Bayesian Logistic Regression Model To Predict Patients Diabetes

Daisy Shi

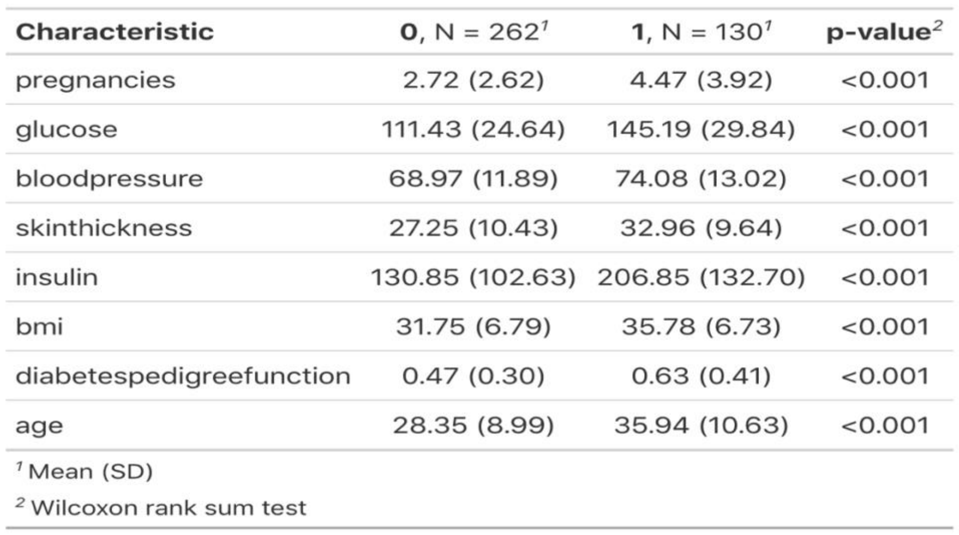
12/10/2021

Abstract: Based on the uncertainty problems in machine learning, it is an effective method to use probability to quantify the uncertainty in the inference of statistical data analysis. The Bayesian approach allows us to make a prior good guess of the intercept and slope, based on our real-life domain knowledge and common sense. In this study, I am interested in build a Bayesian Logistic Regression model to predict whether the patients in the dataset have diabetes or not.
Keywords: Bayesian logistic regression

Introduction

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Total 768 observation and 9 variables, after removing those observation rows with 0 in any of the variables, our study contains 392 observations. The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. All patients here are females at least 21 years old of Pima Indian heritage. In this study, I am going to build a Bayesian Logistic Regression model to predict whether the patients in the dataset have diabetes. Table 1 will help us to understand the overall description of the dataset.

Table 1: summary table and attribute description



Characteristic	**0**, N = 262[1]	**1**, N = 130[1]	**p-value**[2]
pregnancies	2.72 (2.62)	4.47 (3.92)	<0.001
glucose	111.43 (24.64)	145.19 (29.84)	<0.001
bloodpressure	68.97 (11.89)	74.08 (13.02)	<0.001
skinthickness	27.25 (10.43)	32.96 (9.64)	<0.001
insulin	130.85 (102.63)	206.85 (132.70)	<0.001
bmi	31.75 (6.79)	35.78 (6.73)	<0.001
diabetespedigreefunction	0.47 (0.30)	0.63 (0.41)	<0.001
age	28.35 (8.99)	35.94 (10.63)	<0.001

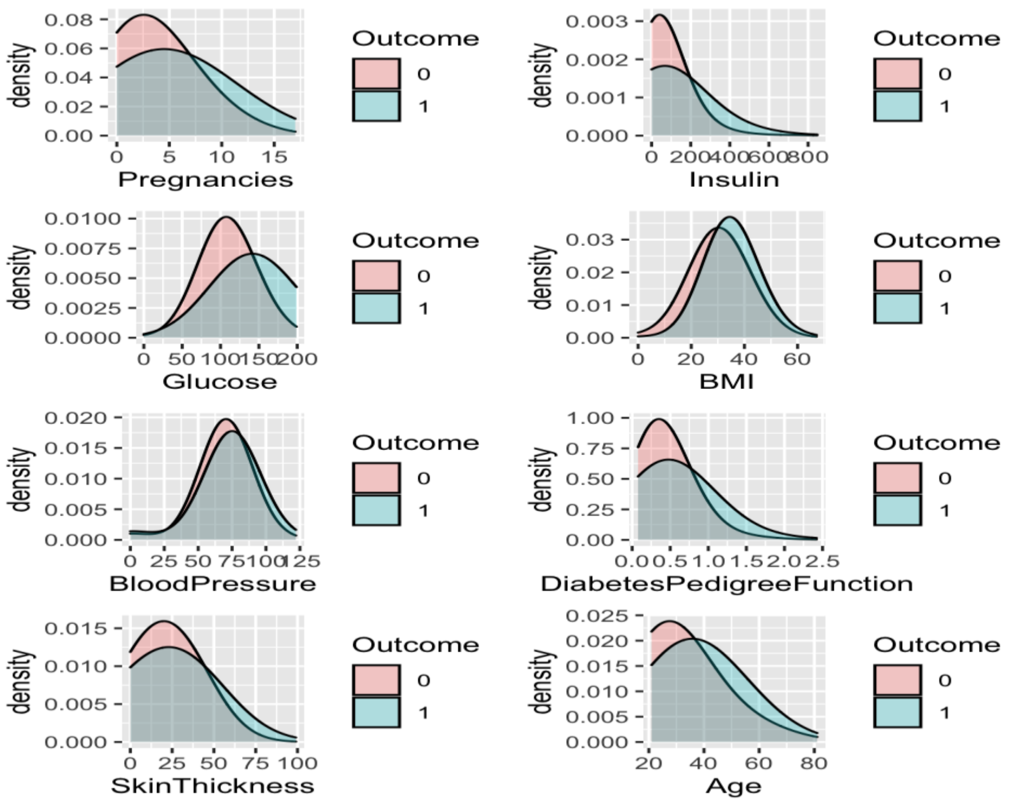
[1] Mean (SD)

[2] Wilcoxon rank sum test

Motivating Data

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on. Table 1 shows the attribute description with mean and standard deviation excluded the missing values.

Figure 1. Density plots for all attributes



Bayesian Logistic Regression Model

In general, there are four steps of a Bayesian analysis in this study [1].

1. Specify a joint distribution for the outcome(s) and all the unknowns, which typically takes the form of a marginal prior distribution for the unknowns multiplied by a likelihood for the outcome(s) conditional on the unknowns. This joint distribution is proportional to a posterior distribution of the unknowns conditional on the observed data.
2. Draw from posterior distribution using Markov Chain Monte Carlo (MCMC)[2].
3. Evaluate how well the model fits the data and possibly revise the model.
4. Draw from the posterior predictive distribution of the outcome(s) given interesting values of the predictors to visualize how a manipulation of a predictor affects (a function of) the outcome(s).

Priors

A full Bayesian analysis requires specifying prior distributions f(α) and f(β) for the intercept and vector of regression coefficients. Suppose we have 8 predictors and believe, prior to seeing the data, that α, β1,…,βK are as likely to be positive as they are to be negative, but are highly unlikely to be far from zero. These beliefs can be represented by normal distributions with mean zero and a small scale (standard deviation 1).

Likelihood

For a binomial GLM the likelihood for one observation y can be written as a conditionally binomial PMF. $\binom{n}{y}\pi^y(1-\pi)^{n-y},$ where n is the known number of trials, π=g−1(η) is the probability of success and η=α+x⊤β is a linear predictor. For a sample of size N , the likelihood of the entire sample is the product of N individual likelihood contributions.

Because π is a probability, for a binomial model the link function g maps between the unit interval (the support of π) and the set of all real numbers ℝ . When applied to a linear predictor η with values in ℝ , the inverse link function g−1(η) therefore returns a valid probability between 0 and 1.

The likelihood for a single observation becomes as follow:

$$\binom{n}{y}\left(\operatorname{logit}^{-1}(\eta)\right)^y\left(1-\operatorname{logit}^{-1}(\eta)\right)^{n-y} = \binom{n}{y}\left(\frac{e^\eta}{1+e^\eta}\right)^y\left(\frac{1}{1+e^\eta}\right)^{n-y}$$

Posterior

With independent prior distributions, the joint posterior distribution for α and β is proportional to the product of the priors and the N likelihood contributions

$$f(\alpha,\boldsymbol{\beta}|\mathbf{y},\mathbf{X}) \propto f(\alpha)\times\prod_{k=1}^{K} f(\beta_k)\times\prod_{i=1}^{N} g^{-1}(\eta_i)^{y_i}\left(1-g^{-1}(\eta_i)\right)^{n_i-y_i}.$$

Main Results

Normal distributions with mean zero and a small scale (standard deviation 1) are used in the model, and it is a reasonable default prior when coefficients should be close to zero but have some chance of being large. The model returns the posterior distribution for the parameters describing the uncertainty related to unknown parameter values. The uncertainty intervals are computed by finding the relevant quantiles of the draws from the posterior distribution.

We got a corresponding posterior median estimate as follows:

```
##   (Intercept)   pregnancies       glucose bloodpressure skinthickness
##         -1.01          0.26          1.20         -0.02          0.12
##       insulin           bmi           dpf           age
##         -0.10          0.50          0.40          0.35
```

95% CI we got results we follows:

```
##                2.5% 97.5%
## (Intercept)   -1.31 -0.74
## pregnancies   -0.09  0.61
## glucose        0.86  1.59
## bloodpressure -0.31  0.29
## skinthickness -0.25  0.48
## insulin       -0.41  0.20
## bmi            0.12  0.89
## dpf            0.12  0.69
## age           -0.03  0.73
```

Using Pareto smoothed leave-one-out cross-validation (PSIS-LOO)[3] to compute expected log predictive density. Computed from 4000 by 392 log-likelihood matrix, we see that PSIS-LOO result is reliable as all Pareto k estimates are small ($k< 0.5$) and Monte Carlo SE of elpd_loo is 0.1. By build a baseline model without covariates and to compare with the PSIS-LOO model, we see that covariates contain clearly useful information for predictions.

Discussion

Bayesian logistic regression is not an algorithm, but a different method of statistical inference. The main advantage is that, with Bayesian processing, you can recover the entire range of extrapolation understanding, rather than the point estimates and confidence intervals found in traditional regression. In addition, the point estimates reported coefficient for this predictor suggests that there was a negative coefficient with insulin, it indicated that the higher the insulin the less likely they were to develop diabetes. In other words, those who took insulin were 9.5% less likely to develop diabetes than those who did not. In conclusion, we have strong evidence that the Bayesian logistic regression model has great performance.

References

1. Gelman, A. and Hill, J. (2007). Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press, Cambridge, UK.
2. Gelman, A., & Shirley, K. (2011). Inference from simulations and monitoring convergence. In S. Brooks, A. Gelman, G. Jones, & X. Meng (Eds.), *Handbook of Markov chain Monte Carlo*. Boca Raton: Chapman & Hall/CRC.
3. Stan Development Team. (2015). *Stan modeling language user's guide and reference manual, Version 2.9.0*. http://mc-stan.org/documentation. See the 'Hamiltonian Monte Carlo Sampling' chapter.
4. Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4), 457 – 472.
5. Gelman, A., & Shirley, K. (2011). Inference from simulations and monitoring convergence. In S. Brooks, A. Gelman, G. Jones, & X. Meng (Eds.), *Handbook of Markov chain Monte Carlo*. Boca Raton: Chapman & Hall/CRC.

Appendix (R Codes)

~~~
```{r,echo=FALSE, warning=FALSE, results='hide',message=FALSE}
library(ggplot2)
diabetes<- read.csv("/学习课件/STA545 Baysian/diabetes.csv",header = TRUE)
summary(diabetes)
str(diabetes)  #768 obs. of  9 variables
diabetes$Outcome <- factor(diabetes$Outcome)
# removing those observation rows with 0 in any of the variables
for (i in 2:6) {
 diabetes <- diabetes[-which(diabetes[, i] == 0), ]
}
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
 library(grid)
 # Make a list from the ... arguments and plotlist
 plots <- c(list(...), plotlist)
 numPlots = length(plots)
 # If layout is NULL, then use 'cols' to determine layout
 if (is.null(layout)) {
  # Make the panel
  # ncol: Number of columns of plots
  # nrow: Number of rows needed, calculated from # of cols
  layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                   ncol = cols, nrow = ceiling(numPlots/cols))
 }

 if (numPlots==1) {
  print(plots[[1]])
 } else {
  # Set up the page
~~~

```
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))


    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))


      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
a<-ggplot(diabetes,aes(Pregnancies,fill=Outcome))+  geom_density(alpha = 0.3,adjust = 5)
+plot_title
b<-ggplot(diabetes,aes(Glucose,fill=Outcome))+  geom_density(alpha = 0.3,adjust = 5)
c<-ggplot(diabetes,aes(BloodPressure,fill=Outcome))+  geom_density(alpha = 0.3,adjust =
5)
d<-ggplot(diabetes,aes(SkinThickness,fill=Outcome))+  geom_density(alpha = 0.3,adjust =
5)
e<-ggplot(diabetes,aes(Insulin,fill=Outcome))+  geom_density(alpha = 0.3,adjust = 5)
f<-ggplot(diabetes,aes(BMI,fill=Outcome))+  geom_density(alpha = 0.3,adjust = 5)
g<-ggplot(diabetes,aes(DiabetesPedigreeFunction,fill=Outcome))+  geom_density(alpha =
0.3,adjust = 5)
h<-ggplot(diabetes,aes(Age,fill=Outcome))+  geom_density(alpha = 0.3,adjust = 5)
plot_title<- ggtitle("Figure 1",
                     "Density plots for all attributes")
MULTI<-multiplot(a,b,c,d,e,f,g,h,cols=2)
# scale the covariates for easier comparison of coefficient posteriors
for (i in 1:8) {
  diabetes[i] <- scale(diabetes[i])
```

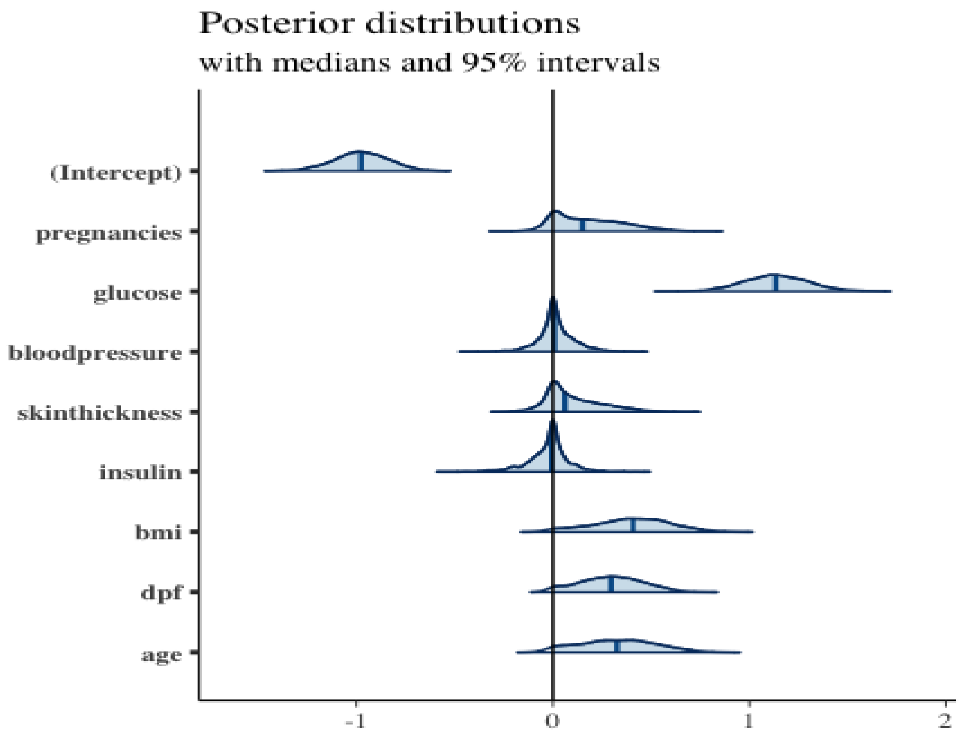
```
}
# modify the data column names slightly for easier typing
names(diabetes) <- tolower(names(diabetes))
n=dim(diabetes)[1]
p=dim(diabetes)[2]
str(diabetes)
print(paste0("number of observations = ", n))
print(paste0("number of predictors = ", p))

# preparing the inputs
x <- model.matrix(outcome ~ . - 1, data = diabetes)
y <- diabetes$outcome
library(rstanarm)
options(mc.cores = parallel::detectCores())
prior = normal(0,1)
post1 <- stan_glm(outcome ~ ., data = diabetes,
          family = binomial(link = "logit"),
          prior = prior, prior_intercept = prior, QR=TRUE,
          seed = 14124869)
library(ggplot2)
pplot<-plot(post1, "areas", prob = 0.95, prob_outer = 1)
round(posterior_interval(post1, prob = 0.95), 2)
plot_title <- ggtitle("Posterior distributions",
            "with medians and 95% intervals")
pplot+ geom_vline(xintercept = 0) + plot_title
round(coef(post1), 1)
round(posterior_interval(post1, prob = 0.95), 2)
library(loo)
loo1 <- loo(post1, save_psis = TRUE)
```

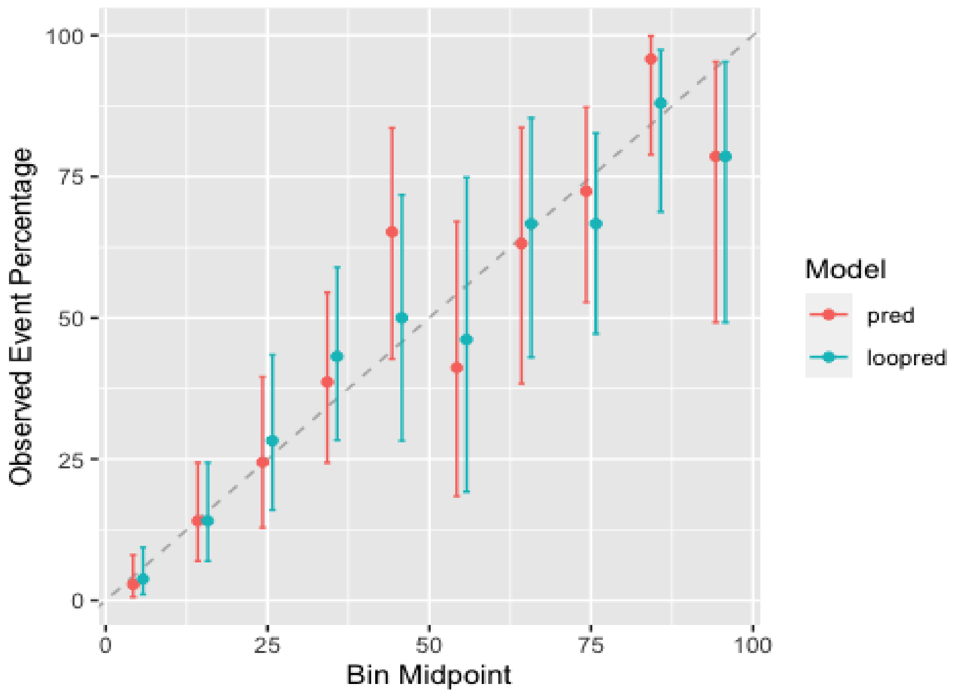
```
post0 <- update(post1, formula = outcome ~ 1, QR = FALSE)#Compute baseline result
without covariates.

loo0 <- loo(post0)

compare<- rstanarm::compare_models(loo0,loo1)

# Predicted probabilities

linpred <- posterior_linpred(post1)

preds <- posterior_linpred(post1, transform=TRUE)

pred <- colMeans(preds)

pr <- as.integer(pred >= 0.5)

table(pr,y)

mean(pr==y)

mean(pr!=y)

# confusion matrix

caret::confusionMatrix(as.factor(as.numeric(pr>0.5)), y)[2]

# posterior classification accuracy

round(mean(xor(pr,as.integer(y==0))),2)

# posterior balanced classification accuracy

round((mean(xor(pr[y==0]>0.5,as.integer(y[y==0])))+mean(xor(pr[y==1]<0.5,as.integer(y
[y==1]))))/2,2)

# PSIS-LOO weights

log_lik=log_lik(post1, parameter_name = "log_lik")

psis=psislw(-log_lik)

#plot(psis$pareto_k)

#plot(psis$lw_smooth[,1],linpred[,1])

# LOO predictive probabilities

ploo=colSums(preds*exp(psis$lw_smooth))

# LOO classification accuracy

round(mean(xor(ploo>0.5,as.integer(y==0))),2)

# LOO balanced classification accuracy
```

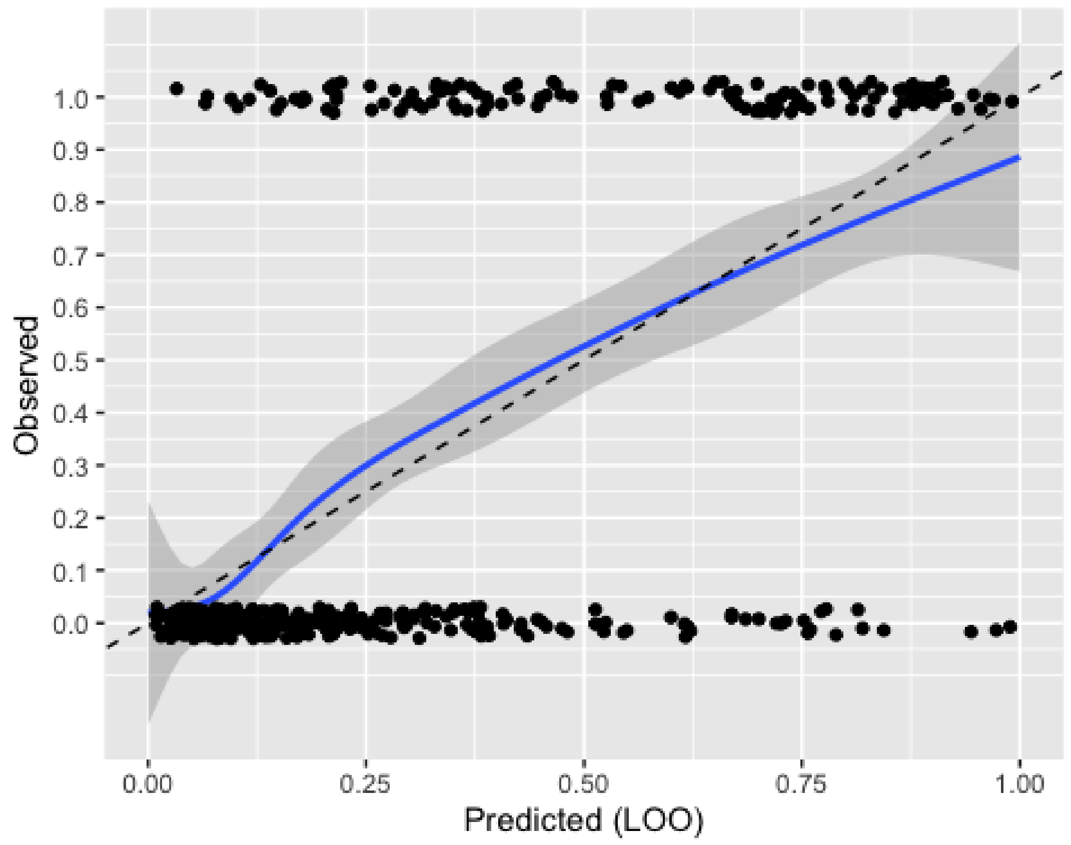
```
round((mean(xor(ploo[y==0]>0.5,as.integer(y[y==0])))+mean(xor(ploo[y==1]<0.5,as.inte
ger(y[y==1]))))/2,2)

plot(pred,ploo)

calPlotData<-caret::calibration(y ~ pred + loopred,

                    data = data.frame(pred=pred,loopred=ploo,y=y),

                    cuts=10, class="1")

plot_title1 <- ggtitle("Posterior Predictive Intervals \nvs Observed Event Percentage"

                )

ggplot(calPlotData, auto.key = list(columns = 2))+plot_title1

library(splines)

library(MASS)

ggplot(data = data.frame(pred=pred,loopred=ploo,y=as.numeric(y)-1), aes(x=loopred,
y=y)) +

  stat_smooth(method='glm', formula = y ~ ns(x, 5), fullrange=TRUE) +

  geom_abline(linetype = 'dashed') + ylab(label = "Observed") + xlab(label = "Predicted
(LOO)") +

  geom_jitter(height=0.03, width=0) + scale_y_continuous(breaks=seq(0,1,by=0.1)) +
xlim(c(0,1))

p0 <- 2 # prior guess for the number of relevant variables

tau0 <- p0/(p-p0) * 1/sqrt(n)

hs_prior <- hs(df=1, global_df=1, global_scale=tau0)

post2 <- stan_glm(outcome ~ ., data = diabetes,

            family = binomial(link = "logit"),

            prior = hs_prior, prior_intercept = prior,

            seed = 14124869, adapt_delta = 0.999)

loo2 <- loo(post2, save_psis = TRUE)

pplot<-plot(post2, "areas", prob = 0.9, prob_outer = 1)

pplot + geom_vline(xintercept = 0)

# Predicted probabilities

linpred2 <- posterior_linpred(post2)

preds2 <- posterior_linpred(post2, transform=TRUE)
```

```
pred2 <- colMeans(preds2)

pr2 <- as.integer(pred2 >= 0.5)

table(pr2,y)

(233+74)/(233+56+29+74)

mean(pr2==y)

mean(pr2!=y)

library(bayesplot)

mcmc_pairs(as.array(post2),pars = c("pregnancies","age","bmi"))
```

Appendix (Plots)







1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Observed
0.00
0.25
0.50
0.75
1.00
Predicted (LOO)