Topic: Analyzing diabetes data using Bayesian logistic regression

Name: Roshan Asim

Matriculation: 3115566

Bioinformatics

Introduction:

Bayesian logistic regression using Stan. The purpose of the code is to fit a logistic regression model to predict the presence or absence of diabetes based on several predictor variables such as age, BMI, glucose level, etc.

Data source:

https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Workflow:

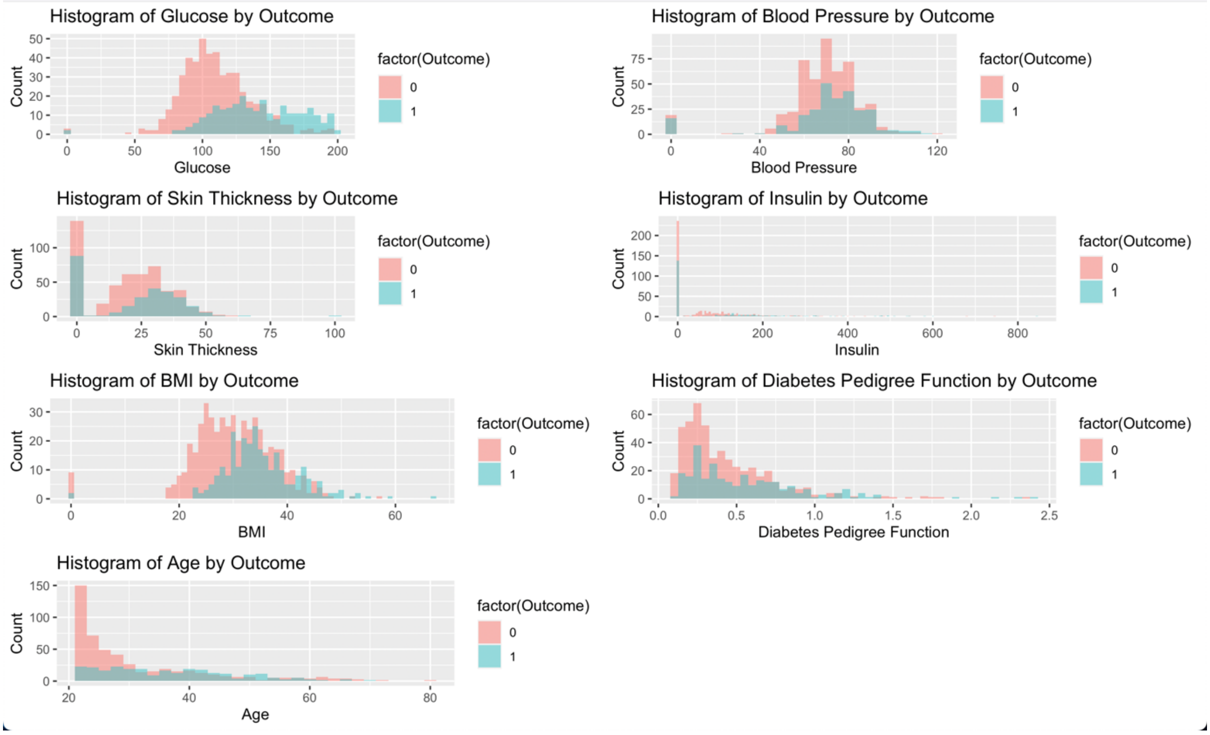
1. The required libraries are loaded at the beginning of the script using the library() function. These libraries are required to perform the subsequent data analysis and visualization steps.

```
library(rstan)
library(dplyr)
library(tidyr)
library(bayesplot)
library(ggplot2)
library(gridExtra)
```

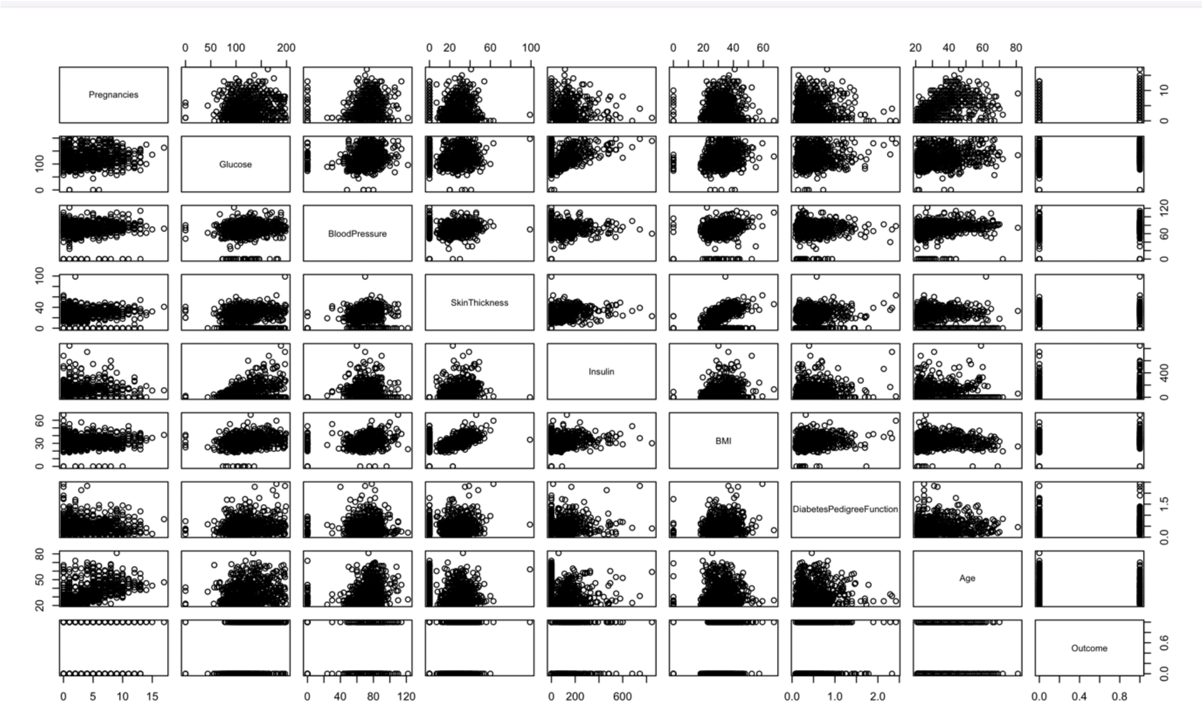
2. The read.csv() function is used to read in a CSV file called "diabetes.csv" which contains data related to diabetes.

```
diabetes_data <- read.csv("diabetes.csv")
```

3. Histograms for each variable in the dataset are created using the ggplot2 package. The histograms are grouped by the variable "Outcome", which indicates whether or not the patient has diabetes. The gridExtra package is used to arrange the plots into a grid.



4. A scatterplot matrix of the dataset is created using the pairs() function, which plots all pairwise combinations of variables.

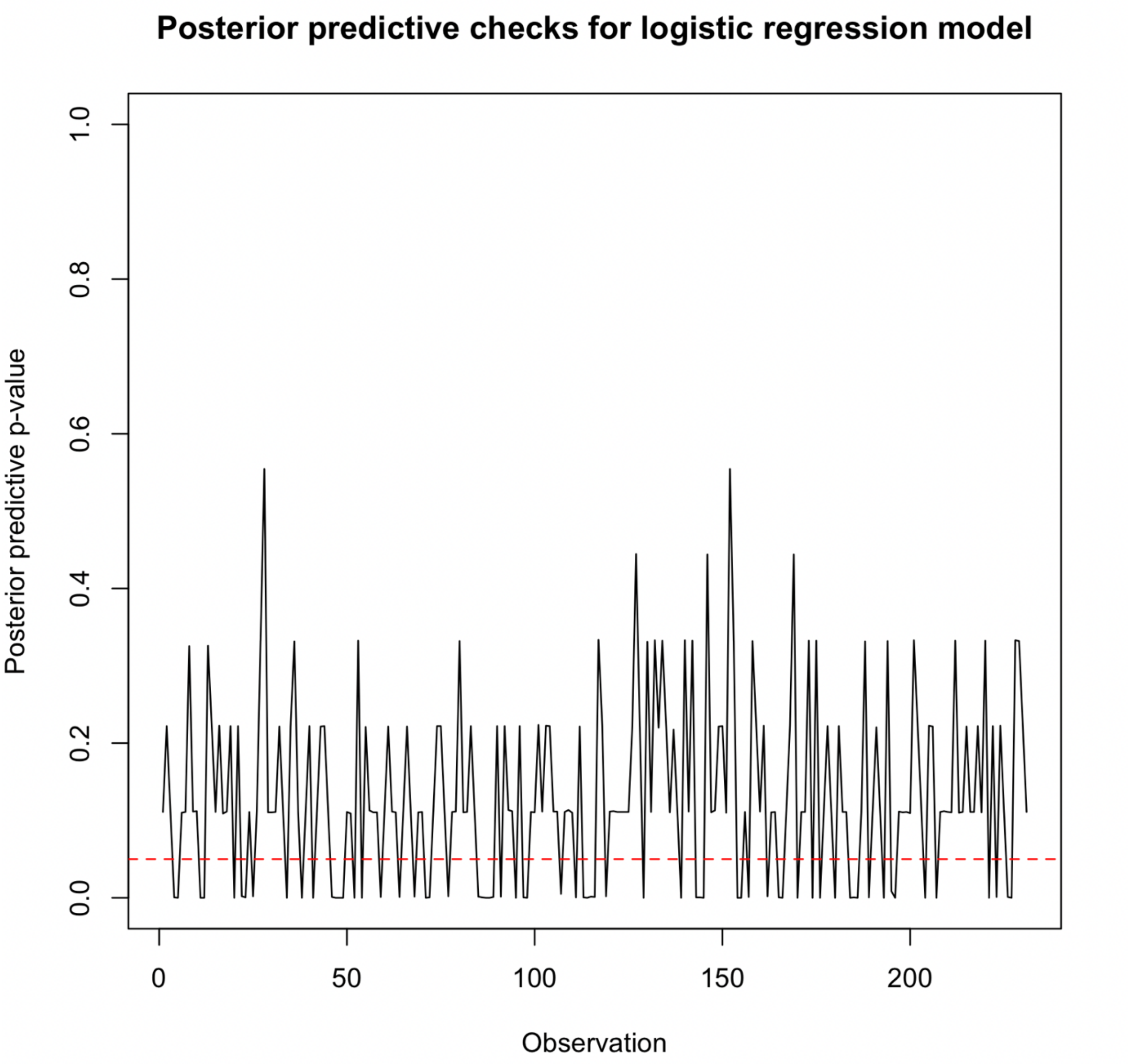


5. The dataset is split into a training set (70% of the data) and a testing set (30% of the data) using the sample() function.

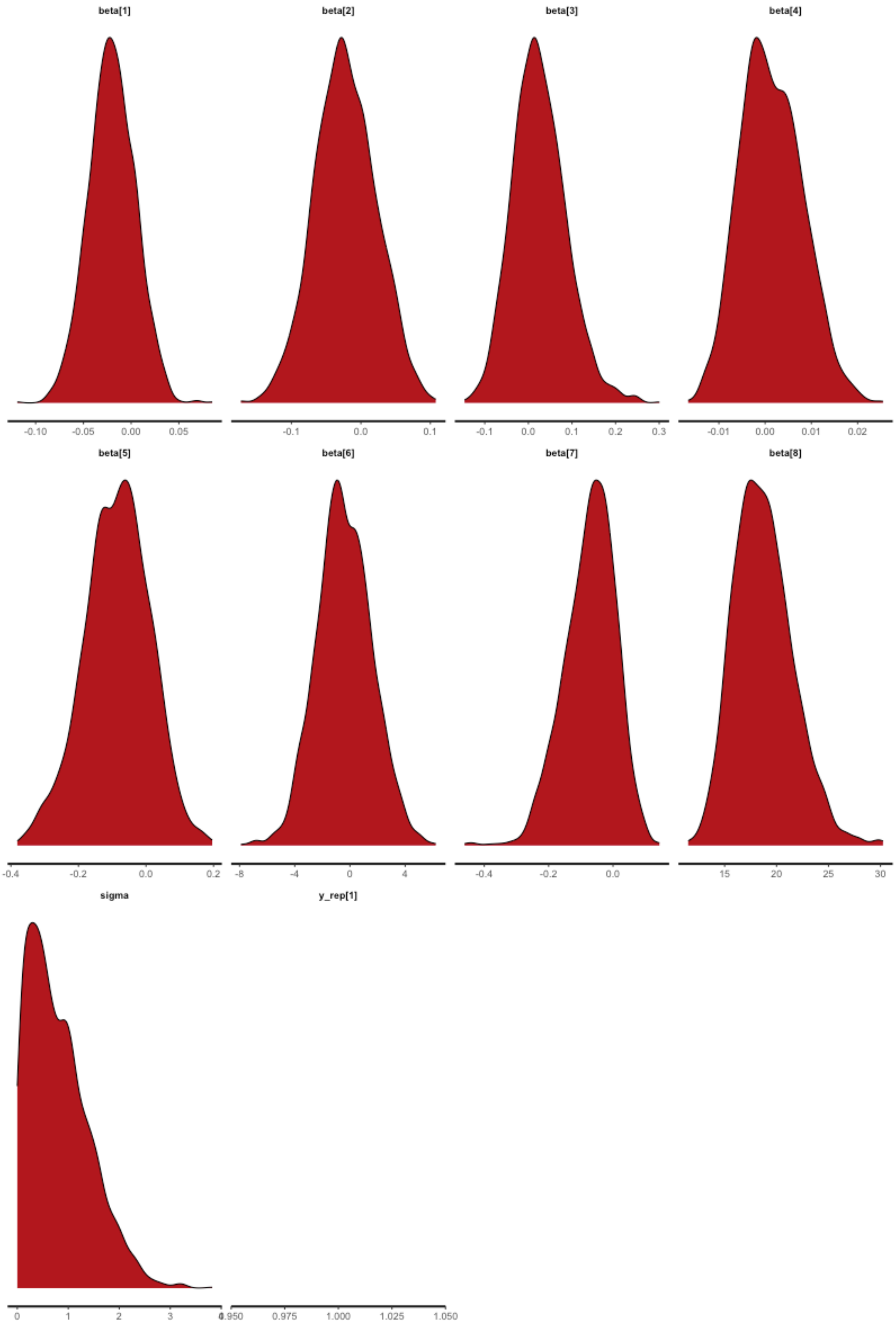
6. The stan() function is used to compile and fit a Bayesian logistic regression model to the training data. The model is specified in a separate file called "bayesian.stan" and model summary is generated.

```
Chain 4:
Chain 4: Iteration:    1 / 1000 [  0%]  (Warmup)
Chain 4: Iteration:  100 / 1000 [ 10%]  (Warmup)
Chain 4: Iteration:  200 / 1000 [ 20%]  (Warmup)
Chain 4: Iteration:  300 / 1000 [ 30%]  (Warmup)
Chain 4: Iteration:  400 / 1000 [ 40%]  (Warmup)
Chain 4: Iteration:  500 / 1000 [ 50%]  (Warmup)
Chain 4: Iteration:  501 / 1000 [ 50%]  (Sampling)
Chain 4: Iteration:  600 / 1000 [ 60%]  (Sampling)
Chain 4: Iteration:  700 / 1000 [ 70%]  (Sampling)
Chain 4: Iteration:  800 / 1000 [ 80%]  (Sampling)
Chain 4: Iteration:  900 / 1000 [ 90%]  (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.809856 seconds (Warm-up)
Chain 4:                0.316504 seconds (Sampling)
Chain 4:                1.12636 seconds (Total)
Chain 4:
$summary
                mean     se_mean          sd        2.5%          25%           50%          75%       97.5%     n_eff      Rhat
beta[1]  -0.019988088 0.0008599581 0.024362770 -0.06748816 -0.036207219 -2.037965e-02 -0.003399846  0.02742704  802.5993 1.0038899
beta[2]  -0.022289832 0.0013353522 0.043930699 -0.10850700 -0.052651351 -2.400699e-02  0.007238833  0.06171803 1082.2923 1.0019178
beta[3]   0.025810767 0.0020280733 0.061628263 -0.08152523 -0.016790825  2.139137e-02  0.063900954  0.15645629  923.4057 1.0018029
beta[4]   0.001401566 0.0001728343 0.006646631 -0.01043074 -0.003381738  9.078532e-04  0.006015679  0.01492865 1478.9159 1.0006921
beta[5]  -0.085563762 0.0030874008 0.096406555 -0.29069886 -0.148072280 -8.157122e-02 -0.019888461  0.09486836  975.0505 1.0032174
beta[6]  -0.407779204 0.0618494681 1.989015012 -4.07617408 -1.702031325 -4.983624e-01  0.916220106  3.60093366 1034.1992 1.0009462
beta[7]  -0.071763353 0.0027532331 0.079419883 -0.23948581 -0.121717602 -6.403241e-02 -0.015931349  0.06382699  832.0942 1.0059556
beta[8]  18.749227812 0.1002139903 2.848250533 13.93015011 16.709967022  1.849031e+01 20.534475978 24.82108379  807.7922 1.0026721
sigma     0.810154086 0.0158010196 0.604282739  0.02839282  0.326330296  6.894384e-01  1.175036832  2.21674921 1462.5481 0.9998399
```

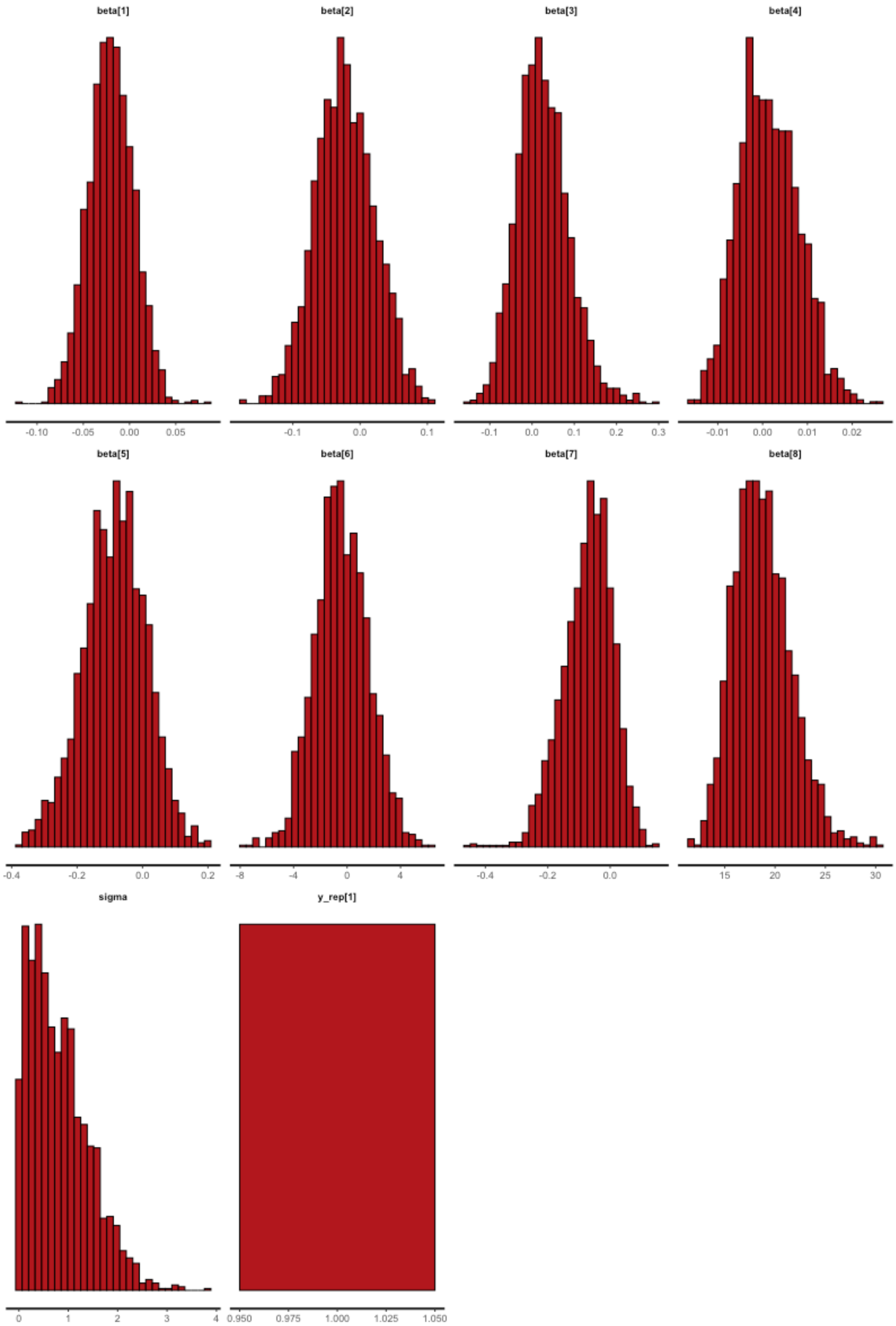
7. The posterior samples are extracted using the extract() function. The posterior predictive distribution is generated using the posterior samples. The posterior predictive checks are then calculated and plotted using the sapply() and plot() functions, respectively. The red line in the plot represents the significance level of 0.05.

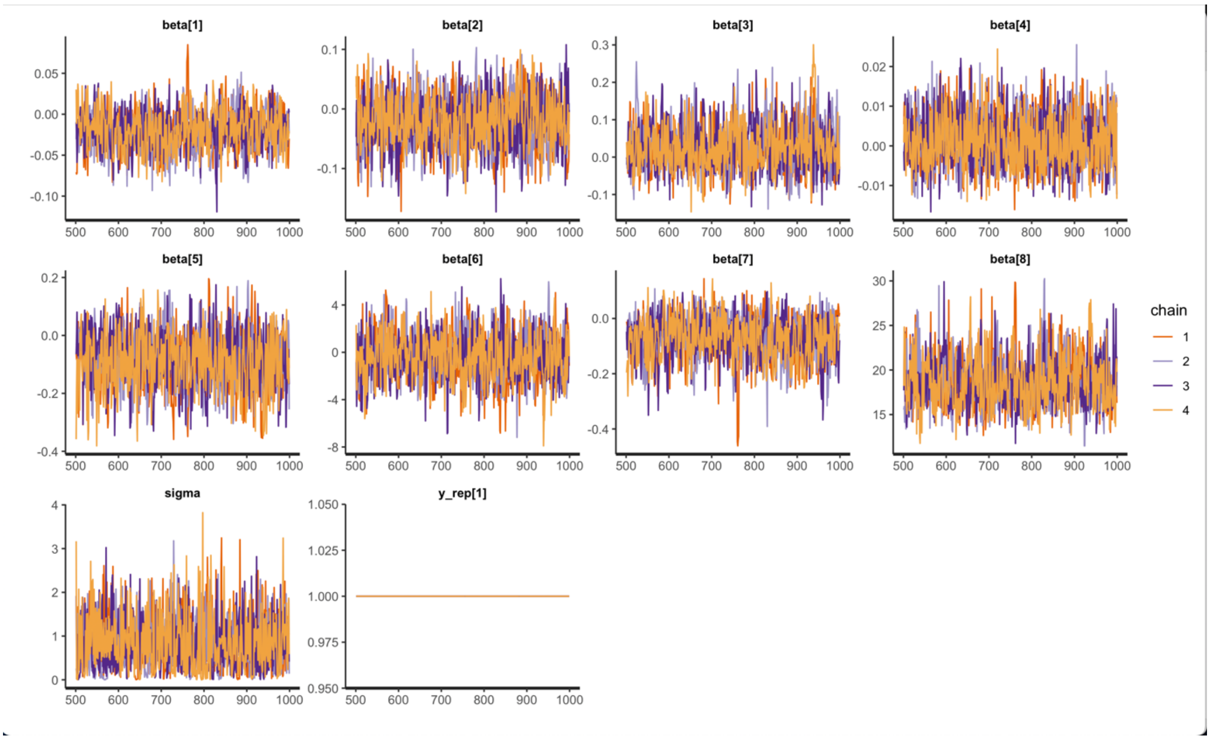


8. Finally, the traceplot(), stan_dens(), and stan_hist() functions are used to visualize the posterior distribution of the model parameters.



beta[1]
beta[2]
beta[3]
beta[4]
-0.10 -0.05 0.00 0.05
-0.1 0.0 0.1
-0.1 0.0 0.1 0.2 0.3
-0.01 0.00 0.01 0.02
beta[5]
beta[6]
beta[7]
beta[8]
-0.4 -0.2 0.0 0.2
-8 -4 0 4
-0.4 -0.2 0.0
15 20 25 30
sigma
y_rep[1]
0 1 2 3 4
0.950 0.975 1.000 1.025 1.050


beta[1]
-0.10 -0.05 0.00 0.05
beta[2]
-0.1 0.0 0.1
beta[3]
-0.1 0.0 0.1 0.2 0.3
beta[4]
-0.01 0.00 0.01 0.02
beta[5]
-0.4 -0.2 0.0 0.2
beta[6]
-8 -4 0 4
beta[7]
-0.4 -0.2 0.0
beta[8]
15 20 25 30
sigma
0 1 2 3 4
y_rep[1]
0.950 0.975 1.000 1.025 1.050



beta[1]
beta[2]
beta[3]
beta[4]
beta[5]
beta[6]
beta[7]
beta[8]
sigma
y_rep[1]
chain
1
2
3
4