Lab 12: Regression

Name here

For this question, we will use a historical data set with NCAA tournament results.

```
library(rjags)
```

```
## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs
```

```
library(knitr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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)
```

```
## Rows: 1248 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Result
## dbl (6): Season, Score.Diff, Seed.Diff, Higher.Seed, SAG.Diff, Higher.SAG
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

1. (4 points) Create two figures to explore the impact of `Seed.Diff` and `SAG.Diff` on `Score.Diff`. 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
modelString = "model {
  for ( i in 1:N ) {
    y[i] ~ dnorm(beta0 + beta1 * x[i], 1/sigma^2) # sampling model
  }
  beta0 ~ dnorm(M0,1/S0^2)
  beta1 ~ dnorm(M1, 1 / S1^2)
  sigma ~ dunif(0,C)
} "
writeLines( modelString, con='NORMmodel.txt')
```

```
# Runs JAGS Model: Seeds
seeds_model <- jags.model( file = "NORMmodel.txt",
                           data =  list(y = ncaa$Score.Diff,
                                        x = ncaa$Seed.Diff,
                                        N = nrow(ncaa),
                                        M0 = 0,
                                        S0 = .001,
                                        M1 = 1,
                                        S1 = 3,
                                        C = 100),
                         n.chains = 2, n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1182
##    Unobserved stochastic nodes: 3
##    Total graph size: 2410
##
## Initializing model
```

```
update(seeds_model, n.iter = 1000)

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

summary(seeds_coda)
```

```
##
## Iterations = 2001:7000
## 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 -1.259e-05 0.0009973 9.973e-06      1.047e-05
## beta1  1.123e+00 0.0449488 4.495e-04      4.495e-04
## sigma  1.169e+01 0.2409318 2.409e-03      3.067e-03
##
```

```
## 2. Quantiles for each variable:
##
##             2.5%        25%        50%       75%     97.5%
## beta0 -0.001952 -0.0006996 -1.995e-05 6.612e-04  0.001959
## beta1  1.034731  1.0929939  1.124e+00 1.154e+00  1.212259
## sigma 11.232163 11.5268095  1.169e+01 1.185e+01 12.176713
```

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

```
## Mean deviance:  9166
## penalty 1.985
## Penalized deviance: 9168
```

```
# Runs JAGS Model: Sagarin
sag_model <- jags.model( file = "NORMmodel.txt",
                           data =  list(y = ncaa$Score.Diff,
                                        x = ncaa$SAG.Diff,
                                        N = nrow(ncaa),
                                        M0 = 0,
                                        S0 = .001,
                                        M1 = .5,
                                        S1 = 3,
                                        C = 100),
                         n.chains = 2, n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1182
##    Unobserved stochastic nodes: 3
##    Total graph size: 2826
##
## Initializing model
```

```
update(sag_model, n.iter = 1000)

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

summary(sag_coda)
```

```
##
## Iterations = 2001:7000
## 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 8.624e-06 0.001009 1.009e-05      1.008e-05
## beta1 1.266e-01 0.004589 4.589e-05      4.529e-05
## sigma 1.134e+01 0.234948 2.349e-03      3.030e-03
##
## 2. Quantiles for each variable:
##
##            2.5%       25%       50%       75%     97.5%
## beta0 -0.001992 -0.000659 1.629e-05 6.917e-04  0.001947
## beta1  0.117552  0.123533 1.267e-01 1.297e-01  0.135638
## sigma 10.894708 11.179136 1.134e+01 1.150e+01 11.804859
```

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

```
## Mean deviance:  9093
## penalty 2.008
## Penalized deviance: 9095
```

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

```
## Difference: 72.58014
## Sample standard error: 23.96074
```

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)