Lab3

May 12, 2019

1. Poisson regression - the MCMC way.

A -> Fitting a regression model on the data (GLM)

```
## MinBidShare       Const      Sealed    VerifyID     MajBlem     LogBook
## -1.89409664  1.07244206  0.44384257 -0.39451647 -0.22087119 -0.12067761
##     LargNeg     Minblem PowerSeller
##  0.07067246 -0.05219829 -0.02054076
```

We have sorted the covariates according to their absolute significance in the output above. Some of the most significant covariates were - "MinBidShare", "Sealed", "VarifyID", "LogBook".

B -> Bayesian analysis of the Poisson regression

```
## MinBidShare       Const      Sealed    VerifyID     MajBlem     LogBook
## -1.89198428  1.06984639  0.44355493 -0.39300356 -0.22129570 -0.12021941
##     LargNeg     Minblem PowerSeller
##  0.07070107 -0.05246432 -0.02051554
```

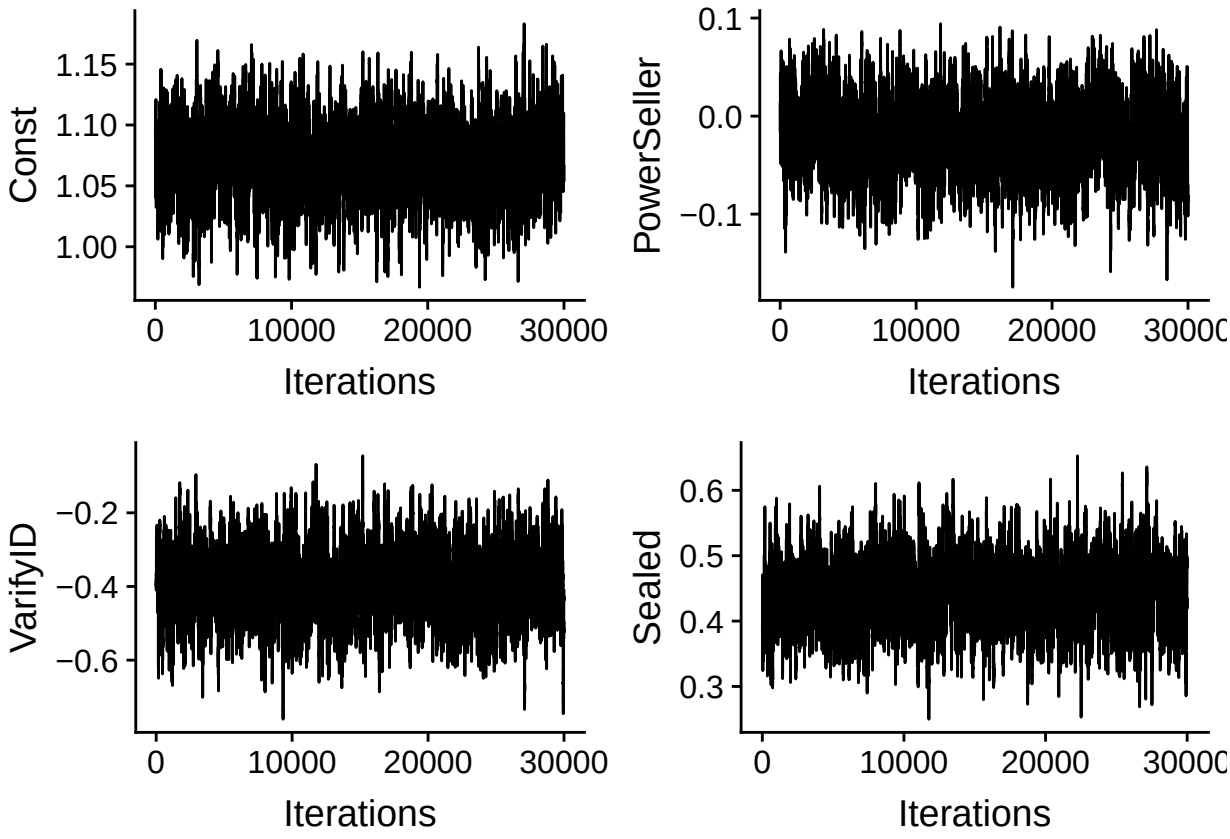
Using numerical optimization(optim function in R) for approximating the posterior distribution of beta as a multivariate normal, we get similar values for the covariates. The order of significance remains the same as we got in the previous question. The covariates "MinBidShare", "Sealed", "VarifyID", "LogBook" are still significant.

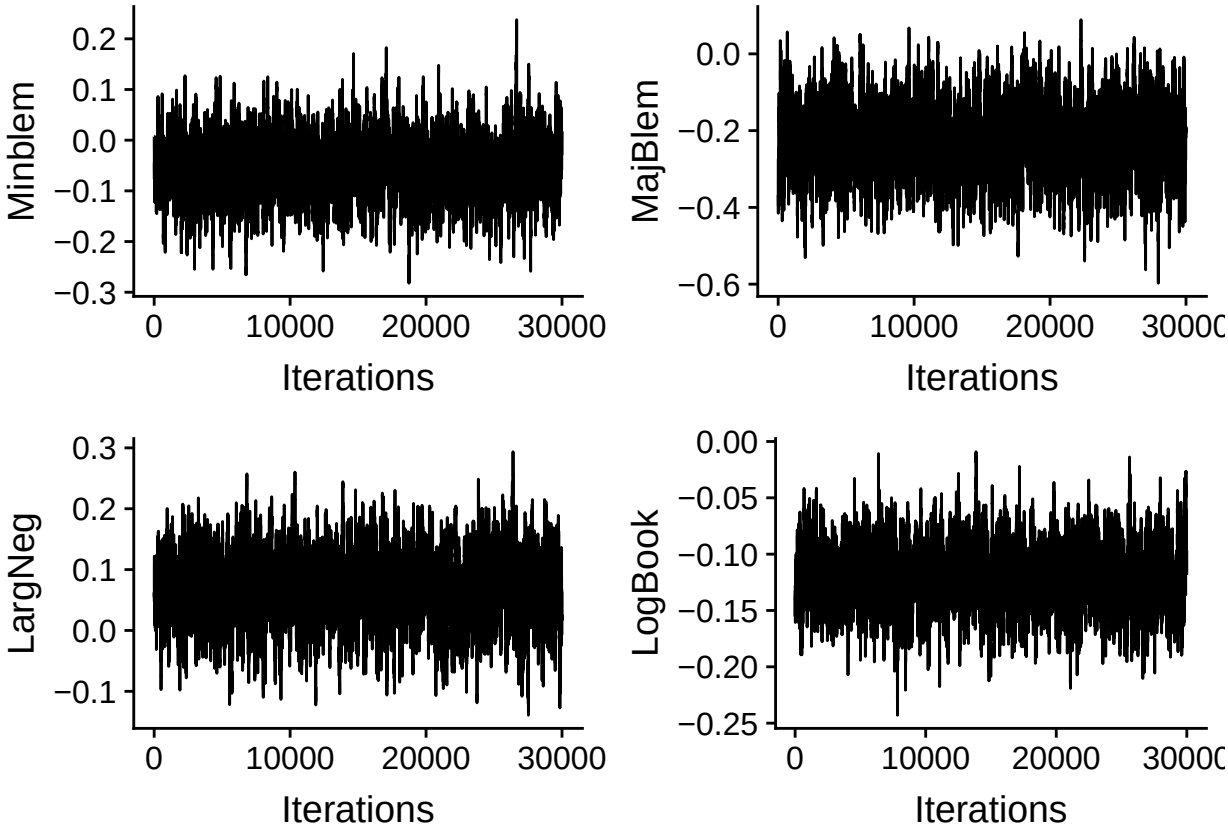
We implemented the logPostProp function for this question, which calculates the sum of log prior and the log likelihood given the initial beta values and the data.

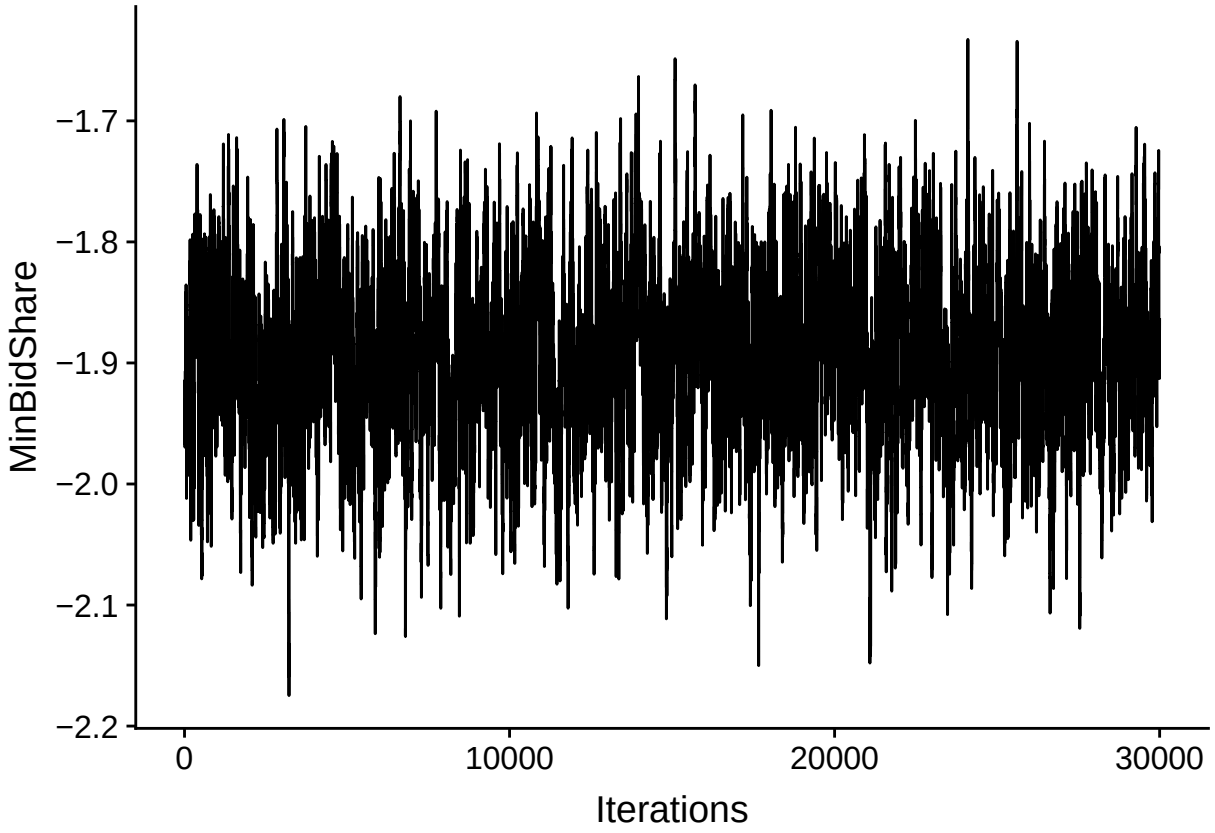
Likelihood values can get very small because we multiply so many small values, and there is a possible loss of numerical precision. This is the reason we use the log of the posterior to prevent this problem.

C -> Simulate Posterior using Metropolis algorithm

```
## Acceptance Rate for Metropolis : 0.4358788
```


Const
1.15
1.10
1.05
1.00
0
10000
20000
30000
Iterations
PowerSeller
0.1
0.0
−0.1
0
10000
20000
30000
Iterations
VarifyID
−0.2
−0.4
−0.6
0
10000
20000
30000
Iterations
Sealed
0.6
0.5
0.4
0.3
0
10000
20000
30000
Iterations


Minblem
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
10000
20000
30000
Iterations
MajBlem
0.0
−0.2
−0.4
−0.6
0
10000
20000
30000
Iterations
LargNeg
0.3
0.2
0.1
0.0
−0.1
0
10000
20000
30000
Iterations
LogBook
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
0
10000
20000
30000
Iterations



We implemented a universal function for Metropolis haistings simulation algorithm. This function takes in as arguments the logPosterior function, the initial parameters for the posterior, and a parameter to control the variability of the simulations being generated. Along with this you can provide any number of additional arguments required for your logPosterior function. In our scenario we are sending in the data X, Y as the additional arguments for the function.

We are making 30,000 draws from the posterior, and we add in the additional burn-in period to this as 10 percent of the number of draws we are makin. So Total draws we are making is 33,000 off which we discard the first 3000 draws.

The parameter "c" that controls the variance of the simulation, was adjusted so that we get an acceptance rate close to 40%. We are getting an acceptance rate of : 0.4351212 for this chain, that is 43% accept rate, which is reasonable.

From the above trace plots for the parameters it looks like the Markov chain has reached the stationary distribution.

```
## Effective sample size for Const  :  882.3291
## Effective sample size for PowerSeller  :  916.9047
## Effective sample size for VerifyID  :  827.5997
## Effective sample size for Sealed  :  882.8641
## Effective sample size for Minblem  :  909.308
## Effective sample size for MajBlem  :  908.6984
## Effective sample size for LargNeg  :  871.627
## Effective sample size for LogBook  :  945.3966
## Effective sample size for MinBidShare  :  901.5065
```

On further analysis for the convergence of the markov chain, we can see that the effective sample size is also

good, approx 850 for all the parameters. So we can get a lot of information from this posterior to make interval estimates about the parameter values.

```
## [1] "Mean parameter values from the simulated posterior:"
```

```
##         Const PowerSeller   VerifyID    Sealed     Minblem    MajBlem
## [1,] 1.070241 -0.02114414 -0.3939983 0.4409227 -0.05500262 -0.2267681
##         LargNeg    LogBook MinBidShare
## [1,] 0.06722392 -0.1210629   -1.892493
```

The parameter values are similar to the ones we got for the first two questions