Week 12 Activity: Regression

On Thursday we will use historical data to predict NCAA basketball games. For this activity we will generate (simulate) synthetic data.

1.

Simulate synthetic data that represents historical NCAA games. In particular, let's consider the model

$$pointdiff = \beta_0 + \beta_1 x_{seeddiff} + \epsilon; \epsilon \sim N(0, \sigma^2)$$

where:

- $\beta_0 = 0$
- $\beta_1 = 2$
- $\sigma = 12$

```
set.seed(04032023)
num_games <- 960 # approximately 15 years
seeddiff <- rep(0:15, each = 60)


beta1 <- 2
sigma <- 12

pointdiff <- rnorm(num_games, mean = beta1 * seeddiff, sd = sigma)

data_out <- tibble(seeddiff = seeddiff,
                   pointdiff = pointdiff)
```

2.

Interpret the three coefficients in the model specified in part 1.

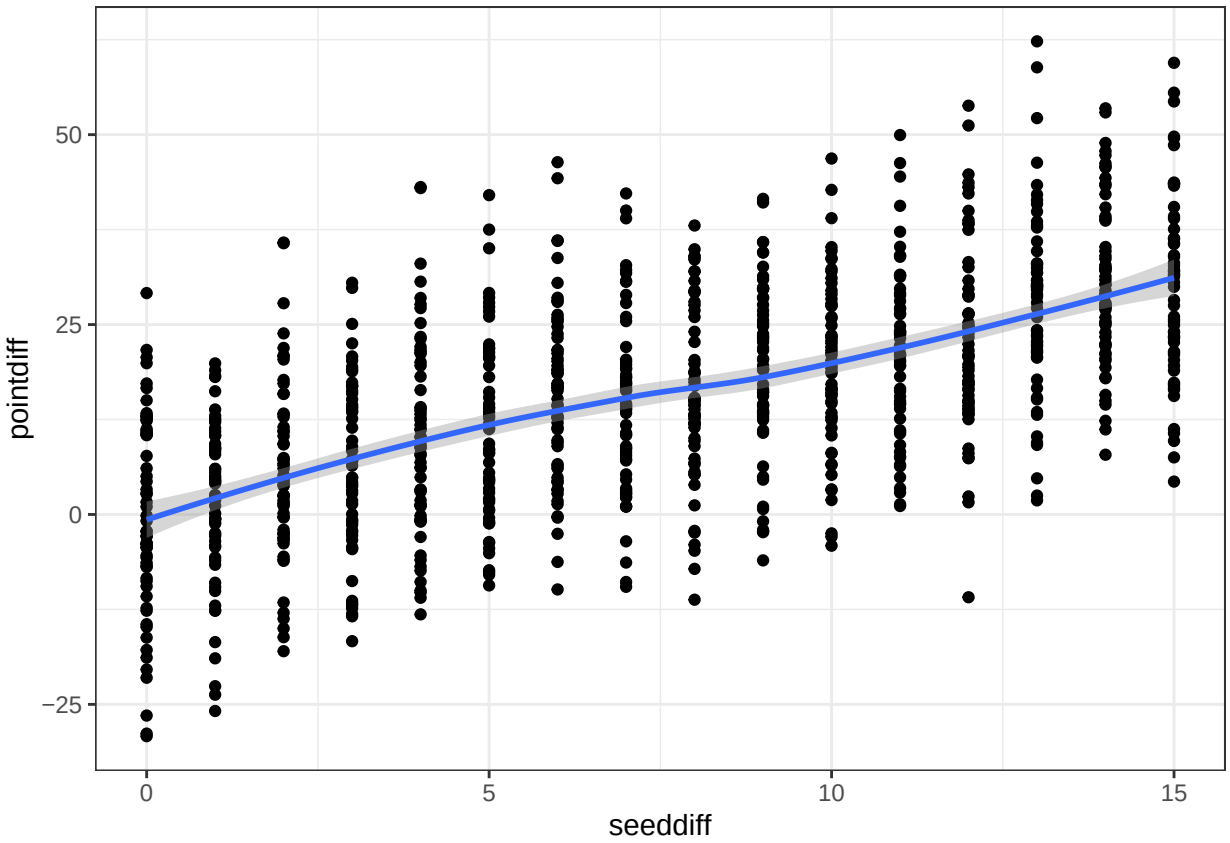
- β_0 = expected point differential for teams with the same seed
- β_1 = multiplicative coefficient for point differential as a function of seed difference
- $\sigma = 12$ = standard deviation in point differential. Most games would be $\pm$ 2 standard deviations (or 24 points) from the mean.

3.

Create a visualization of your point differential versus seed differential.

```
data_out %>% ggplot(aes(y = pointdiff, x = seeddiff), method = 'loess', formula = 'y ~ x') + geom_point
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



4.

Specify prior distributions on your parameters.

- $\beta_0 \sim \text{Normal}(0,.001^2)$
- $\beta_1 \sim \text{Normal}(1,\ 3^2)$
- $\sigma \sim \text{Unif}(0,\ 100)$

5.

Write JAGS code to fit this model. Output the results.

```
#Prior parameters
M0 <- 0
S0 <- .001
M1 <- 1
S1 <- 3
C <- 100

# Store data
dataList = list(y = data_out$pointdiff,
                x = data_out$seeddiff,
                N = nrow(data_out),
                M0 = M0, S0 = S0,
                M1 = M1, S1 = S1, C = C)

# 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
jags.norm <- jags.model( file = "NORMmodel.txt", data = dataList,
                         n.chains = 2, n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 960
##    Unobserved stochastic nodes: 3
##    Total graph size: 1970
##
## Initializing model
```

```
update(jags.norm, n.iter = 1000)

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

summary(coda.norm)
```

```
##
## 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 1.102e-05 0.001005 1.005e-05      9.701e-06
## beta1 2.067e+00 0.043215 4.322e-04      4.456e-04
## sigma 1.185e+01 0.269208 2.692e-03      3.375e-03
##
## 2. Quantiles for each variable:
##
##            2.5%        25%       50%       75%     97.5%
## beta0 -0.001926 -0.0006651 8.190e-07 6.945e-04  0.001995
## beta1  1.982494  2.0372202 2.067e+00 2.096e+00  2.150316
## sigma 11.345158 11.6641006 1.185e+01 1.203e+01 12.399536
```

6.

Summarize your findings. Write a paragraph so that a college basketball coach could understand.

**For each unit difference in seed the expected point differential would increase by about 2 points. For teams of the same seed, the expected point differential would be 0 points. As you know with the NCAA tournament there is the potential for upsets. This is illustrated by the σ term in our model. For a given point spread, it would not be unlikely to see differences of about 20 points, either way, from the expected point spread.

7.

Now we will construct a posterior predictive distribution. This will enable us answers questions like "What is the probability that a one seed is upset by a 16 seed (15 point difference in seeds.) In particular, construct a posterior predictive distribution, conditional on the following scenarios:

- Seed difference = 15
- Seed difference = 7 (most commonly 1 vs. 8)
- Seed difference = 1

Use those distributions to compute the probability of an upset occuring.

```
#Prior parameters
M0 <- 0
S0 <- .001
M1 <- 1
S1 <- 3
C <- 100

# Store data
dataList = list(y = data_out$pointdiff,
                x = data_out$seeddiff,
                N = nrow(data_out),
                M0 = M0, S0 = S0,
                M1 = M1, S1 = S1, C = C)

# Model String
modelString_pp = "model {
  for ( i in 1:N ) {
    y[i] ~ dnorm(beta0 + beta1 * x[i], 1/sigma^2) # sampling model
    pp[i] ~ dnorm(beta0 + beta1 * x[i], 1/sigma^2)
  }
  pp1 ~ dnorm(beta0 + beta1, 1/sigma^2)
  pp7 ~ dnorm(beta0 + beta1 * 7, 1/sigma^2)
  pp15 ~ dnorm(beta0 + beta1 * 15, 1/sigma^2)
  beta0 ~ dnorm(M0,1/S0^2)
  beta1 ~ dnorm(M1, 1 / S1^2)
  sigma ~ dunif(0,C)
} "
writeLines( modelString_pp, con='NORMmodel.txt')


# Runs JAGS Model
jags.norm <- jags.model( file = "NORMmodel.txt", data = dataList,
                         n.chains = 2, n.adapt = 1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
```

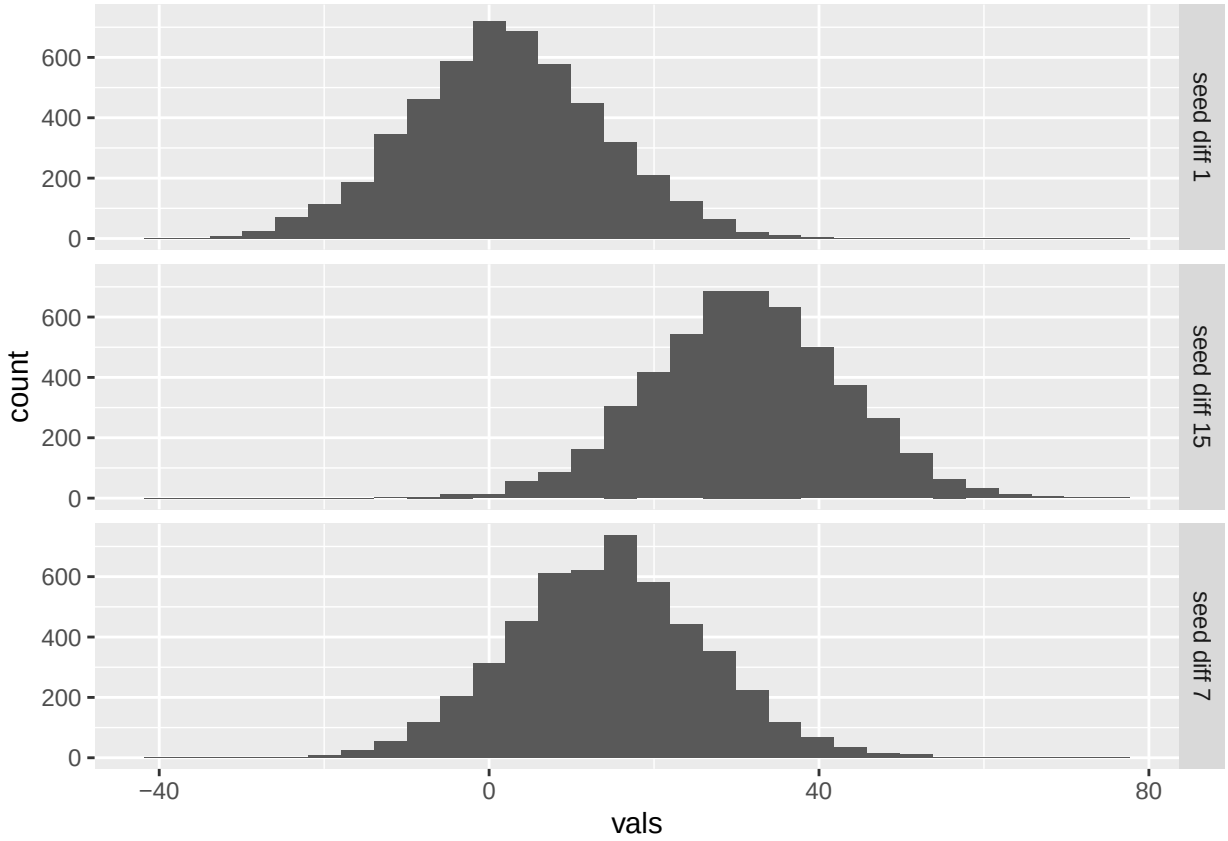
```
##    Observed stochastic nodes: 960
##    Unobserved stochastic nodes: 966
##    Total graph size: 2936
##
## Initializing model
```

```
update(jags.norm, n.iter = 1000)

coda.norm <- coda.samples( jags.norm, variable.names = c('beta0','beta1', 'sigma','pp1','pp7','pp15','p

pp_out <- tibble(vals = c(coda.norm[[1]][,'pp1'],
                          coda.norm[[1]][,'pp7'],
                          coda.norm[[1]][,'pp15']),
                 type = rep(c('seed diff 1',
                              'seed diff 7',
                              'seed diff 15'),
                            each = 5000))

pp_out %>% group_by(type) %>%
  summarize(`upset prob` = mean(vals < 0)) %>%
  kable(digits = 3)
```

type	upset prob
seed diff 1	0.430
seed diff 15	0.005
seed diff 7	0.110

```
pp_out %>% ggplot(aes(x = vals)) +
  geom_histogram() + facet_grid(type~.)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


seed diff 1
seed diff 15
seed diff 7
count
vals
0
200
400
600
−40
0
40
80