STAT 447C: Final Project

Andrew Tran

2024-04-21

GitHub

https://github.com/nartyuh/ubc-stat-447C-project

Introduction

Breast cancer is one the most common and malignant types of cancers. It is life-threatenting to many people around the world, especially women. There are many endeavours that motivate the research to find solutions for breast cancer treatment through the use of computational methods. The Breast Cancer Wisconsin is one of them. The challenge of this dataset is to develop inferential and predictive analyses on 569 images of tumor cell nucleus obtained by Fine-Needle Aspiration. This can help with imaging analysis of cancerous cells and provide a tool for early detection of cancer risk.

Purpose of the project

The project conducts a comparison between the Frequentist and Bayesian approaches to logistic regression on the Breast Cancer Wisconsin dataset. We first look at the perspectives of both approaches in terms of variable selection when there is multicollinearity among the explanatory variables. Then we examine how the Frequentists and the Bayesians approach model comparison. Finally, we will perform model diagnostics using each approach's respective methodologies.

By making the comparison between Frequentist and Bayesian approaches, this project hopes to create a discourse around utilizing Bayesian approach for Breast Cancer Wisconsin.

Data description

Our response variable is `Diagnosis`, which is categorical. The original dataset labels it as **M** (malignant) or **B** (benign). We want to transform it into binary values so that **1** and **0** are equivalent to **M** and **B** respectively.

The Breast Cancer Wisconsin (Diagnostic) data set has 30 explanatory variables. However, in fact, they have 10 main features. For each feature, the mean, standard error, and worst/largest values are recorded. For example, for a single observation in the data, `radius1` is the mean of distances from center to points on the perimeter, `radius2` is the standard error those distances, and `radius3` is largest distance measured. For the sake of length, we are only interested in the means.

These ten real-valued features are measurements taken and computed for each tumor cell nucleus:

Variable	Type	Description
Radius	Quantitative	Distance from center to points on the perimeter of the tumor
Texture	Quantitative	Gray-scale value
Perimeter	Quantitative	Size of the tumor
Area	Quantitative	Area of the tumor
Smoothness	Quantitative	Local variation in radius length
Compactness	Quantitative	$\frac{\text{perimeter}^2}{\text{area}} - 1$
Concavity	Quantitative	Severity of concave portions of the contour
Concave points	Quantitative	Number of concave portions of the contour
Symmetry	Quantitative	Symmetrical measurement of the tumor
Fractal dimension	Quantitative	coastline approximation − 1

Base model

Frequentist

Fitting a base model with every explanatory variables included is a straightforward process with the Frequentist approach.

```r
frequentist.base_reg <-
  glm(Diagnosis ~ ., data = df, family = binomial(link = "logit"))
```

Bayesian

In contrast to the simplicity of the Frequentist approach, fitting a Bayesian regression model is a more hands-on process.

Let β_0 be the intercept parameter and $\beta_1, ...\beta_{10}$ be the slope parameters for `radius`, `texture`, `perimeter`, `area`, `smoothness`, `compactness`, `concavity`, `concave_points`, `symmetry`, and `fractal_dimension` correspondingly.

To fit a Bayesian logistic regression to our data, we need to specify our Bayesian model. One of the important tasks when specifying our model is the selection of prior distributions. We often select generic weakly informative priors like $Normal(0, 1)$ to perform regression task in Bayesian approach. This choice could work in our case as we are attempting to fit a logistic regression model. However, logistic regression can also become unstable from separation, a problem when the outcome variable separates a predictor variable perfectly. (Bayesian Data Analysis, p. 412) To avoid this problem, Gelman et al. (2008) suggested the Cauchy distribution with center 0 and scale set to 2.5 for the slopes and 10 for the intercept. We will employ this choice of prior to model Bayesian logistic regression. To proceed with this approach, it is required that we center and scale our non-binary variables to have mean 0 and standard deviation 0.5. (Gelman et al., 2008)

$$\beta_0 \sim \text{Cauchy}(0, 10) \tag{1}$$

$$\beta_i \overset{\text{iid}}{\sim} \text{Cauchy}(0, 2.5) \qquad \text{for } i \in \{1, ..., 10\} \tag{2}$$

$$y_n | \beta \sim \text{Bern}(\text{logistic}(\beta_0 + \beta_1 x_{n,1} + ... + \beta_{10} x_{n,10})) \qquad n \in \{1, ..., n_{\text{obs}}\} \tag{3}$$

```r
bayesian.base_model <- stan_model("base_logistic.stan")
bayesian.base_reg <- sampling(
  bayesian.base_model,
  data = bayesian.base_reg.dat,
  seed = 123, iter = 1000
)
```

Model with the exclusion of highly correlated explanatory variables

By looking at the correlation matrix of explanatory variables (Appendix A.1), we identify the following highly correlated features: `radius`, `perimeter`, `area`, `compactness`, `concavity`, and `concave_points`.

The above explanatory variables exhibit multicollinearity. In regression analysis, this is undesirable because it undermines the statistical significance of an independent variable, thus leading to skewed or misleading results that can negatively impact a statistical inference task. (Understanding Regression Analysis, p. 176) To resolve this problem, we need to investigate into the source of the high correlation values and remove the collinear variables when it is reasonable to do so.

- `perimeter` and `area` are highly correlated to `radius` and `concave_points`. This can be explained by the fact that both `radius` and `concave_points` determine the shape and outer structure of a cell nucleus so they will affect the calculation of `perimeter` and `area` to a certain proportional degree. Thus, we can exclude both `perimeter` and `area` from our list of features.
- `compactness` is highly correlated with `concavity` and `concave_points`. `compactness` is calculated using $\frac{\text{perimeter}^2}{\text{area}} - 1$. As we have discussed above, `concavity` and `concave_points` could affect the calculation of `perimeter` and `area`. As a result, it is also likely that `compactness` can be explained by `concavity` and `concave_points`. Thus, we can exclude `compactness` from our list of features.
- `concavity` and `concave_points` are highly correlated. `concavity` is the severity degree of concave portions, and `concave_points` is the number of concave points on a cell nucleus. Perhaps, they are proportional. Thus, we can exclude one of them from our list of features.

Note that we are only considering correlation values that are above 0.8. In practice, this choice is subjective. We do not want to remove too many predictors, otherwise we might get undesirable results. It is reasonable to sacrifice small amount of precision by including unimportant predictors for the general validity of our regression model. (Bayesian Data Analysis, p. 367)

Frequentist

```r
frequentist.better_reg <-
  glm(Diagnosis ~ radius1 + texture1 + smoothness1 + concavity1 + symmetry1 +
        fractal_dimension1, data = df, family = binomial(link = "logit"))
```

Bayesian

It is well-known that multicollinearity is undesirable in Frequentist approach to regression analysis. Why is it also undesirable in Bayesian approach? Multicollinearity could lead to high posterior variance of the regression coefficients, thus increasing uncertainty. Additionally, it makes the inference task highly sensitive to the model's assumption that $E(y|x, \theta)$ is linear in x. (Bayesian Data Analysis, p. 366) Thus, Bayesian approach could also benefit from the removal of multicollinear explanatory variables.

$$\beta_0 \sim \text{Cauchy}(0, 10) \tag{4}$$

$$\beta_i \overset{\text{iid}}{\sim} \text{Cauchy}(0, 2.5) \qquad i \in \{1, 2, 5, 7, 9, 10\} \tag{5}$$

$$y_n|\beta \sim \text{Bern}(\text{logistic}(\beta_0 + \beta_1 x_{n,1} + \beta_2 x_{n,2} + \beta_5 x_{n,5} + \beta_7 x_{n,7} + \beta_9 x_{n,9} + \beta_{10} x_{n,10})) \qquad n \in \{1, ..., n_{\text{obs}}\} \tag{6}$$

```r
bayesian.better_model <- stan_model("better_logistic.stan")
bayesian.better_reg <- sampling(
  bayesian.better_model,
  data = bayesian.better_reg.dat,
  seed = 123, iter = 1000
)
```

Model comparison

Frequentist

In the Frequentist approach, we often use Akaike information criterion (AIC) to compare different models on the same dataset. Models with lower AIC are better.

```
AIC(frequentist.base_reg)
```

```
## [1] 168.1304
```

```
AIC(frequentist.better_reg)
```

```
## [1] 170.6327
```

We can see that the AIC for our base model is lower than the one for our model with variable exclusion in the Frequentist approach. Even though the difference is not huge, it provides us insight into our analysis. The lower AIC for base model could imply that we might have removed too many predictors or important ones. With this information, we can adjust our variable selection.

Bayesian

WAIC

AIC works really well in the Frequentist approach. However, it is not a preferable method of model comparison in Bayesian approach. Gelman et al. (2013) suggested that we use Watanabe-Akaike information criteria (WAIC) when comparing Bayesian models for the following reasons:

- WAIC can utilize the full posterior distribution whereas AIC cannot because AIC conditions on a point estimate
- WAIC works well with complex models (e.g. hierarchical) where the number of parameters increases with sample size
- WAIC corrects the effective number of parameters to adjust for overfitting

(Gelman et al., 2013)

In a similar fashion as AIC, models with lower WAIC are better.

```
waic(extract_log_lik(bayesian.base_reg))$waic
```

```
## [1] 167.1968
```

```
waic(extract_log_lik(bayesian.better_reg))$waic
```

```
## [1] 171.3876
```

Similar to the result of AIC for the Frequentist models, the base model has slightly lower WAIC than the model with variable exclusion. The implication is also similar to the Frequentist approach.

PSIS-LOO

Gelman et al. (2016) also suggested Pareto-smoothed importance sampling LOO (PSIS-LOO), or LOOIC, over WAIC for the following reasons:

- PSIS-LOO is more robust in cases with weak priors and influential observations
- PSIS-LOO and WAIC are asymptotically equivalent but WAIC may behave differently for small finite samples

(Gelman et al., 2016)

Again, in similar fashion to AIC and WAIC, models with lower LOOIC are better.

```
loo(extract_log_lik(bayesian.base_reg))$looic
```

```
## [1] 167.8696
```

```
loo(extract_log_lik(bayesian.better_reg))$looic
```

```
## [1] 171.6826
```

Model diagnostics

Frequentist

To diagnose a Frequentist logistic regression model, we can check for linearity, outlier, and multicollinearithy assumption. Below we will demonstrate 2 of the 3 diagnostic assumptions.

Outlier assumption

Frequentist logistic regression assumes that there are no outliers in the data. To test for this, we check if there are any data points that have a standardized residual larger than 3.

```
model.data <- augment(frequentist.better_reg) %>% mutate(index = 1:n())
nrow(model.data %>% filter(abs(.std.resid) > 3))
```

```
## [1] 1
```

As we can see, there is one outlier in our data. This violates the outlier assumption. Thus, we need to remove this observation and fit the regression model again.

Multicollinearity/VIF

Frequentist logistic regression assumes that there is no multicolinearity in our data. To test for this, we check for any VIF values larger than 5. Looking at the output in Appendix A.2, we can see there is no VIF value larger than 5. The multicollinearity assumption is satisfied. This makes sense because we have eliminated some variables with high correlation previously.

Bayesian

To diagnose a Bayesian regression model, we can perform calibration analysis via cross-validation, prior predictive checks, MCMC diagnostics, etc. In general, there are more options in the Bayesian approach. Below we will demonstrate 2 diagnoses of the Bayesian workflow.

Prior predictive checks

Prior predictive checks can help assess the fit of a Bayesian model by evaluating potential replications involving new parameters. Looking at figure in Appendix A.3, we can see that the distribution of the average outcome, $\bar{y}$, stays about the same as we increase the number of predictors. Whether this is desirable or not is up to the objective of the modeler.

MCMC diagnostics

MCMC diagnostics can help test for the speed of convergence. The MCMC trace plot in Appendix A.4 does not show any major deviance between chains. The MCMC rank histogram in Appendix A.5 shows that all histograms are approximately uniform. Thus, we can see that this is a case of fast mixing.

Discussion

The goals of the Frequentist and Bayesian approaches in our case are technically the same: fit a logistic regression model to the Breast Cancer Wisconsin dataset for inference and prediction. However, each approach has a different way of performing the regression fit, thus having different output. The Frequentists assume that the population is fixed with an unknown quantity/parameter. Thus, the Frequentists give you point estimates. The Bayesians treat the unknown probabilistiically and think the world can always change. Thus, the Bayesians give you distributions. Because of this difference, it is difficult to directly compare the Frequentist and Bayesian regression models. This is likely the biggest limitation in this project.

Reference

Allen, M. P. (1997). Understanding regression analysis. Plenum Press.

Gelman, A., Carlin, J. B., Stern, H. S., & Dunson, D. B. (2013). Bayesian data analysis. Chapman and Hall/CRC.

Gelman, A., Jakulin, A., Pittau, M. G., & Su, Y.-S. (2008). A Weakly Informative Default Prior Distribution for Logistic and Other Regression Models. The Annals of Applied Statistics, 2(4), 1360–1383. http://www.jstor.org/stable/30245139

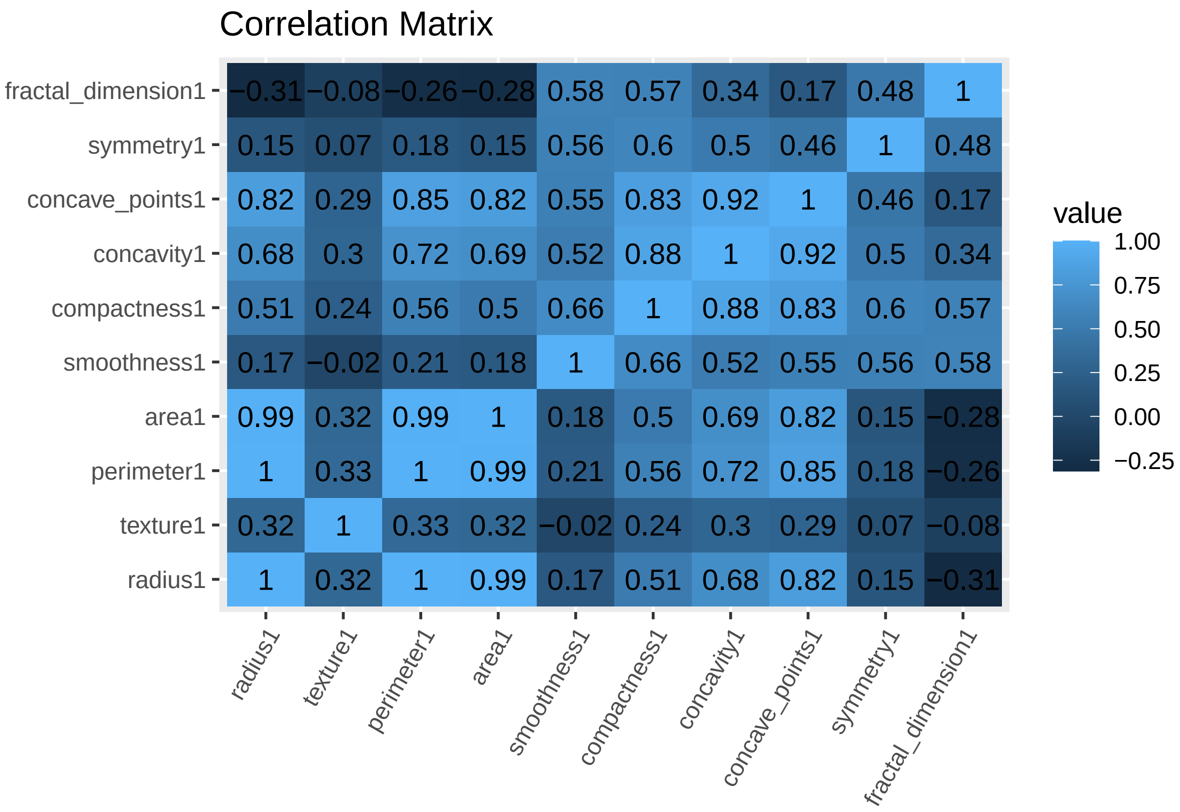
Gelman, A., Hwang, J. & Vehtari, A. Understanding predictive information criteria for Bayesian models. Stat Comput 24, 997–1016 (2013). https://doi.org/10.1007/s11222-013-9416-2

Vehtari, A., Gelman, A. & Gabry, J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat Comput 27, 1413–1432 (2017). https://doi.org/10.1007/s11222-016-9696-4

A. Appendix

A.1 Correlation matrix of explanatory variables

```r
cor_mat <- round(cor(df[, 2:11]), 2)
melted_cor_mat <- melt(cor_mat)
ggplot(data = melted_cor_mat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
  theme(axis.text.x = element_text(angle = 60, vjust = 1, hjust=1)) +
  xlab("") + ylab("") + ggtitle("Correlation Matrix")
```



	radius1	texture1	perimeter1	area1	smoothness1	compactness1	concavity1	concave_points1	symmetry1	fractal_dimension1
fractal_dimension1	−0.31	−0.08	−0.26	−0.28	0.58	0.57	0.34	0.17	0.48	1
symmetry1	0.15	0.07	0.18	0.15	0.56	0.6	0.5	0.46	1	0.48
concave_points1	0.82	0.29	0.85	0.82	0.55	0.83	0.92	1	0.46	0.17
concavity1	0.68	0.3	0.72	0.69	0.52	0.88	1	0.92	0.5	0.34
compactness1	0.51	0.24	0.56	0.5	0.66	1	0.88	0.83	0.6	0.57
smoothness1	0.17	−0.02	0.21	0.18	1	0.66	0.52	0.55	0.56	0.58
area1	0.99	0.32	0.99	1	0.18	0.5	0.69	0.82	0.15	−0.28
perimeter1	1	0.33	1	0.99	0.21	0.56	0.72	0.85	0.18	−0.26
texture1	0.32	1	0.33	0.32	−0.02	0.24	0.3	0.29	0.07	−0.08
radius1	1	0.32	1	0.99	0.17	0.51	0.68	0.82	0.15	−0.31

A.2 VIF for Frequentist model with variable exclusion

```
VIF(frequentist.better_reg)
```

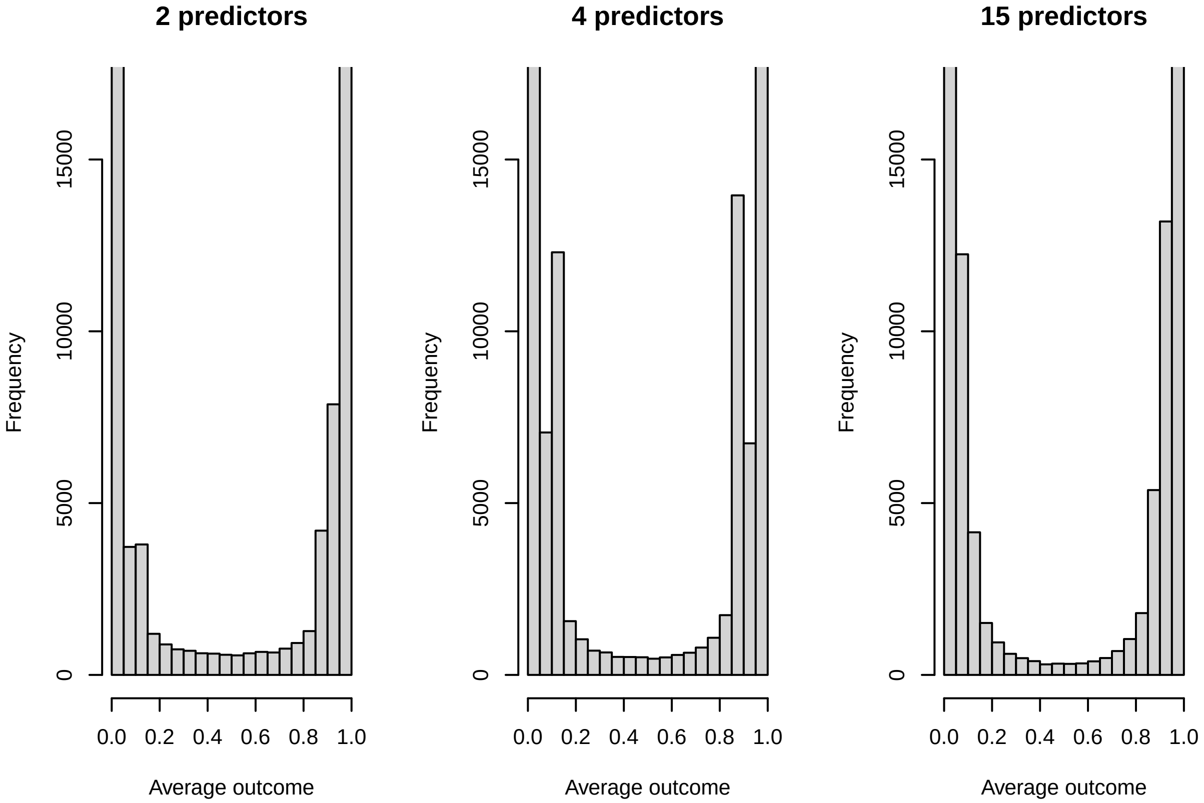
```
##           radius1          texture1        smoothness1          concavity1
##          2.134301          1.686365          2.563388          2.721099
##         symmetry1 fractal_dimension1
##          1.842083          4.759889
```

A.3 Prior predictive checks

```
suppressPackageStartupMessages(library(extraDistr))
```

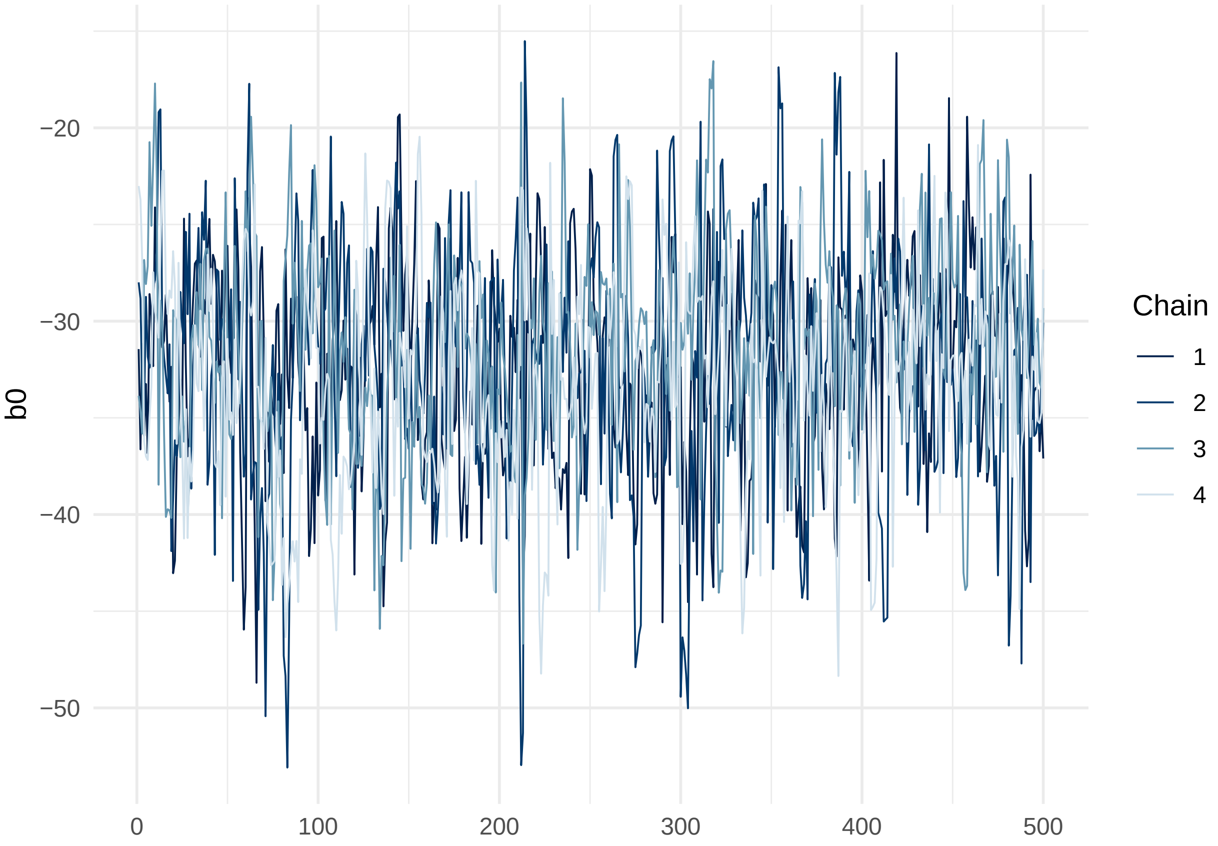
```
logistic_regression <- function(X) {
  b <- append(rcauchy(1, 0, 10), rcauchy(ncol(X) - 1, 0, 2.5))
  p <- plogis(as.vector(X %*% b))
  y <- rbern(nrow(X), p)
  mean(y)
}
```

```
pred_prob = 0.9
n_obs = 100
n_sim = 100000
opar = par(mfrow=c(1,3))
for(n_pred in c(2,4,15)) {
    X = matrix(rbern(n_obs*n_pred, pred_prob), nrow=n_obs)
    simulated_ybars = replicate(n_sim, logistic_regression(X))
    hist(simulated_ybars, breaks=20, ylim=c(0,17000),
         main=paste(n_pred, "predictors"), xlab = "Average outcome")
}
```



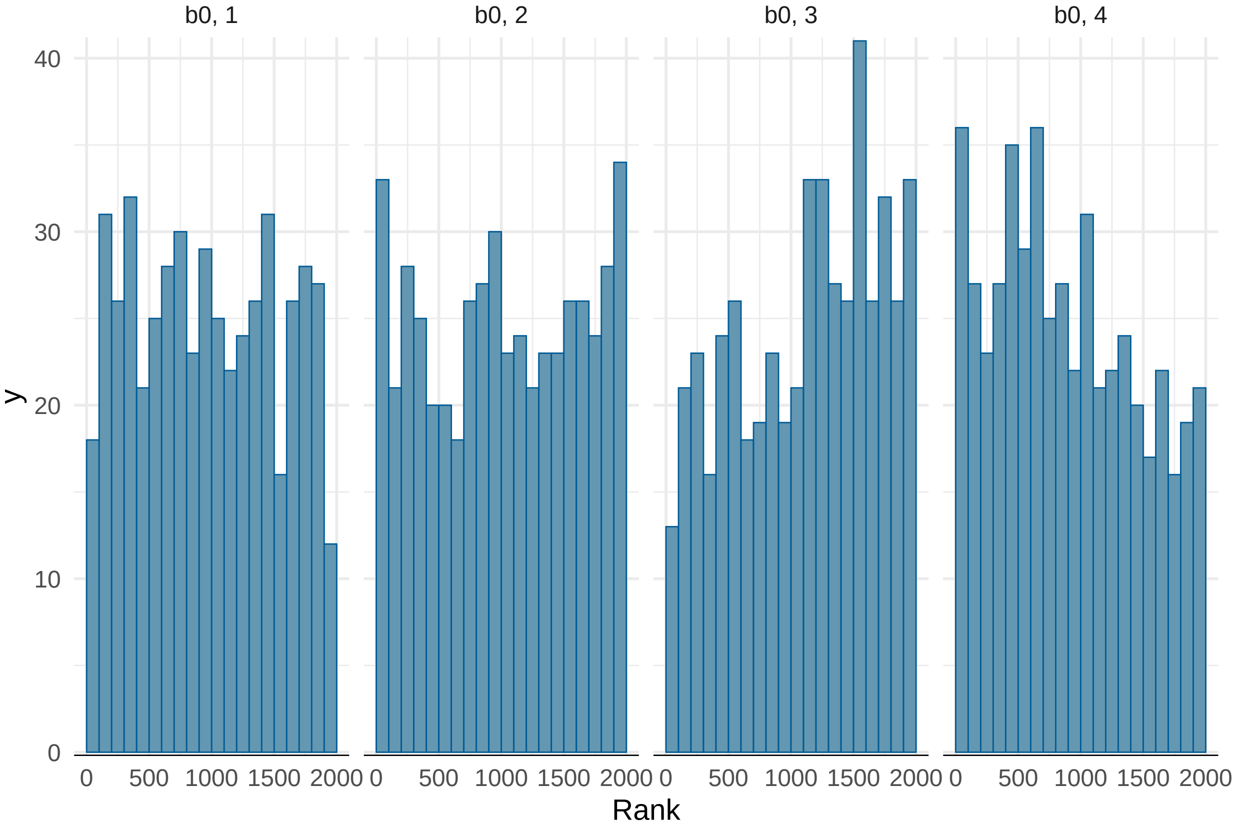
A.4 MCMC trace plot

```
mcmc_trace(bayesian.better_reg, pars = c("b0")) + theme_minimal()
```



A.5 MCMC rank histogram

```r
mcmc_rank_hist(bayesian.better_reg, pars = c("b0")) + theme_minimal()
```



A.6 base_logistic.stan

```
data {
  // train data
  int<lower=0> N;
  vector[N] x1;
  vector[N] x2;
  vector[N] x3;
  vector[N] x4;
  vector[N] x5;
  vector[N] x6;
  vector[N] x7;
  vector[N] x8;
  vector[N] x9;
  vector[N] x10;
  int<lower=0,upper=1> y[N];
}

parameters {
  real b0;
  real b1;
  real b2;
  real b3;
  real b4;
  real b5;
  real b6;
  real b7;
  real b8;
  real b9;
  real b10;
}

model {
  b0 ~ cauchy(0, 10);
  b1 ~ cauchy(0, 2.5);
  b2 ~ cauchy(0, 2.5);
  b3 ~ cauchy(0, 2.5);
  b4 ~ cauchy(0, 2.5);
  b5 ~ cauchy(0, 2.5);
  b6 ~ cauchy(0, 2.5);
  b7 ~ cauchy(0, 2.5);
  b8 ~ cauchy(0, 2.5);
  b9 ~ cauchy(0, 2.5);
  b10 ~ cauchy(0, 2.5);
  y ~ bernoulli_logit(b0 + b1*x1 + b2*x2 + b3*x3 + b4*x4 + b5*x5 + b6*x6 + b7*x7 + b8*x8 + b9*x9 + b10*
}

generated quantities {
  vector[N] log_lik;
  for (i in 1:N) {
    log_lik[i] = bernoulli_logit_lpmf(y[i] | b0 + b1*x1[i] + b2*x2[i] + b3*x3[i] + b4*x4[i] + b5*x5[i]
  }
}
```

A.7 better_logistic.stan

```
data {
  // train data
  int<lower=0> N;
  vector[N] x1;
  vector[N] x2;
  vector[N] x5;
  vector[N] x7;
  vector[N] x9;
  vector[N] x10;
  int<lower=0,upper=1> y[N];
}

parameters {
  real b0;
  real b1;
  real b2;
  real b5;
  real b7;
  real b9;
  real b10;
}

model {
  b0 ~ cauchy(0, 10);
  b1 ~ cauchy(0, 2.5);
  b2 ~ cauchy(0, 2.5);
  b5 ~ cauchy(0, 2.5);
  b7 ~ cauchy(0, 2.5);
  b9 ~ cauchy(0, 2.5);
  b10 ~ cauchy(0, 2.5);
  y ~ bernoulli_logit(b0 + b1*x1 + b2*x2 + b5*x5 + b7*x7 + b9*x9 + b10*x10);
}

generated quantities {
  vector[N] log_lik;
  for (i in 1:N) {
    log_lik[i] = bernoulli_logit_lpmf(y[i] | b0 + b1*x1[i] + b2*x2[i] + b5*x5[i] + b7*x7[i] + b9*x9[i] +
  }
}
```