2012-04-17

Hitters vs. Pitchers: A Comparison of Fantasy Baseball Player Performances Using Hierarchical Bayesian Models

Scott D. Huddleston
Brigham Young University - Provo

Hitters vs. Pitchers: A Comparison of Fantasy Baseball Player Performances Using Hierarchical Bayesian Models

Scott D. Huddleston

A selected project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Gilbert W. Fellingham, Chair
C. Shane Reese
E. Shannon Neeley

Department of Statistics

Brigham Young University

June 2012

ABSTRACT

Hitters vs. Pitchers: A Comparison of Fantasy Baseball Player Performances Using Hierarchical Bayesian Models

Scott D. Huddleston
Department of Statistics, BYU
Master of Science

In recent years, fantasy baseball has seen an explosion in popularity. Major League Baseball, with its long, storied history and the enormous quantity of data available, naturally lends itself to the modern-day recreational activity known as fantasy baseball. Fantasy baseball is a game in which participants manage an imaginary roster of real players and compete against one another using those players' real-life statistics to score points. Early forms of fantasy baseball began in the early 1960s, but beginning in the 1990s, the sport was revolutionized due to the advent of powerful computers and the Internet.

The data used in this project come from an actual fantasy baseball league which uses a head-to-head, points-based scoring system. The data consist of the weekly point totals that were accumulated over the first three-fourths of the 2011 regular season by the top 110 hitters and top 70 pitchers in Major League Baseball. The purpose of this project is analyze the relative value of pitchers versus hitters in this league using hierarchical Bayesian models. Three models will be compared, one which differentiates between hitters and pitchers, another which also differentiates between starting pitchers and relief pitchers, and a third which makes no distinction whatsoever between hitters and pitchers. The models will be compared using the deviance information criterion (DIC). The best model will then be used to predict weekly point totals for the last fourth of the 2011 season. Posterior predictive densities will be compared to actual weekly scores.

Keywords: fantasy baseball, hierarchical Bayesian models, MCMC

ACKNOWLEDGMENTS

I'd like to thank my wonderful wife, Nikki, who encouraged me to go back to school when I would have never imagined it a possibility. I would also like to thank the professors at BYU who have endeavored to show me the way through their diligent instruction. Lastly, I would like to especially thank Dr. Fellingham for his patient guidance throughout this project, and for sharing the same passion that I have for sports.

CONTENTS

CHAPTER 1

INTRODUCTION

In recent years, fantasy baseball has seen an explosion in popularity. With its long, 162 game season and the enormous quantity of data available, Major League Baseball, even more than other sports, seems to be a natural fit for this type of recreational activity. Fantasy baseball is a game in which participants manage an imaginary roster of real Major League baseball players. The participants compete against one another using those players' real life statistics to score points. Early forms of fantasy baseball began in the early 1960s, but beginning in the 1990s, the sport was revolutionized due to the advent of powerful computers and the Internet. This allowed scoring to be done entirely by computer and allowed leagues to develop their own scoring systems. In this way, fantasy baseball has become a sort of real-time simulation of baseball and allowed many fans to develop a more sophisticated understanding of how the real-world game works. According to statistics published in 2009, nearly 11 million people now play fantasy baseball (Greenburg 2009).

The landmark development in fantasy baseball occurred with the introduction of Rotisserie League Baseball in 1980, named after the New York City restaurant La Rotisserie Française, where its founders met for lunch and first played the game (Starr 2008). Rotisserie league baseball, nicknamed *roto*, uses traditional statistics to determine the winning team in a head-to-head matchup. For each player's team, statistics are accrued for each of ten categories—five hitting categories (e.g., RBI and home runs) and five pitching categories (e.g., wins and earned run average). The winning team is determined by which team is superior in the most categories over the course of a scoring period.

The dataset used in this project, by contrast, comes from a head-to-head, points-based scoring system. Each team "owner" in the league is allowed to have a twenty-five

player roster, with eighteen of those players in his active lineup each week. Those eighteen players fill specific positions consisting of eleven hitters and seven pitchers. Of those eleven hitters, each of the standard baseball positions must be filled (i.e. one first baseman, one second baseman, three outfielders, etc.). Since catchers for most teams typically do not play every day, each team may start two catchers. In addition, there are two additional spots—utility and designated hitter—which may be filled by a hitter of any position. Of the seven pitchers, an owner may start either four starting pitchers and three relief pitchers or five starting pitchers and two relief pitchers.

Each fantasy scoring period lasts a week, and teams accumulate points for each of their starting hitters and pitchers for the duration of the week. At the conclusion of the week, the team with the most points wins. The four teams with the best records at the end of the fantasy regular season (approximately two weeks before the end of the real baseball season) make the playoffs, and the playoffs are played out over two more weeks in the usual fashion. The team left standing at the end is the winner of the league.

In each of the last two years, we successfully won the fantasy league in which we participate. Prior to that, we had struggled for more than five years, failing to make the playoffs even once. We became frustrated and decided to change our strategy to focus on having the best pitching in the league. We would aim to have great starting pitchers on our team along with the best possible relief pitchers, and only then would we concentrate on filling out our hitters. This approach apparently paid off, as we won the league for the first time ever in 2010 and were able to successfully defend our title, winning again in 2011. This project will examine the question: what is the value of pitchers versus hitters in this type of points-based league? Was our strategy correct in focusing on drafting pitchers over hitters? And would this strategy continue to pay off in the future? We will attempt to answer these question using hierachical Bayesian models.

The data to be modeled in this project consist of the weekly point totals that were accumulated over the first three-fourths of the 2011 regular season by the top 110 hitters and

top 70 pitchers in Major League Baseball. Three models will be constructed and compared, one which differentiates only between hitters and pitchers, one which also differentiates between starting pitchers and relief pitchers, and one which does not differentiate whatsoever between hitters and pitchers. The model fits will be tested using Bayesian Chi-square goodness-of-fit tests and then will be compared using the deviance information criterion (DIC). The best model will then be used to predict weekly totals for the final fourth of the 2011 regular season. Posterior predictive densities will be compared to actual weekly scores. Our approach to modeling this data is framed within hierarchical Bayesian models using Markov chain Monte Carlo (MCMC) as the computational tool. We will present these methodologies in Chapter 2 and discuss the theoretical underpinnings of each.

CHAPTER 2

LITERATURE REVIEW

The literature review is divided into three parts. Section 2.1 discusses Bayesian methods and hierarchical models. Section 2.2 discusses Markov chain Monte Carlo methods and two techniques used to draw samples from the posterior distribution. Section 2.3 discusses Model Selection.

2.1 Bayesian Methods

Theory of Bayesian Methods

In the Bayesian paradigm, information brought by the data is combined with prior information that is specified in a *prior distribution* and summarized in a probability distribution called the *posterior distribution* (Robert and Casella 2004). The Bayesian approach begins exactly as a traditional frequentist analysis does, with a *sampling model* for the observed data $\mathbf{y} = (y_1, \ldots, y_n)$ given a vector of unknown parameters $\boldsymbol{\theta}$. This sampling model is typically given in the form of a probability distribution $f(\mathbf{y}|\boldsymbol{\theta})$. When viewed as a function of $\boldsymbol{\theta}$ instead of $\mathbf{y}$, this distribution is usually called the *likelihood*, and sometimes written as $L(\boldsymbol{\theta}|\mathbf{y})$ to emphasize our mental reversal of the roles of $\boldsymbol{\theta}$ and $\mathbf{y}$ (Carlin and Louis 2009).

In the Bayesian approach, instead of supposing that $\boldsymbol{\theta}$ is a fixed parameter, we think of it as a *random* quantity as well. This approach can be operationalized by adopting a probability distribution for $\boldsymbol{\theta}$, the vector of unknown parameters, that summarizes any information we have about it not related to that provided by the data $\mathbf{y}$, called the *prior* distribution (or simply the *prior*). Combining the likelihood from the data and the prior distribution, Bayes' Theorem can be used to create the posterior distribution of $\boldsymbol{\theta}$.

In order to handle continuous variables, Bayes' Theorem involves probability calculus, with which some readers may be less comfortable. We thus first present an alternate, discrete version of Bayes' Theorem. In this simpler formulation, we are given an event of interest, A, and a collection of events, B_j, $j = 1, \ldots, J$, that are mutually exclusive and exhaustive (that is, exactly one of them must occur). Given the probabilities of each of these events $P(B_j)$, as well as the conditional probabilities $P(A|B_j)$, from fundamental rules of probability, we have

$$\begin{aligned} P(B_j|A) &= \frac{P(A, B_j)}{P(A)} = \frac{P(A, B_j)}{\sum_{j=1}^{J} P(A, B_j)} \\ &= \frac{P(A|B_j)P(B_j)}{\sum_{j=1}^{J} P(A|B_j)P(B_j)}, \end{aligned} \tag{2.1}$$

where $P(A, B_j)$ indicates the *joint* event where both A and B_j occur; many textbooks write $P(A \cap B_j)$ for $P(A, B_j)$. All four expressions in (2.1) are just discrete finite versions of the corresponding expressions in (2.2), with the B_j playing the role of the parameters θ and A playing the role of the data $\mathbf{y}$ (Carlin and Louis 2009).

Bayes' Theorem first appeared (in a somewhat simplified form) in *An Essay towards solving a Problem in the Doctrine of Chances* (Bayes and Price 1763). Bayes' Theorem is given by

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\ d\boldsymbol{\theta}}, \tag{2.2}$$

where $p(\boldsymbol{\theta}|\mathbf{y})$ is the posterior distribution, $f(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood, $\pi(\boldsymbol{\theta})$ is the prior distribution, $\mathbf{y}$ is the data vector, and $\boldsymbol{\theta}$ is the vector of parameters. Just as the likelihood has parameters $\boldsymbol{\theta}$, the prior may have parameters $\boldsymbol{\eta}$; these are often referred to as *hyperparameters* in order to distinguish them from the likelihood parameters $\boldsymbol{\theta}$. For the moment, we assume that the hyperparameters $\boldsymbol{\eta}$ are known, and thus write the prior as $\pi(\boldsymbol{\theta}) \equiv \pi(\boldsymbol{\theta}|\boldsymbol{\eta})$.

Notice the contribution of both the experimental data (in the form of the likelihood f) and prior opinion (in the form of the prior π). The posterior may be thought of as the product of the likelihood and the prior, renormalized so that it integrates to 1 (and thus is itself a valid probability distribution). The denominator of (2.2) is referred to as the normalizing constant. Often the greatest challenge in evaluating the posterior lies in performing the integral in the denominator. Notice we are writing this as a single integral, but in fact it is a *multiple* integral, having dimension equal to the number of parameters in the $\boldsymbol{\theta}$ vector (Carlin and Louis 2009).

Conjugacy is formally defined as follows: if $\mathcal{F}$ is a class of sampling distributions $f(\mathbf{y}|\boldsymbol{\theta})$, and $\mathcal{P}$ is a class of prior distributions for $\boldsymbol{\theta}$, then the class $\mathcal{P}$ is *conjugate* for $\mathcal{F}$ if

$$p(\boldsymbol{\theta}|\mathbf{y}) \in \mathcal{P} \text{ for all } f(\cdot|\boldsymbol{\theta}) \in \mathcal{F} \text{ and } \pi(\cdot) \in \mathcal{P}. \tag{2.3}$$

This definition is formally vague, since if we choose $\mathcal{P}$ as the class of all distributions, then $\mathcal{P}$ is always conjugate no matter what class of sampling distributions is used. We are most interested in *natural* conjugate prior families, which arise by taking $\mathcal{P}$ to be the set of all densities having the same functional form as the likelihood (Gelman et al. 1995).

The basic justification for the use of conjugate prior distributions is similar to that for using standard models (such as binomial and normal) for the likelihood: it is easy to understand the results, which can often be put in analytical form, they are often a good approximation, and they simplify computations.

Let us consider the simple binomial model, in which the aim is to estimate an unknown population proportion from the results of a sequence of Bernoulli trials; that is, data $y_1, \ldots, y_n$, each of which is either 0 or 1. The binomial distibution provides a natural model for data that arise from a sequence of n trials or draws from a large population where each trial gives rise to one of two possible outcomes, conventionally labeled "success" and "failure." The usual assumption is that the n values y_i may be regarded as *exchangeable*, meaning that the joint probability density $p(y_1, \ldots, y_n)$ should be invariant to permutations of the indexes (Gelman et al. 1995). Because of the exchangeability, the data can be summarized by

the total number of success in the n trials, which we denote here by y. We let the parameter θ represent the proportion of successes in the population or, equivalently, the probability of success in each trial (Gelman et al. 1995). The binomial sampling model states that

$$p(y|\theta) = \text{Bin}(y|n, \theta) = \binom{n}{y}\theta^y(1-\theta)^{n-y}. \tag{2.4}$$

We now use the binomial distribution to demonstrate the concept of conjugacy. Considered as a function of θ, the likelihood is of the form

$$p(y|\theta) \propto \theta^y(1-\theta)^{n-y}. \tag{2.5}$$

If the prior density is of the same form, then the posterior density will also be of this form. We will parameterize such a prior density as

$$p(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}, \tag{2.6}$$

which is a beta distribution with parameters α and β: $\theta \sim \text{Beta}(\alpha,\beta)$. Comparing $p(\theta)$ and $p(y|\theta)$ suggests that this prior density is equivalent to $\alpha - 1$ prior successes and $\beta - 1$ prior failures. For now, assume that we can select reasonable values α and β. The posterior density for θ is

$$\begin{aligned} p(\theta|y) &\propto \theta^y(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{y+\alpha-1}(1-\theta)^{n-y+\beta-1} \\ \Rightarrow p(\theta|y) &= \text{Beta}(\theta|\alpha+y, \beta+n-y). \end{aligned} \tag{2.7}$$

From this example, we see that the beta prior distribution is a *conjugate family* for the binomial likelihood. The conjugate family is mathematically convenient in that the posterior distribution follows a known parametric form. Of course, if information is available that contradicts the conjugate parametric family, it may be necessary to use a more realistic, if inconvenient, prior distribution (Gelman et al. 1995).

Conjugate prior distributions have the practical advantage, in addition to computational convenience, of being interpretable as additional data, as we have seen for the binomial

example and which can also be seen for the normal distribution and other standard models (Gelman et al. 1995).

It turns out that many of the common statistical distributions have a similar form. A density is from the *one-parameter exponential family* if it can be put into the form

$$p(y|\theta) = g(y)h(\theta)\exp\{t(y)\psi(\theta)\}, \tag{2.8}$$

or equivalently, if the likelihood of n independent observations $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ from this distribution is

$$f(\theta|\mathbf{y}) \propto h(\theta)^n \exp\left\{ \sum t(y_i)\psi(\theta) \right\}. \tag{2.9}$$

It follows immediately from Neyman's factorization theorem that $\sum t(y_i)$ is sufficient for θ given $\mathbf{y}$ (Lee 1997).

Probability distributions that belong to an exponential family have natural conjugate prior distributions. The quantity $\sum t(y_i)$ is said to be a *sufficient statistic* for θ because the likelihood for θ depends on the data $\mathbf{y}$ only through the value of $\sum t(y_i)$. Sufficient statistics are useful in algebraic manipulations of likelihoods and posterior distributions. It has been shown that, in general, the exponential families are the only classes of distributions that have natural conjugate prior distributions, since, apart from certain irregular cases, the only distributions having a fixed number of sufficient statistics for all n are of the exponential type (Gelman et al. 1995).

Hierarchical Bayesian Models

The basic Bayesian model considered in equation (2.2) has two stages, one for $f(\mathbf{y}|\boldsymbol{\theta})$, the likelihood of the data $\mathbf{y}$ given the parameters $\boldsymbol{\theta}$, and one for $\pi(\boldsymbol{\theta}|\boldsymbol{\eta})$, the prior distribution of the model parameters $\boldsymbol{\theta}$ given a vector of hyperparameters $\boldsymbol{\eta}$. In many cases, however, we may need to use a model with more than two stages. Suppose we were unsure as to the proper value for the $\boldsymbol{\eta}$ vector. The proper Bayesian solution would be to quantify this uncertainty in a second-stage prior distribution (sometimes called a *hyperprior*). Denoting

this distribution by $h(\boldsymbol{\eta})$, the desired posterior for $\boldsymbol{\theta}$ is now obtained by also marginalizing over $\boldsymbol{\theta}$,

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{y}) &= \frac{p(\mathbf{y},\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{\int p(\mathbf{y},\boldsymbol{\theta},\boldsymbol{\eta})\ d\boldsymbol{\eta}}{\int\int p(\mathbf{y},\mathbf{u},\boldsymbol{\eta})\ d\boldsymbol{\eta}\ d\mathbf{u}} \\ &= \frac{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\eta})h(\boldsymbol{\eta})\ d\boldsymbol{\eta}}{\int\int f(\mathbf{y}|\mathbf{u})\pi(\mathbf{u}|\boldsymbol{\eta})h(\boldsymbol{\eta})\ d\boldsymbol{\eta}\ d\mathbf{u}}. \end{aligned} \tag{2.10}$$

Of course, in principle, there is no reason why the hyperprior for $\boldsymbol{\eta}$ cannot itself depend on a collection of unknown parameters $\boldsymbol{\lambda}$, resulting in a generalization of (2.10) featuring a second-stage prior $h(\boldsymbol{\eta}|\boldsymbol{\lambda})$ and a third-stage prior $g(\boldsymbol{\lambda})$. This enterprise of specifying a model over several levels is called *hierarchical modeling*, with each new distribution forming a new level in the hierarchy. The proper number of levels varies with the problem (Carlin and Louis 2009).

This concept of a "hierarchical model" can be extended to a hierarchy of l levels, where the joint distribution of the data and the parameters is given by

$$f(\boldsymbol{y}|\boldsymbol{\theta}_1)\pi_1(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)\pi_2(\boldsymbol{\theta}_2|\boldsymbol{\theta}_3)\cdots\pi_l(\boldsymbol{\theta}_l|\boldsymbol{\lambda}), \tag{2.11}$$

and we typically seek the marginal posterior of the first stage parameters, $p(\boldsymbol{\theta}_1|\boldsymbol{y})$. In the case where f and π_i are normal distributions with known variance matrices, this marginal posterior is readily available. However, more complicated settings generally require Monte Carlo methods for their solution (Carlin and Louis 2009).

2.2 MCMC Methods

In some simple Bayesian models, it is possible to draw directly from the posterior distribution. This typically happens when the prior distributions are conjugate. Usually, however, Bayesian models yield complex posteriors which cannot be sampled from directly. In such cases, it is now standard Bayesian practice to turn to *Markov chain Monte Carlo (MCMC)* methods. These methods operate by sequentially sampling parameter values from a Markov

chain whose stationary distribution is exactly the desired joint posterior distribution of interest. The great increase in generality of these methods comes at the price of requiring an assessment of *convergence* of the Markov chain to its stationary distribution, something that can sometimes be shown theoretically, but more typically is judged using plots or numerical summaries of the sampled output from the chain. The majority of Bayesian MCMC computing is accomplished using one of two basic algorithms, the *Gibbs sampler* and the *Metropolis-Hastings (M-H)* algorithm (Carlin and Louis 2009).

Markov Chains

We introduce the concept of a *Markov chain*. Consider a stochastic process $\{X_n, n = 0, 1, 2, \ldots\}$ that takes on a finite or countable number of possible values. If $X_n = i$, then the process is said to be in state i at time n. We suppose that whenever the process is in state i there is a fixed probability P_{ij} that it will next be in state j (Ross 1996). That is, we suppose that

$$P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_1 = i_1, X_0 = i_0\} = P_{ij} \tag{2.12}$$

for all states $i_0, i_1, \ldots, i_{n-1}, i, j$ and all $n \geq 0$. Such a stochastic process is known as a *Markov chain*. Equation (2.12) may be interpreted as stating that, for a Markov chain, the conditional distribution of any future state X_{n+1}, given the past states $X_0, X_1, \ldots, X_{n-1}$ and the present state X_n, is independent of the past states and depends only on the present state. This is called the *Markovian* property. The value P_{ij} represents the probability that the process will, when in state i, next make a transition to state j (Ross 1996). Since probabilities are nonnegative and since the process must make a transition into some state, we have that

$$P_{ij} \geq 0, \quad i, j \geq 0; \quad \sum_{j=0}^{\infty} P_{ij} = 1, \quad i = 0, 1, \ldots. \tag{2.13}$$

Let P denote the matrix of one-step transition probabilities P_{ij}, so that

$$P = \begin{pmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ \vdots & & & \\ P_{i0} & P_{i1} & P_{i2} & \cdots \\ \vdots & \vdots & & \vdots \end{pmatrix}.$$

We now define the n-step transition probabilities P_{ij}^n to be the probability that a process in state i will be in state j after n additional transitions (Ross 1996). That is,

$$P_{ij}^n = P\{X_{n+m} = j | X_M = i\}, \qquad n \geq 0, \qquad i, j \geq 0. \tag{2.14}$$

State j is said to be accessible from state i if for some $n \geq 0, P_{ij}^n > 0$. Two states i and j accessible to each other are said to *communicate*, and we write $i \leftrightarrow j$. Two states that communicate are said to be in the same *class*; we say that the Markov chain is *irreducible* if there is only one class—that is, if all states communicate with each other.

State i is said to have period d if $P_{ij}^n = 0$ whenever n is not divisible by d and d is the greatest integer with this property. A state with period 1 is said to be *aperiodic* (Ross 1996).

Markov Chain Simulation

The idea of Markov chain simulation is to simulate a random walk in the space of $\boldsymbol{\theta}$ which converges to a stationary distribution that is the joint posterior distribution, $p(\boldsymbol{\theta}|\mathbf{y})$. The key to Markov chain simulation is to create a Markov process whose stationary distribution is a specified $p(\boldsymbol{\theta}|\mathbf{y})$ and run the simulation long enough that the distribution of the current draws is close enough to the stationary distribution. It turns out that, given $p(\boldsymbol{\theta}|\mathbf{y})$, or an unnormalized density, $q(\boldsymbol{\theta}|\mathbf{y})$, a variety of Markov chains with the desired property can be constructed. Once the simulation algorithm has been implemented, we should iterate until convergence has been approximated—or, if convergence is painfully slow, the algorithm should be altered (Gelman et al. 1995).

Gibbs Sampler

A particular Markov chain algorithm that has been found useful in many multidimensional problems is the *Gibbs sampler*, which is defined in terms of subvectors of $\boldsymbol{\theta}$. Suppose our model features k parameters, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$. Each iteration of the Gibbs sampler cycles through the subvectors of $\boldsymbol{\theta}$, drawing each subset conditional on the value of all the others. There are thus k steps in iteration t. At each iteration t, an ordering of the k subvectors is chosen and, in turn, each $\theta_j^{(t)}$ is sampled from the conditional distribution given all the other components of $\boldsymbol{\theta}$:

$$p(\theta_j^{(t)} | \theta_{-j}^{(t-1)}, \mathbf{y}), \tag{2.15}$$

where $\theta_{-j}^{(t-1)}$ represents all the components of $\boldsymbol{\theta}$ except θ_j, at their current values:

$$\theta_{-j}^{(t-1)} = (\theta_1^{(t)}, \ldots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \ldots, \theta_k^{(t-1)}). \tag{2.16}$$

Thus, each subvector θ_j is updated conditional on the latest value of θ for the other components, which are the iteration t values of $\boldsymbol{\theta}$ for the other components, which are the iteration t values for the components already updated and the iteration $t-1$ values for the others (Gelman et al. 1995).

To implement the Gibbs sampler, we must assume that samples can be generated from each of the *full* (or *complete*) *conditional* distributions $\{p(\theta_i | \boldsymbol{\theta}_{j \neq i}, \mathbf{y}), i = 1, \ldots, k\}$ in the model. Such samples might be available directly (say, if the full conditionals were familiar forms, like normals and gammas) or indirectly (say, via a rejection sampling approach). In either case, the collection of full conditional distributions uniquely determines the joint posterior distribution, $p(\boldsymbol{\theta}|\mathbf{y})$, and hence all marginal posterior distributions $p(\theta_i|\mathbf{y}), i = 1, \ldots, k$.

Given an arbitrary set of starting values $\{\theta_2^{(0)}, \ldots, \theta_k^{(0)}\}$, the algorithm proceeds as follows (Carlin and Louis 2009):

Gibbs Sampler: For $(t = 1, \ldots, T)$, repeat:

1. Draw $\theta_1^{(t)}$ from $p\big(\theta_1|\theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_k^{(t-1)}, \mathbf{y}\big)\big)$.

2. Draw $\theta_2^{(t)}$ from $p\big(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_k^{(t-1)}, \mathbf{y}\big)\big)$.

   $\vdots$

k. Draw $\theta_k^{(t)}$ from $p\big(\theta_k|\theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_k^{(t)}, \mathbf{y}\big)\big)$.

Provided the Markov chain is aperiodic and irreducible, the k-tuple obtained at iteration t, $(\theta_1^{(t)}, \ldots, \theta_k^{(t)})$, converges in distribution to a draw from the true joint posterior distribution $p(\theta_1, \ldots, \theta_k|\mathbf{y})$ (Ross 1996). This means that for t sufficiently large (say, bigger than t_0), $\{\boldsymbol{\theta}^{(t)}, t = t_0 + 1, \ldots, T\}$ is a (correlated) sample from the true posterior, from which any posterior quantities of interest may be estimated (Carlin and Louis 2009).

Metropolis Algorithm

The Gibbs sampler is easy to understand and implement, but requires the ability to readily sample from each of the full conditional distributions $p(\theta_i|\boldsymbol{\theta}_{l\neq i}, \mathbf{y})$. Unfortunately, when the prior $\pi(\boldsymbol{\theta})$ and the likelihood $f(\mathbf{y}|\boldsymbol{\theta})$ are not a conjugate pair, one or more of these full conditionals may not be available in closed form. Even in this setting, however, $p(\theta_i|\boldsymbol{\theta}_{l\neq i}, \mathbf{y})$ *will be* available up to a proportionality constant, because it is proportional to the portion of $f(\mathbf{y}|\boldsymbol{\theta})$ that involves θ_i (Carlin and Louis 2009).

The *Metropolis algorithm* (and its *Metropolis-Hastings algorithm* extension) is a rejection algorithm that attacks precisely this problem, since it requires only a function proportional to the distribution to be sampled, at the cost of requiring a rejection step from a particular *candidate* density.

Like the Gibbs sampler, the Metropolis algorithm was not developed by statisticians for the purpose of estimating posterior distributions. Instead, the development was by nuclear physicists working on the Manhattan Project in the 1940s seeking to understand the particle movement theory underlying the first atomic bomb. One of the coauthors on the

original Metropolis et al. (1953) paper was Edward Teller, who is often referred to as "the father of the hydrogen bomb."

Suppose we wish to generate from a joint posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}) \propto h(\boldsymbol{\theta}) \equiv f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. We begin by specifying a *candidate* (or *proposal*) density $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ that is a valid density function for every possible value of the conditioning variable $\boldsymbol{\theta}^{(t-1)}$, and satisfies $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)$ (i.e., q is *symmetric* in its arguments). Given a starting value $\boldsymbol{\theta}^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows (Carlin and Louis 2009):

Metropolis Algorithm: For $(t = 1, \ldots, T)$, repeat:

1. Draw $\boldsymbol{\theta}^*$ from $q(\cdot|\boldsymbol{\theta}^{(t-1)})$.

2. Compute the ratio $r = h(\boldsymbol{\theta}^*)/h(\boldsymbol{\theta}^{(t-1)}) = \exp[\log\, h(\boldsymbol{\theta}^*) - \log\, h(\boldsymbol{\theta}^{(t-1)})]$.

3. If $r \geq 1$, set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$;

   If $r < 1$, set $\boldsymbol{\theta}^{(t)} = \begin{cases} \boldsymbol{\theta}^* \text{ with probability r} \\ \boldsymbol{\theta}^{(t-1)} \text{ with probability 1 - r} \end{cases}$.

Then under similarly mild conditions as those supporting the Gibbs sampler, a draw $\boldsymbol{\theta}^{(t)}$ converges in distribution to a draw from the true posterior density $p(\boldsymbol{\theta}|\mathbf{y})$ as $t \to \infty$. Note that the shape of h is not used in the candidate generation, unlike the Gibbs sampler, which draws from full conditionals derived from h (Carlin and Louis 2009).

The Metropolis algorithm affords substantial flexibility through the selection of the candidate density q, but this flexibility can be a blessing and a curse. While theoretically we are free to pick almost anything, in practice only a "good" choice will result in sufficiently many candidate acceptances. One might imagine an optimal choice of q would produce an empirical acceptance ratio of 1, the same as the Gibbs sampler (and with no apparent "waste" of candidates). However, accepting all or nearly all of the candidates is often the result of an overly narrow candidate density. Such a density will "baby-step" around the parameter space, leading to high acceptance, but also high autocorrelation in the sampled

chain. An overly wide candidate density will also struggle, proposing leaps to places far from the bulk of the posterior's support, leading to high rejection and, again, high autocorrelation. For the multivariate target density having product form

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{K} g(\theta_i) \tag{2.17}$$

for some one-dimensional density g, the optimal acceptance rate approaches 23.4% as the dimension of the parameter space k goes to infinity (Gelman et al. 1997). This suggests that in high dimensions, it is worth risking an occasional sticking point in the algorithm in order to gain the benefit of an occasional large jump across the parameter space (Carlin and Louis 2009).

In practice, the Metropolis algorithm is often found as a substep in a larger Gibbs sampling algorithm, used to generate from awkward full conditionals. Such hybrid Gibbs-Metropolis applications are sometimes known as "Metropolis within Gibbs" (Carlin and Louis 2009).

Metropolis-Hastings Algorithm

We now present the important generalization of the Metropolis algorithm devised by Hastings (1970). In this variant, we drop the requirement that q be symmetric in its arguments, which is often useful for bounded parameter spaces (say, $\theta > 0$).

Metropolis-Hastings Algorithm: When using a candidate density q for which $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) \neq q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)$, replace the acceptance ratio r in Step 2 of the Metropolis algorithm previously given by

$$r = \frac{h(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}{h(\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}. \tag{2.18}$$

Then again under mild conditions, a draw $\boldsymbol{\theta}^{(t)}$ converges in distribution to a draw from the true posterior density $p(\boldsymbol{\theta}|\mathbf{y})$ as $t \to \infty$ (Carlin and Louis 2009).

2.3 Model Selection

Model selection is the task of selecting a statistical model from a set of candidate models given the data. The Bayesian statistician has several methods available with which to do this. We will focus on the Bayesian χ^2 test for goodness-of-fit and the Deviance Information Criterion (DIC).

Bayesian χ^2 Test for Goodness-of-Fit

To begin, let $y_1, \ldots, y_n (= \mathbf{y})$ denote scalar-valued, continuous, identically distributed, conditionally independent observations drawn from probability density function $f(y|\boldsymbol{\theta})$. Let $\tilde{\boldsymbol{\theta}}$ denote a value of $\boldsymbol{\theta}$ sampled from the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$.

To construct the Bayesian goodness-of-fit statistic, choose quantiles $0 \equiv a_0 < a_1 < \cdots < a_{K-1} \equiv 1$, with $p_K = a_k - a_{k-1}, k = 1, \ldots, K$. Define $\boldsymbol{z}_j(\tilde{\boldsymbol{\theta}})$ to be a vector of length K whose kth element is 0 unless

$$F(y_j|\tilde{\boldsymbol{\theta}}) \in (a_{k-1}, a_k], \tag{2.19}$$

in which case it is 1. Next, define

$$\mathbf{m}(\tilde{\boldsymbol{\theta}}) = \sum_{j=1}^{n} \mathbf{z}_j(\tilde{\boldsymbol{\theta}}). \tag{2.20}$$

It follows that the kth component of $\mathbf{m}(\tilde{\boldsymbol{\theta}})$, $m_k(\tilde{\boldsymbol{\theta}})$, represents the number of observations that fell into the kth bin, where bins are determined by the quantiles of the inverse distribution function evaluated at $\tilde{\boldsymbol{\theta}}$. Finally, define

$$R^B(\tilde{\boldsymbol{\theta}}) = \sum_{k=1}^{K} \left[\frac{(m_k(\tilde{\boldsymbol{\theta}}) - np_k)}{\sqrt{np_k}} \right]^2. \tag{2.21}$$

Assuming that regularity conditions apply, R^B converges to a χ^2 distribution with $K - 1$ degrees of freedom as $n \to \infty$.

Now suppose we want to determine how well each of our models fits the data. For each model, we can compute the proportion of R^B values drawn from the posterior distribution

that exceeds the specified critical value from their nominal χ^2_{K-1} distribution. For a given data vector and probability model, such a procedure might lead to a statement that, say, 90% of R^B values generated from the posterior distribution exceeded the 95th percentile of the reference χ^2 distribution. Large proportions of R^B values indicate a lack of fit for the model in question (Johnson 2004).

Deviance Information Criterion (DIC)

Appropriate statistical selection of the best among a collection of hierarchical models can be problematic, due to the ambiguity in the "size" of such models arising from the posterior shrinkage of their random effects towards a common value. To address this problem, Spiegelhalter et al. (2002) suggest a generalization of the Akaike information criterion (AIC) that is based on the posterior distribution of the *deviance* statistic,

$$D(\boldsymbol{\theta}) = -2 \log f(\mathbf{y}|\boldsymbol{\theta}) + 2 \log h(\mathbf{y}), \tag{2.22}$$

where $f(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood function for the observed data vector $\mathbf{y}$ given the parameter vector $\boldsymbol{\theta}$, and $h(\mathbf{y})$ is some standardizing function of the data alone (which thus has no impact on model selection). In this approach, the fit of a model is summarized by the posterior expectation of the deviance, $\overline{D} = E_{\theta|y}[D]$, while the *complexity* of a model is captured by the effective number of parameters p_D, which is typically less than the total number of model parameters due to the borrowing of strength across individual-level parameters in hierarchical models. It can be shown that a reasonable definition of p_D is the expected deviance minus the deviance evaluated at the posterior expectations,

$$p_D = E_{\theta|y}[D] - D(E_{\theta|y}[\boldsymbol{\theta}]) = \overline{D} - D(\bar{\boldsymbol{\theta}}). \tag{2.23}$$

The *Deviance Information Criterion (DIC)* is then defined as

$$DIC = \overline{D} + p_D = 2\overline{D} - D(\bar{\boldsymbol{\theta}}), \tag{2.24}$$

with smaller values of DIC indicating a better-fitting model (Carlin and Louis 2009). Note that DIC is scale-free; the choice of standardizing function $h(\mathbf{y})$ in (2.21) is arbitrary. Thus values of DIC have no intrinsic meaning; as with AIC, only *differences* in DIC across models are meaningful.

Bayesian Estimation and Prediction

Observe that equation (2.2) may be expressed in the convenient shorthand

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \tag{2.25}$$

or in words, "the posterior is proportional to the likelihood times the prior." The likelihood may be multiplied by any constant (or even any function of $\mathbf{y}$ alone) without altering the posterior.

Bayes' Theorem may also be used *sequentially*: suppose we have two independently collected samples of data, $\mathbf{y}_1$ and $\mathbf{y}_2$. Then

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{y}_1, \mathbf{y}_2) &\propto f(\mathbf{y}_1, \mathbf{y}_2|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\ &= f_2(\mathbf{y}_2|\boldsymbol{\theta}) f_1(\mathbf{y}_1|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\ &\propto f_2(\mathbf{y}_2|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{y}_1). \end{aligned} \tag{2.26}$$

That is, we can obtain the posterior for the full dataset $(\mathbf{y}_1, \mathbf{y}_2)$ by first finding $p(\boldsymbol{\theta}|\mathbf{y}_1)$ and then treating it as a prior for the second portion of the data $\mathbf{y}_2$.

Geisser (1993) and other authors have argued that concentrating on inference for the model parameters is misguided, since $\boldsymbol{\theta}$ is merely an unobservable, theoretical quantity. Switching to a different model for the data may result in an entirely different $\boldsymbol{\theta}$ vector. Moreover, even a perfect understanding of the model does not constitute a direct attack on the problem of *predicting* how the system under study will behave in the future—often the real goal of a statistical analysis. To this end, suppose the y_{n+1} is a future observation,

independent of $\mathbf{y}$ given the underlying $\boldsymbol{\theta}$. Then the *predictive distribution* for y_{n+1} is given by

$$\begin{aligned} p(y_{n+1}|\mathbf{y}) &= \int p(y_{n+1}, \boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \\ &= \int f(y_{n+1}|\boldsymbol{\theta}, \mathbf{y}) p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \\ &= \int f(y_{n+1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}, \end{aligned} \tag{2.27}$$

the last equality holding thanks to the conditional independence of y_{n+1} and $\mathbf{y}$ given the parameters $\boldsymbol{\theta}$ (i.e., the usual independence of observations often assumed in the likelihood model). The predictive distribution summarizes the information concerning the likely value of a new observation, given the likelihood, the prior, and the data we have observed so far (Carlin and Louis 2009).

CHAPTER 3

MODEL COMPARISON

3.1 The Data

The data used in this project come from an actual fantasy baseball league managed on the website cbssportsline.com for the 2011 Major League Baseball season. The format is weekly, head-to-head matchups, and the scoring system is points-based. Points are allocated as shown in Table 3.1. Data for each of the top 110 hitters and top 70 pitchers were used. Given that there were ten teams in the league, each with a weekly starting lineup of eleven hitters and seven pitchers, the data chosen were intended to represent the "theoretical" starting lineups of all ten teams. The raw dataset, therefore, consists of twenty-five weeks of scoring data for 180 professional baseball players. The data were adjusted to account for three non-standard scoring periods: the first scoring period of the season, an eleven-day "week" resulting from opening day falling on a Thursday; the scoring period in the middle of the season, shortened by three days because of the All-Star break; and the final scoring period of the season, a ten-day week resulting from the season ending on a Wednesday. Data for the four-day scoring period occurring All-Star week were thrown out of the analysis, and data for the first and last weeks of the season were standardized to represent seven-day scoring periods.

3.2 Proposed Models

The project that we propose will attempt to answer the questions posed in the Introduction; that is, what is the value of pitchers versus hitters in this particular fantasy baseball league? What is the optimal strategy in relation to emphasizing pitchers over hitters or vice versa? Was our hunch correct in 2009 when we began to draft for pitching over hitting? Will this

Table 3.1: Allocation of fantasy points for hitters and pitchers.

Batting Categories	Points
1B - Singles	1
2B - Doubles	2
3B - Triples	3
BB - Walks	1
CS - Caught Stealings	-1
CSC - Caught Stealing by Catcher	2
CYC - Hitting for the Cycle	2
HP - Hit by Pitch	1
HR - Home Runs	5
KO - Strikeous	-1
R - Runs	1
RBI - Runs Batted In	1
SB - Stolen Bases	2
Pitching Categories	Points
BBI - Walks Issued	-1
BS - Blown Saves	-2
CG - Complete Games	5
ER - Earned Runs	-2
HA - Hits Allowed	-1
HD - Holds	4
INN - Innings	3
K - Strikeouts	1
L - Losses	-5
S - Saves	8
SO - Shutouts	5
W - Wins	10
NH - No-Hitters	5

strategy continue to pay off in the future? We will attempt to answer these questions using hierarchical Bayesian models.

The first model will not differentiate between pitchers and hitters whatsoever. The second model will compare pitchers versus hitters, but will not differentiate between relief pitchers and starting pitchers. The third model will compare players, this time with three distinct groups: starting pitchers, relief pitchers, and hitters. Each model will be a hierarchical Bayesian model, allowing each player to have his own mean and variance.

3.3 Likelihood Function

The data to be modeled in this analysis are weekly points scored, and we will use data from the 2011 season. For the likelihood function, we will use a normal distribution. Typically, players will score positive points for a given week. Hitters, in particular, will typically have positive weeks, since they primarily only score negative points for strikeouts. Occurrences of negative weeks for hitters are rare; however, negative weeks for pitchers are more common. If a starting pitcher loses a game and gets shelled in the process, giving up several runs and getting pulled from the game early, this can lead to a very negative week (perhaps as few as -10 or -20 points, and in some cases even fewer). If a relief pitcher gives up a few runs and blows a save, the negative points can accumulate very quickly. Thus, we need to use a likelihood function that allows for positive and negative values. We feel that the data for each player will usually fall around his average and could be above or below it in a fairly symmetric fashion. Therefore, a normal distribution seems appropriate. The likelihood function for a single observation for player i in week j at position k is thus given as

$$f(x_{ijk}|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{(x_{ijk}-\mu_{ik})^2}{2\sigma_{ik}^2}}, \quad x_{ijk}, \mu_{ik} \in (-\infty, \infty), \quad \sigma_{ik}^2 > 0.$$

3.4 Prior Distributions

For the prior distributions for the parameters of my likelihood function, we choose to use a normal distribution for the μ_{ik} and a gamma distribution for the σ_{ik}^2 (with constant shape parameter α_{σ^2}), so each player will have his own mean and variance. Thus, the prior distributions are:

$$\text{Priors:} \begin{cases} \mu_{ik} \sim \text{N}(\mu_{\mu_k}, \sigma_{\mu_k}^2) \\ \sigma_{ik}^2 \sim \text{Gam}(2, \beta_{\sigma_k^2}) \end{cases}.$$

Parameter Selection for Hyperpriors

Parameters for the hyperpriors were selected based on the knowledge I have gained through my past experience and interaction with fantasy baseball. My belief prior to doing this analysis is that, on average, pitchers will score more points per week than hitters. However, since pitchers have greater potential to score negative points, I also believe that the points scored by pitchers will vary more from week to week.

Let us first consider the prior distribution of μ_{ik}. When considering the top 180 players in fantasy baseball, we expect pitchers' means to vary more than those of hitters. This is based on my belief that there is a greater range of talent between the top echelon of pitchers and those near the bottom of the top 70 than there is between the best hitters and those who are merely very good hitters. We will select hyperpriors to allow for a wide enough range to account for this. We expect pitchers' means to be on average, around 25 points per week, with an overall range of roughly 14 to 35. We expect hitters' means to be, on average, about 22, with a range of approximately 13 to 30. To find a mean for the hyperprior for μ_{μ_k} that is appropriate, we calculate a weighted average of the pitchers' and hitters' overall mean averages, or approximately $\mu_{\mu_\mu} = 23$. We do not expect the overall mean to be much different than this, so we choose a relatively small variance of $\sigma^2_{\mu_\mu} = 16$. Thus, for each position k, we let $\mu_{\mu_k} \sim \text{N}(23, 16)$.

For the variance of the means, $\sigma^2_{\mu_k}$, we would like to cover the full range of means that we would expect, described above as spanning 13 to 35. This is equivalent to a variance of approximately 30. A Gam(12, 2.5) distribution has an expected value of $12 \times 2.5 = 30$ and a variance of $12 \times 2.5^2 = 75$. This distribution, shown in Figure 3.1, seems appropriate for the hyperprior of $\sigma^2_{\mu_k}$, since the distribution is centered around 30 while allowing for some variance in both directions. Thus, we let $\sigma^2_{\mu_k} \sim \text{N}(12, 2.5)$ for each position k.

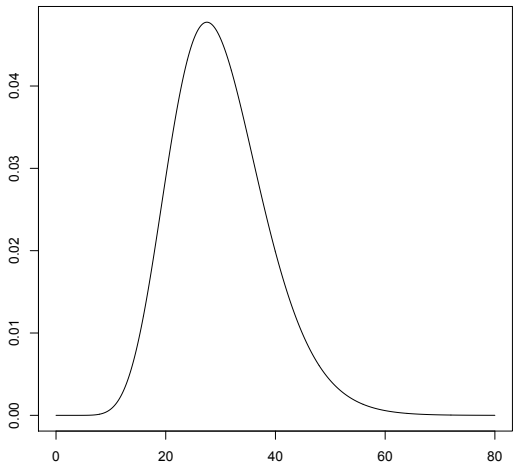

Figure 3.1: Plot of Gam(12, 2.5) distribution.

We now consider the prior distribution of σ_{ik}^2. In our experience, individual players' variances can be quite different. Let us first consider hitters. A player who is more consistent from week to week might have weekly point totals that range between 10 and 30 for the entire season. In contrast, some players are more hot and cold—one week they are on fire and the next they cannot hit anything. This type of player might have point totals that vary between, say, 5 and 60 points over the course of the season (not likely, but certainly possible).

Pitchers' weekly performances fluctuate similarly, but with pitchers, additional variation is introduced because the number of appearances a pitcher makes varies more from week to week. This clearly applies to starting pitchers, whose number of starts in a single week can be either one or two (or in some cases even zero), depending on how the rotation of starting pitchers sets up for his team. In addition, the number of appearances a relief pitcher will make can fluctuate wildly from week to week. Relief pitchers who come into a close game in the ninth inning, generally to protect a small lead, are known as "closers" and are the most valuable type of relief pitcher in fantasy baseball because of their ability to earn saves. The number of saves a particular closer will earn in a given week clearly depends on a number of factors, which includes not only how well he pitches, but also whether his team is

in a position to win very many close games that week where he could be called upon. These various factors surrounding both starting and relief pitchers will lead to greater variation week-to-week than that of hitters.

For our prior distribution of σ_{ik}^2, we fix the shape parameter α_{σ^2} to be a constant of 2, since the shape parameter $\beta_{\sigma_k^2}$ should introduce sufficient variation. As described above, an individual player's variance could range from a number as small as, say, 25, to a larger number, approaching even 200. Since the expected value of a gamma distribution with shape parameter α and scale parameter β is $\alpha\beta$, the expected value of σ_{ik}^2 is equal to $2E(\beta_{\sigma_k^2})$. In light of this calculation, a Gam(2, 20) distribution, shown in Figure 3.2, seems appropriate for $\beta_{\sigma_k^2}$. Thus, we let $\beta_{\sigma_k^2} \sim$ Gamma(2, 20).

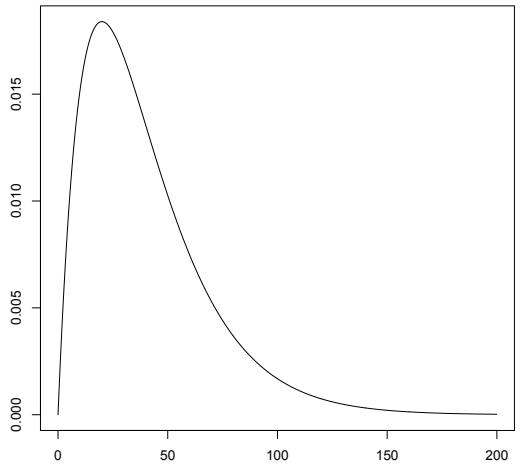

Figure 3.2: Plot of Gam(2, 20) distribution.

In summary, for both hitters and pitchers, the priors and hyperpriors are:

$$\text{Priors:} \begin{cases} \mu_{ik} \sim \text{N}(\mu_{\mu_k}, \sigma^2_{\mu_k}) \\ \sigma^2_{ik} \sim \text{Gam}(2, \beta_{\sigma^2_k}) \end{cases}$$

$$\text{Hyperpriors:} \begin{cases} \mu_{\mu_k} \sim \text{N}(23, 16) \\ \sigma^2_{\mu_k} \sim \text{Gam}(12, 2.5) \\ \beta_{\sigma^2_k} \sim \text{Gam}(2, 20) \end{cases}$$

for player i at position k. To investigate the validity of the priors, we create a *prior predictive* distribution by running a simulation of 100,000 draws from each of these priors and hyperpriors. A plot of this prior predictive distribution is shown in Figure 3.3. The distribution appears to be an accurate reflection of the expectations of fantasy baseball player performances.

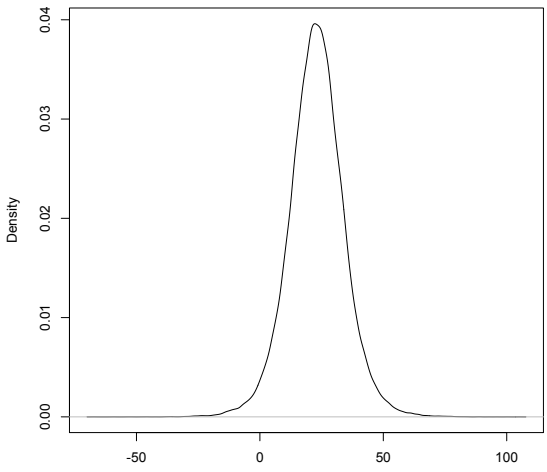

Figure 3.3: Plot of prior predictive distribution based on chosen priors and hyperpriors.

Model 1

Model 1 does not differentiate between hitters and pitchers whatsoever. Thus, $k = 1$.

Model 2

Model 2 differentiates only between hitters and pitchers and allows for no distinction between relief pitchers and starting pitchers. The priors and hyperpriors for both types of players will be the same as those in Model 1, except for $k = 2$.

Model 3

For Model 3, we will also distinguish between starting pitchers and relief pitchers. The priors and hyperpriors for all three types of players will be the same as those in Models 1 and 2, except that $k = 3$.

Model Comparison and Prediction

Each model will be fitted to data from the first three-quarters of the season, or eighteen weeks of data. The model fits will be tested using Bayesian Chi-square goodness-of-fit tests, then compared using the deviance information criterion (DIC). The best model will then be used to predict weekly totals for the final quarter of the data, and posterior predictive densities will be compared to actual weekly scores. This comparison will be made for a sample team of eighteen players constructed to imitate an actual starting lineup for a fantasy baseball team (i.e., one first baseman, one second baseman, five starting pitchers, two relief pitchers, etc.).

CHAPTER 4

RESULTS

4.1 Comparisons of Model Fit

Three models were fitted, as specified in Chapter 3. Originally, 10,000 iterations were used, but due to the presence of excessive autocorrelation, the number of iterations were increased by a factor of 10. The resulting output was thinned back to 10,000 iterations per player, and the amount of autocorrelation was found to be significantly reduced to an acceptable level.

Bayesian χ^2 goodness-of-fit statistics were constructed for each model. In addition, for each model, we computed the proportion of R^B values drawn from the posterior distribution that exceeds the 95th percentile from their χ^2_{K-1} distribution with $K = 25$. The proportions for all three models were 100%, indicating a significant lack of fit for each model. Plots of the densities of each model's R^B values, along with the true χ^2 distribution, are shown in Figure 4.1. We conclude that the data are not normally distributed.

To further examine the non-normality issue, a normal Q-Q plot was constructed and is shown in Figure 4.2. In addition, we conducted tests of normality on our data using the Kolmogorov-Smirnov and Shapiro-Wilk tests. The Kolmogorov-Smirnov test results in a test statistic of D = 0.0774 and a p-value of 0, while the Shapiro-Wilk test results in a test statistic of W = 0.9706 and a p-value of 0. Both strongly indicate that the data are not normally distributed.

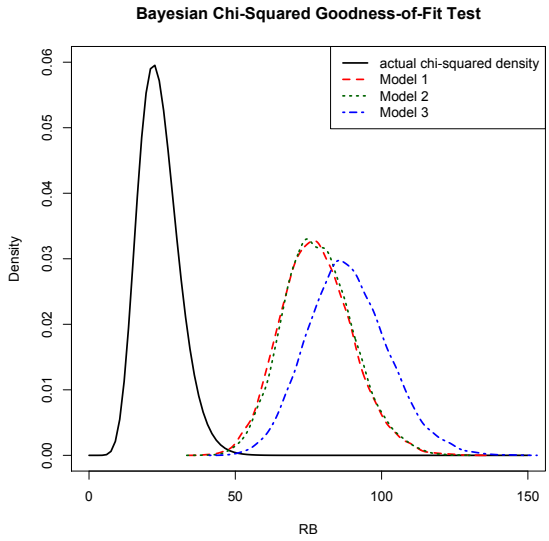

Figure 4.1: A test of model fits using the Bayesian χ^2 goodness-of-fit test. None of the test statistics follow a chi-squared distribution, indicating that our data are not normally distributed.

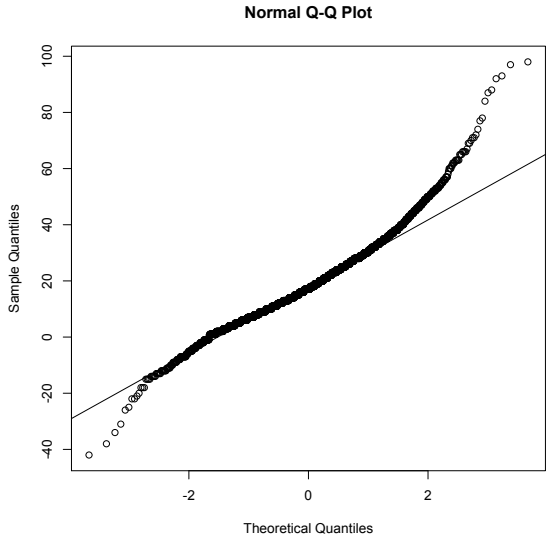

Figure 4.2: A normal Q-Q plot of the data.

In order to explore this further, we constructed a plot including a histogram of the data, the nonparametric density estimate of the data using the Sheather-Jones method to select the bandwidth, and the best fitting normal distribution. This plot, appearing in Figure 4.3, clearly indicates that the data are not normally distributed. The reasons for this are

primarily the peakedness and the slight right-skewness of the data. Despite the data not having a normal distribution, we expect that predictions based on the posterior predictive distributions from our best model will be fairly accurate.

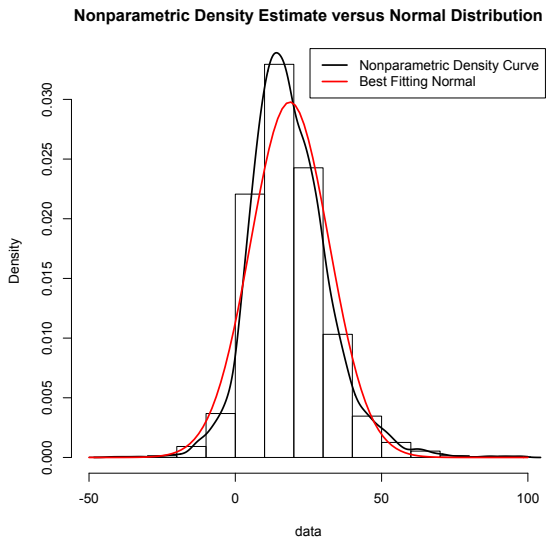

Figure 4.3: A plot including a histogram of the data, the nonparametric density estimate of the data using the Sheather-Jones method, and the best fitting normal distribution.

To select the best of our three models, we use the Deviance Information Criterion (DIC) as discussed in Chapter 2. The results are summarized in Table 4.1. Since smaller values of DIC indicate a better-fitting model, these results indicate that Model 2, which differentiates only between hitters and pitchers, is the best-fitting model. We will therefore proceed with this model in making our predictions.

Table 4.1: Comparison of DIC values for the three models.

Model	DIC
Model 1	-124,382.3
Model 2	-126,681.7
Model 3	-123,409.2

4.2 Comparisons of Pitchers versus Hitters

As a base comparison between hitters and pitchers, let us construct a plot comparing the posterior densities of their means, μ_{μ_k}. This plot is shown in Figure 4.4.

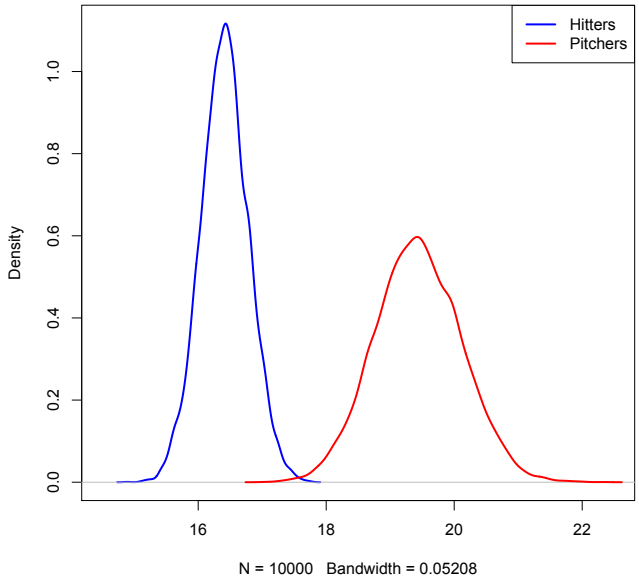

Figure 4.4: Posterior densities of the means of hitters and pitchers, $\mu_{\mu_{hitters}}$ and $\mu_{\mu_{pitchers}}$.

This plot clearly demonstrates that, overall, pitchers do outperform hitters. We ran a simulation of 100,000 draws from each density and found that 100% of the draws favored the pitcher over the hitter.

Next, we examined a plot of the distributions of all 180 players using the posterior densities of their respective μ_{ik}'s and σ^2_{ik}'s. This plot is shown in Figure 4.5, with blue indicating hitters and red indicating pitchers.

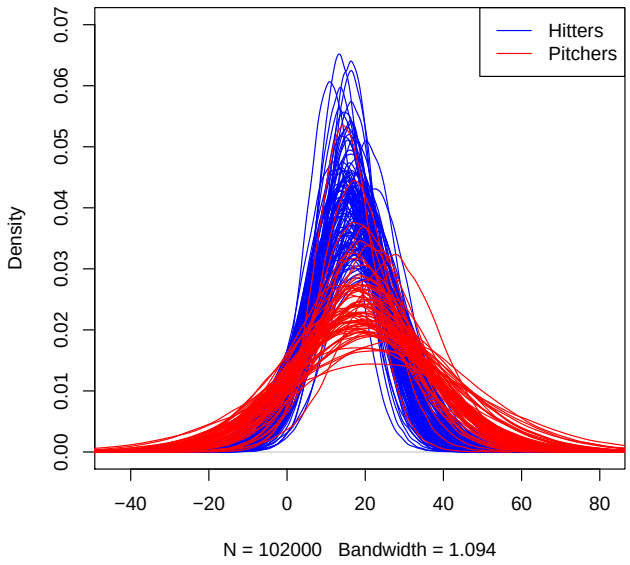

Figure 4.5: Distributions of all 180 players using the posterior densities of their respective μ_{ik}'s and σ^2_{ik}'s.

Again, we see that pitchers have the edge over hitters.

4.3 Prediction

As outlined in Chapter 3, we now use Model 2 to predict weekly totals for the final six weeks of the 2011 season. In order to do this, we compared posterior predictive densities for a sample team of eighteen players to their actual weekly scores. This comparison is made for a sample team that is constructed to imitate an actual starting lineup for a fantasy baseball team (i.e., the appropriate number of players at each position is represented by this lineup). For this comparison, we have constructed the lineup shown Table 4.2.

For each player listed above, we plot the six data points from the final quarter of data from the 2011 season along the x-axis and overlay their posterior predictive density. The plots for each of the eighteen players are shown in Figures 4.6 - 4.8.

From these plots, we see that our posterior predictive densities do a good job of predicting the performances of most of the players over the last six weeks of the season.

Table 4.2: Sample lineup for a fantasy team consisting of eighteen players at each of the appropriate positions.

Hitter	Position	*Pitcher*	Position
Jacoby Ellsbury	OF	Clayton Kershaw	SP
Jose Bautista	3B	Tim Lincecum	SP
Brandon Phillips	2B	Jon Lester	SP
Starlin Castro	SS	Ryan Vogelsong	SP
Matt Wieters	C	Mat Latos	SP
Mike Stanton	OF	Joel Hanrahan	RP
Juan Pierre	OF	Carlos Marmol	RP
Brett Gardner	U		
Eric Hosmer	1B		
Freddie Freeman	DH		
Chris Ianetta	C		

However, a few players significantly over-performed during the last six weeks of the season compared to their prior performances, and similarly, a few players under-performed relative to expectations based on the first three-quarters of the season. These results are not particularly surprising, given the small number of data points we were attempting to predict.

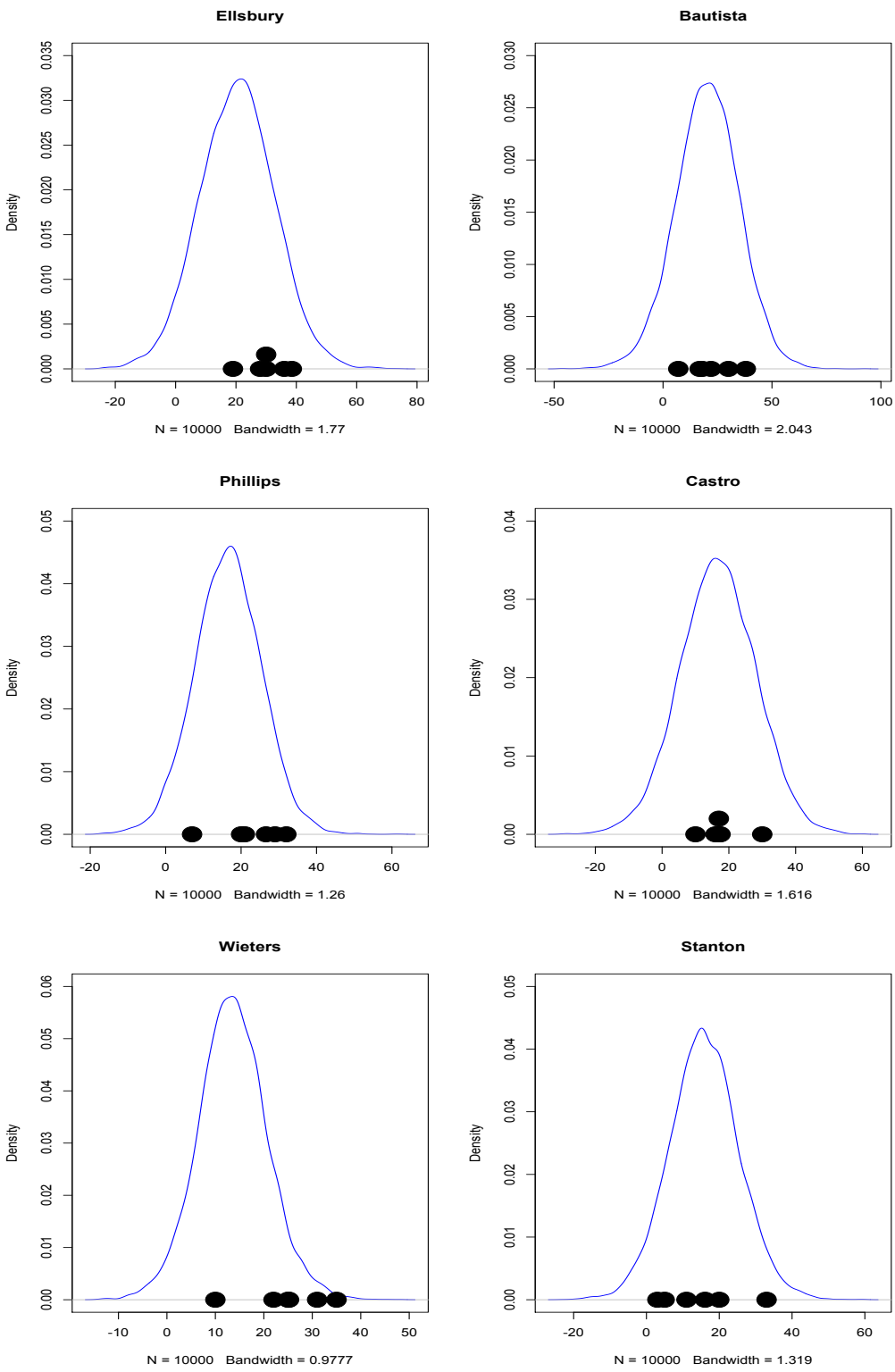

Figure 4.6: Comparisons of actual weekly scores for players 1 - 6 of sample lineup of eighteen players and their posterior predictive densities using Model 2.

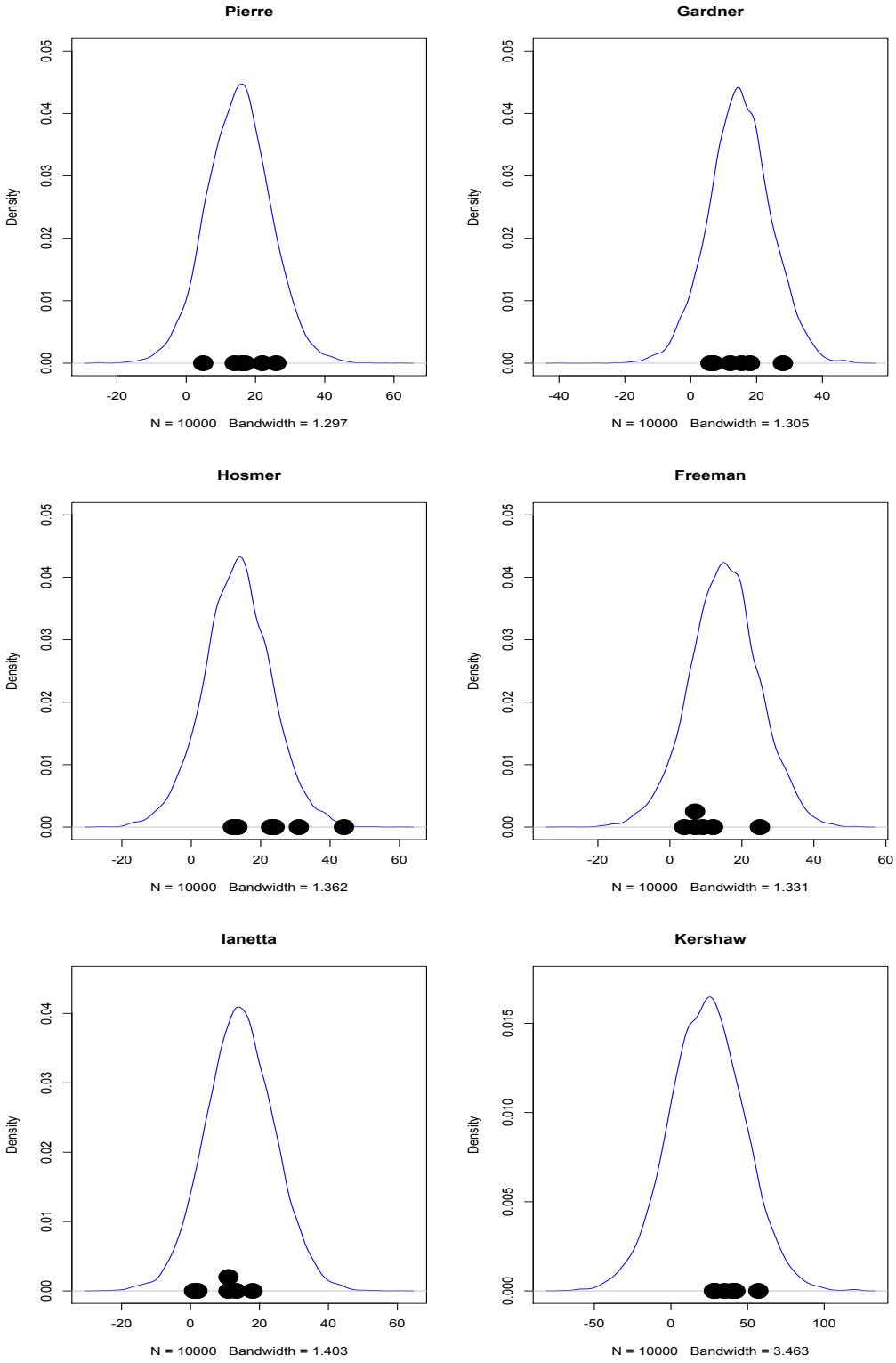

Figure 4.7: Comparisons of actual weekly scores for players 7 - 12 of sample lineup of eighteen players and their posterior predictive densities using Model 2.

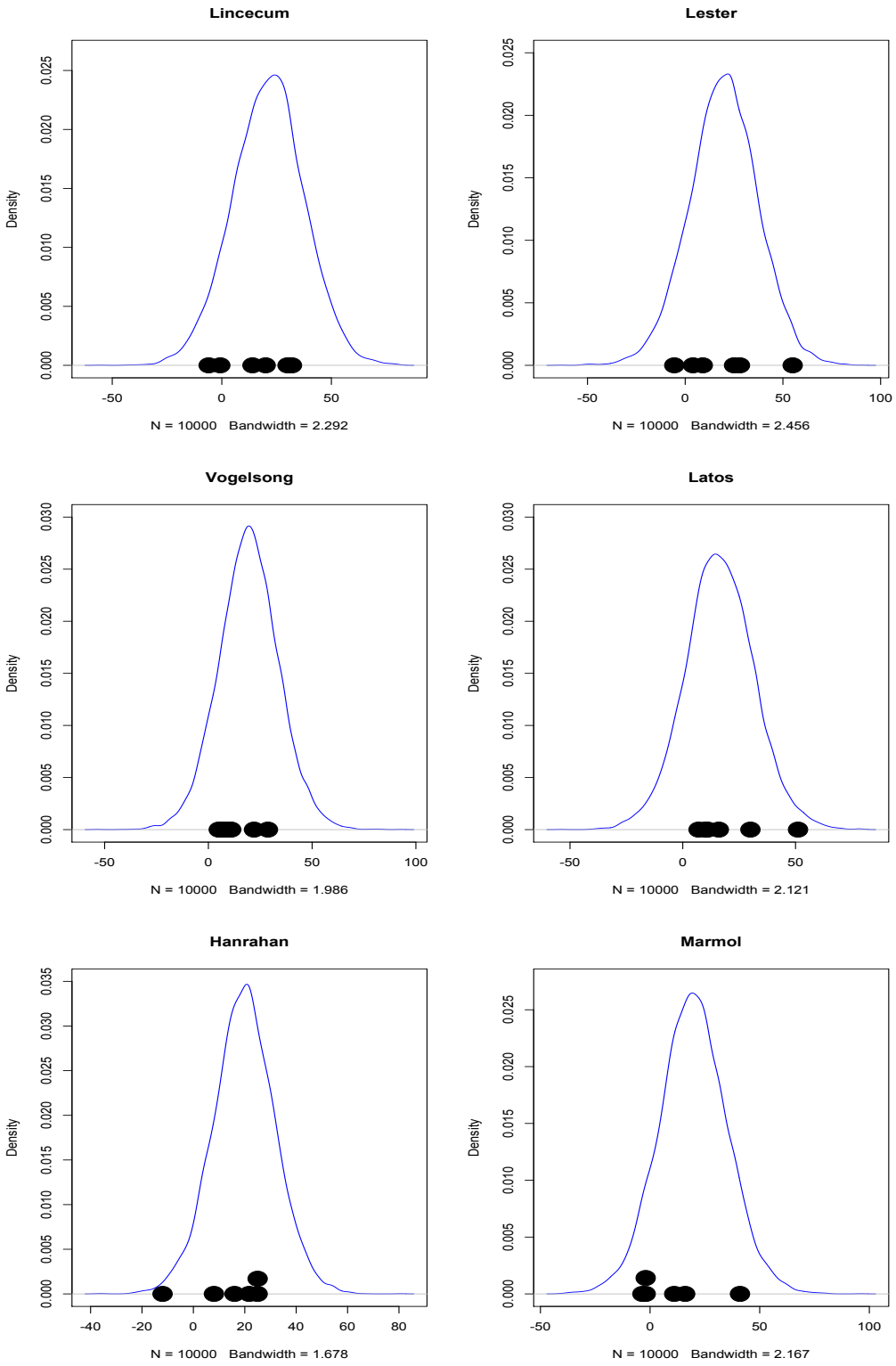

Figure 4.8: Comparisons of actual weekly scores for players 13 - 18 of sample lineup of eighteen players and their posterior predictive densities using Model 2.

CHAPTER 5

DISCUSSION AND CONCLUSIONS

The purpose of this project has been to attempt to answer the question posed herein: what is the value of pitchers versus hitters in this fantasy baseball league? In order to answer these questions, we have employed the use of hierarchical Bayesian models. We have compared three models, the first of which made no distinction between pitchers and hitters, the second of which compared pitchers and hitters with no differentiation between relief pitchers and starting pitchers, and the third of which compared players with the additional distinction between the two types of pitchers.

Using Bayesian χ^2 goodness-of-fit tests, we determined that the data are not normally distributed. Upon examination of a density plot of the data, it appears that the data are slightly right-skewed. This suggests that a right-skewed probability distribution, such as a gamma distribution, would be appropriate. The two-parameter family of gamma distributions does not allow for negative data; thus, a variation of this, such as a three-parameter gamma that allows for negative values of x, seems to be appropriate. This could be the basis for further research using this dataset.

Using DIC to compare these models, we determined that the second model was superior. Using this model, we have constructed posterior predictive densities of eighteen specific players, and we have found our predictions to be reasonably accurate overall.

What does this model say with regards to the questions we posed earlier? We have certainly seen that pitchers score, on average, more points per week than do hitters. For our sample eighteen-player roster, this is clearly seen in Figure 5.1. The posteriors of Kershaw, Lincecum, Lester, and Latos clearly indicate superior weekly performance compared to those

of our hitters. This result occurs despite the fact that two of our hitters, Ellsbury and Bautista, were among the best fantasy hitters in baseball in 2011.

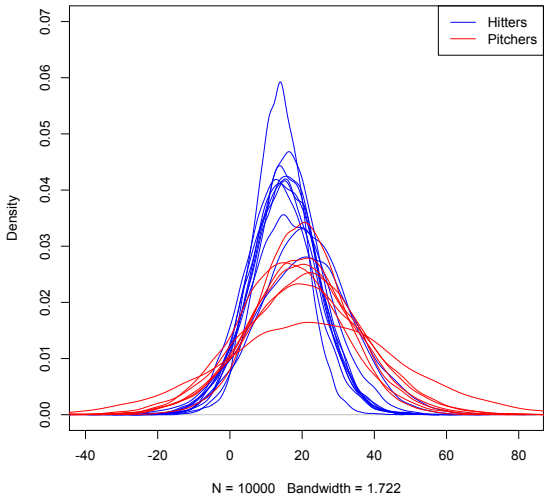

Figure 5.1: Comparison of posterior predictive distributions for our eighteen-player sample roster

The next question we would like to pose is, how can we use this result to gain an advantage in the league? To begin with, we can continue to do what we did in 2010 and 2011, which was to focus on drafting pitchers early in our league drafts. This allowed us to draft pitchers like Kershaw, Lincecum, Lester, and Latos, players we were confident would continue to pitch at a high level and enable us to be competitive in the league. However, we must recognize that if pitching is inherently superior to hitting in this league, then this advantage will carry over to every team owner in the league. To fully exploit this discrepancy, we must focus on drafting the top pitchers in the league first and foremost, enabling us to gain the biggest possible advantage. As was discussed in section 3.4, we believe that the range of talent between the top and bottom echelons of pitchers is greater than that same range for hitters. By drafting players like Kershaw and Lincecum, we were able to secure premium pitchers who were more likely to outperform even the best hitters, and we were still

able to draft high-quality hitters, since there was a larger number of them available relative to other hitters.

To date, this strategy has paid off for us two years in a row, and we enter the 2012 season hoping for a "three-peat." By continuing to employ this strategy, we hope to have success in the league for years to come, and for the memories of the woeful seasons of 2005 to 2009 to continue to be a thing of the past.

CHAPTER 6

APPENDIX

6.1 R Code

```
rm(list=ls())

setwd('/Volumes/My Passport/My Documents/School - BYU/Documents/Classes/Masters
     Project/data analysis')

data <- as.matrix(read.csv('2011 stats 18 weeks.csv',header=T))
data <- matrix(as.numeric(data[,c(4:21)]),ncol=18,nrow=180)

n <- nrow(data); n
n_hitters <- 110
n_pitchers <- 70

alpha <- 2
alpha1 <- 12
beta1 <- 2.5
mu <- 23
sigma2 <- 16
alpha2 <- 2
beta2 <- 20

loglik <- function(dat_i,mu_i,sigma2_i){
sum(-1/(2*sigma2_i)*(sum(dat_i-mu_i)^2) - 1/2*length(dat_i)*log(sigma2_i))
}
```

```
lngmui <- function(dat_i,mu_i,sigma2_i,mu_mu,sigma2_mu){
-1/(2*sigma2_i)*sum((dat_i - mu_i)^2) - 1/(2*sigma2_mu)*(mu_i - mu_mu)^2
}

lngsigma2i <- function(dat_i,mu_i,sigma2_i,alpha_sigma2,beta_sigma2){
-1/(2*sigma2_i)*sum((dat_i - mu_i)^2) + (1 - 1/2*length(dat_i))*log(sigma2_i) -
     sigma2_i/beta_sigma2
}

lngbetasigma2 <- function(alpha_sigma2,beta_sigma2,sigma2_i){
-2*length(sigma2_i)*log(beta_sigma2) - sum(sigma2_i)/beta_sigma2 + log(beta_sigma2) -
     beta_sigma2/20
}

lngmumu <- function(mu_i,mu_mu,sigma2_mu){
-1/(2*sigma2_mu)*sum((mu_i - mu_mu)^2) - 1/(2*16)*(mu_mu - 23)^2
}

lngsigma2mu <- function(mu_i,mu_mu,sigma2_mu){
-1/2*length(mu_i)*log(sigma2_mu) - 1/(2*sigma2_mu)*sum((mu_i - mu_mu)^2) +
     (12-1)*log(sigma2_mu) - (1/2.5)*sigma2_mu
}

# get initial values

mui <- rep(23,n)
sigma2i <- rep(16,n)
betasigma2_1 <- 40
betasigma2_2 <- 40
mumu_1 <- 23
mumu_2 <- 23
sigma2mu_1 <- 16
sigma2mu_2 <- 16
```

```
# assign candidate sigmas

csmui_hitters <- 8
csmui_pitchers <- 10
cssigma2i_hitters <- 188
cssigma2i_pitchers <- 330
csbetasigma2_1 <- 18
csbetasigma2_2 <- 60
csmumu_1 <- 2.0
csmumu_2 <- 2.5
cssigma2mu_1 <- 6
cssigma2mu_2 <- 12

# set up counters

cntbetasigma2_1 <- 0
cntbetasigma2_2 <- 0
cntmumu_1 <- 0
cntmumu_2 <- 0
cntsigma2mu_1 <- 0
cntsigma2mu_2 <- 0

cntmui_hitters <- NA
cntmui_pitchers <- NA
cntsigma2i_hitters <- NA
cntsigma2i_pitchers <- NA
for (i in 1:n){
cntmui_hitters[i] <- 0
cntsigma2i_hitters[i] <-0
cntmui_pitchers[i] <- 0
cntsigma2i_pitchers[i] <-0
}
```

```
Nsim <- 102000
burnin <- 2000

out <- matrix(0,Nsim,6 + 2*n)
loglik_dat <- matrix(0,Nsim,n)

out[1,1] <- betasigma2_1 # hitters
out[1,2] <- betasigma2_2 # pitchers
out[1,3] <- mumu_1 # hitters
out[1,4] <- mumu_2 # pitchers
out[1,5] <- sigma2mu_1 # hitters
out[1,6] <- sigma2mu_2 # pitchers
out[1,(6+1):(6+n)] <- mui
out[1,(6+n+1):(6+2*n)] <- sigma2i

# MCMC sampling

for (i in 2:Nsim){
# candidate draws for betasigma2 for hitters
cand <- rnorm(1,out[i-1,1],csbetasigma2_1)
out[i,1] <- out[i-1,1]
if (cand > 0){
r1 <- lngbetasigma2(2,cand,out[i-1,(6+n+1):(6+n+n_hitters)]) -
     lngbetasigma2(2,out[i-1,1],out[i-1,(6+n+1):(6+n+n_hitters)])
if (r1 > log(runif(1,0,1))){
out[i,1] <- cand
cntbetasigma2_1 <- cntbetasigma2_1 + 1
}
}
# candidate draws for betasigma2 for pitchers
cand <- rnorm(1,out[i-1,2],csbetasigma2_2)
out[i,2] <- out[i-1,2]
```

```
if (cand > 0){
r1 <- lngbetasigma2(2,cand,out[i-1,(6+n+n_hitters+1):(6+2*n)]) -
     lngbetasigma2(2,out[i-1,2],out[i-1,(6+n+n_hitters+1):(6+2*n)])
if (r1 > log(runif(1,0,1))){
out[i,2] <- cand
cntbetasigma2_2 <- cntbetasigma2_2 + 1
}
}
# candidate draws for mumu for hitters
cand <- rnorm(1,out[i-1,3],csmumu_1)
out[i,3] <- out[i-1,3]
if (cand > 0){
r2a <- lngmumu(out[i-1,(6+1):(6+n_hitters)],cand,out[i-1,5]) -
     lngmumu(out[i-1,(6+1):(6+n_hitters)],out[i-1,3],out[i-1,5])
if (r2a > log(runif(1,0,1))){
out[i,3] <- cand
cntmumu_1 <- cntmumu_1 + 1
}
}
# candidate draws for mumu for pitchers
cand <- rnorm(1,out[i-1,4],csmumu_2)
out[i,4] <- out[i-1,4]
if (cand > 0){
r2b <- lngmumu(out[i-1,(6+n_hitters+1):(6+n)],cand,out[i-1,6]) -
     lngmumu(out[i-1,(6+n_hitters+1):(6+n)],out[i-1,4],out[i-1,6])
if (r2b > log(runif(1,0,1))){
out[i,4] <- cand
cntmumu_2 <- cntmumu_2 + 1
}
}
# candidate draws for sigma2mu for hitters
cand <- rnorm(1,out[i-1,5],cssigma2mu_1)
out[i,5] <- out[i-1,5]
```

```
if (cand > 0){
r3a <- lngsigma2mu(out[i-1,(6+1):(6+n_hitters)],out[i-1,3],cand) -
     lngsigma2mu(out[i-1,(6+1):(6+n_hitters)],out[i-1,3],out[i-1,5])
if (r3a > log(runif(1,0,1))){
out[i,5] <- cand
cntsigma2mu_1 <- cntsigma2mu_1 + 1
}
}
# candidate draws for sigma2mu for pitchers
cand <- rnorm(1,out[i-1,6],cssigma2mu_2)
out[i,6] <- out[i-1,6]
if (cand > 0){
r3b <- lngsigma2mu(out[i-1,(6+n_hitters+1):(6+n)],out[i-1,4],cand) -
     lngsigma2mu(out[i-1,(6+n_hitters+1):(6+n)],out[i-1,4],out[i-1,6])
if (r3b > log(runif(1,0,1))){
out[i,6] <- cand
cntsigma2mu_2 <- cntsigma2mu_2 + 1
}
}
out[i,(6+1):(6+n)] <- out[i-1,(6+1):(6+n)]
out[i,(6+n+1):(6+2*n)] <- out[i-1,(6+n+1):(6+2*n)]
# updates for hitters
for (j in 1:n_hitters){
cand <- rnorm(1,out[i-1,6+j],csmui_hitters)
if (cand > 0){
r4a <- lngmui(data[j,],cand,out[i,6+n+j],out[i,3],out[i,5]) -
     lngmui(data[j,],out[i,6+j],out[i,6+n+j],out[i,3],out[i,5])
if (r4a > log(runif(1,0,1))){
out[i,6+j] <- cand
cntmui_hitters[j] <- cntmui_hitters[j] + 1
}
}
cand <- rnorm(1,out[i-1,6+n+j],cssigma2i_hitters)
```

```
if (cand > 0){
r5a <- lngsigma2i(data[j,],out[i,6+j],cand,2,out[i,1]) -
     lngsigma2i(data[j,],out[i,6+j],out[i,6+n+j],2,out[i,1])
if (r5a > log(runif(1,0,1))){
out[i,6+n+j] <- cand
cntsigma2i_hitters[j] <- cntsigma2i_hitters[j] + 1
}
}
loglik_dat[i,j] <- loglik(data[j,],out[i,6+j],out[i,6+n+j]) }
# updates for pitchers
for (j in (n_hitters+1):n){
cand <- rnorm(1,out[i-1,6+j],csmui_pitchers)
if (cand > 0){
r4b <- lngmui(data[j,],cand,out[i,6+n+j],out[i,4],out[i,6]) -
     lngmui(data[j,],out[i,6+j],out[i,6+n+j],out[i,4],out[i,6])
if (r4b > log(runif(1,0,1))){
out[i,6+j] <- cand
cntmui_pitchers[j] <- cntmui_pitchers[j] + 1
}
}
cand <- rnorm(1,out[i-1,6+n+j],cssigma2i_pitchers)
if (cand > 0){
r5b <- lngsigma2i(data[j,],out[i,6+j],cand,2,out[i,2]) -
     lngsigma2i(data[j,],out[i,6+j],out[i,6+n+j],2,out[i,2])
if (r5b > log(runif(1,0,1))){
out[i,6+n+j] <- cand
cntsigma2i_pitchers[j] <- cntsigma2i_pitchers[j] + 1
}
}
loglik_dat[i,j] <- loglik(data[j,],out[i,6+j],out[i,6+n+j]) }
}

###### burn-in ######
```

```
out <- out[-c(1:burnin),]
dim(out)

###### check acceptance rates ######

cntbetasigma2_1 / Nsim
cntbetasigma2_2 / Nsim
cntmumu_1 / Nsim
cntmumu_2 / Nsim
cntsigma2mu_1 / Nsim
cntsigma2mu_2 / Nsim
cntmui_hitters / Nsim
cntmui_pitchers / Nsim
cntsigma2i_hitters / Nsim
cntsigma2i_pitchers / Nsim

###### mcmc plots ######

library(coda)
raftery.diag(as.mcmc(out))
par(mfrow=c(2,4))
for (i in 1:8){
acf(as.mcmc(out[,i]))
}
par(mfrow=c(2,4))
for (i in 1:8){
acf(as.mcmc(out[,i+180]))
}

###### thin data ######

new_Nsim <- 10000
```

```
new_out <- matrix(0,new_Nsim,6 + 2*n)
for (i in 1:new_Nsim){
new_out[i,] <- out[i*10,]
}
dim(new_out)

###### save posterior thetas ######

write.table(new_out,'new_out_table.csv',sep=', ',quote=FALSE,row.names=FALSE,
     col.names=FALSE)
new_out <- read.table('new_out_table.csv',sep=',')
new_out <- as.matrix(new_out,nrow=10000,ncol=180)
dim(new_out)

####### check autocorrelation ########

library(coda)
raftery.diag(as.mcmc(new_out))
par(mfrow=c(2,4))
for (i in 1:8){
acf(as.mcmc(new_out[,i]))
}
par(mfrow=c(2,4))
for (i in 1:8){
acf(as.mcmc(new_out[,i+180]))
}

######### run diagnostics #########

###### DIC ######

loglik_dat <- loglik_dat[-c(1:burnin),]
dim(loglik_dat)
```

```
loglik_vec <- NA
for (i in 1:(nrow(loglik_dat)/10)){
loglik_vec <- sum(loglik_dat[10*i,])
}

dbar1 <- -2*mean(loglik_vec)
dthetahat1 <- -2*loglik(apply(data,1,mean),apply(out[,7:186],2,mean),
     apply(out[,187:366],2,mean))
pd1 <- dbar1 - dthetahat1
dic1 <- dbar1 + pd1

###### Bayes Chi-Square ######

n <- 180*18
K <- round(n^0.4,0)
exp <- rep(n/K,K)
test <- rep(0,dim(new_out)[1])
qs <- seq(0,1,length.out=K+1)
for (k in 1:dim(new_out)[1]){
cnt <- rep(0,K)
for (i in 1:180){
x <- data[i,]
mu <- new_out[k,6+i]
#sigma2 <- new_out[k,5]
sigma2 <- new_out[k,6+180+i]
cnt <- cnt + hist(pnorm(x,mean=mu,sd=sqrt(sigma2)),breaks=qs,plot=F)$counts
}
test[k] <- 1/(n/K)*sum((exp-cnt)^2)
}

crit <- qchisq(0.95,K-1)
sum(test>crit)
```

```
mean(test>crit)

plot(density(test),xlim=c(0,150),ylim=c(0,0.06),col='blue')
curve(dchisq(x,K-1),add=T,col=2)

### explore non-normality ###

data_full <- as.matrix(read.csv('2011 stats full season.csv',header=T))
data_full <- matrix(as.numeric(data_full[,c(4:27)]),ncol=24,nrow=180)
data_full <- as.vector(data_full)

hist(data_full[which(data_full!=0)],freq=F,main='Nonparametric Density Estimate
     versus Normal Distribution',xlab='data')
lines(density(data_full[which(data_full!=0)],bw='SJ-ste'),lwd=2,col=1,main=
     'Density Comparison for Data versus Normal Distribution')
curve(dnorm(x,mean=mean(data_full[which(data_full!=0)]),sd=
     sd(data_full[which(data_full!=0)])),lwd=2,col=2,add=T)
legend('topright',c('Nonparametric Density Curve','Best Fitting Normal')
     ,lwd=c(2,2),col=c(1,2))

###### q-q plot ######
qqnorm(data_full[which(data_full!=0)])
qqline(data_full[which(data_full!=0)])

#### shapiro-wilk and k-s tests
shapiro.test(data_full[which(data_full!=0)])
ks.test(data,"pnorm",mean(data),sd(data))

#################################################################

###### mcmc plots ######
n <- 180
plot(as.mcmc(new_out[,1:(n+6)]))
```

```
plot(as.mcmc(new_out[,(n+7):(2*n+6)]))

###### look at means and variances ######

means <- NA
vars <- NA

for (i in 1:366){
means[i] <- mean(new_out[,i])
vars[i] <- var(new_out[,i])
}

means
vars

mean(means[c(7:186)]) # 17.54562
min(means[c(7:186)]) # 11.34800
max(means[c(7:186)]) # 26.69996

mean(means[c(187:366)]) # 7122.53
min(means[c(187:366)]) # 5531.399
max(means[c(187:366)]) # 8630.241

#############################################################

plot(density(new_out[,3]),type='l',col='blue',xlim=c(14.5,22.5),main='',lwd=2)
lines(density(new_out[,4]),col='red',lwd=2)
legend("topright",c("Hitters","Pitchers"),col=c("blue","red"),lty=c(1,1),lwd=c(2,2))

compare <- sample(new_out[,3],100000,replace=TRUE) < sample(new_out[,4],100000,
     replace=TRUE)
mean(compare)
```

```
#################################################################

Nsim <- 10000
post_draws <- matrix(NA,nrow=(Nsim),ncol=180)

for (i in 1:180){
post_draws[,i] <- rnorm((Nsim),new_out[,i+6],sqrt(new_out[,180+6+i]))
}

plot(density(post_draws[,1]),type='l',ylim=c(0,0.07),col='blue',main='')
for (i in 2:110){
lines(density(post_draws[,i]),col='blue')
}
for (i in 111:180){
lines(density(post_draws[,i]),col=2)
}
legend("topright",c("Hitters","Pitchers"),col=c("blue","red"),lty=c(1,1))

data_last6 <- as.matrix(read.csv('2011 stats last 6 weeks.csv',header=T))
data_last6 <- matrix(as.numeric(data_last6[c(1:180),c(4:9)]),ncol=6,nrow=180)

### plot posterior predictive densities and actual data points ###

par(mfrow=c(1,2))
#ellsbury-1
plot(density(post_draws[,1]),type='l',ylim=c(0,0.035),col='blue',main='Ellsbury')
xs <- data_last6[1,]
ys <- rep(0,6)
points(xs[-5],ys[-5],pch=19,cex=3)
points(xs[5],0.0016,pch=19,cex=3)

#bautista-6
plot(density(post_draws[,6]),type='l',ylim=c(0,0.03),col='blue',main='Bautista')
```

```
xs <- data_last6[6,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#phillips-30
plot(density(post_draws[,30]),type='l',ylim=c(0,0.05),col='blue',main='Phillips')
xs <- data_last6[30,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#castro - 41
plot(density(post_draws[,41]),type='l',ylim=c(0,0.04),col='blue',main='Castro')
xs <- data_last6[41,]
ys <- rep(0,6)
points(xs[-2],ys[-2],pch=19,cex=3)
points(xs[2],0.002,pch=19,cex=3)

#wieters - 47
plot(density(post_draws[,47]),type='l',ylim=c(0,0.06),col='blue',main='Wieters')
xs <- data_last6[47,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#stanton - 62
plot(density(post_draws[,62]),type='l',ylim=c(0,0.05),col='blue',main='Stanton')
xs <- data_last6[62,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#pierre - 66
plot(density(post_draws[,66]),type='l',ylim=c(0,0.05),col='blue',main='Pierre')
xs <- data_last6[66,]
ys <- rep(0,6)
```

```
points(xs,ys,pch=19,cex=3)

#gardner - 71
plot(density(post_draws[,71]),type='l',ylim=c(0,0.05),col='blue',main='Gardner')
xs <- data_last6[71,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#hosmer - 74
plot(density(post_draws[,74]),type='l',ylim=c(0,0.05),col='blue',main='Hosmer')
xs <- data_last6[74,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#freeman - 95
plot(density(post_draws[,95]),type='l',ylim=c(0,0.05),col='blue',main='Freeman')
xs <- data_last6[95,]
ys <- rep(0,6)
points(xs[-2],ys[-2],pch=19,cex=3)
points(xs[2],0.0025,pch=19,cex=3)

#ianetta - 110
plot(density(post_draws[,110]),type='l',ylim=c(0,0.045),col='blue',main='Ianetta')
xs <- data_last6[110,]
ys <- rep(0,6)
points(xs[-3],ys[-3],pch=19,cex=3)
points(xs[3],0.002,pch=19,cex=3)

#kershaw - 112
plot(density(post_draws[,112]),type='l',ylim=c(0,0.0175),col='blue',main='Kershaw')
xs <- data_last6[112,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)
```

```
#lincecum - 128
plot(density(post_draws[,128]),type='l',ylim=c(0,0.0265),col='blue',main='Lincecum')
xs <- data_last6[128,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#lester - 145
plot(density(post_draws[,145]),type='l',ylim=c(0,0.025),col='blue',main='Lester')
xs <- data_last6[145,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#vogelsong - 152
plot(density(post_draws[,152]),type='l',ylim=c(0,0.03),col='blue',main='Vogelsong')
xs <- data_last6[152,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#latos - 169
plot(density(post_draws[,169]),type='l',ylim=c(0,0.03),col='blue',main='Latos')
xs <- data_last6[169,]
ys <- rep(0,6)
points(xs,ys,pch=19,cex=3)

#hanrahan - 146
plot(density(post_draws[,146]),type='l',ylim=c(0,0.035),col='blue',main='Hanrahan')
xs <- data_last6[146,]
ys <- rep(0,6)
points(xs[-3],ys[-3],pch=19,cex=3)
points(xs[3],0.0017,pch=19,cex=3)

#marmol - 172
```

```
plot(density(post_draws[,172]),type='l',ylim=c(0,0.0275),col='blue',main='Marmol')
xs <- data_last6[172,]
ys <- rep(0,6)
points(xs[-3],ys[-3],pch=19,cex=3)
points(xs[3],0.0014,pch=19,cex=3)

##### plot densities of players on roster together #####

plot(density(post_draws[,1]),type='l',ylim=c(0,0.07),col='blue',main='')
for (i in c(6,30,41,47,62,66,71,74,95,110)){
lines(density(post_draws[,i]),col='blue')
}
for (i in c(112,128,145,152,169,146,172)){
lines(density(post_draws[,i]),col=2)
}
legend("topright",c("Hitters","Pitchers"),col=c("blue","red"),lty=c(1,1))
```

BIBLIOGRAPHY

Bayes, T., and Price, R. (1763), "An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, M. A. and F. R. S." *Philosophical Transactions of the Royal Society of London*, 53, 370–418.

Carlin, B. P., and Louis, T. A. (2009), *Bayesian Methods for Data Analysis* (3rd ed.), Chapman & Hall/CRC Press.

Geisser, S. (1993), *Predictive Inference*, London: Chapman and Hall.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995), *Bayesian Data Analysis* (1st ed.), London: Chapman and Hall.

Gelman, A., Gilks, W., and Roberts, G. (1997), "Weak convergence and optimal scaling of random walk Metropolis algorithms," *Ann. Appl. Probab.*, 7, 110–120.

Greenburg, Z. O. (2009), "Tips from fantasy baseball's best," *Forbes.com*.

Hastings, W. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 97–109.

Johnson, V. E. (2004), "A Bayesian χ^2 test for goodness-of-fit," *The Annals of Statistics*, 32, 2361–2384.

Lee, P. (1997), *Bayesian Statistics: An Introduction* (2nd ed.), New York: John Wiley & Sons Inc.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953), "Equations of state calculations by fast computing machines," *J. Chemical Physics*, 21, 1087–1091.

Robert, C. P., and Casella, G. (2004), *Monte Carlo Statistical Methods* (2nd ed.), New York: Springer.

Ross, S. M. (1996), *Stochastic Processes* (2nd ed.), New York: John Wiley & Sons Inc.

Spiegelhalter, D. J., Best, N., Carlin, B., and van der Linde, A. (2002), "Bayesian measures of model complexity and fit (with discussion)," *J. Roy. Statist. Soc., Ser. B*, 64, 583–639.

Starr, M. (2008), "My Baseball Fantasy," *Newsweek*.