OM 386 Advanced Data Analytics in Marketing
Assignment 2
Due: February 28th, 11:59pm

Submission by: Mahika Bansal (mb62835)

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.

Ans:

```
df <- fread('Bank_Retention_Data.csv',stringsAsFactors = TRUE)
df$TractID <- as.factor(df$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.

Ans:

1. Logit Model:

   logitModel <- glm(Churn~Age+Income+HomeVal+Tenure+DirectDeposit+Loan+Dist+MktShare,df,family=binomial(link='logit'))
   summary(logitModel)

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

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
```

2. Probit Model:

   probitModel <- glm(Churn~Age+Income+HomeVal+Tenure+DirectDeposit+Loan+Dist+MktShare,df,family=binomial(link='probit'))
   summary(probitModel)

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

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
```

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's of the logistic and probit models. Which model (logistic or probit) fits the data better based on AIC?

Ans:

The variables β_1, β_2, β_3, β_4, β_5, β_6, β_7, β_8 represent the change in log-odds of churn per unit age, income, home value, tenure, direct deposit, loan, distance and market share respectively. The positive/ negative effect is decided by the sign of the coefficients.

As per both the models, β_1, β_2, β_3, β_4, β_5, β_7 are significant as the p-values are less than 0.05.

AIC(logitModel)
AIC(probitModel)

```
> AIC(logitModel)
[1] 2207.358
> AIC(probitModel)
[1] 2206.626
```

Based on AIC, the probit model is better.

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

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

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

Ans:

glmer.model = glmer(Churn~Age+Income+HomeVal+Tenure+DirectDeposit+Loan+Dist+MktShare+(1|TractID),data=df,family=binomial(link='logit'),glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=100000)))
summary(glmer.model)

```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: Churn ~ Age + Income + HomeVal + Tenure + DirectDeposit + Loan +      Dist + MktShare + (1 | TractID)
   Data: df
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))

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

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.0913 -0.5118 -0.3894 -0.2447  5.3463

Random effects:
 Groups  Name        Variance Std.Dev.
 TractID (Intercept) 0.01988  0.141
Number of obs: 2505, groups:  TractID, 26

Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.561878   0.305951  -1.836   0.0663 .
Age           -0.016503   0.004178  -3.950 7.81e-05 ***
Income         0.106973   0.016078   6.653 2.87e-11 ***
HomeVal       -0.026715   0.005692  -4.693 2.69e-06 ***
Tenure        -0.029232   0.006564  -4.453 8.46e-06 ***
DirectDeposit -0.461198   0.111002  -4.155 3.25e-05 ***
Loan           0.099832   0.124633   0.801   0.4231
Dist           0.266895   0.063377   4.211 2.54e-05 ***
MktShare       0.006009   0.373151   0.016   0.9872
---
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     -0.207 -0.060 -0.534
Tenure       0.014 -0.285 -0.075  0.077
DirectDepst -0.176  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 (bobyqa) convergence code: 0 (OK)
Model failed to converge with max|grad| = 0.0156544 (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( ). Based on the AIC, compare the model fit of this model to the models in (1).

Ans:

The fixed effect estimates retain their significance and direction as per logit and probit models developed earlier.
AIC(glmer.model)
BIC(glmer.model)

```
> AIC(glmer.model)
[1] 2208.686
> BIC(glmer.model)
[1] 2266.947
```

As per AIC, previous models logit and probit performed better.

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.

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?

Ans:

bayesLogit = MCMChlogit(fixed=Churn~Age+Income+HomeVal+Tenure+DirectDeposit+Loan+Dist+MktShare, random=~1, group="TractID", data=df, burnin=10000, mcmc=20000 , thin=20, r=2, R=diag(1))
summary(bayesLogit$mcmc[,1:9])

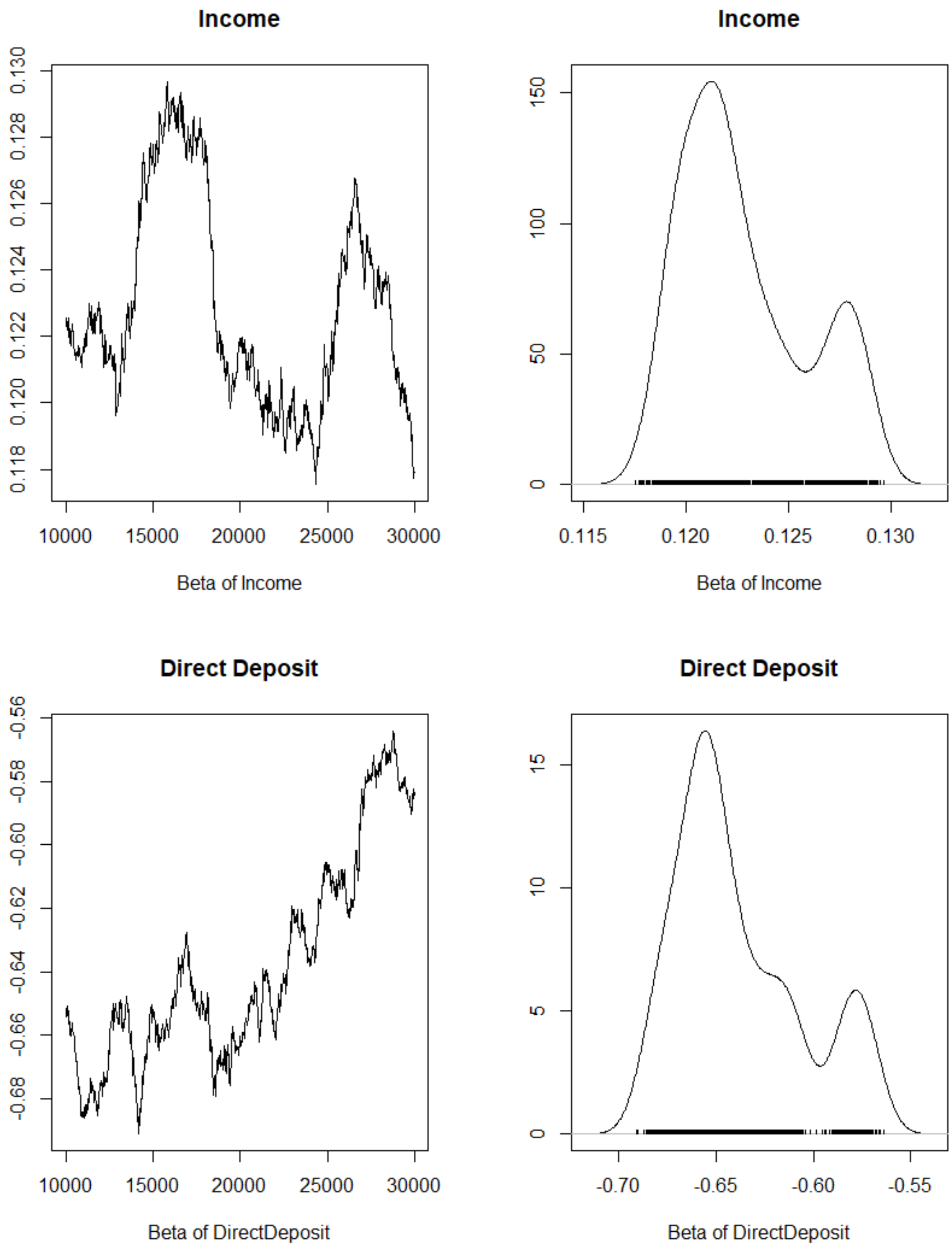
```
Iterations = 10001:29981
Thinning interval = 20
Number of chains = 1
Sample size per chain = 1000

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
```

All variables seem significant at 5% level, unlike previous methods

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.

Ans:

```
plot(bayesLogit$mcmc[,3],xlab = "Beta of Income", main='Income')
plot(bayesLogit$mcmc[,6],xlab = "Beta of DirectDeposit", main = "Direct Deposit")
```


Income
Beta of Income
Income
Beta of Income
Direct Deposit
Beta of DirectDeposit
Direct Deposit
Beta of DirectDeposit

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.

Ans:

```
DataFile = "CreditCard_LatePayment_data.csv"
LP.data = read.csv(DataFile, header=T)
probitModel2 <- glm(Latepay~Usage+Balance,LP.data,family=binomial(link='probit') )
summary(probitModel2)
```

```
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

Please run the completed code. Use the ts.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.

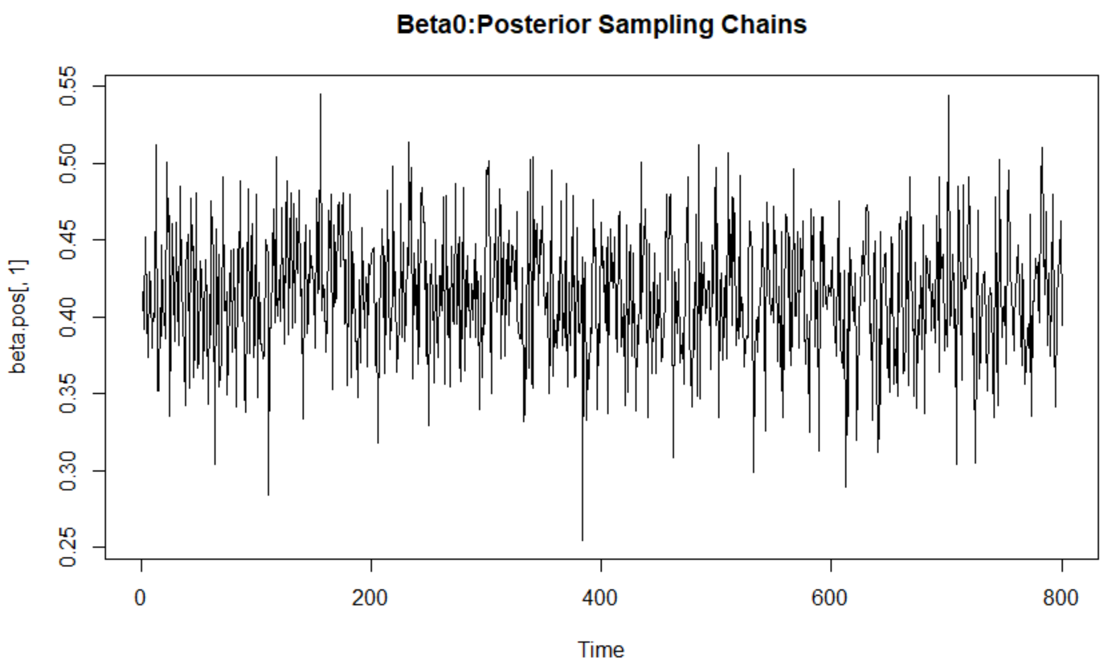
Ans:

```
library(truncnorm)
library(mnormt)
#Bayesian estimation for probit regression
#stage 1. read data into R and create columns for censored data
DataFile = "CreditCard_LatePayment_data.csv"
LP.data = read.csv(DataFile, header=T)

#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
```

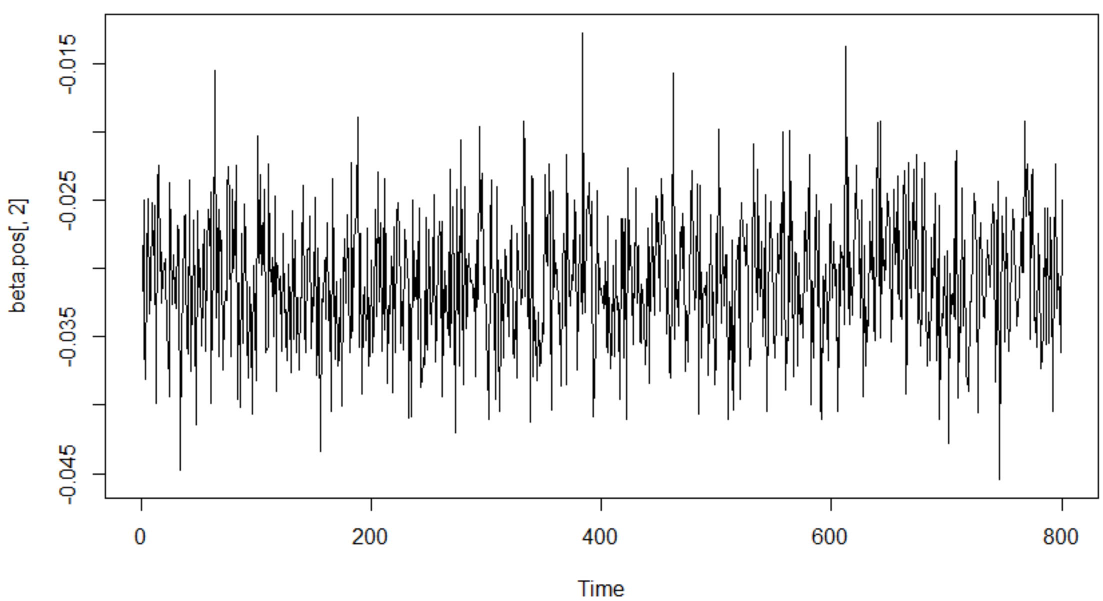
```
n0 = dim(LP.X0)[1]
n1 = dim(LP.X2)[1]
nObs = dim(LP.data)[1]
LP.Y = rep(0, nObs)

#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

#stage 5. Gibbs sampler

#initialize the loop
curBeta = c(0.1, 0, 0) #initial (current) regression coeff beta
g = 1

#main loop
for (m in 1:NIT){
        #step 1. sample the latent variable > 0 if Latepay=1, <0 if Latepay=0
        #use the corresponding truncated normal distribution given curbeta and X variables
        #Please fill in the code
  curY0 = rtruncnorm(n0,b=0, mean = LP.X0%*%curBeta, sd = 1)
  curY1 = rtruncnorm(n1,a=1, mean = LP.X1%*%curBeta, sd = 1)

        #step 2 sample curbeta (same as the linear regression code assuming the error's
variance is 1)
        #Please fill in the code
  LP.Y[LP.data$Latepay==1] = curY0
  LP.Y[LP.data$Latepay==1] = curY1
  sigma.hat = solve(LP.X2 + iSigma.beta)
  betaPos.mn = sigma.hat%*%(t(LP.X)%*%LP.Y + iSigma.beta%*%mu.beta)
  curBeta = as.vector(rmnorm(1, mean=betaPos.mn, varcov=sigma.hat))


        #save thinned sampled beta after burn-ins
        if ((m > nBurn) & (m%%thin.step == 0)) {
```

```
        beta.pos[g,] = curBeta
        g = g+1
    }
}

matplot(beta.pos, type = 'l')
ts.plot(beta.pos[,1], type = 'l', main = paste0('Beta0:Posterior Sampling Chains'))
ts.plot(beta.pos[,2], type = 'l', main = paste0('Beta1:Posterior Sampling Chains'))
ts.plot(beta.pos[,3], type = 'l', main = paste0('Beta2:Posterior Sampling Chains'))
colMeans(beta.pos)
hist(beta.pos[,1], xlab = paste0('Beta0'), main = paste0('Beta0'))
hist(beta.pos[,2], xlab = paste0('Beta0'), main = paste0('Beta1'))
hist(beta.pos[,3], xlab = paste0('Beta0'), main = paste0('Beta2'))
```

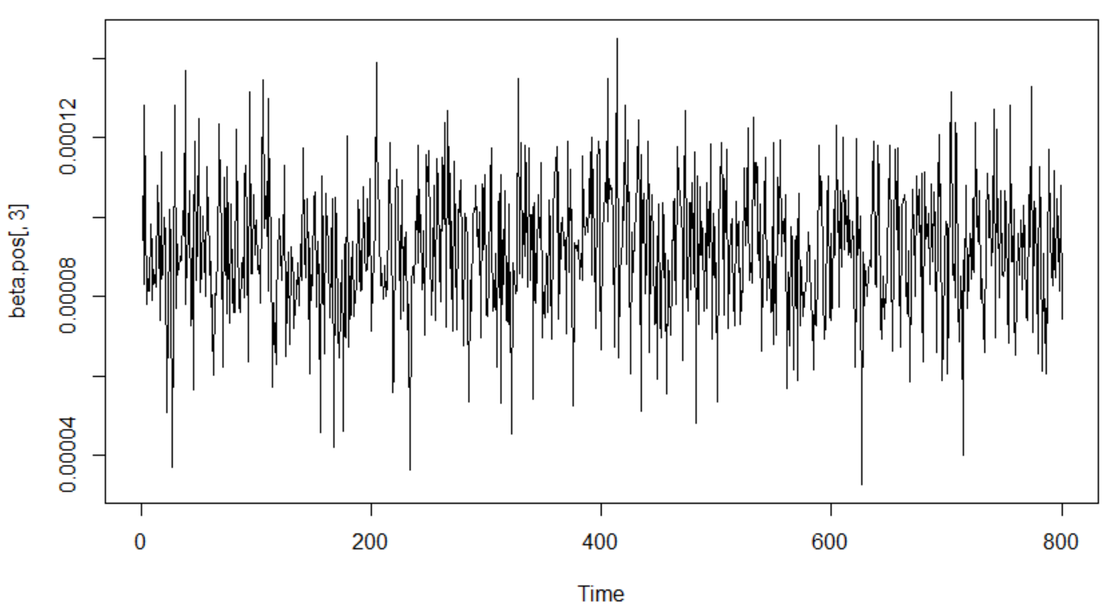
```
> colMeans(beta.pos)
[1]  0.4131450515 -0.0308526545  0.0000917738
```

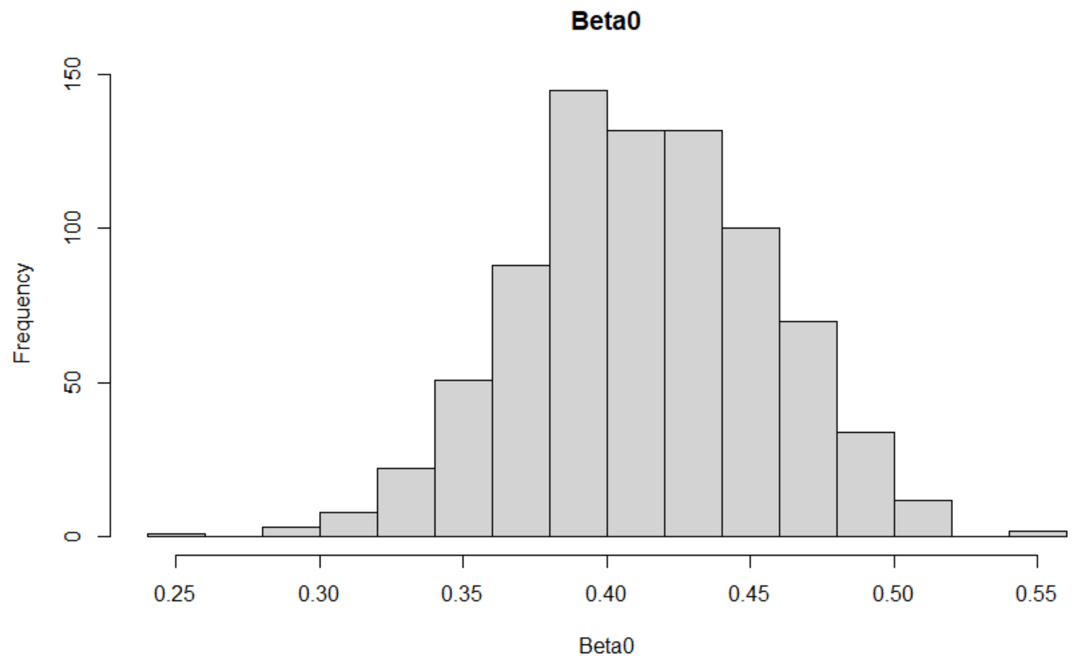


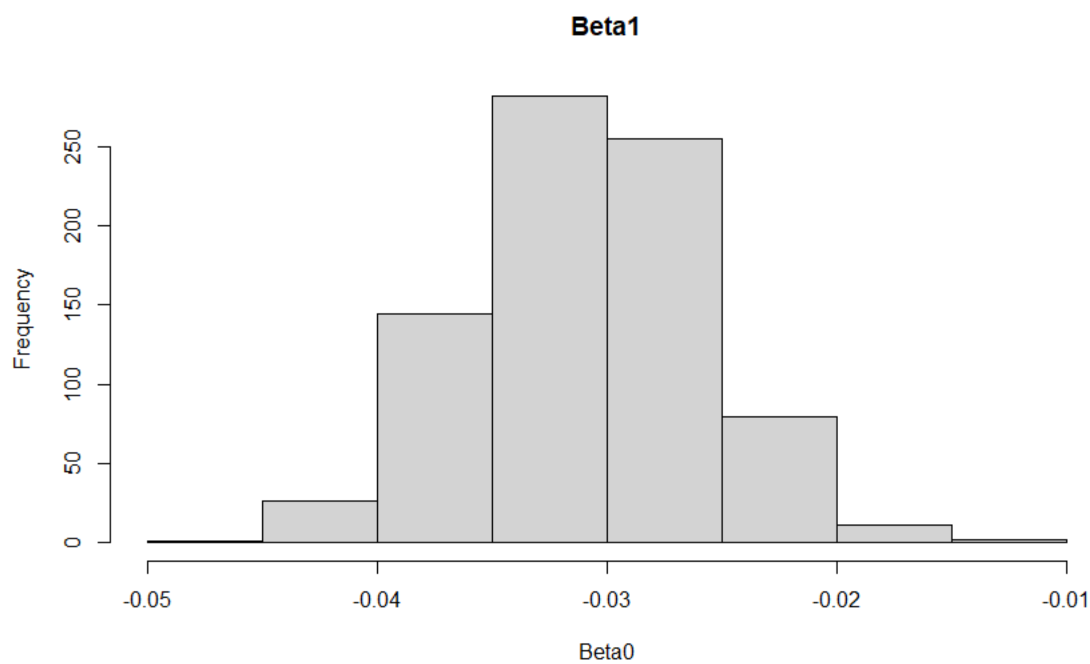
Beta1:Posterior Sampling Chains

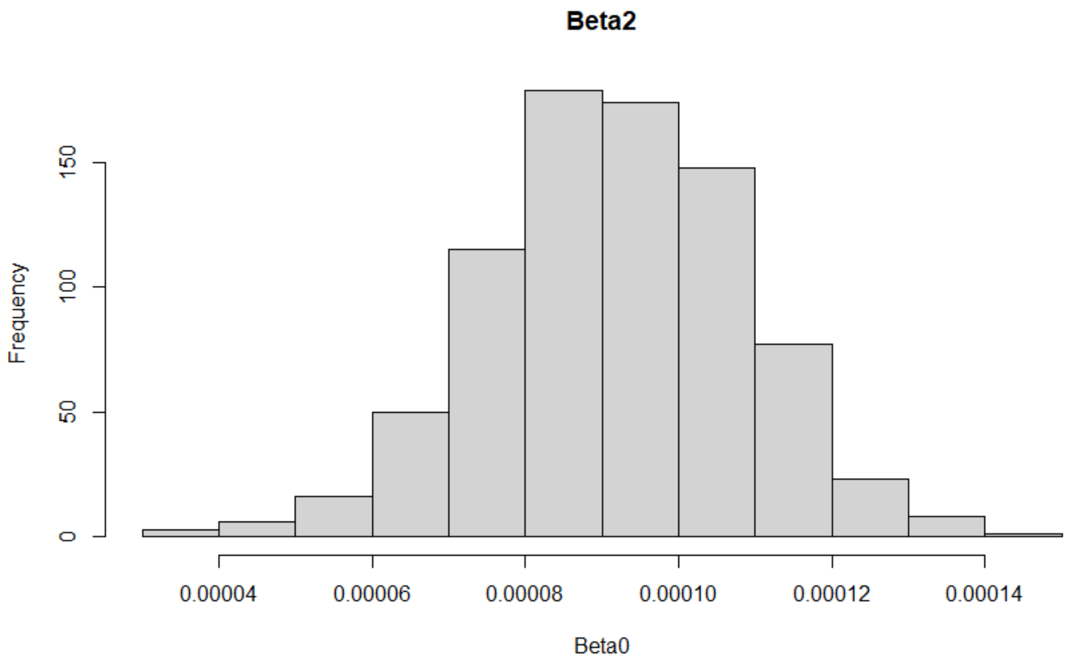


Beta2:Posterior Sampling Chains




Beta0
Frequency
0
50
100
150
0.25
0.30
0.35
0.40
0.45
0.50
0.55
Beta0


Beta1
Frequency
0
50
100
150
200
250
-0.05
-0.04
-0.03
-0.02
-0.01
Beta0




```
> quantile(beta.pos[,1], probs = c(0.025,0.5, 0.975))
     2.5%       50%     97.5% 
0.3345441 0.4121950 0.4952138 
> quantile(beta.pos[,2], probs = c(0.025,0.5, 0.975))
       2.5%         50%       97.5% 
-0.04043338 -0.03086614 -0.02166217 
> quantile(beta.pos[,3], probs = c(0.025,0.5, 0.975))
        2.5%          50%        97.5% 
5.729426e-05 9.170678e-05 1.239400e-04 
```