Lab 13: Logistic Regression

Name here

For this question, we will use a historical data set with NCAA tournament results. However, we will now look at predicting a binary outcome, win or loss by the higher seeded team.

```
library(rjags)
library(knitr)
library(tidyverse)
ncaa <- read_csv('https://raw.githubusercontent.com/stat456/labs/main/Lab12_data.csv') %>%
  filter(Seed.Diff != 0) %>%
  mutate(Seed.Diff = -1 * Seed.Diff,
         SAG.Diff = -1 * SAG.Diff,
         upset = as.numeric(Result == 'Loss'))

ncaa_train <- ncaa %>% filter(Season < 2020)
ncaa_test <- ncaa %>% filter(Season >= 2020)
```

1. (4 points) Create two figures to explore the impact of `Seed.Diff` and `SAG.Diff` on `upset.` Add a smoother line to approximate the relationship.

2. (4 points) Write a short caption to accompany each figure.

3. (4 points) Deviance Information Criteria (DIC) is a Bayesian analog to AIC. Consider the two model below, which do you prefer and why?

```
# Model String
logistic = "model {
  for ( i in 1:N ) {
    y[i] ~ dbern(p[i])
    logit(p[i]) = beta0 + beta1 * x[i]
  }
  beta0 ~ dnorm(M0, 1 / S0^2)
  beta1 ~ dnorm(M1, 1 / S1^2)
} "
writeLines( logistic, con='logistic.txt')
```

```
# Runs JAGS Model: Seeds
seeds_model <- jags.model( file = "logistic.txt",
                           data =  list(y = ncaa_train$upset,
                                        x = ncaa_train$Seed.Diff,
                                        N = nrow(ncaa_train),
                                        M0 = 0,
                                        S0 = .001,
                                        M1 = 0,
```

```r
                                        S1 = 3),
                         n.chains = 2, n.adapt = 5000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1058
##    Unobserved stochastic nodes: 2
##    Total graph size: 2171
##
## Initializing model
```

```r
update(seeds_model, n.iter = 5000)

seeds_coda <- coda.samples(seeds_model,
                          variable.names = c('beta0','beta1'),
                          n.iter = 5000)

summary(seeds_coda)
```

```
##
## Iterations = 10001:15000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean        SD  Naive SE Time-series SE
## beta0 -7.903e-06 0.0009847 9.847e-06      1.226e-05
## beta1 -1.677e-01 0.0112482 1.125e-04      1.413e-04
##
## 2. Quantiles for each variable:
##
##             2.5%        25%        50%        75%     97.5%
## beta0 -0.001962 -0.0006564 -3.836e-06  0.0006586  0.001908
## beta1 -0.190126 -0.1753042 -1.674e-01 -0.1600796 -0.146346
```

```r
summary(glm(upset ~ Seed.Diff -1, family = binomial, data = ncaa_train))
```

```
##
## Call:
## glm(formula = upset ~ Seed.Diff - 1, family = binomial, data = ncaa_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1073  -0.8487  -0.6334   1.2492   2.2742
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## Seed.Diff -0.16717    0.01136  -14.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1466.7  on 1058  degrees of freedom
## Residual deviance: 1162.7  on 1057  degrees of freedom
## AIC: 1164.7
##
## Number of Fisher Scoring iterations: 4
```

```
dic_seed <- dic.samples(seeds_model, 5000)
dic_seed
```

```
## Mean deviance:  1164
## penalty 0.9763
## Penalized deviance: 1165
```

```
# Runs JAGS Model: Sagarin
sag_model <- jags.model( file = "logistic.txt",
                           data =  list(y = ncaa_train$upset,
                                        x = ncaa_train$SAG.Diff,
                                        N = nrow(ncaa_train),
                                        M0 = 0,
                                        S0 = .001,
                                        M1 = 0,
                                        S1 = 3),
                         n.chains = 2, n.adapt = 5000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1058
##    Unobserved stochastic nodes: 2
##    Total graph size: 2771
##
## Initializing model
```

```
update(sag_model, n.iter = 5000)

sag_coda <- coda.samples(sag_model,
                          variable.names = c('beta0','beta1'),
                          n.iter = 5000)

summary(sag_coda)
```

```
##
## Iterations = 10001:15000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean       SD  Naive SE Time-series SE
## beta0 -0.0000158 0.000999 9.990e-06      1.276e-05
```

```
## beta1 -0.0250309 0.001989 1.989e-05      2.470e-05
##
## 2. Quantiles for each variable:
##
##             2.5%        25%        50%        75%    97.5%
## beta0 -0.001999 -0.0006843 -1.611e-05  0.0006413  0.00193
## beta1 -0.028975 -0.0263730 -2.496e-02 -0.0236713 -0.02127
```

```
dic_sag <- dic.samples(sag_model, 5000)
dic_sag
```

```
## Mean deviance:  1146
## penalty 1.012
## Penalized deviance: 1147
```

```
diffdic(dic_seed, dic_sag)
```

```
## Difference: 17.51113
## Sample standard error: 15.20846
```

4. (4 points) Fit the best possible model using DIC as your criteria. You may want to consider interactions and/or non-linearity terms.

5. (4 points) Write out the formal model you've selected in part 4, including all priors.

6. (4 points) Interpret your parameters in the model and summarize your findings. Assume you are telling your parents how statistics can be useful for filling out an NCAA bracket.

7. (4 points) Construct a posterior predictive distribution to calculate the winning probability of the following games, Note your model is likely set to predict the point spread for the higher seed:

1. Montana State (Seed: 14, Sagarin: 133) vs. Kansas State (Seed: 3, Sagarin: 18)
2. Purdue (Seed: 1, Sagarin: 9) vs. Farleigh Dickinson (Seed: 16, Sagarin: 310)
3. Arizona (Seed: 2, Sagarin: 10) vs. Princeton (Seed 15:, Sagarin: 118)
4. Connecticut (Seed: 4, Sagarin: 4) vs. Iona (Seed: 13, Sagarin: 86)

Hint you might need to use `ilogit()` in JAGS.

8. (1 EC point) Rather than using DIC as a model selection criteria, compare your model's ability to predict outcomes during the 2021 and 2022 tournaments (`ncca_test`).

Consider using classification error (win/loss) and Brier score (`brier.scoer()` in the `iterativeBMA` package.)