Bayesball

Andrew Burkard and Mary Peng

Introduction

The baseball analytics community has invented an overwhelming number of metrics to benchmark a baseball pitcher's ability. Some metrics, like the ERA (Earned Run Average), focus on outcomes influenced by both the pitcher's ability and external factors such as defensive performance. A desire to evaluate solely the pitcher's skills led to the advent of FIP (Fielding Independent Pitching). FIP comprises outcomes that only the pitcher controls: number of home runs, walks, strikeouts, and batters hit by pitch; it removes the count of balls hit into the field of play.

FIP constitutes a prime candidate for Bayesian analysis for two reasons. First, one can reasonably believe that each player has an underlying FIP distribution, because FIP proxies for individual performance. Furthermore, a pitcher's FIP distribution likely changes during his or her career, as skills mature during early career, peak, and then decline with aging. Consequently, one can only calculate FIP by individual player-seasons, or groupings of player-seasons, thereby leading to small sample sizes that render frequentist models inaccurate.

This paper develops and compares two Bayesian models for individual pitchers' FIP component distributions during 2011 – 2013. The first model assumes that the FIP component statistics come from a common distribution across the three-year timeframe. The second model assumes that every pitcher's performance follows a different distribution during each of the three seasons. We compare the models along their ability to predict the FIP components and ERA in 2014 – 2016. We find that the pitcher-season specific model outperforms the three-year timeframe model in terms of estimating the FIP parameters. However, both models show improvements over observed FIP when predicting ERA over the upcoming three seasons.

Data and Methodology

Before diving into methodology, let us first define a few key terms referenced throughout the paper:

Number of Batters Faced = HR + HBP + BB + SO + BIP, where HR = # of home runs, HBP = # of hits by pitch, BB = # of walks, SO = # of strikeouts, and BIP = balls in play. The first four statistics depend only on the pitcher's performance (whereas BIP depends on defense performance as well), and are referred to as the "FIP component statistics" in the rest of this paper.

FIP = $\frac{13*HR+3*(HBP+BB)-2SO}{Innings\ Pitched} + FIP\ constant^{1}$; FIP constant varies annually, but is generally around 3.2 and is added to put FIP at the same scale as Earned Run Average (see appendix for formula).

ERA = $\frac{9*Earned\ Runs\ Allowed}{Innings\ Pitched}$

We leverage pitcher performance data at the player and season level from Lahman R package. We consider only pitchers with data for all five metrics (HR, HBP, BB, SO, and BIP) for 2011 – 2016; 2011 – 2013 is used for modelling, and 2014 – 2016 is used for prediction comparison. We build models for a total of 50 pitchers and 150 player-seasons.

Let $\mathbf{z}_i^{(t)}$ = [$z_{1i}^{(t)}$ $z_{2i}^{(t)}$ $z_{3i}^{(t)}$ $z_{4i}^{(t)}$ $z_{5i}^{(t)}$] represent a pitcher *i*'s homeruns, walks, hits by pitch, strikeouts, and balls in play in season *t*. The last component, balls in play, is a nuisance parameter since we are only interested in measuring the four FIP components, but need BIP to ensure that the probabilities add up to 1. We use this likelihood to build two different Bayesian models. See appendix for full derivations of the posterior and details on the sampling method.

The first model assumes that a pitcher's FIP component statistics come from a common distribution during 2011 - 2013: $\mathbf{z}_i^{(t)} \overset{}{_{\text{i.i.d.}}}\sim$ *Multinom*(n_i; ϑ_{1i}, ϑ_{2i}, ϑ_{3i}, ϑ_{4i}, ϑ_{5i}), where n_i represents the total number of batters faced. Multinomial likelihood works well since the observed data $\mathbf{z}_i^{(\mathbf{t})}$ are counts with probabilities that sum up to 1. Let $\boldsymbol{\vartheta}_i$ = [ϑ_{1i}, ϑ_{2i}, ϑ_{3i}, ϑ_{4i}, ϑ_{5i}]' be the vector of rate of homeruns, walks, hits by pitch, strikeouts, and balls in play as percentage of numbers of batters faced. We assume a conjugate prior $\boldsymbol{\theta}_i \sim$ Dir(α_{1i}, α_{2i}, α_{3i}, α_{4i}, α_{5i}) and hyper-prior $\pi(\boldsymbol{\alpha}_i) \propto \frac{1}{\Gamma(\alpha_{1i}+\alpha_{2i}+\alpha_{3i}+\alpha_{4i}+\alpha_{5i})}$. We use Gibbs-MH to sample the marginal posterior of $\boldsymbol{\alpha}_i$ and then draw $\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i$.

In the second model, we assume that the distributions of a pitcher's FIP component statistics change every season: $\mathbf{z}_i^{(t)} \sim$ *Multinom*($n_i^{(t)}$; $\vartheta_{1i}^{(t)}$,$\vartheta_{2i}^{(t)}$,$\vartheta_{3i}^{(t)}$,$\vartheta_{4i}^{(t)}$ $\vartheta_{5i}^{(t)}$). Instead of assuming a joint prior across all three seasons, we designate a separate Dirichlet prior for each player-season. The model also incorporates historical data by designating the weighted sum of previous seasons' observations as the components of the Dirichlet:

$$\boldsymbol{\theta}_i^{(t)} \sim Dir(\textstyle\sum_{k=1}^{t}\frac{z_{1i}^{(k)}}{b^{j-k}}+1, \sum_{k=1}^{t}\frac{z_{2i}^{(k)}}{b^{j-k}}+1, \sum_{k=1}^{t}\frac{z_{3i}^{(k)}}{b^{j-k}}+1, \sum_{k=1}^{j}\frac{z_{4i}^{(k)}}{b^{j-k}}+1, \sum_{k=1}^{j}\frac{z_{5i}^{(k)}}{b^{j-k}}+1)$$

where *b* represents a decay rate between 0 and 1 that weights the more recent years' observations more heavily. In addition to the decay rate, we considered normalizing a player's observed counts in a given season by the number of batters faced in that season. For example, if a pitcher faces 50 batters one season instead of the usual 200, his/her number of walks, strikeouts, etc. will necessarily be lower as well. Ultimately, we decided against normalizing because if a pitcher faced few batters in a given season, the data from that season will have less influence on the Dirichlet prior. Giving less weight to observations from small-sample-size seasons compared to seasons with larger sample size make sense, since the larger sample sizes necessarily contain more information and broader range of observations than the small sample sizes.

The next section compares the two models along the following dimensions:

- Which model more accurately predicts the FIP component statistics observed in 2014 – 2016?
- Combining the posterior samples into a Bayesian FIP, which model more accurately predicts the ERA, and how does its accuracy compare with the frequentist FIP?

Results and Discussion

Table 1: Comparison of Posterior Predictive Values vs. Observed Data

	% of players where 95% CI contains 2014 – 2016 observed # of walks	% of players where 95% CI contains 2014 – 2016 observed # of strikeouts	% of players where 95% CI contains 2014 - 2016 observed # of homeruns	% of players where 95% CI contains 2014 – 2016 observed # of hit by pitch	% of players where 95% CI contains 2014-2016 observed # of balls in play
Model I	24%	8%	16%	20%	10%
Model II	34%	32%	52%	62%	52%

Model 1 refers to the pitcher-specific model
Model 2 refers to the pitcher-season specific model

FIP's denominator is number of innings pitched in the denominator, while our estimated walk rates, strikeout rates etc. use the number of batters faced in the denominator, we multiply all of our estimated rates by the number of batters faced and divide by the number of innings pitched to make the numbers comparable.

Table 32 Comparison of Predictiveness of Adjusted FIP vs. Traditional FIP of ERA

	R^2 when regress ERA on adjusted FIP (median / mode)
Model I	.1528
Model II	.1648
Industry FIP	.1034

While the 95% credible intervals do not contain the observed values as much as one would hope, both models show a stronger correlation against ERA over the subsequent 3-years than using the purely the observed counts values to calculate FIP as seen in Table 2. Interestingly, we see in Table 1 that Model II significantly outperformed Model I when estimating the parameters of the two most unlikely categories, HR and HBP. It is possible that this is due to the hyper-prior in Model I producing overestimates of these values. Model II outperformed Model I when considering the percentage of observed values within the credible intervals for all FIP rate parameters.

Conclusion

Model II outperforms Model I by all measures in our results. One could hypothesize many reasons for this. It may be that year-to-year fluctuations in a pitcher's underlying walk, strikeout, and home run rates are so great that assuming them to come from a common distribution is unwise. It's also possible that using a timeframe other than three seasons to build the model, and three seasons for prediction would improve results. Lastly, the selection of the hyper-prior $\pi(\boldsymbol{\alpha}_i)$ should be examined. One approach could be to use a hybrid version of the two models where the rates are assumed to come from a common distribution, but the prior is informed by the values for previous seasons.

References

1. "Panel Data Modeling and Inference: A Bayesian Primer Chapter 15", Siddhartha Chib, The Econometrics of Panel Data, Springer-Verlag Berlin Heidelberg 2008

2. "Bayesian Poll of Polls for Multi-Party Systems," Miriam Hurtado Bodell, Division of Statistics and Machine Learning, Linkoping University

3. "Bayesian Data Analysis, Third Edition." Andrew Gelman et al. CPC Press, Taylor & Francis Group, 2014

4. "Measuring Pitcher Performance in 2017," Matt Delfing, https://unbalanced.media/measuring-pitcher-performance-in-2017

5. "The Many Flavors of DIPS: A History and Overview," Dan Basco and Michael Davis, Fall 2010 Baseball Research Journal

6. "FIP, Fan Graphs," https://www.fangraphs.com/library/pitching/fip/

Appendix

Section 1: Formula and Definitions:

FIP Constant = lgERA – (((13*lgHR)+(3*(lgBB+lgHBP))-(2*lgK))/lgIP)

Section 2: Analytic Derivations and Sampling Method

For a single pitcher *i* in season *t*, $\mathbf{z}_{\mathrm{i}}^{(\mathrm{t})}$ ~ *Multinom*$(n_i^{(t)}; \vartheta_{1i}^{(t)},\vartheta_{2i}^{(t)},\vartheta_{3i}^{(t)},\vartheta_{4i}^{(t)} \vartheta_{5i}^{(t)})$; $n_i^{(t)}$ represents the total number of batters faced in season *t*. Let $\boldsymbol{\vartheta}_i^{(t)} = [\vartheta_{1i}^{(t)},\vartheta_{2}^{(t)},\vartheta_{3i}^{(t)},\vartheta_{4i}^{(t)}, \vartheta_{5i}^{(t)}]'$.

<u>Model 1: Pitcher-Specific Model</u>

For a single pitcher *i* in season *t*, $\mathbf{z}_{\mathrm{i}}^{(\mathrm{t})}$~ Multinom$(\mathrm{n}_{\mathrm{i}}^{(\mathrm{t})}; \theta_{1i},\theta_{2i},\theta_{3i},\theta_{4i,} \theta_{5i})$; $n_i^{(t)}$ represents the total number of batters faced in season *t*.

The likelihood function for the observed FIP components in each season for pitcher *i* is:

$$\mathrm{L}(\boldsymbol{z}_{\boldsymbol{i}}^{(\boldsymbol{t})} \,|\, \boldsymbol{\theta}_{\boldsymbol{i}}) = \binom{n_i^{(t)}}{z_{1i}^{(t)}\ z_{2i}^{(t)}\ z_{3i}^{(t)}\ z_{4i}^{(t)}\ z_{5i}^{(t)}} (\theta_{1i})^{z_{1i}^{(t)}}\ (\theta_{2i})^{z_{2i}^{(t)}} (\theta_{3i})^{z_{3i}^{(t)}} (\theta_{4i})^{z_{4i}^{(t)}} (\theta_{5i})^{z_{5i}^{(t)}}$$

The joint likelihood across all three seasons is:

$$\mathrm{L}(\boldsymbol{z}_{\boldsymbol{i}}^{(\mathbf{2011})}, \boldsymbol{z}_{\boldsymbol{i}}^{(\mathbf{2012})},\ \boldsymbol{z}_{\boldsymbol{i}}^{(\mathbf{2013})} \,|\, \boldsymbol{\theta}_{\boldsymbol{i}}) = \prod_{t=2011}^{2013} \binom{n_i^{(t)}}{z_{1i}^{(t)}\ z_{2i}^{(t)}\ z_{3i}^{(t)}\ z_{4i}^{(t)}\ z_{5i}^{(t)}} (\theta_{1i})^{z_{1i}^{(t)}}\ (\theta_{2i})^{z_{2i}^{(t)}} (\theta_{3i})^{z_{3i}^{(t)}} (\theta_{4i})^{z_{4i}^{(t)}} (\theta_{5i})^{z_{5i}^{(t)}}$$

$$= [\prod_{t=2011}^{2013} \binom{n_i^{(t)}}{z_{1i}^{(t)}\ z_{2i}^{(t)}\ z_{3i}^{(t)}\ z_{4i}^{(t)}\ z_{5i}^{(t)}}][(\theta_{1i})^{\sum_{t=2011}^{2013} z_{1i}^{(t)}}\ (\theta_{2i})^{\sum_{t=2011}^{2013} z_{2i}^{(t)}} (\theta_{3i})^{\sum_{t=2011}^{2013} z_{3i}^{(t)}} (\theta_{4i})^{\sum_{t=2011}^{2013} z_{4i}^{(t)}} (\theta_{5i})^{\sum_{t=2011}^{2013} z_{5i}^{(t)}}]$$

$\boldsymbol{\theta}_i \sim \mathrm{Dir}(\alpha_{1i}, \alpha_{2i}, \alpha_{3i}, \alpha_{4i}, \alpha_{5i})$, so $\pi(\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i) = \frac{\Gamma(\alpha_{1i}+ \alpha_{2i}+ \alpha_{3i}+ \alpha_{4i}+ \alpha_{5i})}{\Gamma(\alpha_{1i})\Gamma(\alpha_{2i})\Gamma(\alpha_{3i})\Gamma(\alpha_{4i})\Gamma(\alpha_{5i})}(\theta_{1i})^{\alpha_{1i}}(\theta_{2i})^{\alpha_{2i}}(\theta_{3i})^{\alpha_{3i}}(\theta_{4i})^{\alpha_{4i}}(\theta_{5i})^{\alpha_{5i}}$

We also assume hyper-prior $\pi(\boldsymbol{\alpha}_i) \propto \frac{1}{\Gamma(\alpha_{1i}+ \alpha_{2i}+ \alpha_{3i}+ \alpha_{4i}+ \alpha_{5i})}$

We previously considered using a flat hyper-prior $\pi(\boldsymbol{\alpha}_i) \propto c$, but that resulted in the Gibbs-MH sampler failing to converge. This was due to the fact that infinitely large values of **α**, can generate plausible **θ** so long as the $\boldsymbol{\alpha}_i$ remain in relative proportion with each other.

The full posterior for pitcher *i*'s statistics are:

$$p(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i|\boldsymbol{z}_i^{(2011)}, \boldsymbol{z}_i^{(2012)}, \boldsymbol{z}_i^{(2013)}) \propto L(\boldsymbol{z}_i^{(2011)}, \boldsymbol{z}_i^{(2012)}, \boldsymbol{z}_i^{(2013)}|\boldsymbol{\theta}_i) * \pi(\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i) * \pi(\boldsymbol{\alpha}_i)$$

$$\propto [(\theta_{1i})^{\sum_{t=2011}^{2013} z_{1i}^{(t)}}(\theta_{2i})^{\sum_{t=2011}^{2013} z_{2i}^{(t)}}(\theta_{3i})^{\sum_{t=2011}^{2013} z_{3i}^{(t)}}(\theta_{4i})^{\sum_{t=2011}^{2013} z_{4i}^{(t)}}(\theta_{5i})^{\sum_{t=2011}^{2013} z_{5i}^{(t)}}] *$$

$$\frac{\Gamma(\alpha_{1i}+ \alpha_{2i}+ \alpha_{3i}+ \alpha_{4i}+ \alpha_{5i})}{\Gamma(\alpha_{1i})\Gamma(\alpha_{2i})\Gamma(\alpha_{3i})\Gamma(\alpha_{4i})\Gamma(\alpha_{5i})}(\theta_{1i})^{\alpha_{1i}}(\theta_{2i})^{\alpha_{2i}}(\theta_{3i})^{\alpha_{3i}}(\theta_{4i})^{\alpha_{4i}}(\theta_{5i})^{\alpha_{5i}} * \frac{1}{\Gamma(\alpha_{1i}+ \alpha_{2i}+ \alpha_{3i}+ \alpha_{4i}+ \alpha_{5i})}$$

$$= \frac{1}{\Gamma(\alpha_{1i})\Gamma(\alpha_{2i})\Gamma(\alpha_{3i})\Gamma(\alpha_{4i})\Gamma(\alpha_{5i})} *$$
$$[(\theta_{1i})^{\sum_{t=2011}^{2013} z_{1i}^{(t)}+\alpha_{1i}}(\theta_{2i})^{\sum_{t=2011}^{2013} z_{2i}^{(t)}+\alpha_{2i}}(\theta_{3i})^{\sum_{t=2011}^{2013} z_{3i}^{(t)}+\alpha_{3i}}(\theta_{4i})^{\sum_{t=2011}^{2013} z_{4i}^{(t)}+\alpha_{4i}}(\theta_{5i})^{\sum_{t=2011}^{2013} z_{5i}^{(t)}+\alpha_{5i}}]$$

The conditional $p(\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i, \boldsymbol{z}_i^{(2011)}, \boldsymbol{z}_i^{(2012)}, \boldsymbol{z}_i^{(2013)})$

$$\propto (\theta_{1i})^{\sum_{t=2011}^{2013} z_{1i}^{(t)}+\alpha_{1i}}(\theta_{2i})^{\sum_{t=2011}^{2013} z_{2i}^{(t)}+\alpha_{2i}}(\theta_{3i})^{\sum_{t=2011}^{2013} z_{3i}^{(t)}+\alpha_{3i}}(\theta_{4i})^{\sum_{t=2011}^{2013} z_{4i}^{(t)}+\alpha_{4i}}(\theta_{5i})^{\sum_{t=2011}^{2013} z_{5i}^{(t)}+\alpha_{5i}}$$

The marginal posterior $p(\boldsymbol{\alpha}_i|\boldsymbol{z}_i^{(2011)}, \boldsymbol{z}_i^{(2012)}, \boldsymbol{z}_i^{(2013)})$

$$\propto \int \frac{1}{\Gamma(\alpha_{1i})\Gamma(\alpha_{2i})\Gamma(\alpha_{3i})\Gamma(\alpha_{4i})\Gamma(\alpha_{5i})}$$
$$* [(\theta_{1i})^{\sum_{t=2011}^{2013} z_{1i}^{(t)}+\alpha_{1i}}(\theta_{2i})^{\sum_{t=2011}^{2013} z_{2i}^{(t)}+\alpha_{2i}}(\theta_{3i})^{\sum_{t=2011}^{2013} z_{3i}^{(t)}+\alpha_{3i}}(\theta_{4i})^{\sum_{t=2011}^{2013} z_{4i}^{(t)}+\alpha_{4i}}(\theta_{5i})^{\sum_{t=2011}^{2013} z_{5i}^{(t)}+\alpha_{5i}}]d\boldsymbol{\theta}_i$$

$$\propto \frac{1}{\Gamma(\alpha_{1i})\Gamma(\alpha_{2i})\Gamma(\alpha_{3i})\Gamma(\alpha_{4i})\Gamma(\alpha_{5i})} * \frac{\Gamma(\sum_{t=2011}^{2013} z_{1i}^{(t)}+\alpha_{1i})\Gamma(\sum_{t=2011}^{2013} z_{2i}^{(t)}+\alpha_{2i})\Gamma(\sum_{t=2011}^{2013} z_{3i}^{(t)}+\alpha_{3i})\Gamma(\sum_{t=2011}^{2013} z_{4i}^{(t)}+\alpha_{4i})\Gamma()^{\sum_{t=2011}^{2013} z_{5i}^{(t)}+\alpha_{5i}})}{\Gamma[\left(\sum_{t=2011}^{2013} z_{1i}^{(t)}+\alpha_{1i}\right)+\left(\sum_{t=2011}^{2013} z_{2i}^{(t)}+\alpha_{2i}\right)+\left(\sum_{t=2011}^{2013} z_{3i}^{(t)}+\alpha_{3i}\right)+\left(\sum_{t=2011}^{2013} z_{4i}^{(t)}+\alpha_{4i}\right)+\left(\,)^{\sum_{t=2011}^{2013} z_{5i}^{(t)}+\alpha_{5i}}\right)]}$$

To draw samples of **α** and **θ**, we use a Gibbs sampler. First we sample each $\boldsymbol{\alpha}_i$ component of **α** using a Metropolis-Hastings step. We use a Gamma($\boldsymbol{\alpha}_i$, 1) as our proposal density, that is a gamma distribution centered at the previous $\boldsymbol{\alpha}_i$. Once **α** is sampled, we can sample **θ** directly from the Dirichlet distribution.

Model 2: Pitcher-Season-Specific Model

For a single player *i* in season t:

$$L(\boldsymbol{z}_i^{(t)}|\boldsymbol{\theta}_i^{(t)}) = \binom{n_i^{(t)}}{z_{1i}^{(t)}\ z_{2i}^{(t)}\ z_{3i}^{(t)}\ z_{4i}^{(t)}\ z_{5i}^{(t)}}(\theta_{1i}^{(t)})^{z_{1i}^{(t)}}(\theta_{2i}^{(t)})^{z_{2i}^{(t)}}(\theta_{3i}^{(t)})^{z_{3i}^{(t)}}(\theta_{4i}^{(t)})^{z_{4i}^{(t)}}(\theta_{5i}^{(t)})^{z_{5i}^{(t)}}$$

We select Dirichlet as our conjugate prior. To account for the data from previous seasons, we designate the weighted sum of previous seasons' observations as the components of the Dirichlet. *b* represents a decay rate between 0 and 1 that weights the more recent years' observations more heavily.

$$\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t})} \sim Dir(\sum_{k=1}^{t} \frac{z_{1i}^{(k)}}{b^{j-k}}, \sum_{k=1}^{t} \frac{z_{2i}^{(k)}}{b^{j-k}}, \sum_{k=1}^{t} \frac{z_{3i}^{(k)}}{b^{j-k}}, \sum_{k=1}^{j} \frac{z_{4i}^{(k)}}{b^{j-k}}, \sum_{k=1}^{j} \frac{z_{5i}^{(k)}}{b^{j-k}})$$

We also considered normalizing the counts in the components of the Dirichlet prior to account for differences in the number of batters faced by a pitcher from year to year. For example, if a pitcher faces 50 batters one season instead of the usual 200, his/her number of walks, strikeouts, etc. will necessarily be lower as well.

Ultimately, we decided to not normalize because smaller sample size from one season would also result in that data point being weighted less in the Dirichlet.

We selected a decay rate of 1/2. This has the effect of weighting the most recent season equal to all previous seasons combined. If we had more time, a good sensitivity check would have been to test the effectiveness of the models when using different decay rates.

For an individual player *i* season *t*, the posterior is:

$p(\boldsymbol{\theta}_i^{(t)} | \mathbf{z}_i^{(1)} \ldots \mathbf{z}_1^{(t)})$

$\propto L(\mathbf{z}_i^{(t)} | \boldsymbol{\theta}_i^{(t)}) * \pi(\boldsymbol{\theta}_i^{(t)})$

$$\propto (\theta_{1i}^{(t)})^{z_{1i}^{(t)}} (\theta_{2i}^{(t)})^{z_{2i}^{(t)}} (\theta_{3i}^{(t)})^{z_{3i}^{(t)}} (\theta_{4i}^{(t)})^{z_{4i}^{(t)}} (\theta_{5i}^{(t)})^{z_{5i}^{(t)}}$$

$$*(\theta_{1i}^{(t)})^{\sum_{k=1}^{t} \frac{z_{1i}^{(k)}}{b^{j-k}} - 1} (\theta_{2i}^{(t)})^{\sum_{k=1}^{t} \frac{z_{2i}^{(k)}}{b^{j-k}} - 1} (\theta_{3i}^{(t)})^{\sum_{k=1}^{t} \frac{z_{3i}^{(k)}}{b^{j-k}} - 1} (\theta_{4i}^{(t)})^{\sum_{k=1}^{t} \frac{z_{4i}^{(k)}}{b^{j-k}} - 1} (\theta_{5i}^{(t)})^{\sum_{k=1}^{t} \frac{z_{5i}^{(k)}}{b^{j-k}} - 1}$$

$$\propto [(\theta_{1i}^{(t)})^{z_{1i}^{(t)} + \sum_{k=1}^{t} \frac{z_{1i}^{(k)}}{b^{j-k}} - 1} (\theta_{2i}^{(t)})^{z_{2i}^{(t)} + \sum_{k=1}^{t} \frac{z_{2i}^{(k)}}{b^{j-k}} - 1} (\theta_{3i}^{(t)})^{z_{3i}^{(t)} + \sum_{k=1}^{t} \frac{z_{3i}^{(k)}}{b^{j-k}} - 1} (\theta_{4i}^{(t)})^{z_{4i}^{(t)} + \sum_{k=1}^{t} \frac{z_{4i}^{(k)}}{b^{j-k}} - 1} (\theta_{5i}^{(t)})^{z_{4i}^{(t)} + \sum_{k=1}^{t} \frac{z_{5i}^{(k)}}{b^{j-k}} - 1}$$

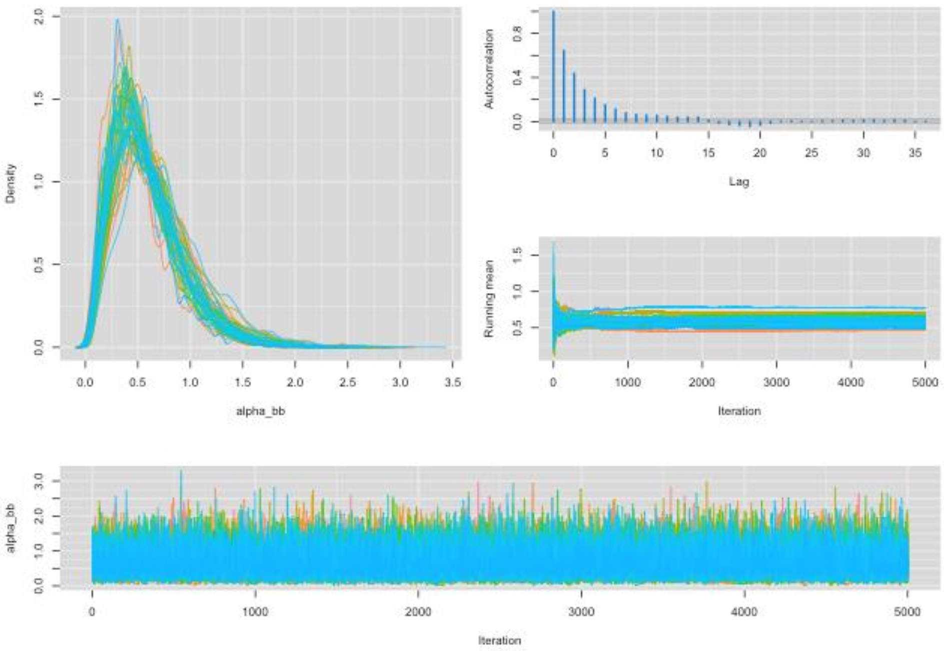
Hence $\boldsymbol{\theta}_i^{(t)} | \mathbf{z}_i^{(1)} \ldots \mathbf{z}_1^{(t)} \sim$ Dirichlet($z_{1i}^{(t)} + \sum_{k=1}^{t} \frac{z_{1i}^{(k)}}{b^{j-k}}$, $z_{2i}^{(t)} + \sum_{k=1}^{t} \frac{z_{2i}^{(k)}}{b^{j-k}}$, $z_{3i}^{(t)} + \sum_{k=1}^{t} \frac{z_{3i}^{(k)}}{b^{j-k}}$, $z_{4i}^{(t)} + \sum_{k=1}^{t} \frac{z_{4i}^{(k)}}{b^{j-k}}$, $z_{5i}^{(t)} + \sum_{k=1}^{t} \frac{z_{5i}^{(k)}}{b^{j-k}}$)

We assume that the $\boldsymbol{\theta}_i^{(t)}$ are independent, and so we will sample the posterior for each player season separately using the custom Dirichlet distribution.

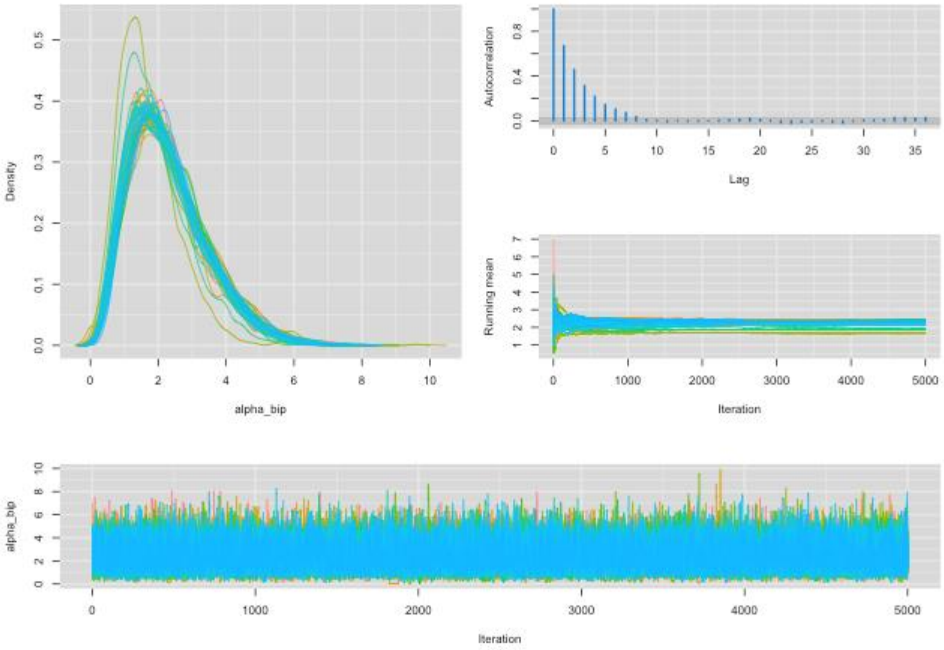
Section 3: Convergence of Posterior Samples in Models 1 and 2

Overlayed convergence plots across all pitchers. From the Autocorrelation plots, we determine that a thinning rate of 10 is reasonable.

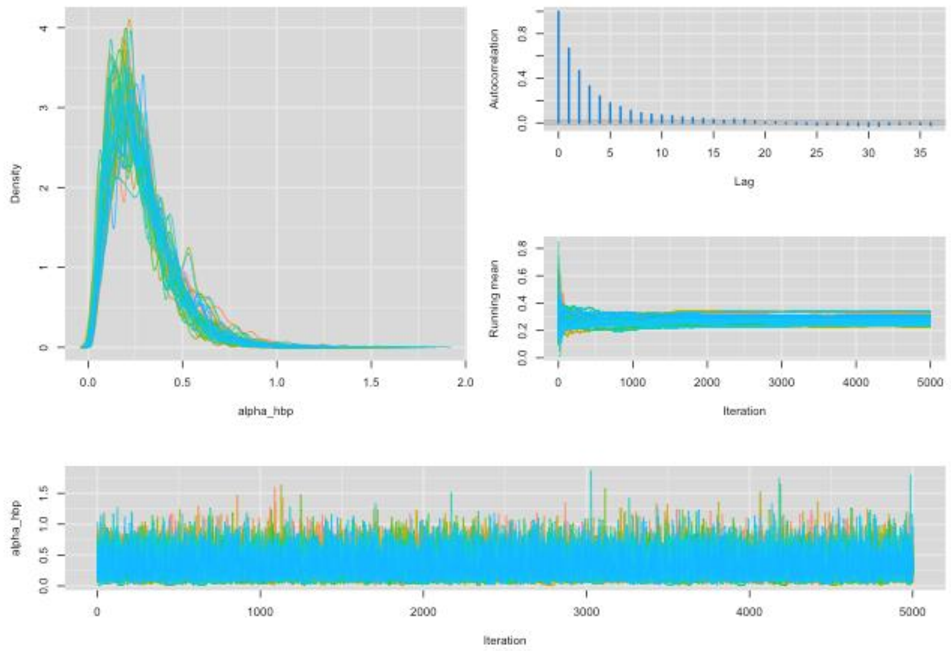
Diagnostics for alpha_bb



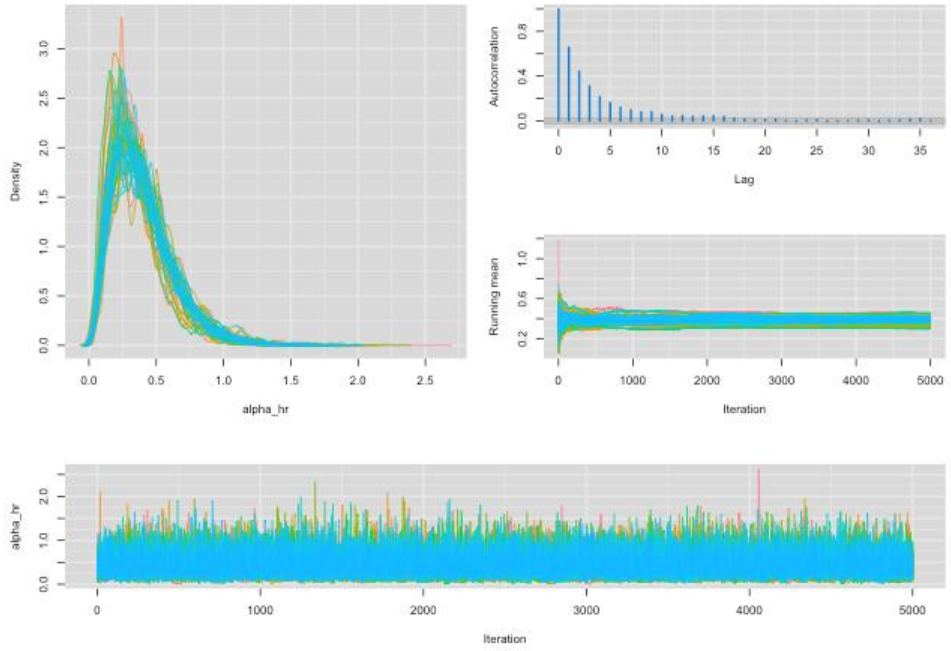
Diagnostics for alpha_bip

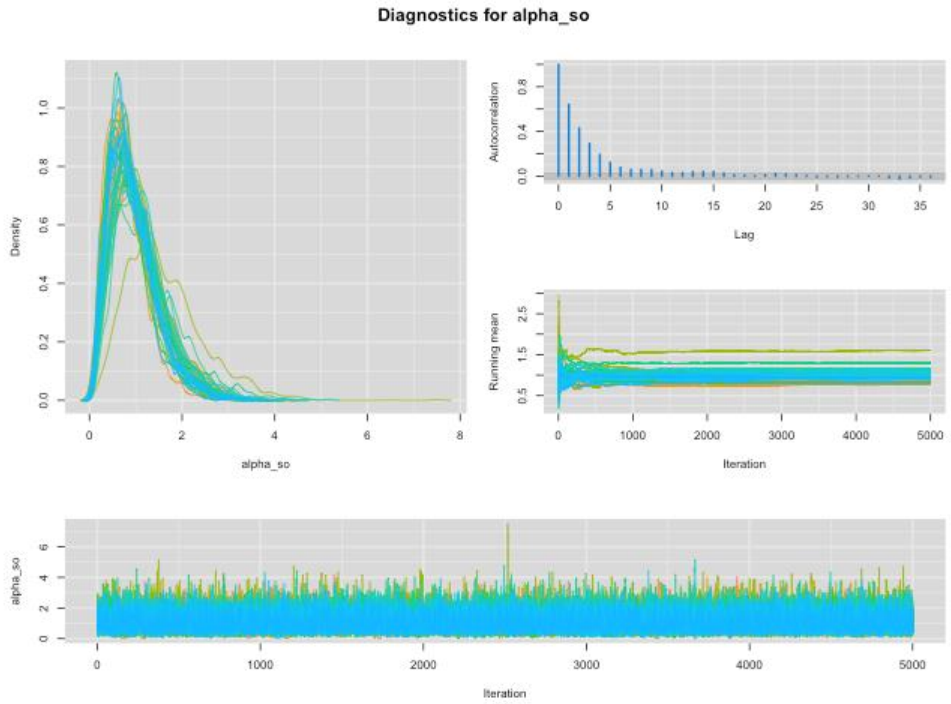


Diagnostics for alpha_hbp



Diagnostics for alpha_hr





Section 4: R Code Appendix

# Also available at github.com/aburkard/bayesball

~~~
```{r}

############# Model 1

library(MCMCpack)

library(mcmcplots)

library(Lahman)

library(dplyr)


# Note: dplyr must be loaded after MCMCpack, otherwise some functions get overridden. You can use unloadNamespace on both packages if this happens.


B <- 1000


# log marginal density of alpha|Z
~~~

```
ldalpha <- function(alpha, z, prior_exp=0) {
  J <- nrow(z)
  z_plus_alpha <- sweep(z, 2, alpha, "+")
  log_density <- (lgamma(sum(alpha))*(J-prior_exp) - sum(lgamma(alpha))*J) +
    sum(lgamma(z_plus_alpha)) -
    sum(lgamma(rowSums(z_plus_alpha)))

  log_density
}

# density of alpha|Z
dalpha <- function(alpha, z) {
  exp(ldalpha(alpha, z))
}

# Generates samples given a dataframe of pitchers
# Should be used with a dataframe containing multiple rows all of one pitcher
generate_samples <- function(x) {
  #z <- matrix(c(x$HR, x$SO), ncol=N_BINS)
  z <- matrix(c(x$HR[1:3], x$SO[1:3], x$BB[1:3], x$HBP[1:3], x$BIP[1:3]), ncol=5)
  z_predict_actual <- matrix(c(x$HR[4:6], x$SO[4:6], x$BB[4:6], x$HBP[4:6], x$BIP[4:6]), ncol=5)
  z_predict_actual <- colSums(z_predict_actual)
  set.seed(1234)
  samples = sample_a_theta(z, z_predict_actual, sum(x$IPouts[4:6]/3))
  #print(samples["Theta"])
  #print(samples["alpha"])

  alpha <- samples$alpha
  colnames(alpha) <- c("alpha_hr","alpha_so","alpha_bb","alpha_hbp","alpha_bip")
  #print(colMeans(alpha))
  #print(colMeans(samples$acceptance_rate))
  alpha_chain = mcmc(alpha)

  #alpha_chains[[length(alpha_chains)+1]] <<- alpha_chain
  #mcmcplot(alpha_chain)


  #Theta <- samples$Theta
  #theta_so <- unlist(Theta)[seq(nrow(z)*1+1, B*5*nrow(z), 5*nrow(z))]
  #print(Theta)
  #print(theta_so)
  #plot(density(theta_so))
  return(x)
}

# Gibbs MH sampler for generating alpha, theta
sample_a_theta <- function(z, z_predict_actual, innings) {
  N_BINS <- NCOL(z)
  J <- nrow(z)
```

```
alpha <- matrix(0, nrow=B, ncol=N_BINS)
alpha_acceptance <- matrix(0, nrow=B, ncol=N_BINS)
Theta <- list(matrix(0, nrow=J, ncol=N_BINS))
mean_z <- colMeans(z)
var_z <- apply(z, 2, var)
prop_alpha <- mean_z^2/var_z
prop_beta <- mean_z/var_z
alpha[1,] <- rep(1,N_BINS) #c(2231,21294,5630,1090,54396)#

z_plus_alpha <- sweep(z, 2, alpha[1,], "+")
theta <- t(apply(z_plus_alpha, 1, function(x) rdirichlet(1, x)))
Theta[[1]] <- theta

for (i in 2:B) {
  for (k in 1:N_BINS) {
    J_alpha <- alpha[i-1, k]
    J_beta <- 1
    alpha_star <- rgamma(1, J_alpha, J_beta)
    alpha_star_vec <- alpha[i-1,]
    alpha_star_vec[k] <- alpha_star

    log_rho <- ldalpha(alpha_star_vec, z) -
      log(dgamma(alpha_star, J_alpha, J_beta)) -
      ldalpha(alpha[i-1, ], z) +
      log(dgamma(alpha[i-1, k], J_alpha, J_beta))
    rho <- exp(log_rho)

    # Print diagnostics
    #print("")
    #print(paste("alpha_star ", alpha_star))
    #print(paste("alpha_star prob ", exp(ldalpha(alpha_star_vec, z))))
    #print(paste("proposal prob ", exp(log(dgamma(alpha_star, J_alpha, J_beta)))))
    #print(paste("prev alpha prob ", exp(ldalpha(alpha[i-1, ], z))))
    #print(paste("prev proposal prob ", exp(log(dgamma(alpha[i-1, k], J_alpha, J_beta)))))
    #print(paste("log rho ", log_rho))

    u <- runif(1)
    if (u < min(rho, 1)) {
      alpha[i, k] <- alpha_star
      alpha_acceptance[i, k] = 1
    }
    else {
      alpha[i, k] <- alpha[i-1, k]
    }
  }

  z_plus_alpha <- sweep(z, 2, alpha[i,], "+")
  #theta <- t(apply(z_plus_alpha, 1, function(x) rdirichlet(1, x)))
  #Theta[[i]] <- theta
```

```
  }
  #print(tail(alpha))
  #print(colMeans(alpha))
  #print(colMeans(alpha_acceptance))
  #print(Theta[[B]])
  #theta_list <- unlist(Theta)
  #s1_HR <- theta_list[seq(1+J*1, length(theta_list), J*N_BINS)]
  #plot(density(s1_HR))

  alpha <- alpha[(B/2):B,]
  Theta <- Theta[(B/2):B]
  #in_cis <<- in_cis + predict_in_ci(alpha, z_predict_actual)
  fip <- calculate_fips(alpha, z_predict_actual, innings)
  fips_1 <<- c(fips_1, fip)
  list("Theta"=Theta, "alpha"=alpha, "acceptance_rate"=alpha_acceptance)
}

#alpha_chains <- list()
in_cis <- c(0,0,0,0,0)
fips_1 <- c()

pitchers <- Pitching %>%
  filter(yearID >= 2011) %>%
  #filter(playerID == "salech01") %>%
  filter(BFP >= 200) %>%
```

~~~~
  mutate_if(is.factor, as.character) %>%
  mutate(BIP = BFP-(HR+BB+HBP+SO)) %>%
  filter(!is.na(HR) && !is.na(BB) && !is.na(HBP) && !is.na(SO) && !is.na(BFP)) %>%
  select(playerID, yearID, IPouts, HR, BB, HBP, SO, BIP, BFP) %>%
  group_by(playerID, yearID) %>%
  summarise_all(funs(sum)) %>%
  group_by(playerID) %>%
  filter(min(yearID) == 2011 & max(yearID) == 2016 & length(yearID) == 6 & min(HBP)>0) %>%
  filter(yearID >=2011 & yearID <= 2016) %>%
  do(generate_samples(.))

pitchers
nrow(pitchers)
mcmcplot(alpha_chains)
```
```{r}
calculate_fips <- function(alpha, z_predict_actual, innings) {
  #chain <- alpha_chains[[1]]
  #chain <- chain[5001:10000,]
  #mcmcplot(chain)
  #n_2014 <- 1000
  n <- sum(z_predict_actual)
  theta <- t(apply(alpha, 1, function(x) rdirichlet(1, x)))
  z_pred <- t(apply(theta, 1, function(x) rmultinom(1, n, x)))
~~~~

~~~
  meds <- apply(z_pred, 2, median)
  meds
  fip <- (13*meds[1] + 3*(meds[4]+meds[3])-2*meds[2])/innings
  fip
}


predict_in_ci <- function(alpha, z_predict_actual) {
  #chain <- alpha_chains[[1]]
  #chain <- chain[5001:10000,]
  #mcmcplot(chain)
  #n_2014 <- 1000
  n <- sum(z_predict_actual)
  theta <- t(apply(alpha, 1, function(x) rdirichlet(1, x)))
  z_pred <- t(apply(theta, 1, function(x) rmultinom(1, n, x)))
  #colMeans(theta_2014)
  #quantile(theta_2014[,2],probs=c(.025,.975))
  #print(apply(z_pred, 2, function(x) quantile(x, probs=c(.025,.975))))
  #print(z_predict_actual)
  in_ci <- z_predict_actual >= z_pred[1,] & z_predict_actual <= z_pred[2,]
  in_ci
}


calculate_fips(blah_alpha, z_predict_actual, 450)
```
~~~

~~~
```{r}
plot(fips_actual, eras, main="Model 1 FIP 2014-16 ERA Regression", ylab="2014-16 ERA", xlab="Median Posterior FIP")
fit1 <- lsfit(fips_1, eras)
abline(fit1$coefficients[1], fit1$coefficients[2], col="blue")
cor(eras,fips_1)^2
```


```{r}
library(inline)
sign <- signature(x="numeric", n="integer", d="numeric")

code <- "
  for (int i=1; i < *n; i++) {
    x[i] = x[i-1]*d[0] + x[i];
  }"

c_fn <- cfunction(sign,
                  code,
                  convention=".C"
)

weighted_sum <- function(vector, decay){
  c_fn(x=vector, n=length(vector), d=decay)$x
}
```
~~~

~~~
```{r}
# Model II

library(MCMCpack)
library(Lahman)
library(dplyr)

# Note: dplyr must be loaded after MCMCpack, otherwise some functions get overridden. You can use unloadNamespace on both packages if this happens.

B <- 10

pitchers <- Pitching %>%
  filter(yearID >= 2011) %>%
  #filter(playerID == "salech01") %>%
  filter(BFP >= 200) %>%
  mutate_if(is.factor, as.character) %>%
  mutate(BIP = BFP-(HR+BB+HBP+SO)) %>%
  filter(!is.na(HR) && !is.na(BB) && !is.na(HBP) && !is.na(SO) && !is.na(BFP)) %>%
  select(playerID, yearID, IPouts, HR, BB, HBP, SO, BIP, BFP, ERA, IPouts) %>%
  group_by(playerID, yearID) %>%
  summarise_all(funs(sum)) %>%
  group_by(playerID) %>%
  filter(min(yearID) == 2011 & max(yearID) == 2016 & length(yearID) == 6 & min(HBP)>0) %>%
  filter(yearID >=2011 & yearID <= 2016) %>%
~~~

```
  ungroup()

waic <- function(samples, obs) {
  llpd <- log( 1/B * sum(apply(samples, 1, function(theta) ddirichlet(obs, theta))))

  inner_sum <- 1/B*sum(apply(samples, 1, function(theta_j) {
      log(ddirichlet(obs, theta_j))
    }))

  pwaic <- 1/(B-1)* sum(apply(samples, 1, function(theta_b) {
    s <- log(ddirichlet(obs, theta_b)) - inner_sum
    s^2
  }))

  -2*llpd + 2*pwaic
}

get_val <- function(total_HR,total_SO,total_BB,total_HBP,total_BIP, aHR, aSO, aBB, aHBP, aBIP) {
  samples <- rdirichlet(B, c(total_HR,total_SO,total_BB,total_HBP,total_BIP))
  obs <- c(aHR, aSO, aBB, aHBP, aBIP) / sum(c(aHR, aSO, aBB, aHBP, aBIP))
  #s_waic <- waic(samples, obs)
  df_list <- apply(samples, 2, function(x) quantile(x, probs=c(.025, .975)))  %>%
    setNames(c("HR_lwr", "HR_upr", "SO_lwr", "SO_upr", "BB_lwr", "BB_upr", "HBP_lwr", "HBP_upr", "BIP_lwr", "BIP_upr")) %>%
    as.list()
  #df_list["waic"] <- s_waic
  df <- df_list %>% as.data.frame()

  df$medHR <- median(samples[,1])
  df$medSO <- median(samples[,2])
  df$medBB <- median(samples[,3])
  df$medHBP <- median(samples[,4])
  df$medBIP <- median(samples[,5])
  df
}

sample_all <- function(decay_rate=1/2) {
  result <- suppressWarnings(pitchers %>%
    select(playerID, yearID, BFP, HR, BB, HBP, SO, ERA, IPouts) %>%
    mutate(BIP = BFP-(HR+BB+HBP+SO)) %>%
    #filter(playerID=="salech01") %>%
    group_by(playerID) %>%
    mutate(start_year = min(yearID))  %>%
    mutate(total_BFP = weighted_sum(BFP, decay_rate),
           total_HR = weighted_sum(HR, decay_rate),
           total_SO = weighted_sum(SO, decay_rate),
           total_BB = weighted_sum(BB, decay_rate),
           total_HBP = weighted_sum(HBP, decay_rate),
           total_BIP = weighted_sum(BIP, decay_rate)
           ) %>%
    mutate(aHR = HR/BFP, aSO = SO/BFP, aBB = BB/BFP, aHBP = HBP/BFP, aBIP = BIP/BFP) %>%
    rowwise() %>%
    do(cbind(., get_val(.$total_HR,.$total_SO,.$total_BB,.$total_HBP,.$total_BIP,
                        .$aHR,.$aSO,.$aBB,.$aHBP,.$aBIP))) %>%
    mutate(
      inHR = aHR >= HR_lwr & aHR <= HR_upr,
      inSO = aSO >= SO_lwr & aSO <= SO_upr,
      inBB = aBB >= BB_lwr & aBB <= BB_upr,
      inHBP = aHBP >= HBP_lwr & aHBP <= HBP_upr,
      inBIP = aBIP >= BIP_lwr & aBIP <= BIP_upr
      )
    )
  result
}

#for (decay_rate in c(0.2, 0.5, 0.8))
x <- sample_all(decay_rate=1/2)
sum(x$inHR)/nrow(x)
sum(x$inSO)/nrow(x)
sum(x$inBB)/nrow(x)
sum(x$inHBP)/nrow(x)
sum(x$inBIP)/nrow(x)
#x2<- sample_all(decay_rate=.2)
#x3 <- sample_all(decay_rate=.8)
x
#x2
```

~~~~
#x3
```


```{r}
model2_in_cis <- c(0,0,0,0,0)
is_in_ci <- function(x) {
  x = c(
    between(x$aHR[6], x$HR_lwr[3], x$HR_upr[3]),
    between(x$aSO[6], x$SO_lwr[3], x$SO_upr[3]),
    between(x$aBB[6], x$BB_lwr[3], x$BB_upr[3]),
    between(x$aHBP[6], x$HBP_lwr[3], x$HBP_upr[3]),
    between(x$aBIP[6], x$BIP_lwr[3], x$BIP_upr[3])
  )
  model2_in_cis <<- model2_in_cis+x
}
x
x %>%
  group_by(playerID) %>%
  do(is_in_ci(.))


model2_in_cis/50
```
```{r}
model2_in_cis <- c(0,0,0,0,0)
is_in_ci <- function(x) {
~~~~

~~~
  x = c(
    between(x$aHR[6], x$HR_lwr[3], x$HR_upr[3]),
    between(x$aSO[6], x$SO_lwr[3], x$SO_upr[3]),
    between(x$aBB[6], x$BB_lwr[3], x$BB_upr[3]),
    between(x$aHBP[6], x$HBP_lwr[3], x$HBP_upr[3]),
    between(x$aBIP[6], x$BIP_lwr[3], x$BIP_upr[3])
  )
  model2_in_cis <<- model2_in_cis+x
}
x
x %>%
  group_by(playerID) %>%
  do(is_in_ci(.))

model2_in_cis/50
```

```{r}
eras <- x$ERA[seq(4, length(x$ERA), 6)]
fips <-(13*x$medHR + 3*(x$medHBP+x$medBB)-2*x$medSO)*x$BFP/(x$IPouts/3)
fips <-fips[seq(3, length(x$ERA), 6)]
plot(fips, eras, main="Model 2 2014-16 ERA Regression", ylab="2014-16 ERA", xlab="Median Posterior FIP")
fit<- lsfit(fips, eras)
abline(fit$coefficients[1], fit$coefficients[2], col="blue")
cor(eras,fips)^2
~~~

~~~
fips_actual <- (13*x$HR + 3*(x$HBP+x$BB)-2*x$SO)/(x$IPouts/3)

fips_actual <-fips_actual[seq(3, length(x$ERA), 6)]

plot(fips_actual, eras, main="Industry FIP 2014-16 ERA Regression", ylab="2014-16 ERA", xlab="FIP")

fit2<- lsfit(fips_actual, eras)

abline(fit2$coefficients[1], fit2$coefficients[2], col="blue")

cor(eras,fips_actual)^2


```
~~~