# STAT 447C: Final Project

Andrew Tran

2024-04-21

## Introduction

## Goals of the project

### Methodology

## Data description

```
df <- read.csv("data/breast_cancer_wisconsin.csv")
df <- as_tibble(df)
```

Our response variable is `Diagnosis`, which is categorical. Currently, it's being labelled 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.

```
df$Diagnosis <- factor(df$Diagnosis)
df$Diagnosis <- as.numeric(df$Diagnosis) - 1
```

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 the means.

```
df <- df[, 1:11]
```

These ten real-valued features are measurements taken and computed for each 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 $-1$ |

We will also divide the dataset into a training dataset of 500 observations and a testing dataset of 69 observations.

```
train_data <- df[1:500,]
test_data <- df[501:569,]
```

## Base model

### Frequentist

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

```
frequentist.base_reg <-
  glm(Diagnosis ~ ., data = train_data, 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 $\beta_0$ 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 approach, it is required that we center and scale our nonbinary variables to have mean 0 and standard deviation 0.5. (Gelman et al., 2008)

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

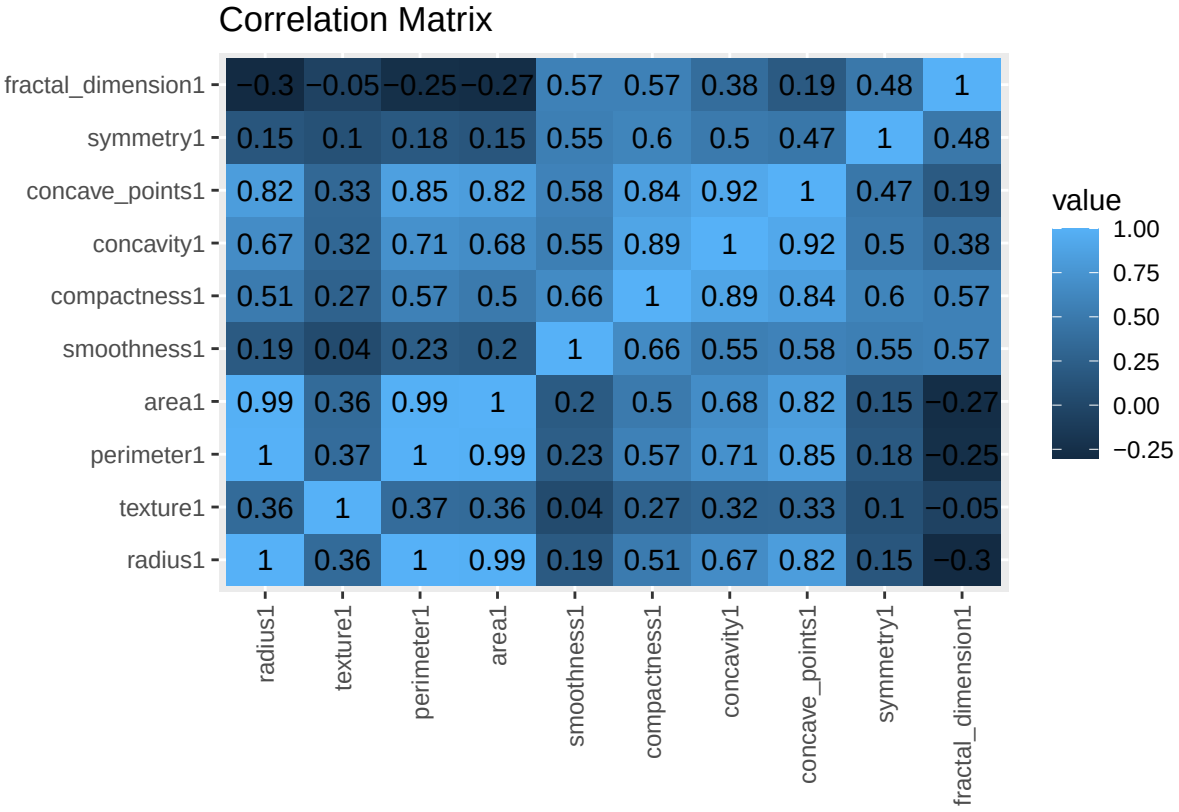
$$\beta_i \overset{\text{iid}}{\sim} \text{Cauchy}(0, 2.5) \qquad \text{for } i \in \{1, ..., 10\} \tag{2}$$

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

```
bayesian.base_reg <- stan(
  seed = 123,
  file = "base_logistic.stan",
  data = bayesian.base_reg.dat,
  iter = 1000
)
```

```
## Trying to compile a simple C file
```

### Model with the exclusion of high correlated explanatory variables




Highly correlated variables:

- `radius` and `perimeter`: 1
- `radius` and `area`: 0.99
- `radius` and `concave_points`: 0.82
- `perimeter` and `area`: 0.99
- `perimeter` and `concavity`: 0.85
- `perimeter` and `concave_points`: 0.71
- `area` and `concave_points`: 0.82
- `compactness` and `concavity`: 0.89
- `compactness` and `concave_points`: 0.84
- `concavity` and `concave_points`: 0.92

The possible loss of precision when including unimportant predictors is usually viewed as a relatively small price to pay for the general validity of predictions and inferences about estimands of interest. (Bayesian Data Analysis, p. 367)

#### Frequentist

```
frequentist.better_reg <-
  glm(Diagnosis ~ radius1 + texture1 + smoothness1 + concavity1 + symmetry1 + fractal_dimension1, data =
```

#### Bayesian

why colllinearity is bad in Bayesian The near-collinearity of the data means that the posterior variance of the regression coefficients would be high in this hypothetical case. Another problem in addition to increased uncertainty conditional on the regression model is that in practice the inferences would be highly sensitive to

the model's assumption that $E(y|x,\theta)$ is linear in x. (Bayesian Data Analysis, p. 366)

$$\beta_0 \sim \text{Cauchy}(0, 10) \tag{4}$$
$$\beta_1 \sim \text{Cauchy}(0, 2.5) \tag{5}$$
$$\beta_2 \sim \text{Cauchy}(0, 2.5) \tag{6}$$
$$\beta_5 \sim \text{Cauchy}(0, 2.5) \tag{7}$$
$$\beta_7 \sim \text{Cauchy}(0, 2.5) \tag{8}$$
$$\beta_9 \sim \text{Cauchy}(0, 2.5) \tag{9}$$
$$\beta_10 \sim \text{Cauchy}(0, 2.5) \tag{10}$$
$$y_n|\beta \sim \text{Bern}(\text{logistic}(\beta_0 + \beta_1 x_{n,1} + \beta_2 x_{n,2} + \beta_5 x_{n,5} + \beta_7 x_{n,7} + \beta_9 x_{n,9} + \beta_{10} x_{n,10})) \quad n \in \{1, ..., n_{\text{obs}}\} \tag{11}$$

```
bayesian.better_reg <- stan(
  seed = 123,
  file = "better_logistic.stan",
  data = bayesian.better_reg.dat,
  iter = 1000
)
```

```
## Trying to compile a simple C file
```

## Model comparison

### Frequentist

### AIC \

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

```
## [1] 148.8762
```

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

```
## [1] 148.3286
```

### ROC \

### Bayesian

### WAIC \

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

```
## [1] 149.4277
```

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

```
## [1] 148.2578
```

### LOOIC \

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

```
## [1] 149.7657
```

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

```
## [1] 148.4246
```

## Model diagnostics

### Frequentist

Influential

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

```
## # A tibble: 1 x 14
##   Diagnosis radius1 texture1 smoothness1 concavity1 symmetry1 fractal_dimension1
##       <dbl>   <dbl>    <dbl>       <dbl>      <dbl>     <dbl>              <dbl>
## 1         1    11.8     18.1      0.0997     0.0268     0.162             0.0629
## # i 7 more variables: .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>,
## #   .cooksd <dbl>, .std.resid <dbl>, index <int>
```

Multicollinearity/VIF

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

```
##            radius1           texture1        smoothness1         concavity1
##           2.313972           1.772572           2.321514           3.045440
##          symmetry1 fractal_dimension1
##           1.764508           4.918610
```

### Bayesian

## Confidence vs Credible

## Direct comparison between Frequentist and Bayesian

## Appendix

```
summary(frequentist.base_reg)
```

```
##
## Call:
## glm(formula = Diagnosis ~ ., family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.00429  -0.14481  -0.03502   0.00671   2.86097
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -12.988495  13.841757  -0.938  0.34806
## radius1             -2.378525   3.984187  -0.597  0.55051
## texture1             0.434221   0.073155   5.936 2.93e-09 ***
## perimeter1           0.004415   0.554275   0.008  0.99365
## area1                0.039313   0.017889   2.198  0.02798 *
## smoothness1         96.827294  36.988329   2.618  0.00885 **
## compactness1        -1.754591  22.250910  -0.079  0.93715
## concavity1           6.176023   9.051688   0.682  0.49505
## concave_points1     53.397630  31.147958   1.714  0.08647 .
## symmetry1           14.180834  11.321459   1.253  0.21036
## fractal_dimension1 -34.328041  90.545340  -0.379  0.70459
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 668.75  on 499  degrees of freedom
## Residual deviance: 126.88  on 489  degrees of freedom
## AIC: 148.88
##
## Number of Fisher Scoring iterations: 9
```

```
summary(frequentist.better_reg)
```

```
##
## Call:
## glm(formula = Diagnosis ~ radius1 + texture1 + smoothness1 +
##     concavity1 + symmetry1 + fractal_dimension1, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2392  -0.1297  -0.0271   0.0266   3.2057
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -38.54804    7.07841  -5.446 5.16e-08 ***
## radius1              1.20246    0.20007   6.010 1.85e-09 ***
## texture1             0.43687    0.07197   6.070 1.28e-09 ***
## smoothness1        132.39617   27.03813   4.897 9.75e-07 ***
## concavity1          21.59574    6.51814   3.313 0.000922 ***
## symmetry1           13.88770   11.78453   1.178 0.238610
## fractal_dimension1 -74.37037   67.89893  -1.095 0.273381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 668.75  on 499  degrees of freedom
## Residual deviance: 134.33  on 493  degrees of freedom
## AIC: 148.33
##
## Number of Fisher Scoring iterations: 8
```