, 10848];

/*array to calculate U matrix*/
array U[9, 10848];
```

```
/*array to store sums for sample moments for observations*/
array UNtemp[9,1];

/*vector with sample moments*/
array UN[9,1];
array UNT[1,9];

/*at each iteration the sum of sample moments starts at 0 */
call fillmatrix(UNtemp, 0);

/*arrays that will help calculate optimal weight matrix at each
iteration*/
ARRAY U2TEMP[9,9];
ARRAY UTEMP2[9,1];
ARRAY TRANSVECTOR[1,9];
ARRAY UIS[9,9];

/*arrays for optimal weight matrix*/
ARRAY WN[9,9];
ARRAY IWN[9,9];

/*arrays that will help calculate quasi-likelihood at each iteration*/
ARRAY FIRST_FACTOR[1, 9];
ARRAY QN [1,1];

/*steps to store dataset into arrays that will be used to get the
quasi-likelihood*/
begincnst;
rc=read_array("mcmc_bmi",y, "bmi");
rc=read_array("mcmc_bmi", data, "intercept", "RACE_",
"feelingscale_0", "ACTIVITYSCALE_0", "feelingscale_1",
"ACTIVITYSCALE_1",
  "feelingscale_2", "ACTIVITYSCALE_2", "ACTIVITYSCALE_3");
endcnst;

/*declaring parameters to be estimated (the beta vector)*/
parms (beta:);

/*prior distributions for regression coefficients*/
prior beta1~ normal(0, var=10000); /*non-informative prior for b0
regression coefficient*/
prior beta2~ normal(-0.113, var=0.001089); /*race*/
prior beta3~ normal(0.457, var=0.007744);/*depression score at cross*/
prior beta4~ normal(-0.073, var=0.001156); /*physical act at cross*/
prior beta5~ normal(0.457, var=0.007744); /*depression at lag-1*/
prior beta6~ normal(-0.073, var=0.001156); /*physical act at lag-1*/
prior beta7~ normal(0.457, var=0.00774); /*depression at lag-2*/
prior beta8~ normal(-0.073, var=0.001156); /*physical act at lag-2*/
prior beta9~ normal(-0.073, var=0.001156); /*physical act at lag-3*/
```

```
/*calculating logit link using beta coefficients and design matrix
(X)*/
call mult(data, beta, logitss);

/*calculating yhat, residuals, D matrix, U matrix that will be used to
obtain sample moments*/
do i=1 to 10848;
 yhat[i]=exp(logitss[i])/(1+exp(logitss[i]));
 resid[i]=y[i]-yhat[i];
  do j=1 to 9;
    D[j,i]=yhat[i]*data[i,j]*(1-yhat[i]);
    U[j,i]=D[j,i]*resid[i];
    /*will add U[j,i] to UNtemp which will be used to calculate mean
sample moments*/
    call addmatrix(UNtemp, U[j,i], UNtemp);
  end;
end;
/*calculate mean of UNtemp, this is UN and it is the vector of sample
moments for each regression coefficient*/
call mult(1/2712, UNtemp, UN); /*there were 2712 subjects in our
data*/

/*will use UN and UNT to calculate quadratic function, that will serve
as quasi-posterior distribution*/
call transpose(UN, UNT);

/*Will calculate sample covariance matrix for each subject*/
/*will save sum of sample covariace matrix across subjects in U2TEMP*/
CALL FILLMATRIX (U2TEMP, 0);
do i=1 to 10848;
     do j=1 to 9;
      /*UTEMP2 keeps vector of moment conditions for subject i*/
      UTEMP2[j,1]=U[j,I];
     end;
     /*TRANSVECTOR is the transpose of UTEMP2 vector of moment
conditions for subject i*/
     CALL TRANSPOSE(UTEMP2, TRANSVECTOR);
      /*UIS is the covariance matrix for vector of moment conditions
for subject i*/
     CALL MULT(UTEMP2, TRANSVECTOR, UIS);
      /*Will add covariance matrix for subject i to sum of covariance
matrices for all subjects*/
     call addmatrix(u2temp, uis, u2temp);
END;


/*calculate mean of covariance matrices for moment conditions of all
subjects*/
CALL MULT(1/2712, U2TEMP, WN); /*there were 2712 subjects in our
data*/

/*calculate optimal weight matrix*/
```

```
CALL INV(WN, IWN);

/*make calculations to get quasi-likelihood function*/
CALL MULT(UNT, IWN, FIRST_FACTOR);

/*QN is the objective quadratic form in GMM*/
CALL MULT(FIRST_FACTOR, UN, QN);

/*-1/2QN is the quasi-likelihood function from which posterior beta
coefficients will be obtained*/

MODEL GENERAL(-0.5*QN[1,1]);

run;
```

The following is part of the output obtained by running code above: posterior summaries, posterior intervals (95% posterior CI for odds ratios shown in poster come form equal tail CI) , posterior autcorrelations, Geweke diagnostics, effective sample sizes and diagnostics plots.

The MCMC Procedure

Posterior Summaries

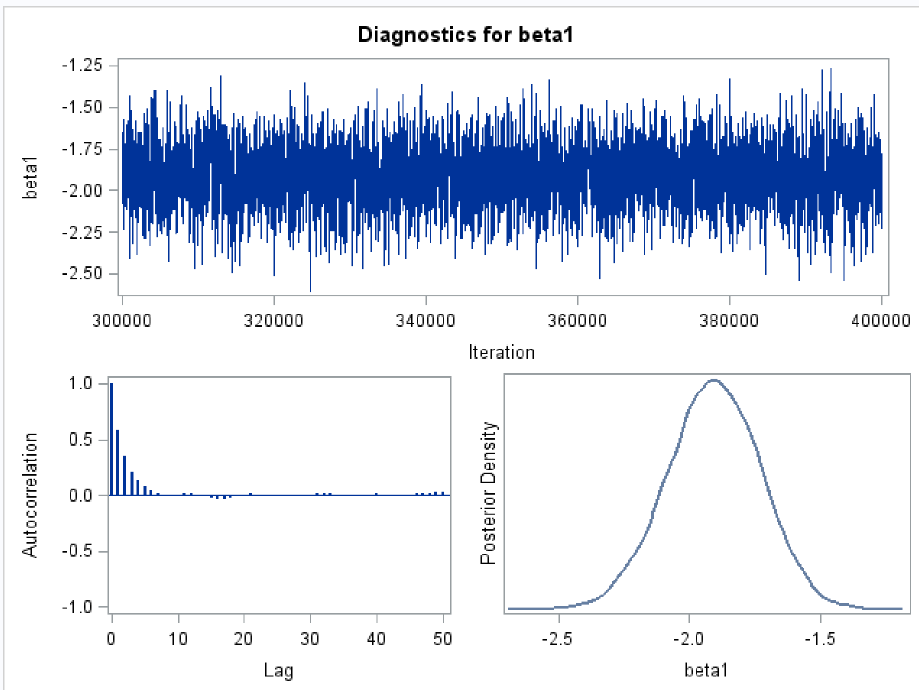
Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta1	10000	-1.9079	0.1770	-2.0240	-1.9055	-1.7874
beta2	10000	-0.1129	0.0324	-0.1347	-0.1133	-0.0910
beta3	10000	0.4584	0.0882	0.3978	0.4596	0.5168
beta4	10000	-0.0749	0.0332	-0.0971	-0.0750	-0.0520
beta5	10000	0.4580	0.0886	0.3994	0.4589	0.5183
beta6	10000	-0.0729	0.0341	-0.0960	-0.0737	-0.0495
beta7	10000	0.4584	0.0878	0.3990	0.4566	0.5179
beta8	10000	-0.0740	0.0338	-0.0965	-0.0741	-0.0519
beta9	10000	-0.0740	0.0342	-0.0969	-0.0738	-0.0511

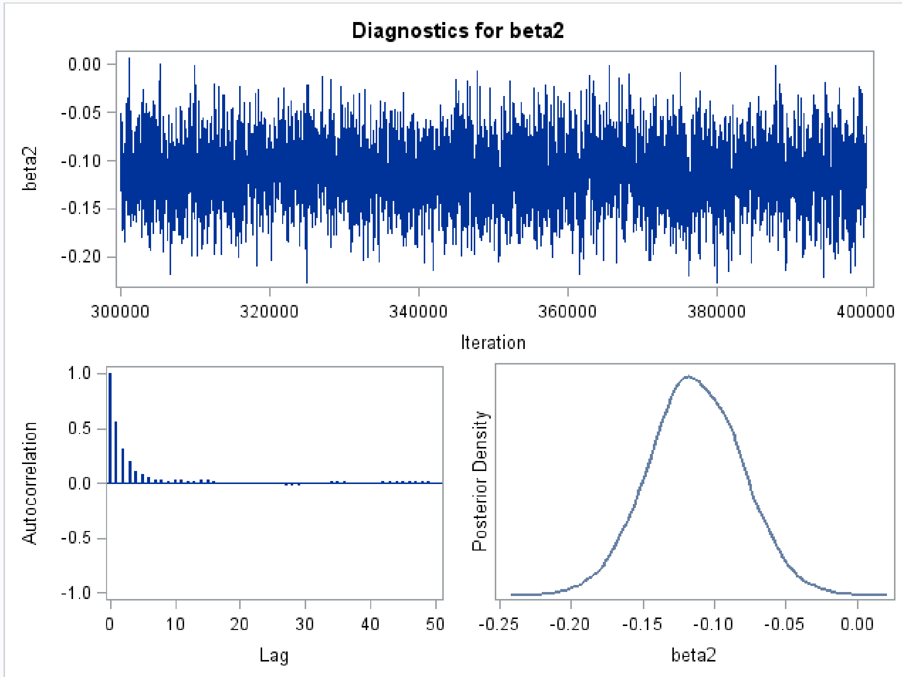
Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta1	0.050	-2.2617	-1.5705	-2.2566	-1.5669
beta2	0.050	-0.1761	-0.0493	-0.1769	-0.0506
beta3	0.050	0.2892	0.6347	0.2867	0.6310
beta4	0.050	-0.1406	-0.0105	-0.1407	-0.0105
beta5	0.050	0.2807	0.6293	0.2807	0.6293
beta6	0.050	-0.1391	-0.00585	-0.1386	-0.00565
beta7	0.050	0.2885	0.6349	0.2832	0.6283
beta8	0.050	-0.1413	-0.00692	-0.1421	-0.00821
beta9	0.050	-0.1413	-0.00815	-0.1392	-0.00635

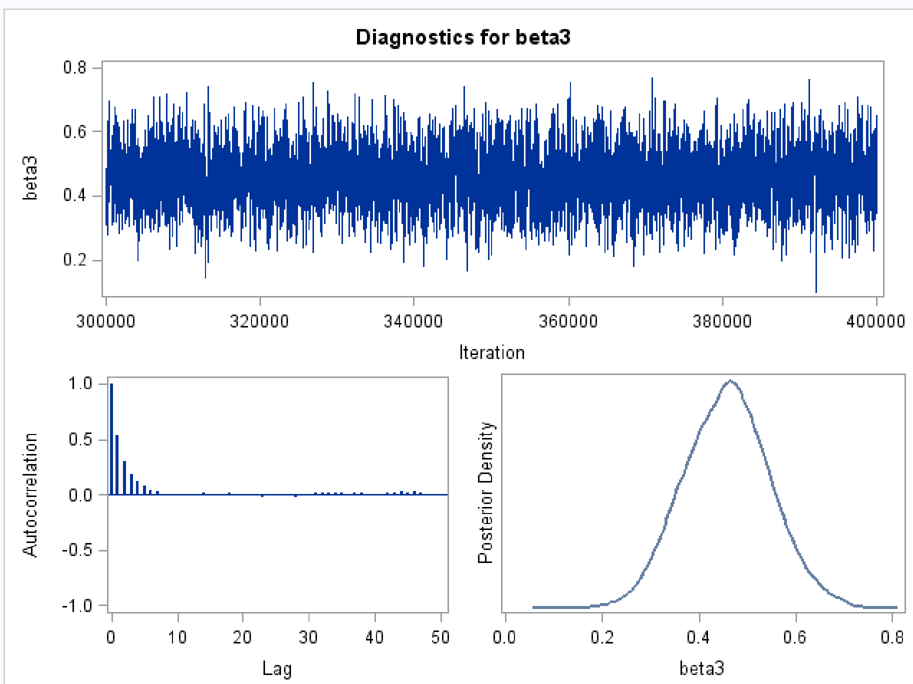
Posterior Autocorrelations				
Parameter	Lag 1	Lag 5	Lag 10	Lag 50
beta1	0.5887	0.0797	0.0057	0.0254
beta2	0.5545	0.0733	0.0244	0.0049
beta3	0.5333	0.0736	-0.0060	-0.0079
beta4	0.5503	0.0703	0.0219	0.0071
beta5	0.6508	0.1378	0.0077	0.0267
beta6	0.5300	0.0687	0.0322	0.0219
beta7	0.5753	0.1382	0.0371	0.0209
beta8	0.5066	0.0566	0.0359	0.0087
beta9	0.3975	0.0465	0.0016	0.0082

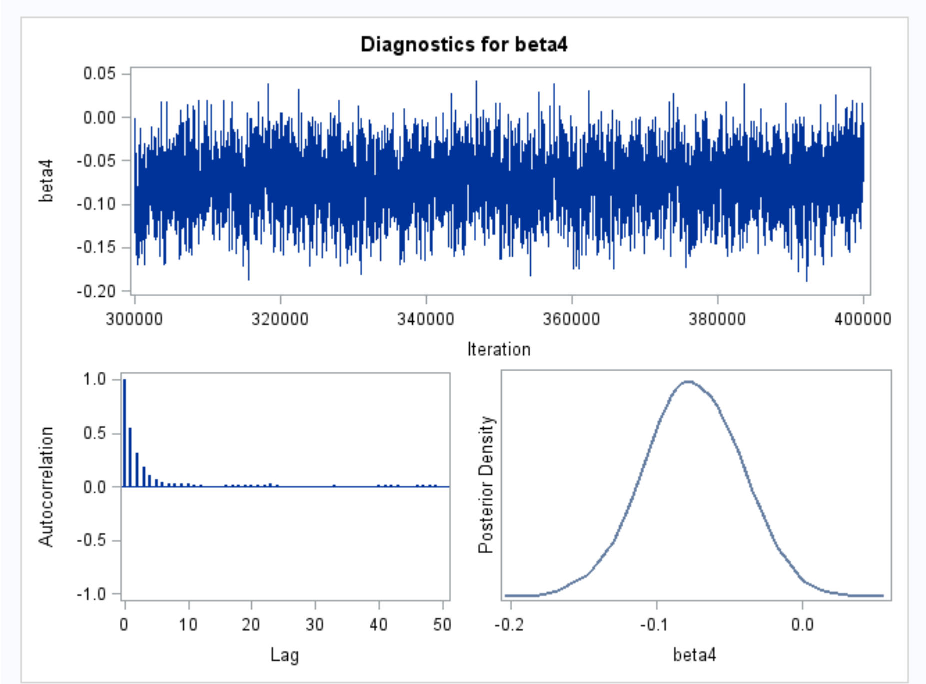
Geweke Diagnostics		
Parameter	z	Pr > |z|
beta1	-0.2702	0.7870
beta2	-0.8218	0.4112
beta3	1.4456	0.1483
beta4	-0.4580	0.6469
beta5	0.4948	0.6208
beta6	-0.5310	0.5954
beta7	-0.1477	0.8826
beta8	-0.6486	0.5166
beta9	-0.3202	0.7488

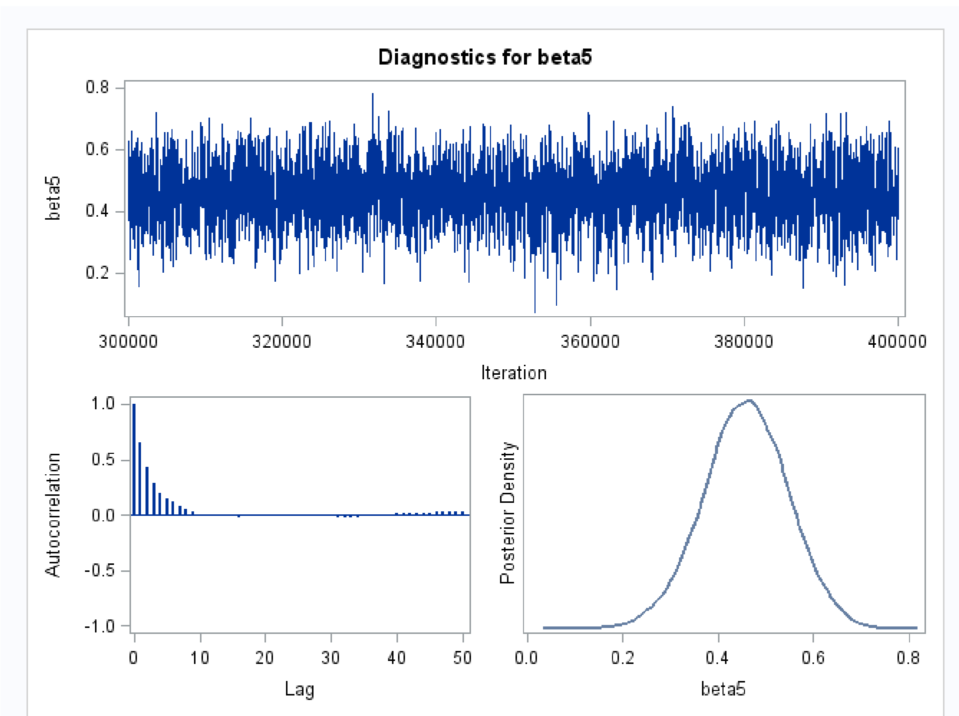
Effective Sample Sizes			
Parameter	ESS	Autocorrelation Time	Efficiency
beta1	2614.2	3.8253	0.2614
beta2	2491.4	4.0138	0.2491
beta3	2816.7	3.5502	0.2817
beta4	2660.8	3.7583	0.2661
beta5	2014.8	4.9634	0.2015
beta6	2391.5	4.1814	0.2392
beta7	2034.1	4.9162	0.2034
beta8	2773.2	3.6059	0.2773
beta9	3715.8	2.6912	0.3716

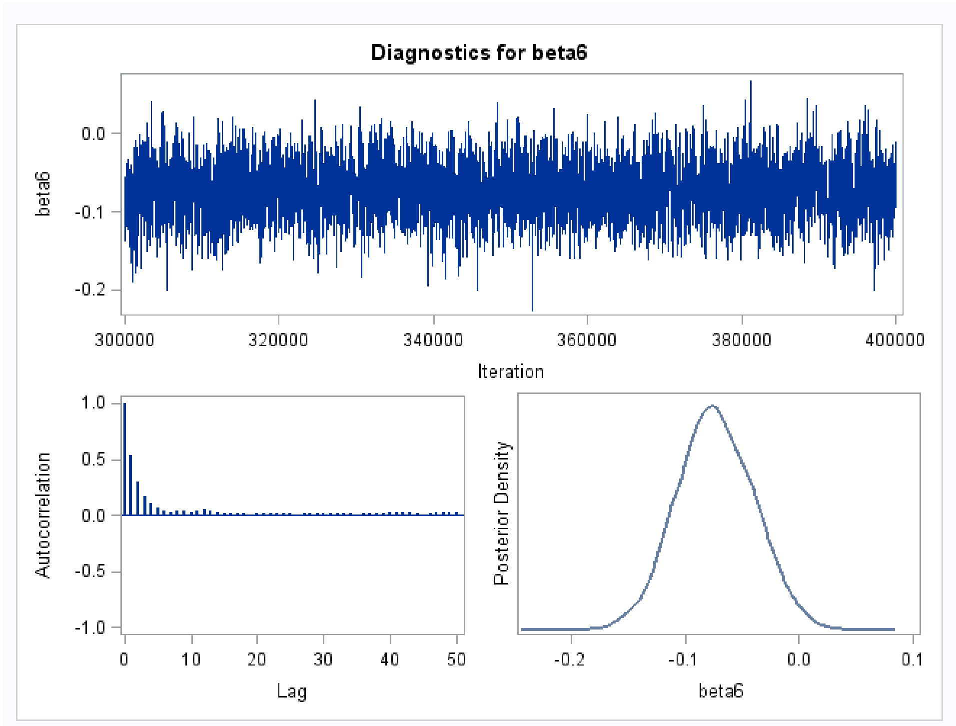


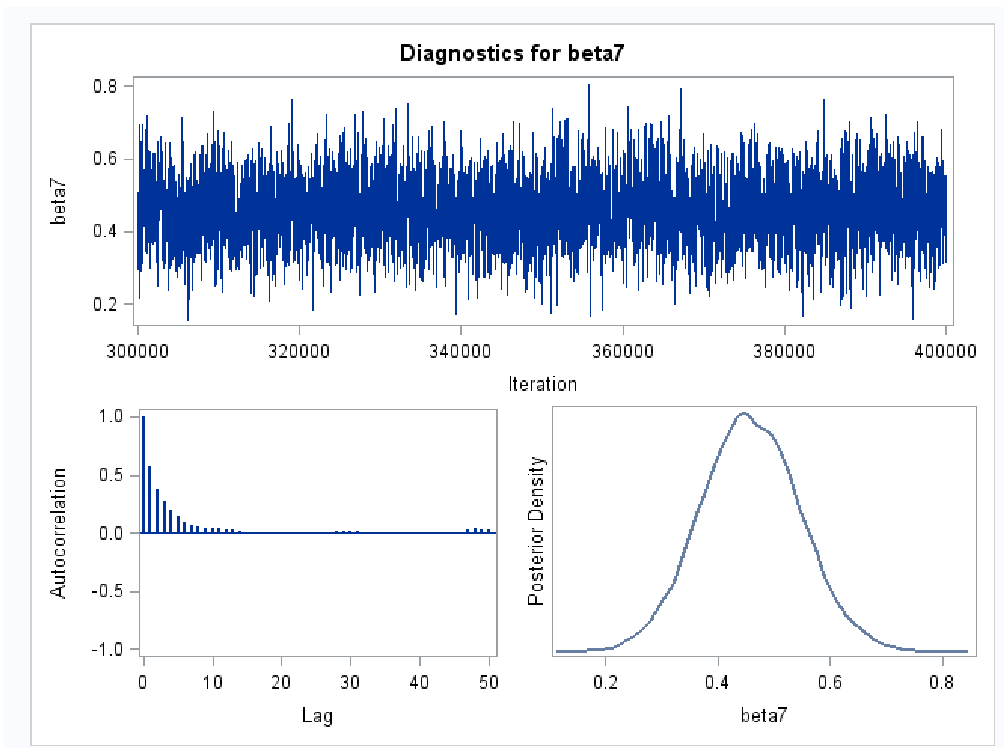

Diagnostics for beta2
beta2
Iteration
Autocorrelation
Lag
Posterior Density
beta2

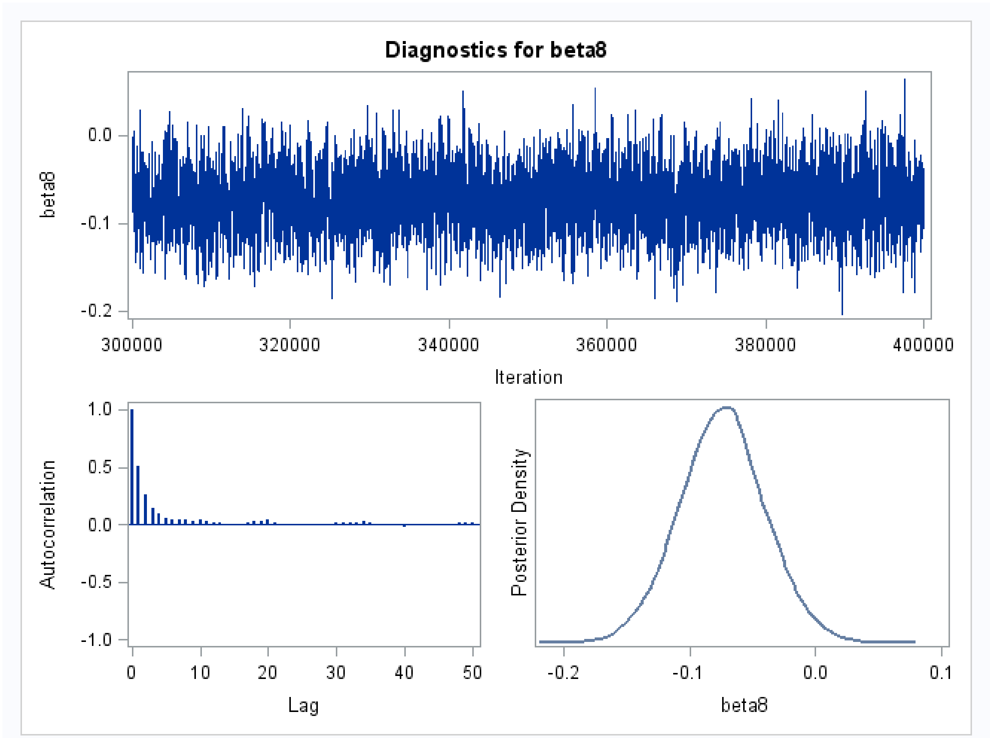

Diagnostics for beta3
beta3
Iteration
Autocorrelation
Lag
Posterior Density
beta3

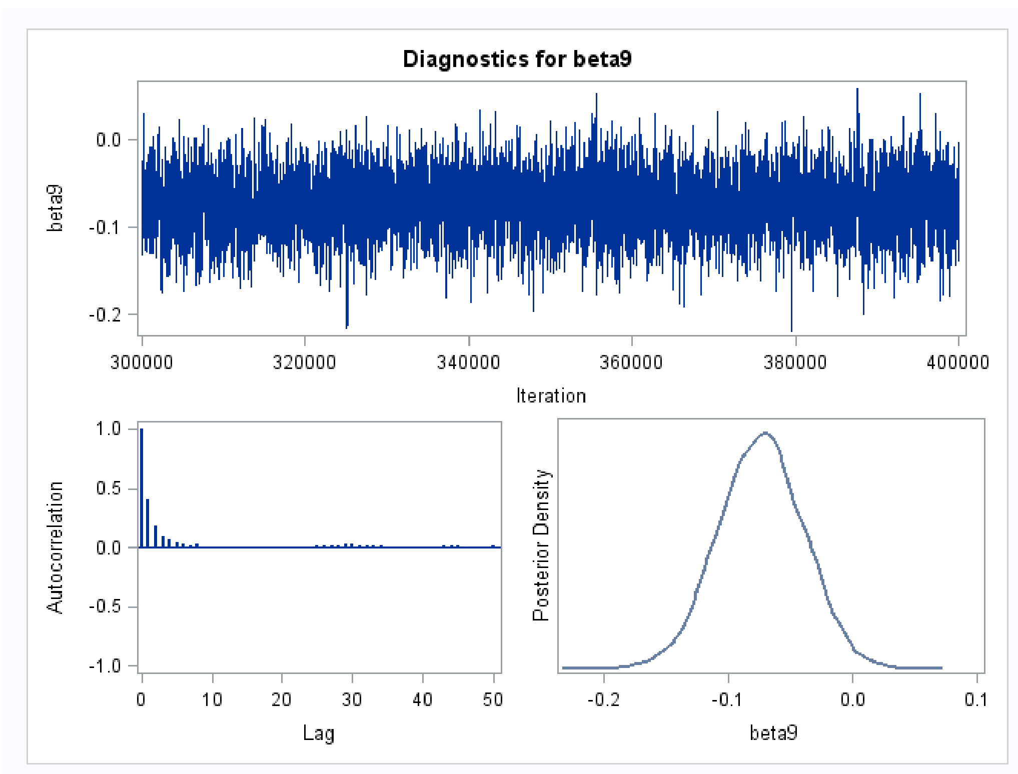

Diagnostics for beta4
0.05
0.00
-0.05
-0.10
-0.15
-0.20
beta4
300000
320000
340000
360000
380000
400000
Iteration
1.0
0.5
0.0
-0.5
-1.0
Autocorrelation
0
10
20
30
40
50
Lag
Posterior Density
-0.2
-0.1
0.0
beta4


Diagnostics for beta5
0.8
0.6
0.4
0.2
beta5
300000
320000
340000
360000
380000
400000
Iteration
1.0
0.5
0.0
-0.5
-1.0
Autocorrelation
0
10
20
30
40
50
Lag
Posterior Density
0.0
0.2
0.4
0.6
0.8
beta5


Diagnostics for beta6
beta6
0.0
-0.1
-0.2
300000
320000
340000
360000
380000
400000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
-0.2
-0.1
0.0
0.1
beta6


Diagnostics for beta7
beta7
0.8
0.6
0.4
0.2
300000
320000
340000
360000
380000
400000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
0.2
0.4
0.6
0.8
beta7


Diagnostics for beta8
beta8
0.0
-0.1
-0.2
300000
320000
340000
360000
380000
400000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
-0.2
-0.1
0.0
0.1
beta8


Diagnostics for beta9
beta9
0.0
-0.1
-0.2
300000
320000
340000
360000
380000
400000
Iteration
Autocorrelation
1.0
0.5
0.0
-0.5
-1.0
0
10
20
30
40
50
Lag
Posterior Density
-0.2
-0.1
0.0
0.1
beta9