Bayesian Modeling with Application to the PGA Tour

Nick Bigelow

Introduction

The primary purpose of the following document is to discuss and describe various Bayesian model methodologies with application to player specific scores in a PGA Tour round. We will explore a general hierarchical Bayesian model as well as two mixed effects Poisson regression models written in Stan and implemented in R. We will briefly touch on the data, problem statement, limitations, and alternative methodologies.

Golf scores in a round of a PGA tournament can be thought of as the number of strokes it took for a player to finish the round, which can be thought of as count data and modeled using a Poisson distribution.

Data

Data was obtained using the datagolf API. There are various granularities which can be used in this dataset. Two which come to mind are round level data and tournament level data.

Round level data gives us player specific metrics such as rolling averages of strokes gained approach, strokes gained tee-to-green, strokes gained putting, etc.

Tournament level data gives us player tournament metrics such as whether or not they were cut, the course where the tournament was played, the name of the tournament, etc.

We will not dive too much further into data preparation as the focus is modeling, however there were variables derived to quantify players recent performance. Including, rolling averages of player specific greens in regulation, strokes gained putting, strokes gained driving, etc.

Limitations

A thorough variable selection was not performed, variables were chosen using the developers discretion. Thorough back testing has not been performed.

More complex hierarchical relationships such as interactions between players and courses, or global score across courses was not explored.

May of 2024 was the latest available month for data for this project.

Model Specification

Mixed Effects Poisson Regression: Player Specific Intercepts

The idea is to model score as a poisson distributed variable, with each golfer has their own intercept. We are also choosing to model a global fixed effect that is based on a 6 round rolling average of a golfers strokes gained tee-to-green, and strokes gained putting. This fixed effect is used as a way to quantify a receny bias. We expecet the coefficients to be negative, as a player is on average gaining strokes on the field over their last 6 rounds we expect them to shoot lower scores. The functional form of our model is shown next:

$$\text{Likelihood Function: } y_{i,g} \sim \text{poisson}(exp(\mu_g + x_{i,p} * \beta_{i,p}))$$

$$\text{Player level prior: } \mu_g \sim \text{normal}(\mu_{0g}, \sigma_{0g})$$

$$\text{Fixed coefficient prior: } \beta_p \sim \text{normal(-1,1)}$$

The priors for μ_g are selected based on historic data respective of each golfer. μ_{0g} is the average score for golfer g from 2023-01-01 to 2024-05-01 and σ_{0g} is the corresponding standard deviation of score for golfer g during that time period.

The priors for β_p were chosen based on an intuition that coefficients should be negative. Our predictor variables, 6 round rolling averages of strokes gained putting and strokes gained tee-to-green, both are expected to have a negative coefficient.

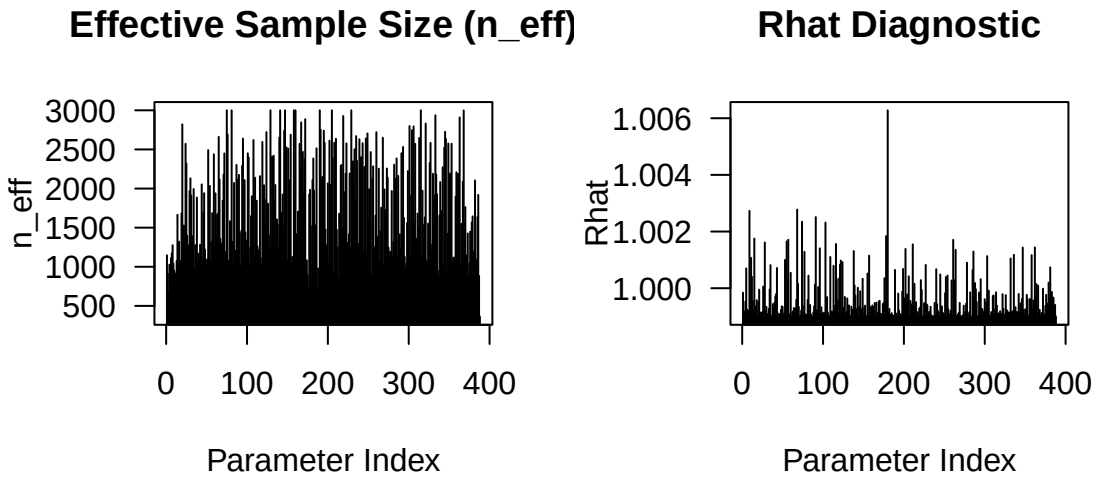
The stan code for this model is shown below and some of the R code used to fit this stan model is also shown below. The full R code can be found in the github repository under _____.

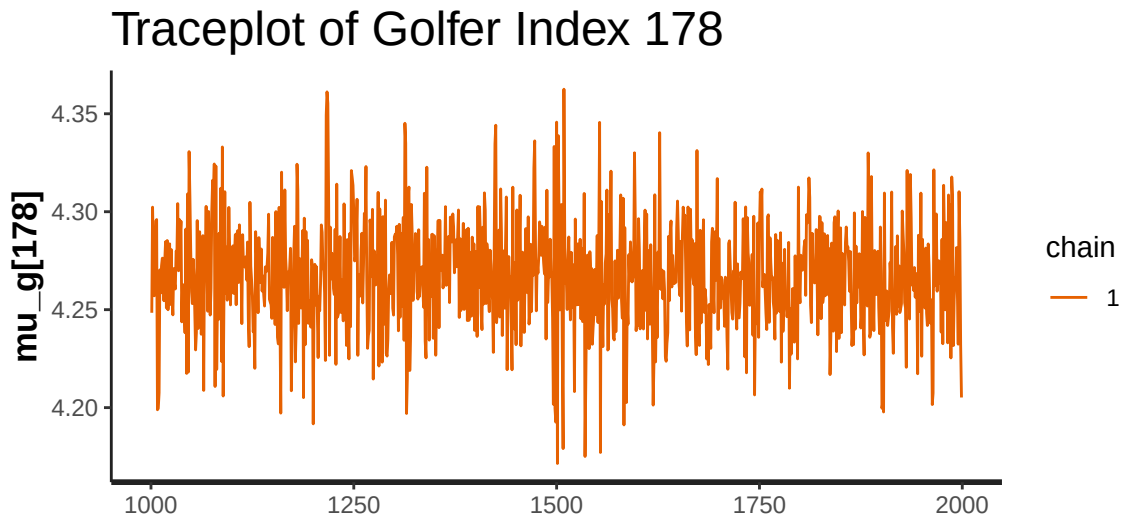
```
data {
  int<lower=0> N; // number of observations
  int<lower=1> G; // number of unique players
  int<lower=0> y[N]; // response variable, scores
  int<lower=1> P; // number of predictors
  matrix[N, P] x; // predictor matrix
  int<lower=1,upper=G> g[N]; // player assignment

  vector<lower=0>[G] mu0; // prior mean for each player's intercept
  vector<lower=0>[G] sigma0; // prior standard deviation for each player's intercept
}

parameters {
  vector[P] beta; // fixed-effect coefficients
  vector<lower=0>[G] mu_g; // player specific intercepts
}

model {

  // Coefficient priors
  beta ~ normal(-1, 1);

  // Player specific priors
  for (j in 1:G){
    mu_g[j] ~ normal(mu0[j],sigma0[j]);
  }

  // Likelihood
  y ~ poisson_log(mu_g[g] + x * beta);
}
```

```
model <- stan_model("fit_playerL.stan")
data <- list(N = nrow(train_sample),
             G = length(unique(train_sample$dg_id_mapped)),
             y = train_sample$score,
             P = 2,
             x = train_sample %>% select(sg_putt_mean_6,
                                         sg_t2g_mean_6),
```

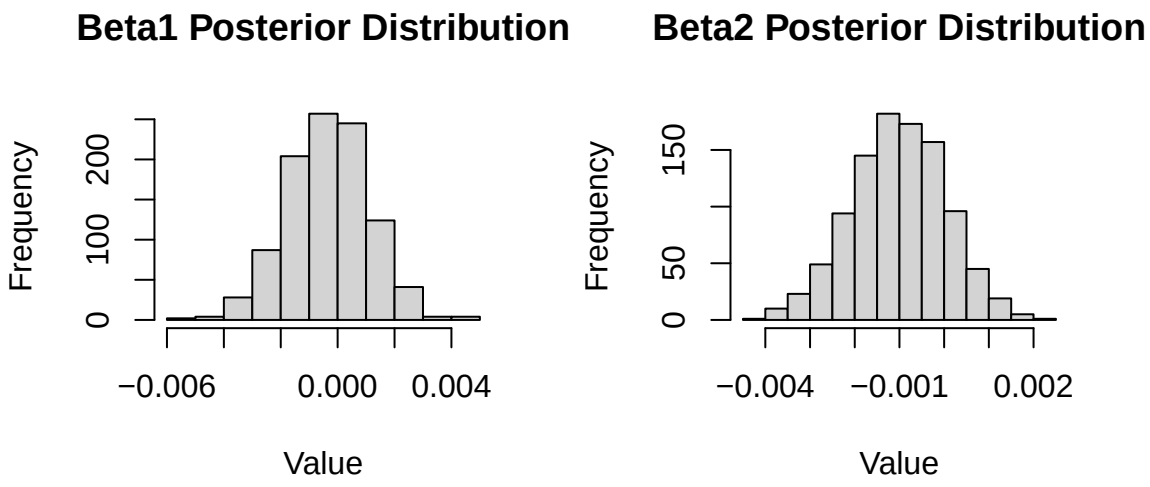
```
              g = train_sample$dg_id_mapped,
              mu0 = log(mu0_prior$meanscore),
              sigma0 = log(sigma0_prior$sdscore))

fit <- sampling(model,data,chains=1,iter=2000)
```

In the data list, mu0 and sigma0 are input in log form. This is because poisson regression is specified using the log link. Next we will look at some model diagnostics including the number of effective samples, Rhats, and a trace plots of the parameter with the highest Rhat to see how well our estimation has converged.





The parameter estimates appear to have converged well with Rhats generally around 1 and the highest Rhat still showing signs of convergence in its trace plot. Next we will take a look regression coefficient estimates:



Both β_1 and β_2 have posterior distributions whose 95% credible intervals overlap 0 and we will not use these elements as predictors. This was somewhat expected as a formal variable selection procedure was not performed.

The goal of these predictor variables was to add a performance recency bias to our model. Instead, we will use a period of 2022-01-01 to 2024-05-01 as a model calibration window. We can then use a prior μ_{0g} and σ_{0g} which are the player level mean and standard deviation of a players score from 2024-01-01 to present.

Using a prior based on recent data and a likelihood based on a longer period of data will give us this desired recency bias effect. In reality, there will be players who haven't played in 2024 and we will have to use data from earlier for this specific players. For this exercise we will just drop those players from our sample. This then brings us to developing a hierarchical Bayesian model.

Hierarchical Bayesian: Player Specific Intercepts

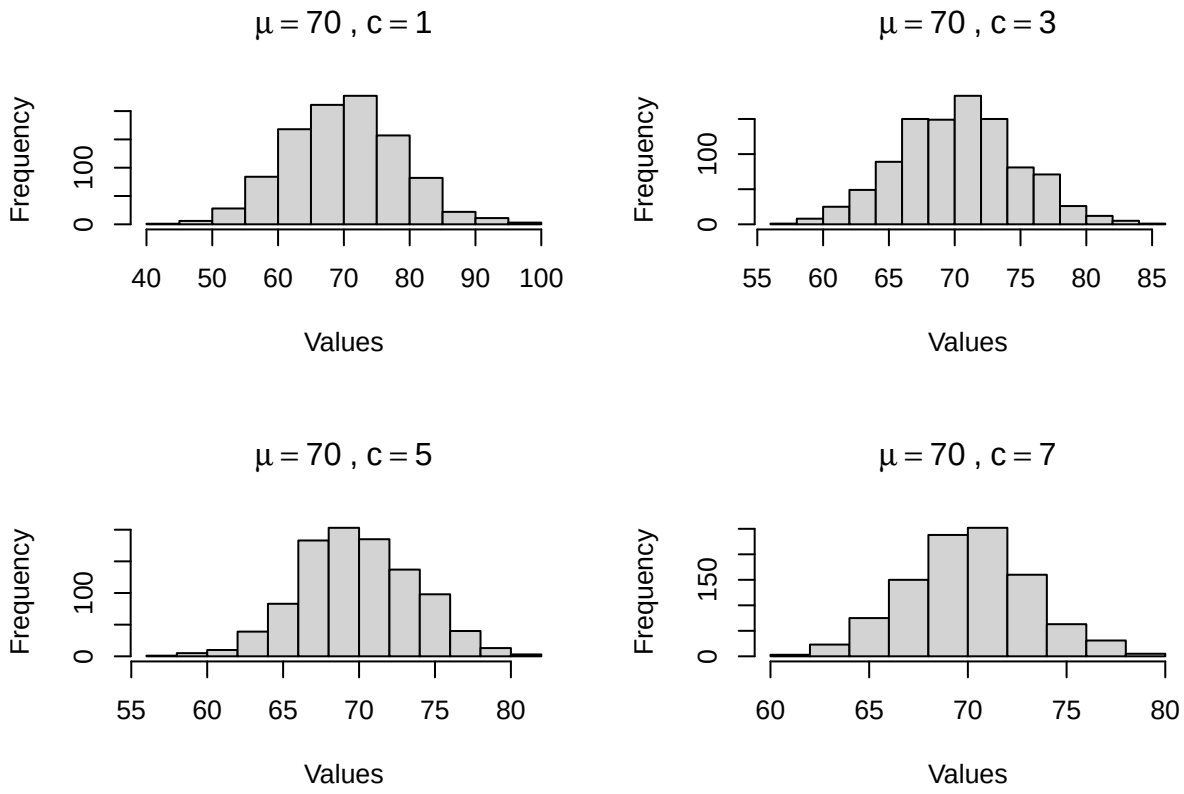
We will tweak our model specification slightly in light of our predictor variables being insignificant.

$$\text{Likelihood Function: } y_{i,g} \sim \text{poisson}(\lambda_g)$$
$$\text{Player level prior: } \lambda_g \sim \text{gamma}(\mu_{0g} * 3, 3)$$

We have changed the prior distribution to a gamma for our player specific rate, or lambda parameter. Consider the following:

$$\lambda_g \sim \text{gamma}(\alpha, \beta)$$
$$E(\lambda_g) = \frac{\alpha}{\beta} = \frac{3*\mu_{0g}}{3} = \mu_{0g}$$

By using the prior specifications of $\alpha = 3 * \mu_{0g}$ and $\beta = 3$ we are specifying the mean of the scoring distribution of player g to be μ_g. This is then scaled by the value of 3 which we will call "c". Lets take a look at some prior predictive plots to see the effect of this scaling constant "c".



As we hold μ_0 constant and change our scaling parameter c, the spread of the prior distribution we are specifying tightens.

The lowest score ever in PGA Tour history was a 58 by Jim Furyk at the Travelers Championship in 2016. The highest ever I am not sure of but id venture to that a PGA Tour players average score is less then 80, certainly 85. Because of this range, I am okay with using the tighter c = 7 parameter, while letting player specific averages control μ.

The stan model for the specifications discussed above is shown below.

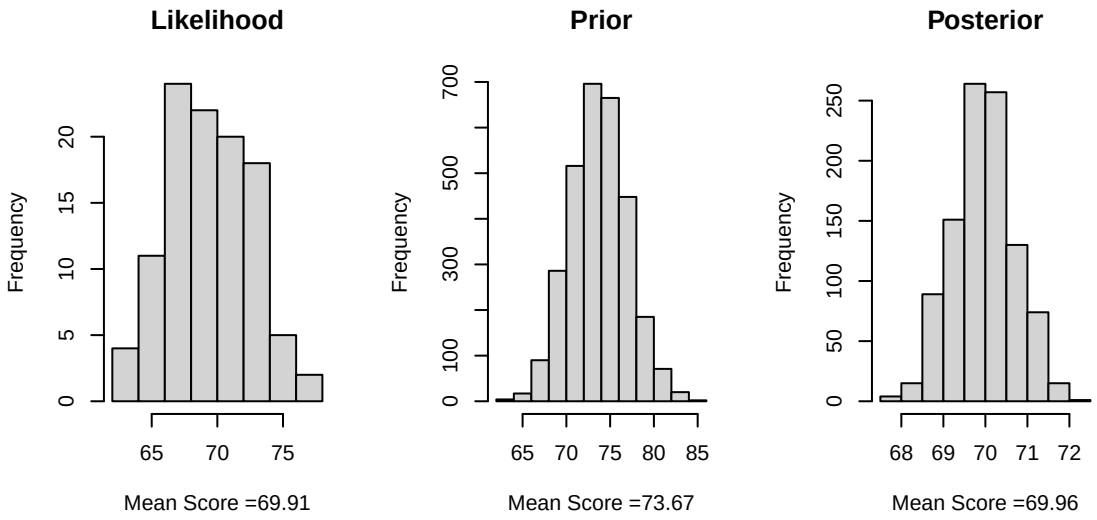
```
data {
  int<lower=0> N; // number of observations
  int<lower=1> G; // number of unique players
  int<lower=0> y[N]; // response variable, scores
  int<lower=1,upper=G> g[N]; // player assignment

  vector<lower=0>[G] mu0; // prior mean for each player's intercept
}

parameters {
  vector<lower=0>[G] lambda; // player specific means
}

model {

  // Player specific priors
  for (j in 1:G){
```

```
    lambda[j] ~ gamma(mu0[j]*7,7);
  }

  // Likelihood
  y ~ poisson(lambda[g]);
}
```

```
model <- stan_model("C:/Users/npbig/Downloads/player_level_fit_player.rds")
data <- list(N = nrow(train_sample),
             G = length(unique(train_sample$dg_id_mapped)),
             y = train_sample$score,
             g = train_sample$dg_id_mapped,
             mu0 = mu0_prior$meanscore)

fit <- sampling(model,data,chains=1,iter=2000)
```

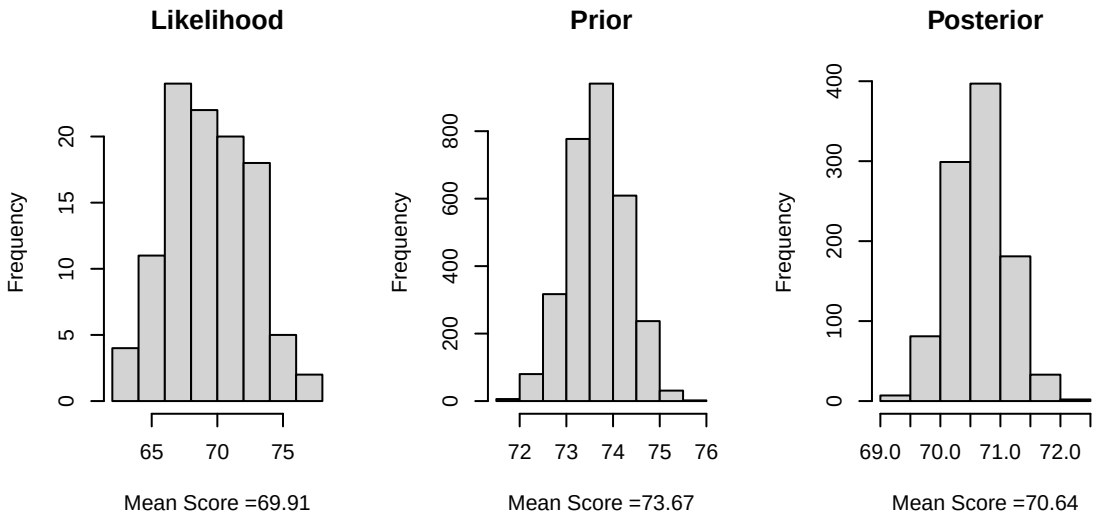
The chains mixed well and all appear to have converged. Next, we will look at the scoring data from 2022-01-01 to present (likelihood), 2024-01-01 to present (prior), and samples from the approximated posterior distribution for a specific golfer.

The golfer we will be looking at is Jon Rahm. During 2022-01-01 to 2024-05-01, our likelihood period, Rahm has an average score on the PGA Tour of 69.91. In more recent history, our prior period, Rahm has had an average score of 73.67. For our exercise we would like the posterior distribution to be weighted with a higher average score to reflect this recent struggle. Plots are shown below.



The posterior distribution appears to be significantly impacted by the likelihood. Lets see what happens when we increase our scaling parameter, c, to 200 in the stan code below.

```
model {
// Player specific priors
for (j in 1:G){
 lambda[j] ~ gamma(mu0[j]*200,200);
 }
}
```



As you can see, increasing our scaling constant to 200 significantly narrowed the range of our prior specification. This also contributed to the increase of the mean of the posterior distribution and very tight range of posterior distribution.

In reality this needs some tweaking, the posterior range is roughly between 69 and 72, a wider range probably represents reality better.

After tweaking this model by potentially using a small scaling constant, one can use it to compare 95% credible intervals for all golfers in the sample.

Mixed Effects Poisson Regression: Player & Course Specific Intercepts and Course Specific effects

An interesting extension of the Bayesian specification could be to include course specific effects and have interaction terms with golfers. The idea being that some golfers score lower at specific courses. A course level intercept could also convey that certain courses have lower scores in general.

Another interesting extension could be random effects specific to each course. The idea being that some courses are better suited towards golfers who are strong putters, or long drivers, or very accurate in hitting greens in regulation. Using predictor variables such as rolling averages of statistics related to greens in regulation, and strokes gained metrics could be fit using course specific specifications. The functional form of this and the corresponding stan model is shown in the section below.

$$\text{Likelihood Function: } y_{i,g} \sim \text{poisson}(exp(\mu_g + \mu_c + (x_{i,p} * \beta_{i,p})) + (x_{i,p,c} * \beta_{i,p,c})))$$

$$\text{Player level prior: } \mu_g \sim \text{normal}(\mu_{0g}, \sigma_{0g})$$

$$\text{Course level prior: } \mu_c \sim \text{normal}(0,4)$$

$$\text{Fixed coefficient prior: } \beta_p \sim \text{normal}(0,2)$$

$$\text{Course coefficient prior: } \beta_{c,p} \sim \text{normal}(0,2)$$

The priors remained roughly the same. A normal(0,4) was chosen for the course level intercept.

This model specification does not include any interaction terms between player and course but it would be an interesting extension to explore. This model was not fit in R however the corresponding stan code can be found below.

```
data {
  int<lower=0> N;           // number of data points
  int<lower=1> G;           // number of players
  int<lower=1> C;           // number of courses
  int<lower=0> y[N];        // scores (count data)
  int<lower=1> P;           // number of predictors
  matrix[N, P] x;           // predictor matrix
  int<lower=1,upper=G> g[N]; // player assignment for each observation
  int<lower=1,upper=C> c[N]; // course assignment for each observation

  vector[G] mu0;            // prior means for player intercepts
  vector[G] sigma0;         // prior std devs for player intercepts
}

parameters {
  // Fixed effects
  vector[P] beta;           // fixed-effect coefficients

  // Player-level effects
  vector[G] mu_g;           // player-specific intercepts

  // Course-level effects
  vector[C] mu_c;           // course-specific intercepts
  vector[P] beta_c[C];      // course-specific slopes
}

model {
  // Priors on fixed effects
  beta ~ normal(0, 2);

  // Player-level priors
  for (l in 1:G) {
    mu_g[l] ~ normal(mu0[l], sigma0[l]);
  }

  // Course-level priors
  mu_c ~ normal(0, 2);      // course intercept priors

  for (k in 1:C) {
    beta_c[k] ~ normal(0, 4);
  }

  // Likelihood
  for (n in 1:N) {
    // linear predictor:
    // include player intercept, course intercept, fixed effects,
    // fixed coefficients, and course-specific coefficients
    real lambda = mu_g[g[n]]
                + mu_c[c[n]]
                + dot_product(beta, x[n])
                + dot_product(beta_c[c[n]], x[n]);

    y[n] ~ poisson_log(lambda);
```

```
  }
}
```

Conclusions

The goal of this document was to display skills in specifying Bayesian models. We successfully built a hierarchical bayesian model which one can use to simulate a round of golf conditional on golfer. We could compare 95% credible intervals for all golfers in the sample to identify the best based on who has the lowest credible interval.