OM 386 Assignment 2

Due: March 3rd, 11:59pm
By: Callie Gilmore (cgg756)

Binary Data Regression Models for Bank Customer Attrition

This exercise is similar to the bank customer acquisition problem that we discussed in our class. Imagine that you are hired as a consultant. For the analysis, the management has given you access to 2505 customers, among whom 449 (about 18%) have closed their accounts within one year. As a consultant, you would like to know what demographic and behavioral variables contribute to higher attrition/churn rates among these customers.

The data file is "Bank_Retention_Data.csv" on Canvas. It has the following variables:

Age	The customer's age
Income	The customer's income
HomeVal	The customer's home value
TractID	A label/ID of the census tract of the customer's residence
Tenure	How long this person has been a customer of the bank
DirectDeposit	Indicator dummy=1 if the customer uses direct deposit and 0 otherwise
LoanInd	Loan indicator dummy = 1 if the customer has ever taken loans from her bank and 0 if not
Dist	Distance from customer's home to the nearest bank branch
MktShare	Bank's market share in the customer's market
Churn	Indicator dummy = 1 if the customer has closed her/his accounts (s/he has churned) with the bank and 0 if not

1). Read the data into R. Convert TractID into a factor variable. (15 points)

```
> bankdata = read.csv('Bank_Retention_Data.csv')
> bankdata$TractID = as.factor(bankdata$TractID)
```

Estimate the following binary data regression model using the R function glm( ).

$$Churn_i \sim \beta_0 + \beta_1 \times Age_i + \beta_2 \times Income_i + \beta_3 \times HomeVal_i + \beta_4 \times Tenure_i + \beta_5 \times DirectDeposit_i + \beta_6 \times LoanInd_i + \beta_7 \times Dist_i + \beta_8 \times MktShare_i$$

Use both of the logit (for logistic regression) and probit (for probit regression) link functions of the binomial family and paste results here.

```
> ### Logit Regression
> churnlogit = glm(Churn ~ Age + Income + HomeVal + Tenure + DirectDeposit +
+                  Loan + Dist + MktShare, data = bankdata, family = binomial(link = "logit"))
> summary(churnlogit)

Call:
glm(formula = Churn ~ Age + Income + HomeVal + Tenure + DirectDeposit +
    Loan + Dist + MktShare, family = binomial(link = "logit"),
    data = bankdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2054  -0.6823  -0.5328  -0.3401   2.6266

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.606224   0.296596  -2.044 0.040960 *
Age           -0.016103   0.004150  -3.881 0.000104 ***
Income         0.107067   0.015985   6.698 2.11e-11 ***
HomeVal       -0.026059   0.005477  -4.758 1.95e-06 ***
Tenure        -0.029709   0.006549  -4.536 5.73e-06 ***
DirectDeposit -0.465836   0.110617  -4.211 2.54e-05 ***
Loan           0.099376   0.124380   0.799 0.424310
Dist           0.267618   0.061958   4.319 1.57e-05 ***
MktShare      -0.082440   0.325551  -0.253 0.800089
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2355.9  on 2504  degrees of freedom
Residual deviance: 2189.4  on 2496  degrees of freedom
AIC: 2207.4

Number of Fisher Scoring iterations: 5
```

```
> ### Probit Regression
> churnprobit = glm(Churn ~ Age + Income + HomeVal + Tenure + DirectDeposit +
+                   Loan + Dist + MktShare, data = bankdata, family = binomial(link = "probit"))
> summary(churnprobit)

Call:
glm(formula = Churn ~ Age + Income + HomeVal + Tenure + DirectDeposit +
    Loan + Dist + MktShare, family = binomial(link = "probit"),
    data = bankdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1714  -0.6886  -0.5374  -0.3252   2.7140

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.397967   0.168825  -2.357   0.0184 *
Age           -0.009050   0.002314  -3.910 9.22e-05 ***
Income         0.059194   0.008871   6.673 2.51e-11 ***
HomeVal       -0.014360   0.002922  -4.914 8.90e-07 ***
Tenure        -0.016430   0.003550  -4.628 3.69e-06 ***
DirectDeposit -0.263070   0.062851  -4.186 2.84e-05 ***
Loan           0.057756   0.070224   0.822   0.4108
Dist           0.154712   0.036313   4.261 2.04e-05 ***
MktShare      -0.045443   0.184547  -0.246   0.8055
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2355.9  on 2504  degrees of freedom
Residual deviance: 2188.6  on 2496  degrees of freedom
AIC: 2206.6

Number of Fisher Scoring iterations: 6
```

How do you interpret β_1, β_2, β_3, β_4, β_5, β_6, β_7, β_8? Are they statistically significant in the logistic and probit models? Please also calculate the AIC and BIC of the logistic and probit models using the R functions AIC( ) and BIC( ). Which model (logistic or probit) fits the data better based on AIC and BIC?

From the results above, the following variables are statistically significant as their p-values are lower than 0.05: Age, Income, HomeVal, Tenure, DirectDeposit and Dist.

```
> ### AIC and BIC
> cat('Logit AIC:', AIC(churnlogit))
Logit AIC: 2207.358
> cat('Logit BIC:', BIC(churnlogit))
Logit BIC: 2259.793
> cat('Probit AIC:', AIC(churnprobit))
Probit AIC: 2206.626
> cat('Probit BIC:', BIC(churnprobit))
Probit BIC: 2259.06
```

Based on results above, the Probit Model is better because its AIC and BIC are lower; however, it is important to note that they are not much lower so the Probit Model barely outperformed Logit Model.

2). Next we will use a random effect grouped by TractID in the logistic regression. Use the function glmer( ) in the "lme4" package in R to fit

$$Churn_i \sim \beta_{0p} + \beta_1 \times Age_i + \beta_2 \times Income_i + \beta_3 \times HomeVal_i + \beta_4 \times Tenure_i$$
$$+ \beta_5 \times DirectDeposit_i + \beta_6 \times LoanInd_i + \beta_7 \times Dist_i + \beta_8 \times MktShare_i$$

where β_{0p} is the random effect for the p-th census tract (TractID). Paste results here.

```
> ## Question 2
> library(lme4)
Loading required package: Matrix
Warning message:
package 'lme4' was built under R version 4.0.3
> churnrand = glmer(Churn ~ (1|TractID) + Age + Income + HomeVal + Tenure + DirectDeposit + Loan + Dist + MktShare, data=bankdata, family = binomial)
Warning messages:
1: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model failed to converge with max|grad| = 0.00217361 (tol = 0.002, component 1)
2: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model is nearly unidentifiable: very large eigenvalue
 - Rescale variables?
```

```
> summary(churnrand)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: Churn ~ (1 | TractID) + Age + Income + HomeVal + Tenure + DirectDeposit +      Loan + Dist + MktShare
   Data: bankdata

     AIC      BIC   logLik deviance df.resid
  2208.7   2266.9  -1094.3   2188.7     2495

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.0912 -0.5118 -0.3895 -0.2447  5.3475

Random effects:
 Groups  Name        Variance Std.Dev.
 TractID (Intercept) 0.01994  0.1412
Number of obs: 2505, groups:  TractID, 26

Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.564391   0.305960  -1.845   0.0651 .
Age           -0.016479   0.004178  -3.944 8.00e-05 ***
Income         0.107015   0.016078   6.656 2.81e-11 ***
HomeVal       -0.026706   0.005691  -4.693 2.69e-06 ***
Tenure        -0.029231   0.006564  -4.453 8.46e-06 ***
DirectDeposit -0.461463   0.111004  -4.157 3.22e-05 ***
Loan           0.099944   0.124635   0.802   0.4226
Dist           0.266979   0.063386   4.212 2.53e-05 ***
MktShare       0.007963   0.373353   0.021   0.9830
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) Age    Income HomeVl Tenure DrctDp Loan   Dist
Age         -0.647
Income      -0.221  0.055
HomeVal     -0.206 -0.060 -0.534
Tenure       0.014 -0.285 -0.075  0.077
DirectDepst -0.175  0.012 -0.050  0.081 -0.115
Loan        -0.073  0.073 -0.007 -0.059 -0.105 -0.083
Dist        -0.324  0.000 -0.012 -0.150 -0.013 -0.008 -0.012
MktShare    -0.359 -0.006 -0.031  0.060 -0.140  0.005 -0.008  0.260
optimizer (Nelder_Mead) convergence code: 0 (OK)
Model failed to converge with max|grad| = 0.00217361 (tol = 0.002, component 1)
Model is nearly unidentifiable: very large eigenvalue
 - Rescale variables?
```

Check the fixed effect estimates of β_1, β_2, β_3, β_4, β_5, β_6, β_7, β_8 again. Are they still statistically significant? Please also calculate the AIC and BIC of this model using the R functions AIC( ) and BIC( ). Based on the AIC and BIC, compare the model fit of this model to the models in (1). (15 points)

```
> cat('Random AIC:', AIC(churnrand))
Random AIC: 2208.686
> cat('Random BIC:', BIC(churnrand))
Random BIC: 2266.947
```

From the results above, the following variables are still statistically significant: Age, Income, HomeVal, Tenure, DirectDeposit, Dist.

3). For the model in (1), use the MCMCpack function MCMChlogit() to estimate the same parameters with Bayesian estimation. Because the model only has a random intercept, specify random=~1 and r=2, R=1 in the MCMChlogit() function. Please also set burnin=10000, mcmc=20000 and thin=20.

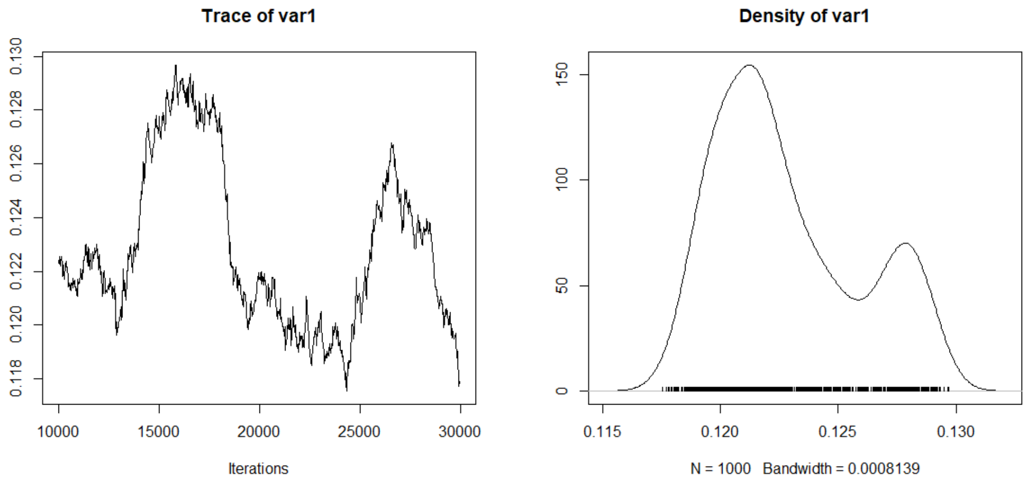
```
> ## Question 3
> library(MCMCpack)
Loading required package: coda
Loading required package: MASS
##
## Markov Chain Monte Carlo Package (MCMCpack)
## Copyright (C) 2003-2021 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
##
## Support provided by the U.S. National Science Foundation
## (Grants SES-0350646 and SES-0350613)
##
Warning messages:
1: package ‘MCMCpack’ was built under R version 4.0.3
2: package ‘coda’ was built under R version 4.0.3
> churnmcmclogit = MCMChlogit(Churn ~ Age + Income + HomeVal + Tenure + DirectDeposit + Loan + Dist + MktShare, data=bankdata, group = 'TractID', random=~1, r=2, R=1, burnin=10000, mcmc=20000, thin=20)

Running the Gibbs sampler. It may be long, keep cool :)

**********:10.0%, mean accept. rate=0.444
**********:20.0%, mean accept. rate=0.489
**********:30.0%, mean accept. rate=0.458
**********:40.0%, mean accept. rate=0.528
**********:50.0%, mean accept. rate=0.504
**********:60.0%, mean accept. rate=0.534
**********:70.0%, mean accept. rate=0.547
**********:80.0%, mean accept. rate=0.440
**********:90.0%, mean accept. rate=0.605
**********:100.0%, mean accept. rate=0.398
```

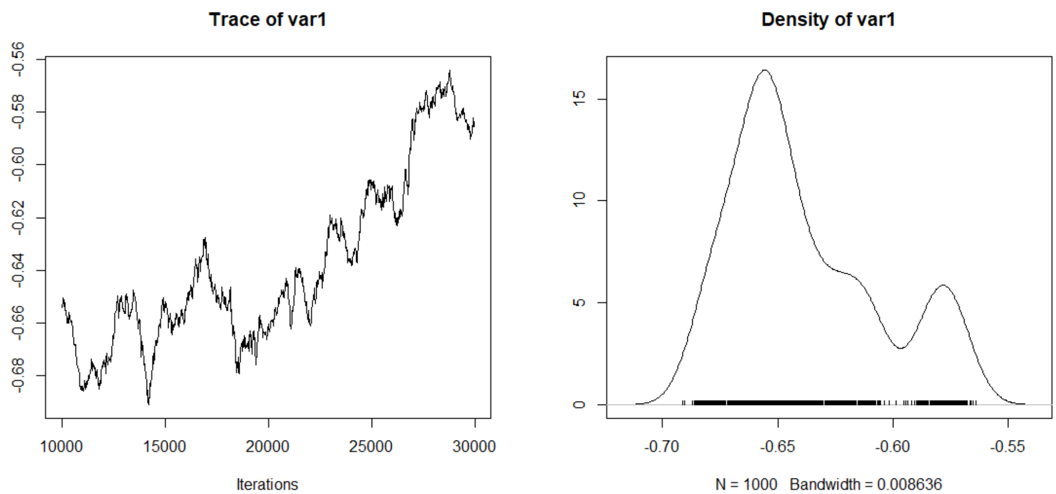
Please copy and paste the Bayesian estimation results of the fixed effects (same fixed effects as in (1)) in the model using summary("*yourBayesianModelName*"$mcmc[,1:9]). From the Bayesian posterior intervals, are the fixed effects significant at the 5% level?

```
> summary(churnmcmclogit$mcmc[,1:9])

Iterations = 10001:29981
Thinning interval = 20
Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                        Mean        SD  Naive SE Time-series SE
beta.(Intercept)   -0.25947 0.0815724 2.580e-03      0.0452272
beta.Age           -0.01804 0.0007226 2.285e-05      0.0003740
beta.Income         0.12285 0.0030567 9.666e-05      0.0019092
beta.HomeVal       -0.03213 0.0006686 2.114e-05      0.0004355
beta.Tenure        -0.03938 0.0019498 6.166e-05      0.0017056
beta.DirectDeposit -0.63830 0.0324330 1.026e-03      0.0241419
beta.Loan           0.27031 0.0337348 1.067e-03      0.0274172
beta.Dist           0.24827 0.0170810 5.401e-04      0.0096881
beta.MktShare      -0.22866 0.1150054 3.637e-03      0.0840372

2. Quantiles for each variable:

                       2.5%      25%      50%      75%    97.5%
beta.(Intercept)   -0.40488 -0.31840 -0.27135 -0.19533 -0.09033
beta.Age           -0.01918 -0.01868 -0.01817 -0.01744 -0.01674
beta.Income         0.11867  0.12046  0.12197  0.12492  0.12881
beta.HomeVal       -0.03312 -0.03271 -0.03224 -0.03178 -0.03095
beta.Tenure        -0.04248 -0.04139 -0.03895 -0.03752 -0.03695
beta.DirectDeposit -0.68343 -0.66094 -0.64954 -0.61730 -0.57135
beta.Loan           0.22642  0.24463  0.25666  0.30730  0.33526
beta.Dist           0.22056  0.23051  0.25124  0.26070  0.28204
beta.MktShare      -0.43719 -0.33105 -0.20048 -0.13206 -0.06770
```

From the results above, the following variables are statistically significant at the 5% level: Age, Income, HomeVal, Tenure, DirectDeposit, Loan, Dist and MarketShare. So, Betas1-8 are significant at 5%.

Use the plot() function to plot the posterior sampling chains and posterior densities for β_2 and β_5; copy and paste the results here. (15 points)


Trace of var1
0.118
0.120
0.122
0.124
0.126
0.128
0.130
10000
15000
20000
25000
30000
Iterations
Density of var1
0
50
100
150
0.115
0.120
0.125
0.130
N = 1000 Bandwidth = 0.0008139


Trace of var1
-0.68
-0.66
-0.64
-0.62
-0.60
-0.58
-0.56
10000
15000
20000
25000
30000
Iterations
Density of var1
0
5
10
15
-0.70
-0.65
-0.60
-0.55
N = 1000 Bandwidth = 0.008636

Probit Regression: Bayesian Estimation

In this exercise, we will practice coding the Gibbs sampler for a probit regression model using the dataset "CreditCard_LatePayment_Data.csv". The dataset has the following variables.

ConsumerID	ID's of the sampled consumers
Latepay	Whether the consumer makes a late payment in the month
Usage	Monthly credit usage activities
Balance	The customer's outstanding balance in the month

1). We would like fit the following probit regression model

$Y_{ij}^{*} = \beta_0 + \beta_1 \times Usage_{ij} + \beta_2 \times Balance_{ij} + \varepsilon_{ij}$
$Latepay_{ij} = 0 \quad \text{if} \quad Y_{ij}^{*} \leq 0$
$Latepay_{ij} = 1 \quad \text{if} \quad Y_{ij}^{*} > 0$
$\varepsilon_{ij} \sim N(0, 1)$

Please use the R function glm( ) to fit this model by MLE. Copy and paste the summary of the results here.

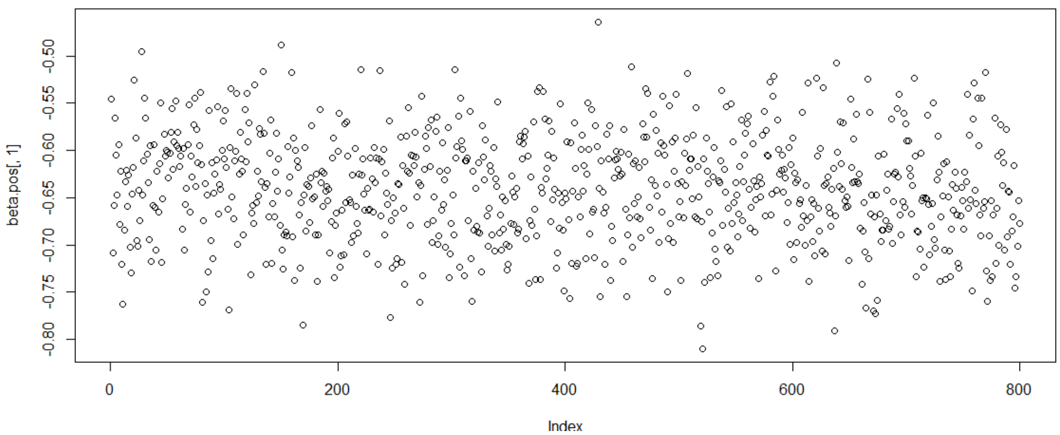
```
> # Probit Regression: Bayesian Estimation
> ## Question 1
> DataFile = "CreditCard_LatePayment_data.csv"
> LP.data = read.csv(DataFile, header=T)
> glmmodel = glm(Latepay ~ Usage + Balance, data = LP.data, family = binomial(link="probit"))
> summary(glmmodel)

Call:
glm(formula = Latepay ~ Usage + Balance, family = binomial(link = "probit"),
    data = LP.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3937  -0.6988  -0.5283  -0.4022   2.3487

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.426e-01  5.939e-02 -10.820  < 2e-16 ***
Usage       -7.368e-02  7.391e-03  -9.969  < 2e-16 ***
Balance      1.878e-04  2.311e-05   8.126 4.42e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3512.4  on 3599  degrees of freedom
Residual deviance: 3317.6  on 3597  degrees of freedom
AIC: 3323.6

Number of Fisher Scoring iterations: 4
```

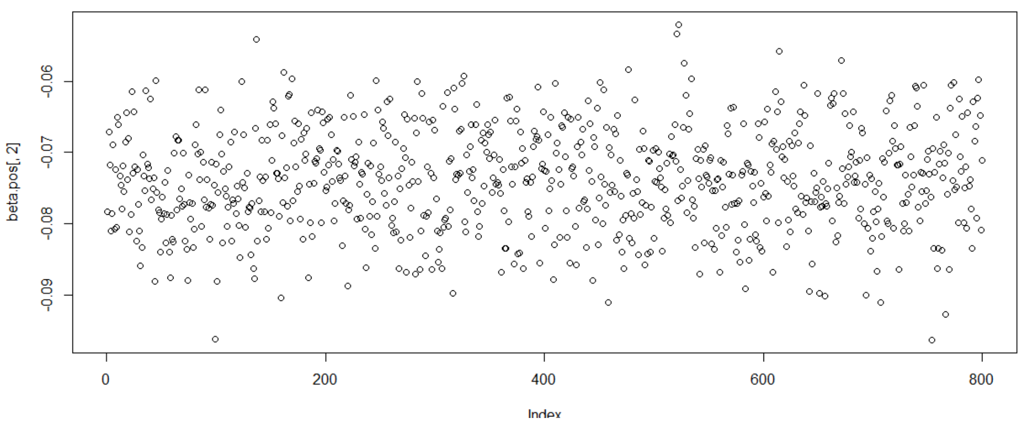
2). Next, we will fit the model above using a Gibbs sampler for Bayesian inference, which involves sampling the latent Y_{ij}^{*}. Parts of the R code are in "Assignment-2_Probit-code_blanks.r". Please read the code carefully and fill in the code in the blanks in the file. You may use the rtruncnorm( ) function in the library(truncnorm) to sample from truncated normal distributions. For the linear regression part given the sampled latent Y_{ij}^{*} in the main loop, please refer to the code BayesianLM.r on Canvas

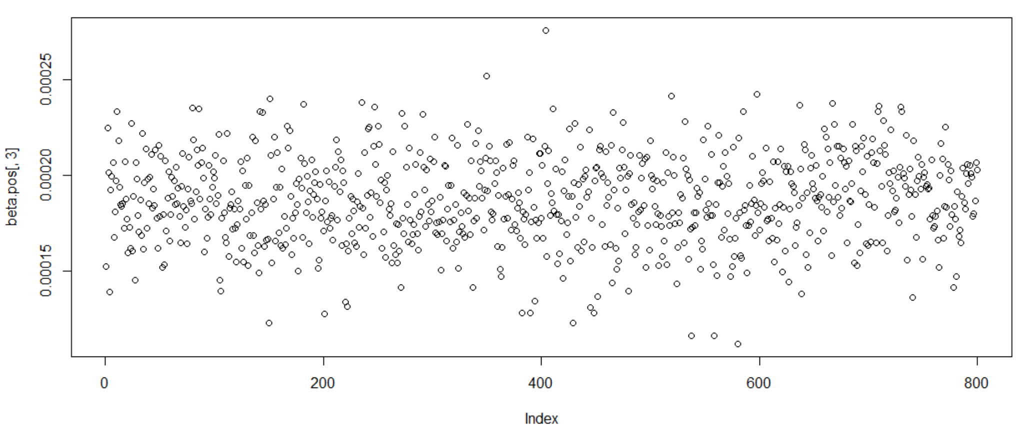
```
> ## Question 2
> library(truncnorm)
> library(mnormt)
> library(mvtnorm)
> ### stage 1. subset the data for Latepay = 1 and =0
> LP.X0 = cbind(1, as.matrix(LP.data[LP.data$Latepay==0, 3:4]))
> LP.X1 = cbind(1, as.matrix(LP.data[LP.data$Latepay==1, 3:4]))
> LP.X = cbind(1, as.matrix(LP.data[, 3:4]))
> LP.X2 = t(LP.X)%*%LP.X
> n0 = dim(LP.X0)[1]
> n1 = dim(LP.X2)[1]
> nObs = dim(LP.data)[1]
> LP.Y = rep(0, nObs)
> ### stage 2. Initial Setup for the algorithm
> NIT = 10000       #num of interations
> nBurn = 2000      #num of burn-ins
> NIT.eff = NIT - nBurn    #effective sample size
> thin.step = 10           #thinning
> NIT.thin = floor(NIT.eff/thin.step)   #effective sample size after thinning
> ### stage 3. Record Posterior samples
> beta.dim = 3
> beta.pos = matrix(0, NIT.thin, beta.dim)
> ### stage 4. priors
> #### for Beta: mNormal(mu.beta, sigma.beta)
> mu.beta = rep(0,beta.dim)
> sigma.beta = 1E6 * diag(beta.dim)
> iSigma.beta = 1E-6 * diag(beta.dim)  #inverse prior covariance matrix
> ##### initialize the loop
> curBeta = c(0.1, 0, 0)
> g = 1
> ##### main loop
> for (m in 1:NIT){
+   ###### step 1. sample the latent variable > 0 if Latepay=1, <0 if Latepay=0
+   #Please fill in the code
+   LP.Y[LP.data$Latepay==1] = rtruncnorm(n=1,a=0, b=Inf, mean = LP.X1%*%curBeta)
+   LP.Y[LP.data$Latepay==0] = rtruncnorm(n=1,a=-Inf, b=0, mean = LP.X0%*%curBeta)
+
+   ###### step 2 sample beta
+   #Please fill in the code
+   beta.pos.var = solve(LP.X2 + iSigma.beta)
+   beta.pos.mean = beta.pos.var%*%(t(LP.X)%*%LP.Y + iSigma.beta%*%mu.beta)
+   curBeta = as.vector(rmvnorm(1, mean=beta.pos.mean, sigma=beta.pos.var))
+
+   ###### save thinned samples after burn-ins
+   if ((m > nBurn) & (m%%thin.step == 0)) {
+     beta.pos[g,] = curBeta
+     g = g+1
+   }
+ }
```

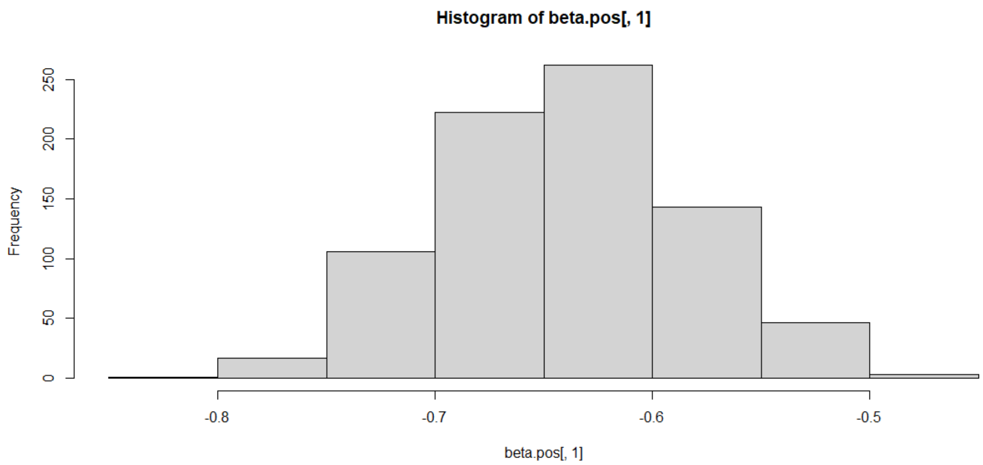
Please run the completed code. Use the plot() function to plot the posterior sampling chains and hist() to plot posterior histograms for β_0, β_1, β_2 . Copy and paste the results here. Please also calculate the 95% posterior intervals for β_0, β_1, β_2 . Copy and paste the results here.

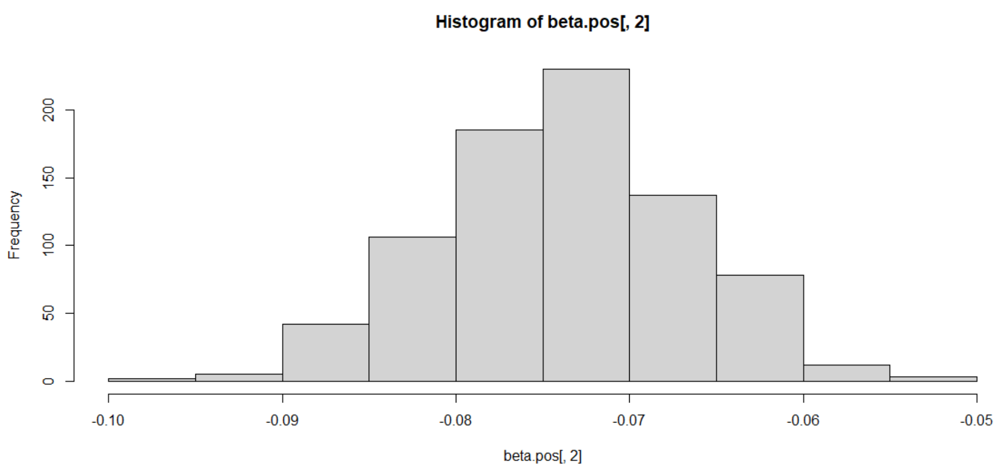
```
plot(beta.pos[,1])
plot(beta.pos[,2])
plot(beta.pos[,3])

hist(beta.pos[,1])
hist(beta.pos[,2])
hist(beta.pos[,3])

quantile(beta.pos[,1], 0.95)
quantile(beta.pos[,2], 0.95)
quantile(beta.pos[,3], 0.95)
```

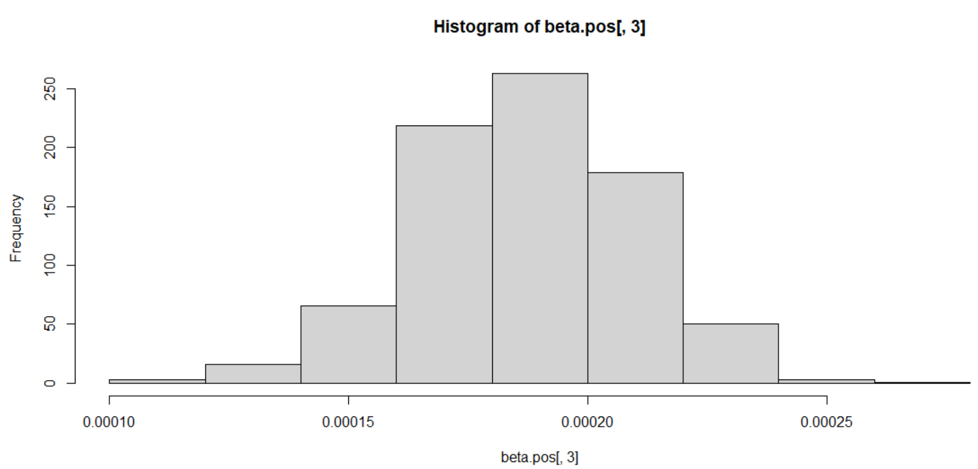

beta.pos[, 1]
-0.50
-0.55
-0.60
-0.65
-0.70
-0.75
-0.80
0
200
400
600
800
Index


beta.pos[, 2]
-0.06
-0.07
-0.08
-0.09
0
200
400
600
800
Index


beta.pos[, 3]
0.00025
0.00020
0.00015
0
200
400
600
800
Index








```
> quantile(beta.pos[,1], 0.95)
       95%
-0.5444697
> quantile(beta.pos[,2], 0.95)
        95%
-0.06201557
> quantile(beta.pos[,3], 0.95)
         95%
0.0002243032
```