Bayesian Data Analysis - Mini Project

Group 1:

1. 2602135446 - Meisa Kamilia
2. 2602141871- Kayla Masayuningtyas
3. 2602158784 - Jeffrey Wijaya
4. 2602097454 - Mohamad Ridho Farhan
5. 2602162522 - Muhammad Athariq Naufal

1. Introduction (of the data)

US Health Insurance Dataset:
https://www.kaggle.com/datasets/teertha/ushealthinsurancedataset

This dataset contains 7 columns and 1338 rows of insured data, where the Insurance charges are given against the following attributes of the insured: Age, Sex, BMI, Number of Children, Smoker and Region. There are no missing or undefined values in the dataset. Here's an overview of each attributes:

-Age: Age of primary beneficiary

-Sex: Insurance contractor gender, female / male

-BMI: Body mass index, providing an understanding of body weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9.

-Children: Number of children covered by health insurance / Number of dependents

-Smoker: Smoker / Non - smoker

-Region: The beneficiary's residential area in the US, northeast, southeast, southwest, northwest.

-Charges: Individual medical costs billed by health insurance.

2. Models (formulation of the models, likelihood, prior)

We use Gaussian Multiple Linear Regression for the assumption that we made and the data is Normally Distributed.

$$Y_i = \alpha + \sum_{j=1}^{6} X_{ij}\beta_j$$

Likelihood:

$$Y_i \sim Normal(\alpha + \sum_{j=1}^{6} X_{ij}\beta_j,\ \sigma^2)$$

Prior uninformative:

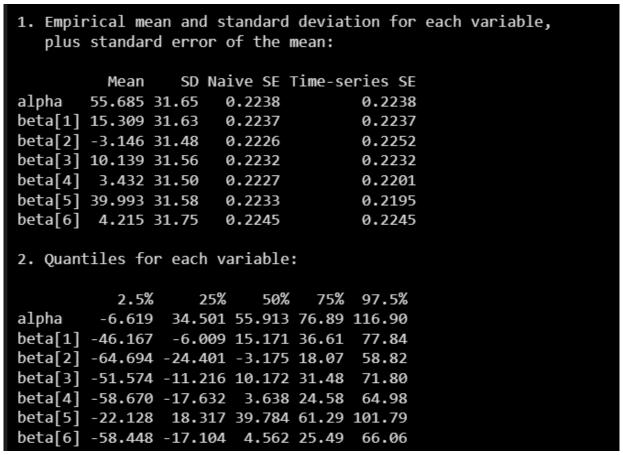
$$\beta_j \sim Normal(0, 1000)$$

$$\sigma^2 \sim InvGamma(0.1, 0.1)$$

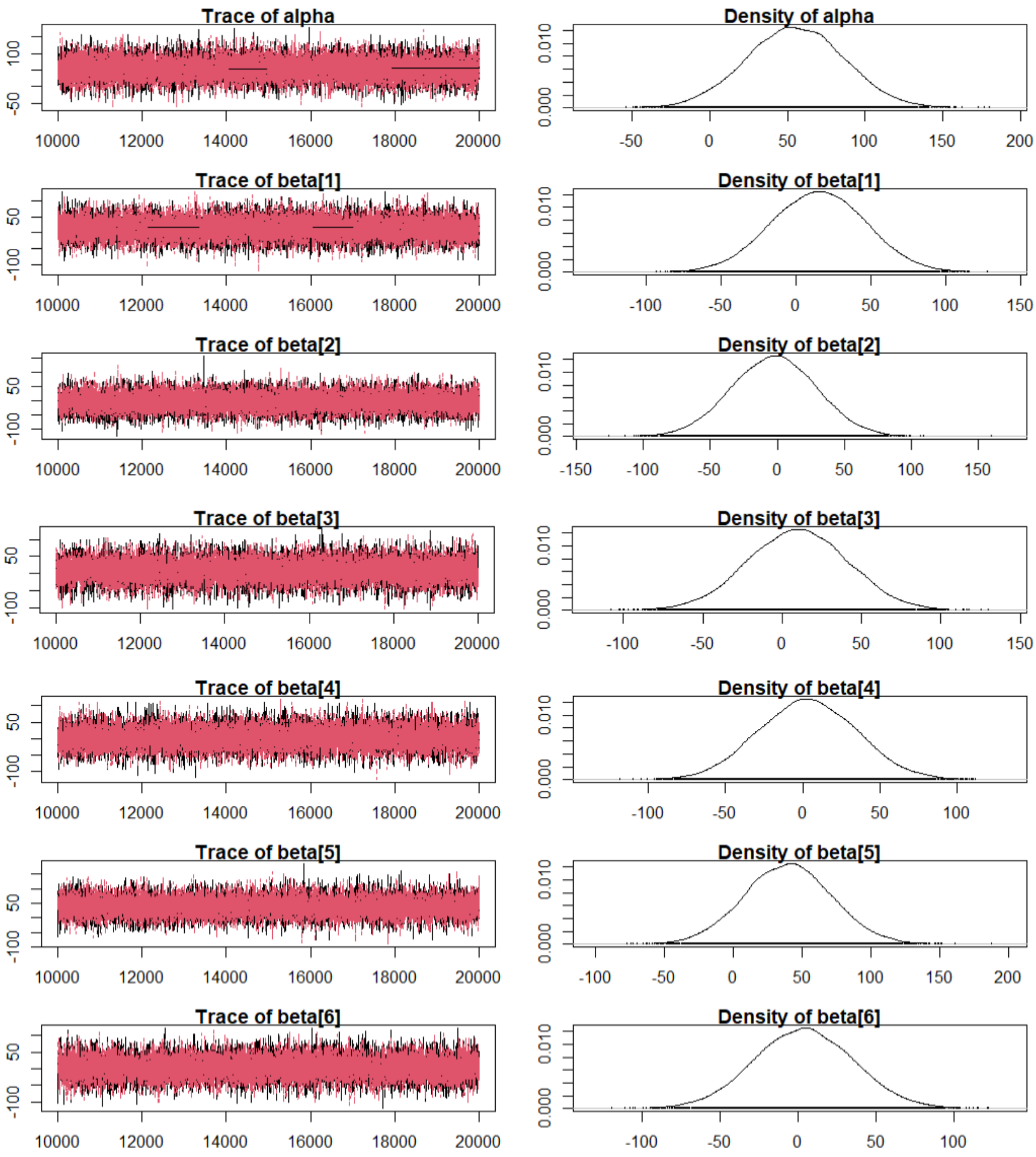
3. Computation (number burn-in samples, number of post-burn-in samples, number of chains, thinning intervals)

Number of burns in samples	: 10000
Number of post burns in sample	: 20000
Thinning interval	: 1
Number of chains	: 2
Sample size per chain	: 10000

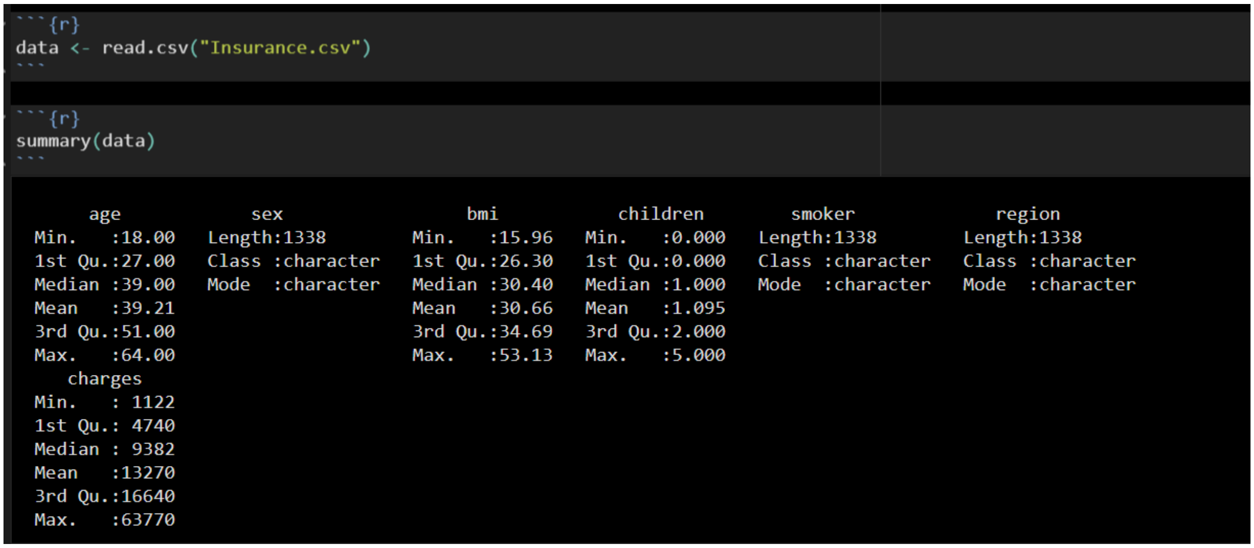
4. At least fitting one model



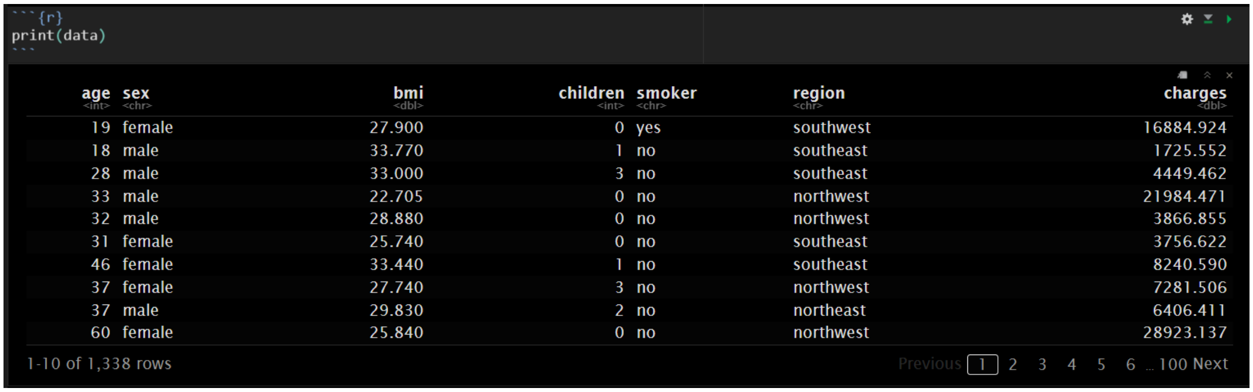
```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean    SD Naive SE Time-series SE
alpha   55.685 31.65   0.2238         0.2238
beta[1] 15.309 31.63   0.2237         0.2237
beta[2] -3.146 31.48   0.2226         0.2252
beta[3] 10.139 31.56   0.2232         0.2232
beta[4]  3.432 31.50   0.2227         0.2201
beta[5] 39.993 31.58   0.2233         0.2195
beta[6]  4.215 31.75   0.2245         0.2245

2. Quantiles for each variable:

           2.5%     25%    50%   75%  97.5%
alpha    -6.619  34.501 55.913 76.89 116.90
beta[1] -46.167  -6.009 15.171 36.61  77.84
beta[2] -64.694 -24.401 -3.175 18.07  58.82
beta[3] -51.574 -11.216 10.172 31.48  71.80
beta[4] -58.670 -17.632  3.638 24.58  64.98
beta[5] -22.128  18.317 39.784 61.29 101.79
beta[6] -58.448 -17.104  4.562 25.49  66.06
```



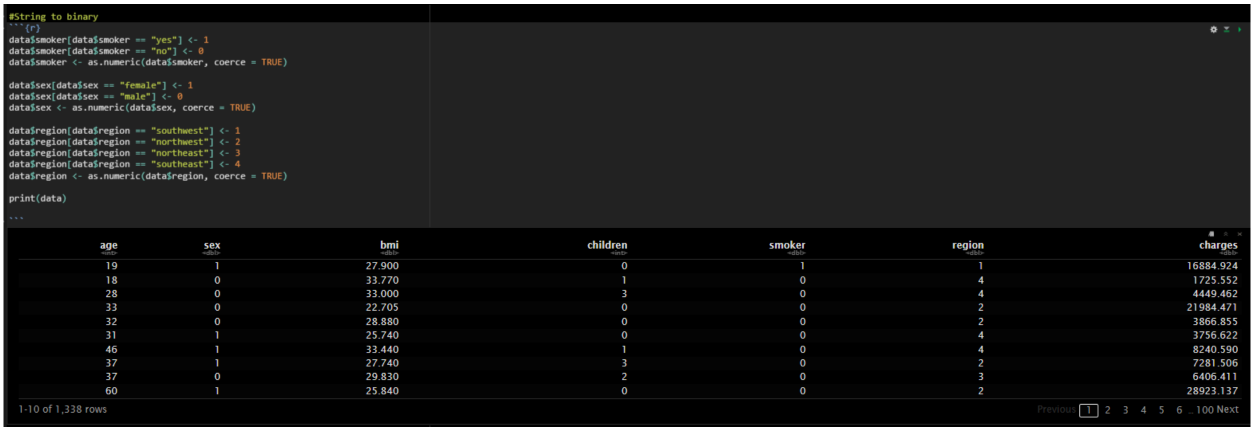
The trace of beta graph shows how the values of regression parameters (Beta) change over time or iterations in the optimization process. From the plot above, the shape of all the density tends to be the same with a value range of -100 to 100, and it can be concluded that the distribution shape is normal because it resembles a bell-shaped.



~~~
```{r}
data <- read.csv("Insurance.csv")
```

```{r}
summary(data)
```

      age            sex                 bmi           children        smoker             region         
 Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000   Length:1338        Length:1338       
 1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000   Class :character   Class :character  
 Median :39.00   Mode  :character   Median :30.40   Median :1.000   Mode  :character   Mode  :character  
 Mean   :39.21                      Mean   :30.66   Mean   :1.095                                        
 3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000                                        
 Max.   :64.00                      Max.   :53.13   Max.   :5.000                                        
    charges     
 Min.   : 1122  
 1st Qu.: 4740  
 Median : 9382  
 Mean   :13270  
 3rd Qu.:16640  
 Max.   :63770  
~~~

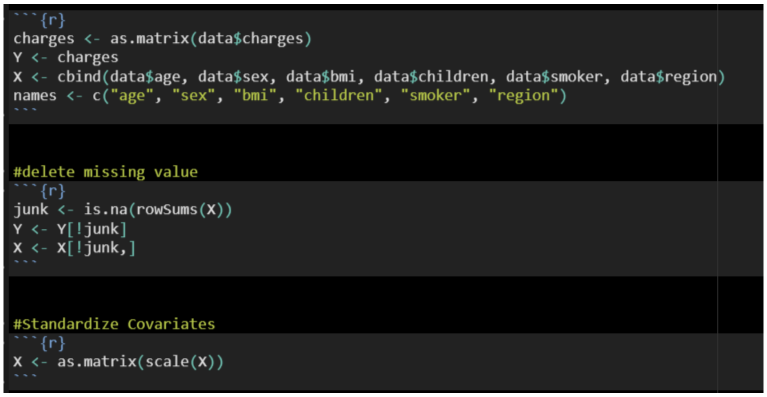


~~~
```{r}
print(data)
```

  age   sex                bmi   children  smoker   region            charges
<int>  <chr>             <dbl>      <int>  <chr>    <chr>               <dbl>
   19  female           27.900          0  yes      southwest       16884.924
   18  male             33.770          1  no       southeast        1725.552
   28  male             33.000          3  no       southeast        4449.462
   33  male             22.705          0  no       northwest       21984.471
   32  male             28.880          0  no       northwest        3866.855
   31  female           25.740          0  no       southeast        3756.622
   46  female           33.440          1  no       southeast        8240.590
   37  female           27.740          3  no       northwest        7281.506
   37  male             29.830          2  no       northeast        6406.411
   60  female           25.840          0  no       northwest       28923.137
1-10 of 1,338 rows                            Previous 1 2 3 4 5 6 ... 100 Next
~~~



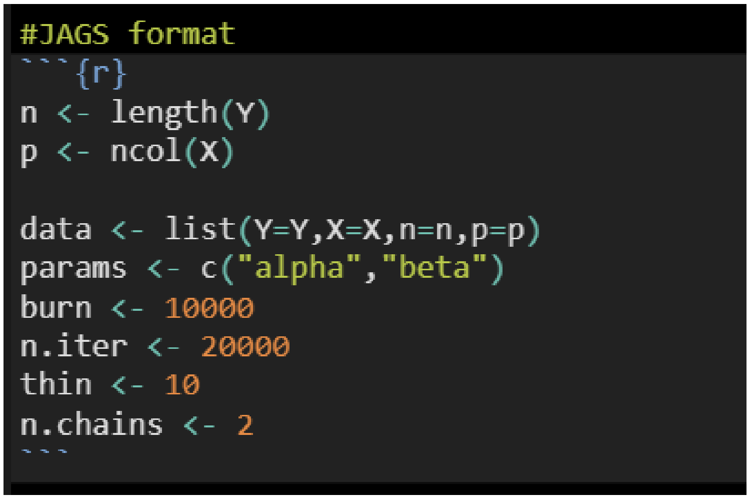
~~~
#String to binary
```{r}
data$smoker[data$smoker == "yes"] <- 1
data$smoker[data$smoker == "no"] <- 0
data$smoker <- as.numeric(data$smoker, coerce = TRUE)

data$sex[data$sex == "female"] <- 1
data$sex[data$sex == "male"] <- 0
data$sex <- as.numeric(data$sex, coerce = TRUE)

data$region[data$region == "southwest"] <- 1
data$region[data$region == "northwest"] <- 2
data$region[data$region == "northeast"] <- 3
data$region[data$region == "southeast"] <- 4
data$region <- as.numeric(data$region, coerce = TRUE)

print(data)

```

  age    sex      bmi   children   smoker   region     charges
<int>  <dbl>    <dbl>      <int>    <dbl>    <dbl>       <dbl>
   19      1   27.900          0        1        1   16884.924
   18      0   33.770          1        0        4    1725.552
   28      0   33.000          3        0        4    4449.462
   33      0   22.705          0        0        2   21984.471
   32      0   28.880          0        0        2    3866.855
   31      1   25.740          0        0        4    3756.622
   46      1   33.440          1        0        4    8240.590
   37      1   27.740          3        0        2    7281.506
   37      0   29.830          2        0        3    6406.411
   60      1   25.840          0        0        2   28923.137
1-10 of 1,338 rows             Previous 1 2 3 4 5 6 ... 100 Next
~~~

The provided R code aims to preprocess a dataset named `data` by converting categorical variables into a numeric format suitable for analysis or modeling. The variables `smoker` and `sex` are transformed into binary numeric representations, with "yes" and "female" coded as 1, and "no" and "male" coded as 0, respectively. Additionally, the variable `region` is encoded into numeric values based on the regions "southwest," "northwest," "northeast," and "southeast," with each region assigned a corresponding numeric code (1, 2, 3, and 4). The resulting dataset is then printed to facilitate further examination or utilization in statistical analyses.

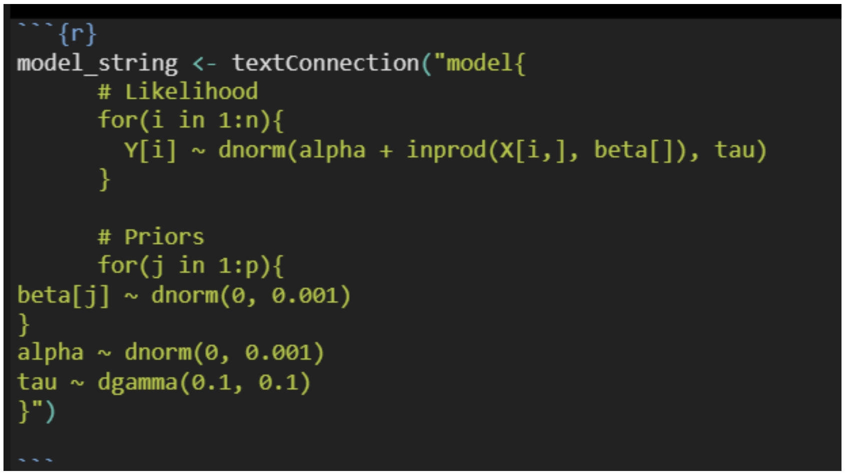


~~~
```{r}
charges <- as.matrix(data$charges)
Y <- charges
X <- cbind(data$age, data$sex, data$bmi, data$children, data$smoker, data$region)
names <- c("age", "sex", "bmi", "children", "smoker", "region")
~~~

#delete missing value
```{r}
junk <- is.na(rowSums(X))
Y <- Y[!junk]
X <- X[!junk,]
```

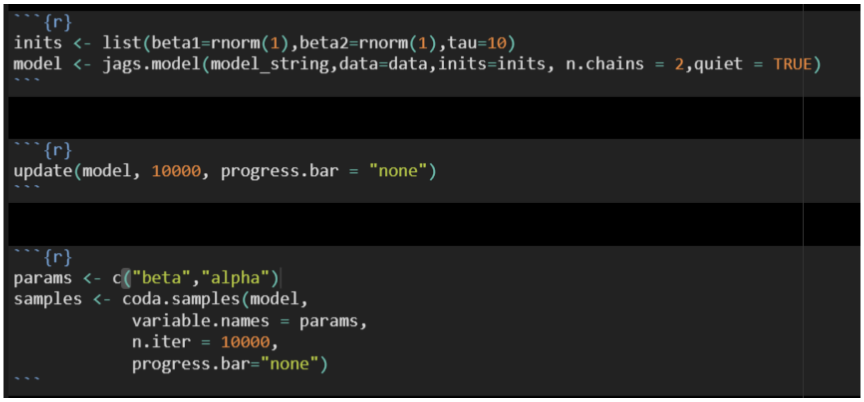
#Standardize Covariates
```{r}
X <- as.matrix(scale(X))
```
```

The R code provided is involved in preparing data for regression analysis. Initially, it extracts the 'charges' variable from the dataset and assigns it to the matrix 'Y'. The independent variables, including age, sex, BMI, children, smoker status, and region, are combined into a matrix 'X'. Subsequently, any rows with missing values in the independent variables or 'Y' are removed. Finally, the independent variables in 'X' are standardized by centering and scaling. The resulting matrices, 'X' and 'Y', along with variable names, are set up for further use in regression modeling or analysis. This code is particularly useful for handling missing values and ensuring that the independent variables are on a comparable scale for regression analysis.



```
#JAGS format
```{r}
n <- length(Y)
p <- ncol(X)

data <- list(Y=Y,X=X,n=n,p=p)
params <- c("alpha","beta")
burn <- 10000
n.iter <- 20000
thin <- 10
n.chains <- 2
```
```

The provided R code is setting up data and parameters in the format required for a JAGS (Just Another Gibbs Sampler) analysis, which is commonly used for Bayesian statistical modeling. The length of the response variable 'Y' is assigned to 'n,' and the number of columns in the matrix of independent variables 'X' is assigned to 'p.' These variables, along with 'Y,' 'X,' 'n,' and 'p,' are organized into a list named 'data.' The parameters to be estimated in the Bayesian model, namely the intercept 'alpha' and regression coefficients 'beta,' are specified in the 'params' vector. The code also defines additional parameters such as burn-in iterations ('burn'), total number of iterations ('n.iter'), thinning parameter
```

('thin'), and the number of chains for the Gibbs sampler ('n.chains'). This code snippet establishes the groundwork for implementing a Bayesian regression model using JAGS with the specified data and parameters.



~~~~
```{r}
model_string <- textConnection("model{
      # Likelihood
      for(i in 1:n){
        Y[i] ~ dnorm(alpha + inprod(X[i,], beta[]), tau)
      }

      # Priors
      for(j in 1:p){
beta[j] ~ dnorm(0, 0.001)
}
alpha ~ dnorm(0, 0.001)
tau ~ dgamma(0.1, 0.1)
}")

```
~~~~

The R code defines a Bayesian regression model in JAGS (Just Another Gibbs Sampler) using a text connection. The model includes a likelihood term for the response variable 'Y' given the predictors 'X' with coefficients 'alpha' and 'beta.' The prior distributions for the regression coefficients and intercept are specified as normal distributions with mean 0 and precision of 0.001. The precision parameter 'tau' for the likelihood has a gamma prior with shape and rate parameters both set to 0.1. This Bayesian model is designed to estimate the regression parameters and allows for making probabilistic inferences about the relationship between the predictors and the response variable. It provides a framework for Bayesian analysis using a Gibbs sampler.

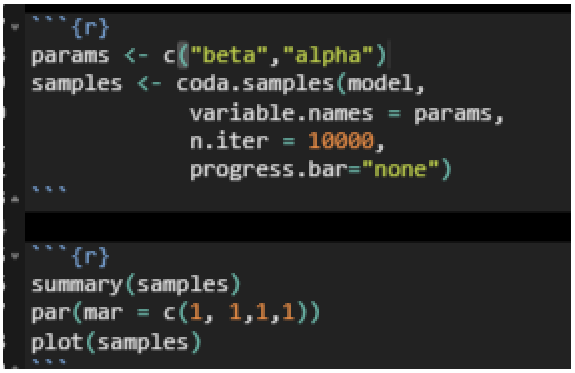


~~~~
```{r}
inits <- list(beta1=rnorm(1),beta2=rnorm(1),tau=10)
model <- jags.model(model_string,data=data,inits=inits, n.chains = 2,quiet = TRUE)
```

```{r}
update(model, 10000, progress.bar = "none")
```

```{r}
params <- c("beta","alpha")
samples <- coda.samples(model,
            variable.names = params,
            n.iter = 10000,
            progress.bar="none")
```
~~~~

The provided code segment employs the JAGS (Just Another Gibbs Sampler) software within R for conducting Bayesian analysis. Initially, it sets up a model with specified initial parameter values for beta1, beta2, and tau. Subsequently, it executes a burn-in phase of 10,000 iterations to enhance convergence without displaying a progress bar. Following this, it proceeds to sample from the model, focusing on the parameters beta and alpha, utilizing the MCMC (Markov Chain Monte Carlo) technique over 100,000 iterations. This process aims to derive the posterior distributions of the specified parameters based on the defined model and input data.



~~~
```{r}
params <- c("beta","alpha")
samples <- coda.samples(model,
            variable.names = params,
            n.iter = 10000,
            progress.bar="none")
```

```{r}
summary(samples)
par(mar = c(1, 1,1,1))
plot(samples)
```
~~~

This code extends the Bayesian analysis by sampling beta and alpha parameters over 10,000 iterations. It then summarizes these samples, likely showcasing key statistics, and generates plots—possibly trace or density plots—to visualize the behavior and convergence of these parameters. Marginal adjustments refine the appearance of the plots. Overall, it helps explore the characteristics of beta and alpha within the Bayesian framework.

The code sets up a Bayesian analysis using JAGS in R. It initializes a model with specific values for parameters and runs a 10,000 iteration burn-in phase. Then, it samples beta and alpha parameters for 100,000 iterations using MCMC. The goal is to estimate the posterior distributions of these parameters based on the defined model and input data.