STAT 4350
Final Project

Zetong Jin
University of Georgia
Athens, GA 30602

April 15, 2022

1 Introduction

1.1 Data Summary

The data cats.txt includes 40 rows and 6 variables and there are no missing value in the data-set. First we need to import the data and understand each variables by reading the data description. Based on the information that we have, we can see that there are totally five types of Breeds and we define Breeds A, B, and C are classified as domestic (Domestic = 1) and breeds D, E are non- domestic (Domestic = 0). Thus, we can create a new column which named "Domestic" and the table below will show the first 6 rows of the data:

	DeltaPCV	Type	Dose	Breed	Domestic
1	-5.40	0	2	A	1.00
2	-4.80	1	6	A	1.00
3	-6.50	1	7	A	1.00
4	-7.80	0	9	A	1.00
5	-8.30	0	9	A	1.00
6	-7.20	1	6	A	1.00

The table above indicates that the "Type", "Breed" and "Domestic" are all categorical variables and the "Dose" and "DeltaPCV" are quantitative variables.

1.2 Fit Linear Model

Now, we need to fit the linear model and see the p value and the linear regression.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-1.7133	0.5497	-3.12	0.0036
Dose	-0.3646	0.0774	-4.71	0.0000
Type	0.2600	0.3622	0.72	0.4775
Domestic	-2.1396	0.3693	-5.79	0.0000

Based on the P-value, we can see that the variable "Type" has a large p-value which is larger than our critical value (0.05). Thus, the "Type" is not significant and the rest of the variables are significant in this case.

1.3 Research Problems

After the Data Cleaning Method, we need to introduce our Research Problems:

1. Is a higher dose of hydroxyurea associated with a greater reduction in packed cell volume?

2. Controlling for the dose of hydroxyurea administered and the type of erythrocytosis, do domesticbreeds of cats have a different expected change in pack cell volume than non-domestic cats?

Those two research problems are straightforward and the first one seeks to find the relationship between the "Dose" and the "DeltaPCV" and the second problem seeks to see the relationship between the "Domestic" and the "DeltaPCV". Thus, we want to use the posterior distribution to estimate those variables.

The technique we will mainly use for this report would be: Gibbs Sampling with JAGS, Linear Regression model, finding the posterior/prior distribution and the Trace/Density plot of MCMC chains for specific parameters of interest.

2 Methodology

2.1 Model Specification/Sampling Model

By seeing the data, we can notice that the "Type", "Breed" and "Domestic" are all categorical variable and we can also clearly see that the "DeltaPCV" would be our response variable in this case. Thus, we can simply use those variables to have a linear regression model which would be:

$$\mu_i = \beta_1 + \beta_2(Domestic)_i + \beta_3(Type)_i + \beta_4(Dose)_i$$

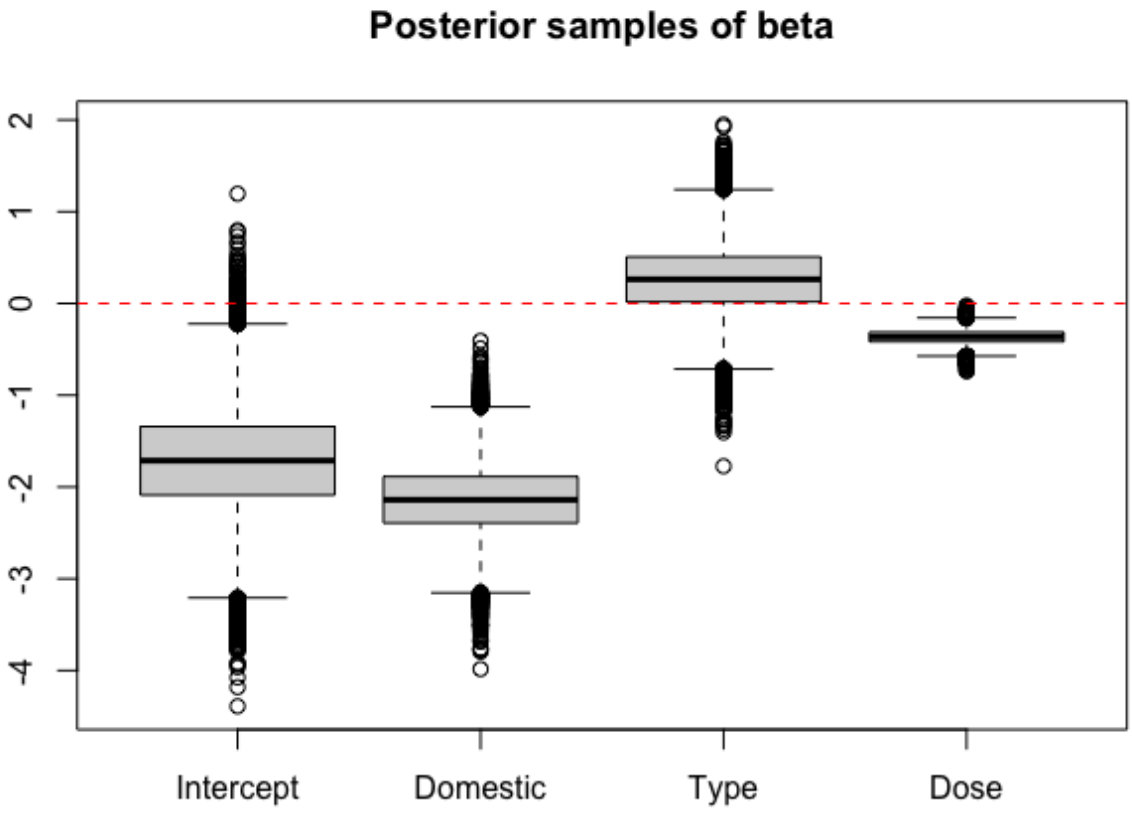
where $i = 1, ..., n$ and the β here will be the regression coefficients.
We make the Y_i be the values of the response variable and based on the linear assumption, we will make this sampling model follows a normal distribution with μ_i and σ^2. Then, we will use this model as our sampling model. Since the "Domestic" variable and "Breeds" variable are highly correlated, we might want to only include "Domestic" in the model to avoid the redundant predictive variable.

2.2 Prior model Specification

Next, we need to figure out our prior model. The unknown values in the sampling model are μ_i and σ^2. Thus, our parameters would be $\beta = (\beta_1, ..., \beta_4), \sigma^2$. Although we do not have any information about our parameters, we know that σ^2 should follow the Inverse Gamma distribution with small value for the parameter A and B. Thus, we will have our $\sigma^2 \sim IG(0.01, 0.01)$. $A = 0.01, B = 0.01$ is a common choice for the hyper-parameters of the prior on σ^2.
For the $\beta_1, ..., \beta_4$, we plan to use flat prior. In this case, we will assume the normal distribution with the coefficients will center at zero and with the large variance. This would almost give us a uniform prior. Since the uniform prior would consider as improper prior, the $\beta_1, ..., \beta_4 \sim N(0, 1e-10)$ will give us the proper but virtually "flat" prior. At the end, we will also assume that the β and σ^2 are independent with each other.

2.3 JAGS/Gibbs Sampler

JAGS is a program for analysis of Bayesian hierarchical models using Markov Chain Monte Carlo (MCMC) simulation and the Gibbs Sampler is one example of a MCMC sampling method. Once we figure out the sampling and prior models. We need to create a txt file for JAGs [See the Appendix for the Jags file]. In this txt file, we need to include our sampling model and prior which we specify earlier. Then, we need to run the JAGS into R to get the posterior samples of beta. Since the normal distribution is parameterized by mean and τ^2 in JAGS, we need to have the model calculated by $\sigma^2 = 1/\tau^2$. When we tried to run the JAGS model, we need to have a good chain settings and make sure that all the variable has been specified in the data-list and initialize the suitable value for our β and τ^2. Since we want to identify the value of β and σ^2, we need to specify the β and σ^2 in our parameters.



By seeing the plot above, we can figure that only the "Type" variable seems most positively related to the "DeltaPCV" and the rest of them seem negatively related to the "DeltaPCV".

2.4 Trace/Density Plot

One way to access if this dependent sampling method is sufficiently estimating the correct posterior distribution is to see the trace plots for β and σ^2. We can diagnose convergence via trace plots and if we can see the random oscillations in the graph, we can conclude that our algorithm has converged. Based on the trace plot [See the appendix for the Trace/Density Plot], we can see that it is converged and the density plots shows that Beta[1] - Beta[4] follows normal distribution which is the same as our expectations. The σ^2 is defined on positive number and the shape looks like a Inverse-Gamma density function which is also the same with our expectations.

3 Conclusion

3.1 Model Checking

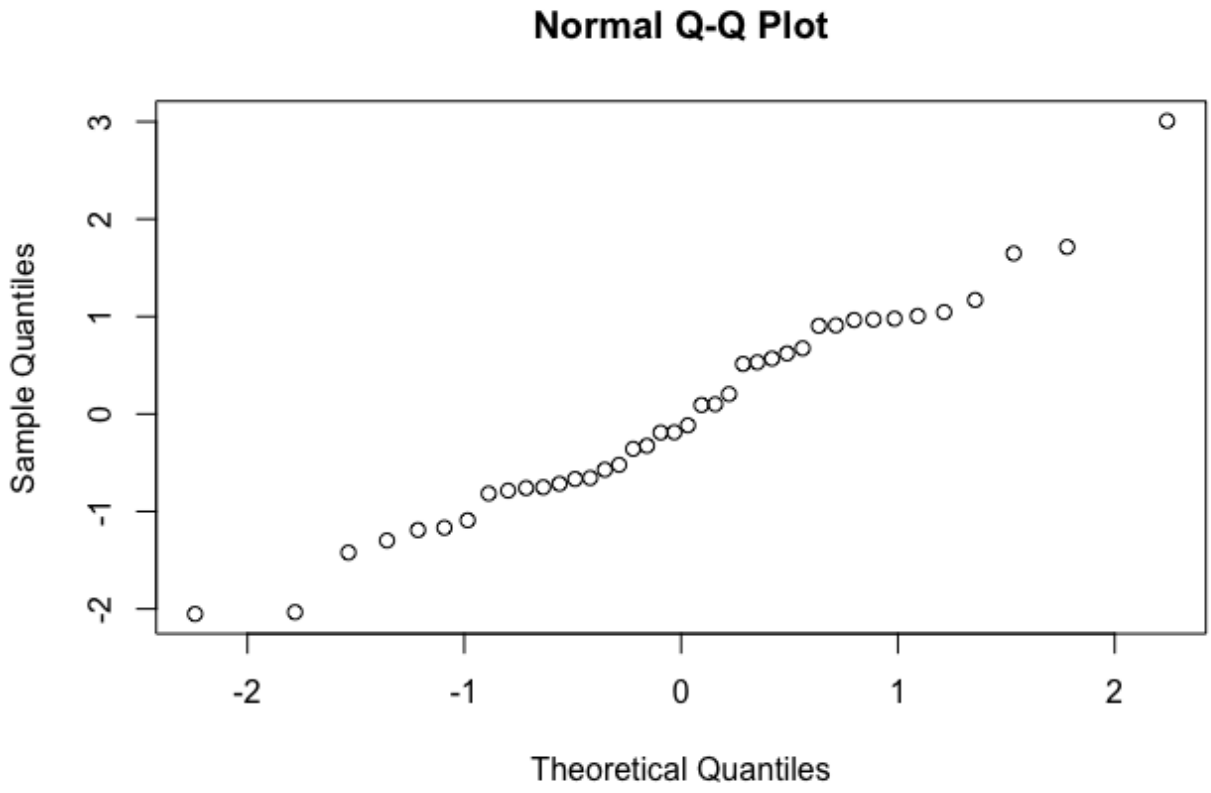
```
> mean(mcmcChain3[,1])
[1] -1.715377
> var(mcmcChain3[,1])
[1] 0.3169907
> mean(mcmcChain3[,2])
[1] -2.140205
> var(mcmcChain3[,2])
[1] 0.1446403
> mean(mcmcChain3[,3])
[1] 0.262279
> var(mcmcChain3[,3])
[1] 0.1374457
> mean(mcmcChain3[,4])
[1] -0.3641218
> var(mcmcChain3[,4])
[1] 0.006256957
```

Based on the density plot, we know that the Beta follows normal distribution and each Beta can be shown as follows: Beta[1] $\sim N(-1.715, 0.317)$, Beta[2] $\sim N(-2.14, 0.144)$, Beta[3] $\sim N(0.262, 0.137)$, Beta[4] $\sim N(-0.364, 0.006)$. Based on the results above for the mean and variance for each Beta, our model can be shown as :

$$\mu_i = -1.72 - 2.14 * (Domestic)_i + 0.262 * (Type)_i - 0.36 * (Dose)_i$$

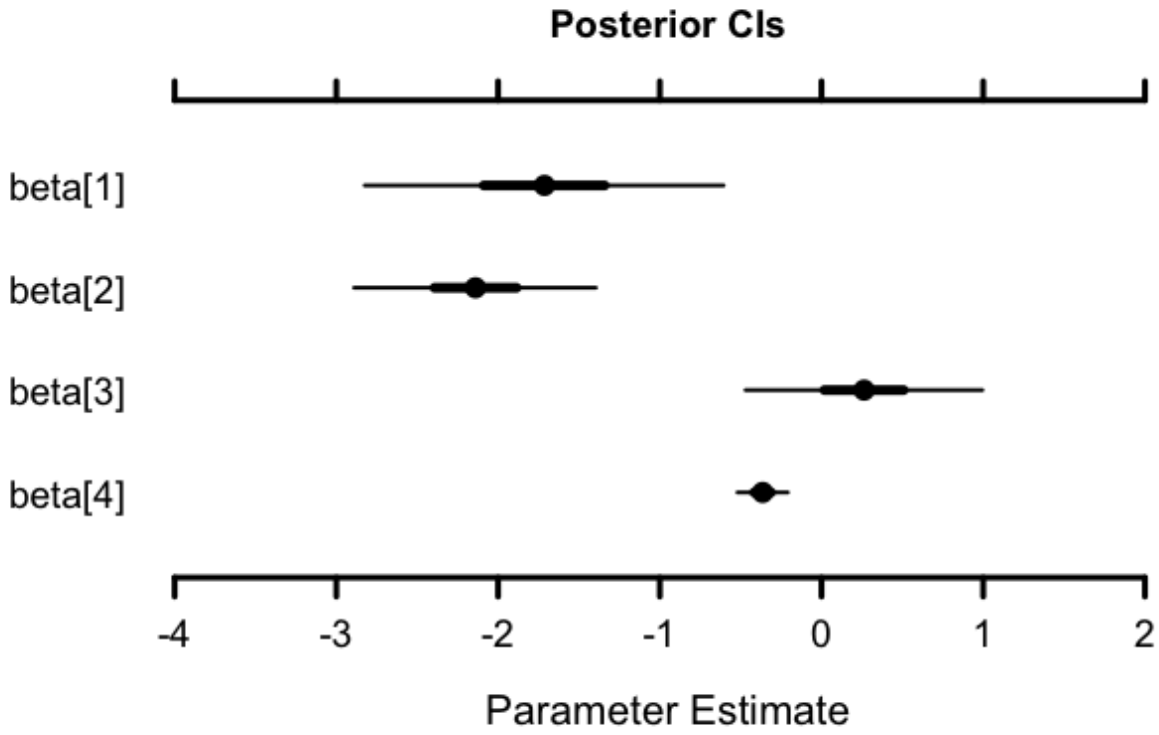
We can also assess model fit in regression to consider how well the model predicts. By plugging the observations into the given model, we have the predicted value and the residuals are given by the difference between the true value and the predicted value of the "DeltaPCV". By viewing the Normal Q-Q plot and the scatter plot, we can conclude that it is a good model and meets expectations.

Now, we need to go back to our model fit and check if the coefficients are similar or not. Since we have a "flat" prior, the coefficients for each variable should be similar otherwise there must be something wrong in the model. By checking the coefficients, we can see that each value is close to our final model. Thus, we can conclude the model is accurate.



3.2 Posterior Credible Interval

A credible interval is an interval within which an unobserved parameter value falls with a particular probability. It is an interval in the domain of a posterior probability distribution or a predictive distribution. We will use the posterior credible interval here to Estimate the beta and thus get the result for our research questions.



Based on this plot, we can answer the two research questions. For the first question, we can see that beta[4] is significantly negative which means higher dose will cause lower PCV. For the second question, we are focusing on the relationship between the "Domestic" and the "DeltaPCV". We can conclude that they are significantly different since the credible interval does not contain zero. Thus, the domestic breeds tend to have more reduction than non-domestic breeds.

4 Appendices

4.1 JAGS File

```
model
{

  # Likelihood - in JAGS, normal distribution is parameterized by
  # mean theta and precision = tau2 = 1/sig2
  for(i in 1:n){
   Y[i] ~ dnorm(mu[i],tau2)
        mu[i] = beta[1]+beta[2]*Domestic[i]+beta[3]*Type[i]+beta[4]*Dose[i]
  }

  # Priors
for(i in 1:p){

beta[i] ~ dnorm(0,1e-10)
}

  tau2 ~ dgamma(0.01,0.01)

  # Need to have model calculate sig2 = 1/tau2
  sig2 = 1/tau2

}
```

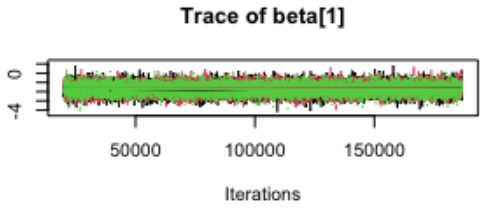
4.2 R Code

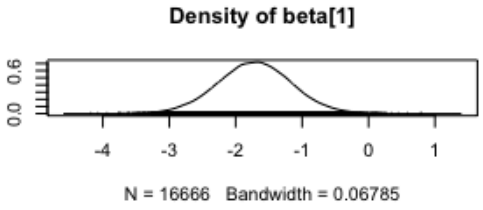
```
cats = read.table("/Users/Thomas/Desktop/cats.txt",header = T)
cats$Domestic<- cats$Breed
cats$Domestic[cats$Domestic == "A"] = 1
cats$Domestic[cats$Domestic == "B"] = 1
cats$Domestic[cats$Domestic == "C"] = 1
cats$Domestic[cats$Domestic== "D"] = 0
cats$Domestic[cats$Domestic == "E"] = 0
cats$Domestic = as.numeric(cats$Domestic)
attach(cats)
head(cats)
g = lm(DeltaPCV~Dose+Type+Domestic)
summary(g)
library(rjags)
library(MCMCvis)
```

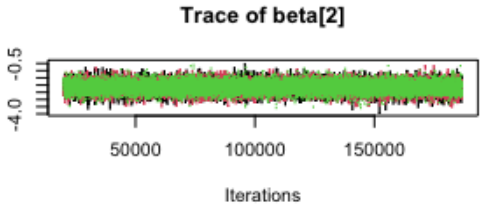
```
n = length(cats$DeltaPCV)
parameters <- c("beta","sig2")
# Chain settings
adaptSteps <- 10000            # number of steps to "tune" the samplers
burnInSteps <- 20000             # number of steps to "burn-in" the samplers
nChains <- 3                    # number of chains to run
numSavedSteps <- 50000          # total number of steps in chains to save
thinSteps <- 10                  # number of steps to "thin" (1 = keep every step)
nIter <- ceiling((numSavedSteps*thinSteps)/nChains)  # steps per chain

############
# Model   #
############
p <- 4   # number of regression coefficients

dataList1 <- list(
  "n" = n,
  "p" = p,
  "Domestic" = Domestic,
  "Type" = Type,
  "Dose" = Dose,
  "Y" = DeltaPCV)

initsValues1 <- list(
  "beta" = rep(0,p),
  "tau2" = 1
)

jagsModel1 <- jags.model("/Users/Thomas/Desktop/PJ_Jags.txt",
                         data = dataList1,
                         inits = initsValues1,
                         n.chains = nChains,
                         n.adapt = adaptSteps)

if(burnInSteps>0){
  update(jagsModel1, n.iter = burnInSteps)
}

codaSamples1 <- coda.samples(jagsModel1,
                             variable.names = parameters,
                             n.iter = nIter,
                             thin = thinSteps)
```

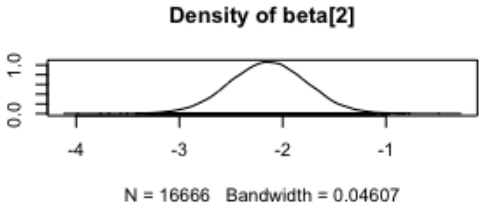
```
mcmcChain3 <- as.matrix(codaSamples1)
MCMCplot(mcmcChain3, params = "beta",
         main = "Posterior CIs", ref = NULL)
codaSamples1
mean(mcmcChain3[,1])
var(mcmcChain3[,1])
mean(mcmcChain3[,2])
var(mcmcChain3[,2])
mean(mcmcChain3[,3])
var(mcmcChain3[,3])
mean(mcmcChain3[,4])
var(mcmcChain3[,4])

ypred=mean(mcmcChain3[,1])+mean(mcmcChain3[,2])*cats$Domestic+
mean(mcmcChain3[,3])*cats$Type+mean(mcmcChain3[,4])*cats$Dose
red=cats$DeltaPCV-ypred
qqnorm(red)
plot(red)

par(ask=T)
plot(codaSamples1)
{boxplot(mcmcChain3[,1:4], main = "Posterior samples of beta",
        names=c("Intercept","Domestic","Type","Dose"))
abline(h = 0, lty = 2, col="red")}
```

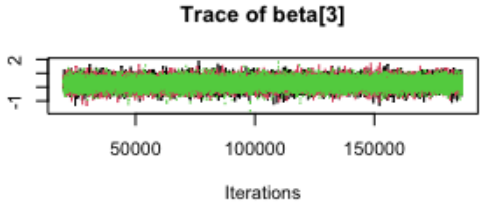
4.3 Figures

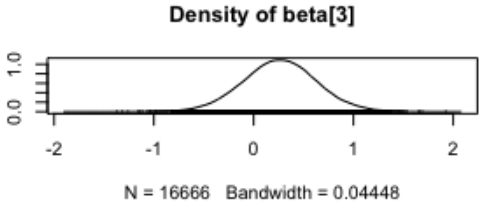

Trace of beta[1]
-4
0
50000
100000
150000
Iterations

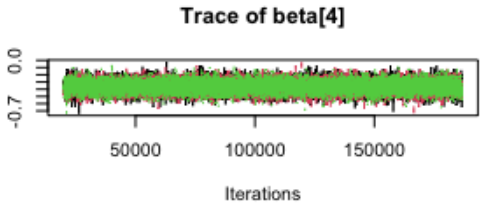

Density of beta[1]
0.0
0.6
-4
-3
-2
-1
0
1
N = 16666 Bandwidth = 0.06785

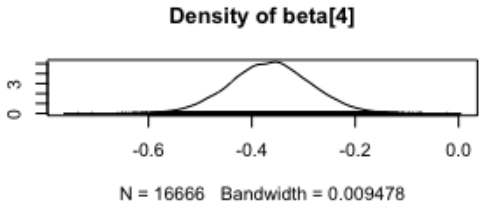

Trace of beta[2]
-4.0
-0.5
50000
100000
150000
Iterations

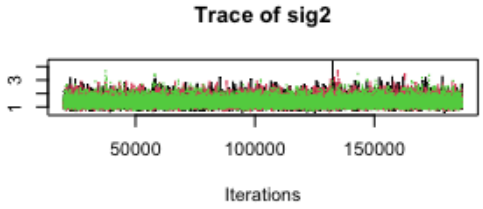

Density of beta[2]
0.0
1.0
-4
-3
-2
-1
N = 16666 Bandwidth = 0.04607

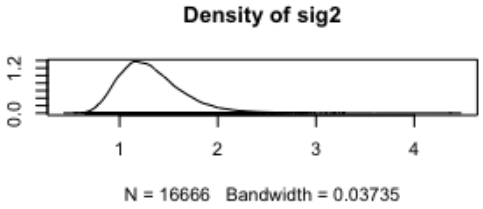

Trace of beta[3]
-1
2
50000
100000
150000
Iterations


Density of beta[3]
0.0
1.0
-2
-1
0
1
2
N = 16666 Bandwidth = 0.04448


Trace of beta[4]
-0.7
0.0
50000
100000
150000
Iterations


Density of beta[4]
0
3
-0.6
-0.4
-0.2
0.0
N = 16666 Bandwidth = 0.009478


Trace of sig2
1
3
50000
100000
150000
Iterations


Density of sig2
0.0
1.2
1
2
3
4
N = 16666 Bandwidth = 0.03735