Contents

Predicting NBA Player Performance Using Bayesian Models

Ben Moolman, Craig Orman, Ethan Pross

STAT 5440: Bayesian Statistics

Iowa State University

May 2025

1 Introduction

1.1 Project Description

In the National Basketball Association (NBA), player performance is influenced by a wide array of factors, including fatigue, game context, opposing team defense, and individual variance. While traditional sports analytics approaches often rely on point estimates or historical averages, these methods may fail to capture the uncertainty and game-to-game variability inherent in athletic performance.

In this project, we aim to study and predict field goals made (FGM) by NBA starters across games from the 2023-2024 regular season. Our interest centers on understanding how much of a player's scoring can be attributed to their own underlying shooting ability versus the strength of the defense they face. By using publicly available game-level box score data, we analyze how the performance of a given player fluctuates depending on the matchup.

We plan to restrict our attention to "starters"—players who were in the starting lineup for at least one game during the season. Starters are commonly the best players on their team at their respective positions. This allows us to focus on players who are expected to have a significant impact on the game, and whose performance is more likely to be influenced by the opposing team's defensive strategy.

We use a publicly available dataset from Kaggle, titled *NBA Boxscore - Season 2023 / 2024*[1], which contains box score statistics for every player-game combination from the 2023-2024 NBA regular season. The full dataset includes 32,385 observations and 30 variables. After filtering for starters, our working dataset contains 12,300 observations. This step ensures we are modeling players with consistent playing time, which improves model reliability and interpretability.

To account for variation in opponent strength, we construct a proxy for team-level defensive effectiveness using defensive rating (DRTG): the average number of points allowed by a team across the season. This allows us to factor in how opposing defenses affect the likelihood of a player making a field goal on any given night.

Our overall goal is to create a probabilistic modeling framework that incorporates both player-specific characteristics and opponent-level effects to estimate and predict field goal outcomes in future games. This framework can help highlight patterns in performance and inform matchup-based expectations in contexts such as playoff forecasting, daily fantasy sports, or team scouting.

1.2 Prior Analysis and Literature Review

The majority of literature considers two metrics, Box plus/minus, and Player Efficiency Rating (PER). PER is supposed to be an overall metric that, given the teams likelihood of winning a game, determines if a particular player is likely to increase or decrease that chance. There have been many attempts to expand upon this, adding more team effects, or trying to overcome the lack of direct interpretation. However, Box plus/minus takes a different approach, and calculates a numbed defined as the points contributed by a player per 100 possessions above an average player. Both of these metrics take into account a variety of covariates, both offense and defensive. However, critics frequently cite that defensive play is undervalued in these measures, as well that they do not take into account any specifics of the actual games played. In Williams et al's analysis titled *Expected Points Above Average: A Novel NBA Player Metric Based on Bayesian Hierarchical Modeling*[2] they predict expected points, using a factorized version of location on the field as a covariate, as well as factoring in player, and overall team skills.

[1] https://www.kaggle.com/datasets/albi9702/nba-boxscore-season-2023-2024?resource=download

[2] https://arxiv.org/html/2405.10453v1

2 Exploratory Data Analysis

2.1 Dataset Overview

As stated in the Introduction, we filtered the dataset to include only players that started a game during the 2023-2024 NBA season. The resulting dataset has 30 columns and 12300 rows, with each row corresponding to a unique player-game entry. 7 columns are identifiers, 1 column is a row index, 2 are characteristic and comment columns, and then we have 20 numerical statistic columns.

2.2 Key Variables for Modeling

From the full box score, we focus on the following variables relevant to our modeling goals:

- `FGM` - Field goals made
- `FGA` - Field goals attempted
- `FG_PCT` - Field goal percentage
- `TEAM_ID` - Player's team
- `OPP_TEAM_ID` - Opposing team (added manually through data wrangling)
- `GAME_ID` - Unique game identifier
- `START_POSITION` - Indicator for whether a player started

We also constructed a derived variable, `DRTG_proxy`, to quantify the defensive strength of each team. This is computed as the average number of points allowed by each team across all games, based on the `PTS` column. We then center this value to produce `centered_OPP_DRTG` for use in the model.

2.3 Summary Statistics

To understand the general trends in field goal performance among NBA starters, we compute basic summary statistics for three key variables: `FGM`, `FGA` & `FG_PCT`

The table below reports the sample mean, standard deviation, minimum, and maximum for each variable across all player-game observations in our filtered dataset.

Table 1: Summary Statistics for Key Shooting Variables for NBA Starters

Variable	Mean	SD	Min	Max
FGM	5.888	3.378	0	25
FGA	12.208	5.918	0	47
FG_PCT	0.481	0.179	0	1

The average NBA starter attempts roughly 12 field goals per game, making about 5.9 of them. This corresponds to a mean field goal percentage of 48.1%, which aligns with league-wide expectations for starters. However, the large standard deviations, particularly for FGA (SD $\approx$ 5.9), indicate substantial variation in shot volume across players. The wide range of FG% values (from 0 to 1) reflects outliers due to either perfect shooting games or very low attempt games, highlighting the importance of modeling game-level variability.

We also visualize the distribution of field goal attempts (FGA) to assess its skewness and variability in figure (1).

Starters take an average of 12 shots per game, but the distribution is skewed right, with some high-volume players attempting over 40 shots in a single game—a rare but influential occurrence.

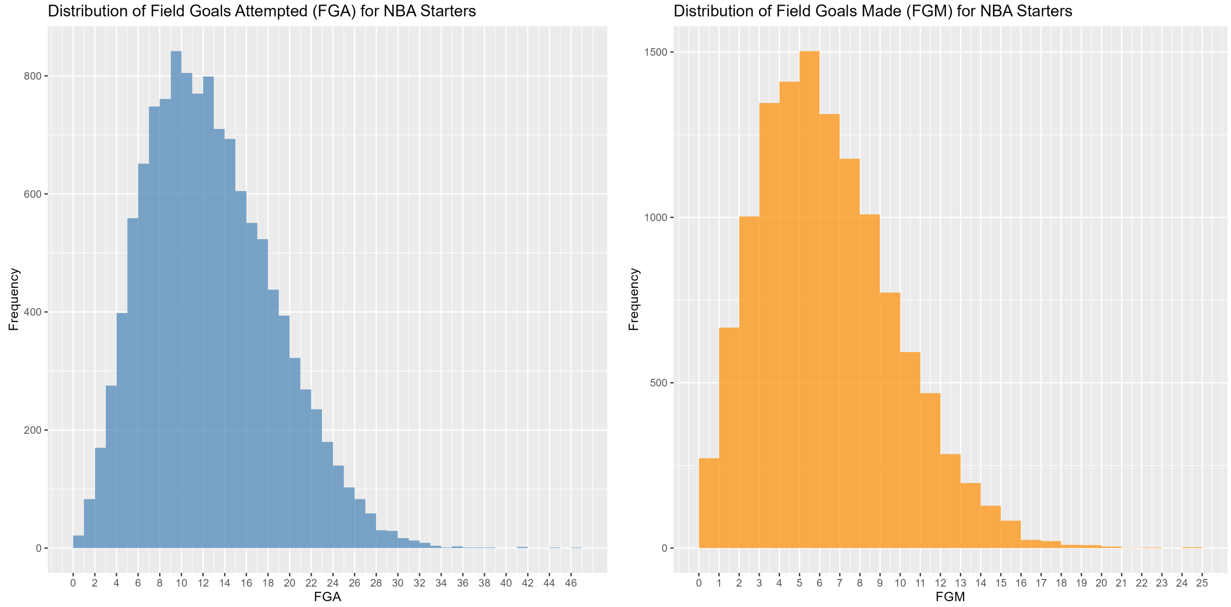

Figure 1: 2.3

And we can do the same for field goals made (FGM), shown in figure (1).

Although starters average nearly 6 made field goals per game, the range extends from 0 to 25. The majority fall below 10 FGM, justifying a Binomial model with varying success probability.

To better understand player-level variability in scoring, we highlight the distribution of field goals made (FGM) for two prominent NBA starters: LeBron James and Stephen Curry. These players offer a useful contrast—both are prolific scorers, but they play different roles and have different shot profiles, as seen in figure (2)

While both LeBron James and Stephen Curry are elite scorers, their distributions of field goals made (FGM) reveal meaningful differences in scoring consistency. LeBron's FGM distribution is more concentrated around 10-12 made field goals per game, reflecting a stable and reliable scoring output. In contrast, Curry's distribution is more spread out, with a wider range of outcomes and a longer right tail—he has both low-FGM games (e.g., 2-4) and explosive performances exceeding 17 FGM. This contrast aligns with their playing styles: LeBron tends to score efficiently and consistently through drives and post-ups, while Curry's reliance on high-variance three-point shooting introduces more game-to-game fluctuation. These insights highlight why modeling player-specific shot distributions is important when forecasting performance against different opponents.

These distributions support a Binomial modeling approach for FGM given FGA, and also motivate the inclusion of player-specific shot attempt distributions in later models.

2.4 Focus Player: LeBron James

To illustrate our modeling framework in a concrete setting, we focus on LeBron James as a case study. LeBron is a high-usage, consistent starter with a well-documented performance profile, making him an ideal player to highlight the effects of matchup strength on field goal outcomes. In the 2023-2024 season, LeBron has started in 71 games for his team, the Los Angeles Lakers, out of 82 total games played in the season.

Below, we plot the distribution of LeBron's field goal attempts (FGA) per game during the 2023-2024 regular season. This provides context for later modeling decisions that incorporate fixed or random shot attempt distributions.

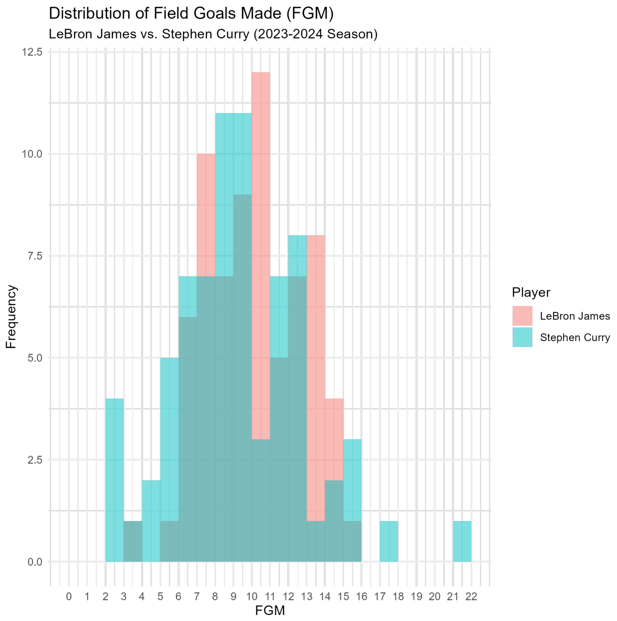

Figure 2: 2.3d

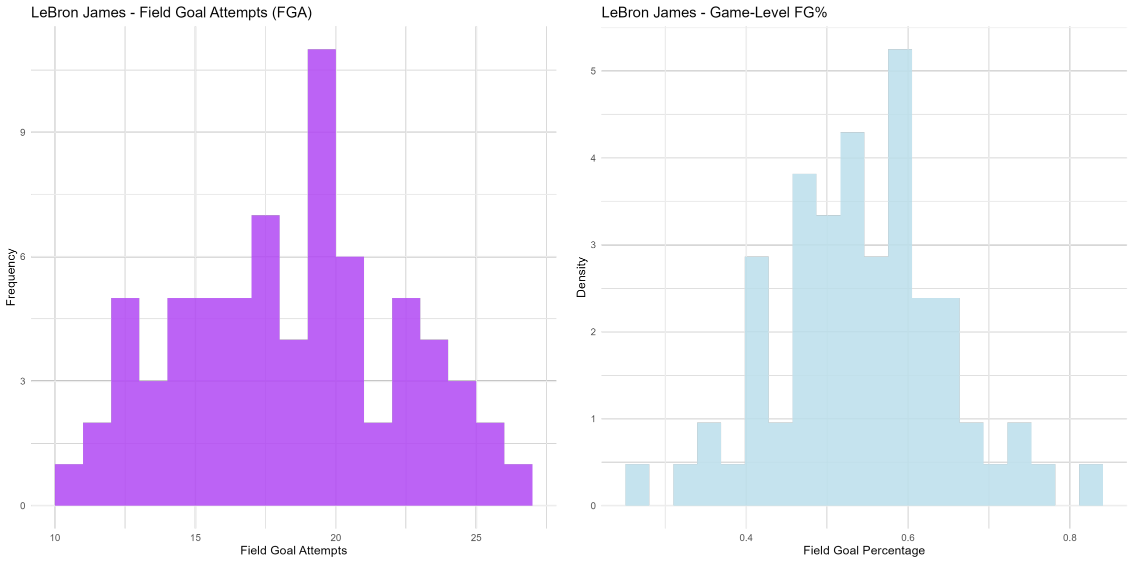

Figure 3: 2.4

LeBron's FGA distribution is moderately concentrated around 18 attempts per game, with occasional high-usage games exceeding 25 shots. His observed minimum, mean, median, and maximum FGA values are summarized below:

Table 2: Summary of LeBron James' Field Goal Attempts per Game (2023-2024)

Statistic	FGA
Minimum	10
Mean	18
Median	18
Maximum	27

These values serve as inputs for different fixed-attempt scenarios in our first model. Additionally, they help motivate more flexible approaches in Models 2 and 3, which allow for variability in shot volume across games.

Finally, the histogram below shows LeBron's game-level field goal percentage (FG%) across the season. This distribution informs our prior belief about his shooting efficiency as shown in figure (3).

LeBron's game-level field goal percentages are concentrated between 45% and 65%, with a clear unimodal structure and light tails on both ends. The distribution is slightly right-skewed but remains relatively symmetric overall. This shape supports the use of a Beta prior centered near 0.54, with moderate dispersion to account for game-to-game variability. The lack of extreme values (very low or very high FG%) suggests stable shooting performance, justifying a prior that reflects both consistency and moderate uncertainty.

3 Bayesian Model Specification

We begin by justifying our modeling choices for both the likelihood and the prior distribution. Because our response variable — field goals made (FGM) — is a count of successful outcomes out of a known number of field goal attempts (FGA), it is natural to model this as a Binomial outcome. Specifically, for each game, a player either makes or misses each of their field goal attempts, and these trials can reasonably be assumed to be conditionally independent given the player's underlying shooting ability and the strength of the opposing defense. This motivates a Binomial likelihood of the form:

$$y_{ijk} \sim \text{Binomial}(n_{ijk}, p_{ik})$$

where:

- y_{ijk}: number of field goals made by player i in game j against opponent team k
- n_{ijk}: number of field goal attempts by player i in game j against opponent k
- p_{ik}: probability that player i makes a shot against team k, which we later model as being influenced by opponent defense
- i: index for player (e.g., LeBron James)
- j: index for game
- k: index for opposing team

We assume that field goals made arise as a Binomial sample out of shot attempts. The number of trials n_{ijk} (FGA) may be fixed or random depending on the model. The probability of success p_{ik} varies across players and across matchups due to the influence of opposing defenses.

We assume that player i's base shooting ability is represented by a parameter $p_i \in (0, 1)$, and we model this with a Beta prior distribution:

$$p_i \sim \text{Beta}(a_i, b_i)$$

To account for opponent defensive strength, we assume the actual game-level shot success probability is a multiplicative function of the baseline ability and an opponent adjustment factor:

$$p_{ik} = p_i \cdot \exp\left[\gamma(\text{DRTG}_k - \overline{\text{DRTG}})\right]$$

where:

- DRTG_k: the defensive rating for team k
- $\overline{\text{DRTG}}$: the average defensive rating across all teams
- γ: a sensitivity parameter controlling how much defensive rating shifts shot percentage

This structure allows opponent strength to scale each player's shooting probability up or down. Because the exponential function is always positive, it ensures that $p_{ik} > 0$, though we enforce an upper bound to ensure $p_{ik} \in [0, 1]$.

3.1 Prior Estimation for Baseline Shooting Ability p_i

To determine the parameters a_i and b_i of the Beta prior for each player's baseline FG%, we use the method of moments. For a Beta distribution:

$$\mathbb{E}[p_i] = \frac{a_i}{a_i + b_i}, \quad \text{Var}(p_i) = \frac{a_i b_i}{(a_i + b_i)^2(a_i + b_i + 1)}$$

Rearranging these equations gives the method of moments estimators:

$$\hat{a}_i = \bar{p}_i \left(\frac{\bar{p}_i(1 - \bar{p}_i)}{s_i^2} - 1 \right), \quad \hat{b}_i = (1 - \bar{p}_i) \left(\frac{\bar{p}_i(1 - \bar{p}_i)}{s_i^2} - 1 \right)$$

where $\bar{p}_i$ is the sample mean FG% and s_i^2 is the sample variance. For example, for LeBron James, the sample mean FG% is 0.5422451 and the sample variance is 0.01134541, giving us:

$$a_i = 11.32102, \quad b_i = 9.557027$$

This yields a moderately informative prior centered around his historical shooting percentage, as can be seen in Fig. (4). We can see LeBron's empirical field goal percentages compared to the fitted Beta distribution. This prior centers the distribution near his historical performance, with moderate concentration, allowing for uncertainty due to game-to-game variability.

For other players, we apply the same method to estimate player-specific a_i and b_i, ensuring priors reflect each player's shooting profile. Code to obtain these estimates for other players (provided some minor data wrangling is done) is given in the Appendix.

3.2 Proposed Models

We propose three models that differ in how they treat the number of shot attempts n_{ijk}. We will hold the following constant for all three models.

$$\begin{aligned} y_{ijk} \mid n_{ijk}, p_{ik} &\sim \text{Binomial}(n_{ijk}, p_{ik}) \\ p_{ik} &= p_i \cdot \exp\left[\gamma(\text{DRTG}_k - \overline{\text{DRTG}})\right] = p_i \cdot c_k \\ p_i &\sim \text{Beta}(a_i, b_i) \end{aligned}$$

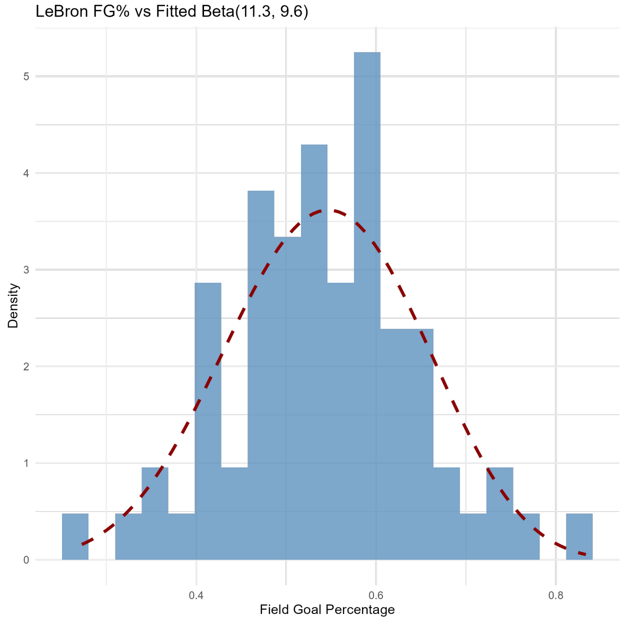

Figure 4: 3.1

3.2.1 Model 1: Fixed Shot Attempts

This model assumes that the number of shot attempts per game is fixed for each player: $n_{ijk} = n_i$. We fit the model for three different choices of n_i: mean, median, and max number of FGA across the season.

A concern with this model is that n_{ijk} is in reality a random variable and not fixed.

3.2.2 Model 2: Poisson Shot Attempts

Model 2 assumes that the number of shot attempts n_{ijk} varies across games and follows a Poisson distribution:

$$n_{ijk} \sim \text{Poisson}(\lambda_i)$$

This reflects the reality that shot attempts are non-negative count data that fluctuate based on in-game context such as team strategy, opponent defense, and game pace. To model this variability, each player i is assigned their own Poisson rate parameter λ_i, which governs the expected number of attempts per game. We use a the non-informative Jeffreys prior for this parameter:

$$p(\lambda_i) \propto \lambda_i^{-1/2}$$

The full model structure is summarized below:

- $y_{ijk} \mid n_{ijk}, p_{ik} \sim \text{Binomial}(n_{ijk}, p_{ik})$
- $p_{ik} = p_i \cdot c_k$, where $c_k = \exp[\gamma(DRTG_k - \overline{DRTG})]$
- $p_i \sim \text{Beta}(a_i, b_i)$
- $n_{ijk} \sim \text{Poisson}(\lambda_i)$
- $\lambda_i \sim \text{Jeffreys prior} \propto \lambda_i^{-1/2}$

3.2.2.1 Model Limitation A significant drawback of using the Poisson model is its assumption that the mean and variance of n_{ijk} are equal. In reality, players often show overdispersion in shot attempts — that is, the variance exceeds the mean. In our dataset, this assumption is violated for roughly 67% of players. Only 33% of players have |mean(FGA) – var(FGA)| < 2, suggesting the need for a more flexible alternative. This motivates the use of a Negative Binomial distribution for n_{ijk} in Model 3.

3.2.3 Model 3: Negative Binomial Shot Attempts

Model 3 generalizes Model 2 by allowing for overdispersion in field goal attempt counts. Rather than assuming that the number of field goal attempts per game n_{ijk} follows a Poisson distribution (which implies mean = variance), we instead assume a Negative Binomial distribution:

$$n_{ijk} \sim \text{NegBin}(r_i, \theta_i)$$

This formulation allows the variance of n_{ijk} to exceed the mean, which better reflects observed variability in player shot attempts across games. Specifically, the mean and variance of the Negative Binomial are:

$$\mathbb{E}[n_{ijk}] = \frac{r_i(1-\theta_i)}{\theta_i}, \quad \text{Var}(n_{ijk}) = \frac{r_i(1-\theta_i)}{\theta_i^2}$$

This relaxes the restrictive Poisson assumption and enables us to accommodate heterogeneity in game-to-game usage. However, the Negative Binomial imposes its own limitation: it cannot accommodate underdispersion, i.e., situations where $\text{Var}(n_{ijk}) < \mathbb{E}[n_{ijk}]$. In our data, roughly 14% of players fall into this category, which could reduce model fit for those cases.

For the Negative Binomial parameters r_i and θ_i, we assume a joint non-informative Jeffreys-type prior:

$$p(r_i, \theta_i) \propto \sqrt{\frac{r_i}{\theta_i^2(1-\theta_i)}}$$

3.2.3.1 Model Summary Model 3 allows for greater flexibility in modeling shot attempt variability across games. While the formulation accounts for overdispersion, it does not permit underdispersion (i.e., $\text{Var}(n_{ijk}) < \mathbb{E}[n_{ijk}]$). In our data, 53% of players had Var(FGA) – Mean(FGA) > 2, supporting the need for this model. However, 14% of players showed the opposite pattern, for which this model may not be appropriate.

3.2.4 Prior Justification

As mentioned, in all models the prior of p_{ik} was chosen to be informative to better reflect each player; while the hyperparameters for n_{ijk} were chosen to be non-informative jeffrey's priors.

3.3 Sensitivity Analysis

To investigate how LeBron's field goal performance might vary based on opponent defensive strength, we conduct a sensitivity analysis under Model 1, where the number of shots attempted is fixed at $n = 18$. We explore how adjusting the shooting percentage p in response to an opponent's Defensive Rating (DRTG) affects the posterior predictive distribution.

We define the adjustment as:

$$p_{\text{adjusted}} = p + \gamma \cdot (\text{DRTG}_{\text{opponent}} - \bar{\text{DRTG}})$$

where:

- γ is a global sensitivity parameter,
- $\text{DRTG}_{\text{opponent}}$ is the opposing team's average points allowed (higher = worse defense),
- $\overline{\text{DRTG}}$ is the league average.

The summary below shows the Warriors (DRTG = 115.16) are slightly weaker than average (114.21):

Table 3: Summary Statistics for Team Defensive Rating (DRTG Proxy)

Statistic	Value
Min.	106.5244
1st Qu.	110.8811
Median	113.4817
Mean	114.2114
3rd Qu.	117.2866
Max.	123.0366

This modest difference suggests that only small adjustments to LeBron's FG% are warranted. For instance, setting $\gamma = 0.1$ causes FG% to spike dramatically against weak defenses, which may lead to unrealistic outcomes. Smaller values (e.g., $\gamma = 0.01$) produce more believable shifts.

In figure (8), we visualize the posterior predictive distribution of field goals made using 4 gamma values against the Warriors.

These plots show:

- With $\gamma = 0$, we recover the baseline predictive distribution
- As γ increases, the center of the distribution shifts right
- At $\gamma = 0.1$, predictions become unrealistically high

Thus, small positive γ values (like 0.01) can reasonably incorporate opponent defensive effects without inflating predictions. Since the Warriors' defense is near average, our adjustment has a modest and appropriate impact.

3.4 Concluding Remarks for Section 3

Section 3 outlined a flexible Bayesian framework for modeling and predicting NBA player shooting performance. The Binomial likelihood and Beta prior accommodate uncertainty in shooting ability, while opponent defense is incorporated multiplicatively. Three models offer different assumptions about shot attempt variability. Model 1 assumes fixed shot volume, while Models 2 and 3 allow for player-specific randomness. Posterior predictive simulations offer insights into likely outcomes against specific opponents.

4 Results

4.1 MCMC Sampler

4.1.1 Model 1

We implemented a custom Metropolis-Hastings (MH) algorithm to draw posterior samples for p_i, given a fixed vector of shot attempts n_i and observed field goals made y_{ijk}. The posterior log-density is:

$$\log q(p) = \sum_j \log \Pr(y_{ij} \mid n_i, p) + \log \pi(p)$$

where $\pi(p) \sim \text{Beta}(11.32102, 9.557027)$ is the prior based on LeBron's historical field goal percentages via method-of-moments.

The algorithm uses a Normal random walk proposal:

- Proposal: $p^* \sim \mathcal{N}(p_{\text{current}}, \sigma^2)$
- Accept with probability $\min(1, \exp(\log q(p^*) - \log q(p_{\text{current}})))$

We ran 3 parallel chains with starting values 0.3, 0.5, and 0.7. Each chain contains 1000 iterations.

Model 1 is evaluated using three different values for the fixed shot attempt count n_i:

$$n_i \in \{\text{mean}, \text{median}, \text{max}\}$$

For LeBron James, these values are:

$$\text{mean}(n_i) = 18, \quad \text{median}(n_i) = 18, \quad \max(n_i) = 27$$

Posterior samples of p_i were used to compute matchup-adjusted shooting probabilities:

$$p_{ik} = p_i \cdot \exp[\gamma(\text{DRTG}_k - \overline{\text{DRTG}})]$$

These estimates allow us to simulate posterior predictive distributions for field goals made against each team.

For example, Fig. 9 displays LeBron's posterior mean FG% by opposing team, based on the posterior samples under the mean-n scenario, with 95% credible intervals:

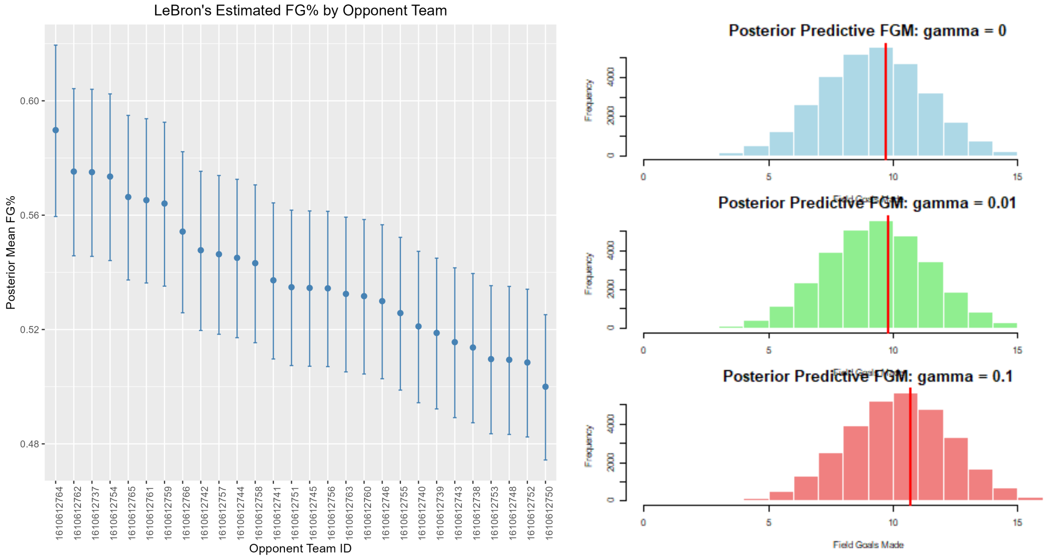

Figure 5: 4.1.1

4.1.2 Model 2

Model 2 incorporates uncertainty in LeBron's field goal attempts by treating them as Poisson-distributed: $n_{ijk} \sim \text{Poisson}(\lambda_i)$, with λ_i estimated jointly alongside p_i. We implement a blocked Metropolis-within-Gibbs sampler targeting the posterior distribution of $(p_i, \lambda_i, \mathbf{n})$.

Each iteration includes the following updates:

- Sample $p_i \sim \text{Beta}(11.32102 + \sum y_{ijk},\ 9.557027 + \sum(n_{ijk} - y_{ijk}))$
- Sample $\lambda_i \sim \text{Gamma}(\sum n_{ijk} - \frac{1}{2}, J)$
- Update each latent count n_{ijk} via a Metropolis step with a normal proposal

We ran 10,000 iterations of this sampler using LeBron's game-level data. Posterior draws of p_i were adjusted for matchup strength against each opposing team using the same scaling factor as in Model 1:

$$p_{ik}^{(s)} = p_i^{(s)} \cdot \exp\left[\gamma(\text{DRTG}_k - \overline{\text{DRTG}})\right], \quad \gamma = 0.01$$

We focus on predicting LeBron's field goals made (FGM) in a future game against the Golden State Warriors. In each posterior iteration, we:

1. Draw $\tilde{n}_{ijk}^{(s)} \sim \text{Poisson}(\lambda_i^{(s)})$
2. Compute matchup-adjusted shooting probability: $p_{ik}^{(s)}$
3. Draw $\tilde{y}_{ijk}^{(s)} \sim \text{Binomial}(\tilde{n}_{ijk}^{(s)}, p_{ik}^{(s)})$

The resulting distribution incorporates both efficiency and usage uncertainty.

4.1.3 Model 3

Model 3 extends Model 2 by accounting for overdispersion in shot attempts. Rather than assuming Poisson variability, we assume:

$$n_{ijk} \sim \text{NegBin}(r_i, \theta_i)$$

with r_i (number of failures) and θ_i (success probability) learned from the data. This allows for greater flexibility in modeling game-to-game variation in usage.

We implement a blocked Metropolis-within-Gibbs sampler to estimate $p_i, r_i, \theta_i, \mathbf{n}$, using the following updates per iteration:

- Sample $p_i \sim \text{Beta}(11.32102 + \sum y_{ijk},\ 9.557027 + \sum(n_{ijk} - y_{ijk}))$
- Sample $\theta_i \sim \text{Beta}(\sum n_{ijk} + \frac{1}{2}, r_i - 1)$
- Update r_i via Metropolis step using normal proposal
- Update each n_{ijk} via Metropolis step using normal proposal

Posterior draws of p_i are adjusted for opponent difficulty using:

$$p_{ik}^{(s)} = p_i^{(s)} \cdot \exp\left[\gamma(\text{DRTG}_k - \overline{\text{DRTG}})\right], \quad \gamma = 0.01$$

To generate posterior predictive draws for a future game against the Golden State Warriors, we simulate:

1. $n_{\text{new}}^{(s)} \sim \text{NegBin}(r_i^{(s)}, \theta_i^{(s)})$
2. $p_{\text{GSW}}^{(s)} = p_i^{(s)} \cdot \exp[\gamma(\text{DRTG}_{\text{GSW}} - \overline{\text{DRTG}})]$
3. $\tilde{y}_{ijk}^{(s)} \sim \text{Binomial}(n_{\text{new}}^{(s)}, p_{\text{GSW}}^{(s)})$

This approach propagates uncertainty in both shot attempts and shooting efficiency, while allowing for greater variability in usage patterns due to the negative binomial structure.

4.2 MCMC Diagnostics

4.2.1 4.2.1 Model 1

In Fig. 10 are the traceplots and autocorrelation functions (ACF) for each of the three MCMC chains. These help assess mixing and convergence.

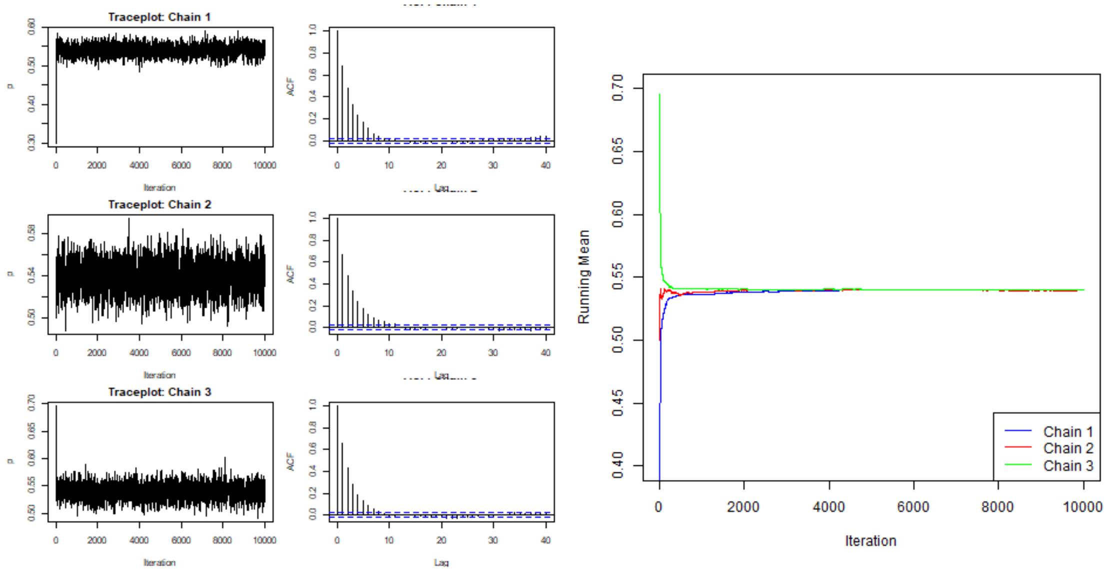

Figure 6: 4.2.1a

We also visualize the running means for each chain to assess convergence to a stable posterior distribution.

Given the traceplots above, and the ESS all being over 2000 below, we find that the model using the fixed values looks very reasonable, and is good for future analysis.

```
## [1] "ESS of model 1,  Max is:  3000 Median is:  3000 Mean is:  2835"
```

4.2.2 4.2.1 Model 2

Observing the traceplots in Fig. 11, we notice that all of the variables have a relative convergence. All of the n variables and p variables converge to the same area which is good. Results summarized in the area below. The traceplot however, shows a good value at 6441

4.2.3 4.2.2 Model 3

With an effective sample size of 41, we have great reason to believe that this model did not converge, however the n values went to 22 after 100k iterations, and the p value was at 0.57, both of which are very reasonable values compared to the LeBron's sampled n and p. One of the main issues encountered in this model is that all of the values involved large factorials which caused problems as the variables grew. The r parameter breaks down around 140 as it produces a NaN or infinity.

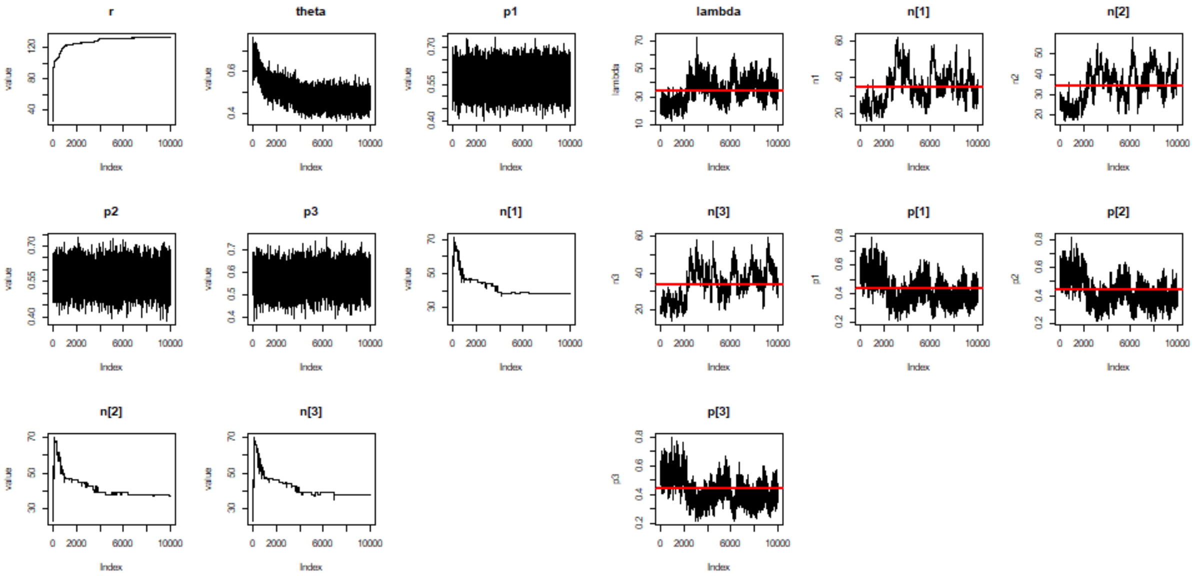

Figure 7: 4.2.2a

4.3 Model Comparison

In order to compare the models, we considered two primary methods. The first is the Bayes Factor, which will indicate which models are more likely given the data that has been collected in an empirical fashion. The second is a visual comparison of the Posterior Predictive Distribution.

4.3.1 Bayes Factors

We see below that the Fixed n model seems to have the highest weight, making it the best candidate model. Derivations are in TODO

FixedBF	PoissonBF	NegBinomBF
0.9913269	0.0086731	0

4.3.2 Posterior Predictive Distribution

In Fig. 13, the comparison of the options for model 1, the mean and median underpredict the FGA considerably, and we believe that is because of the three games played, Lebron had a significant outlier game. The GSW have almost an average defensive rating, but there is a known rivalry between LeBron James and the GSW team. Therefore, we see the mean and median are more likely to under predict the number of shots attempted.

Next, we compare using the fixed max against the other two models. The poisson model underpredicts and gives too much weight to the tail cases.

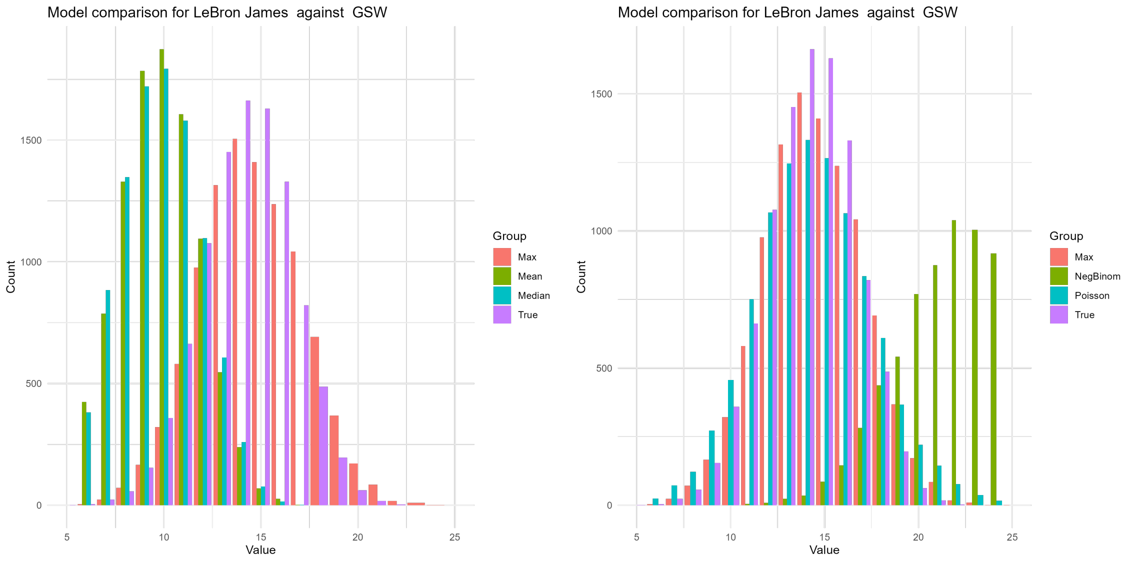

Figure 8: 4.3.2

4.3.3 Posterior Predictive Checking

We generate replicated observations

$$y_{ijk}^{rep(s)} \sim \text{Binomial}(n_{ijk}^{(s)}, p_{ik}^{(s)})$$

for each posterior draw s = 1,. . . ,S.

We define a test statistic T(), for example to check if $|y_{iJk} - S_{iJk}| \leq \phi$, where y_{iJk} denotes the sample mean, S_{iJk} denotes the sample standard deviation, and ϕ denotes a precision parameter; i.e. to see that the mean & variance are reasonably close to each other. The goal is to compare $\{T(y^{rep(s)})\}_{s=1}^{S}$ vs T(y)

The model has no fitting concerns if p_{ppp} = P(T($y_{rep(s)}$) ≥ T(y) | y) is near 0.5.

We approximate

$$p(\tilde{y}|y_{i,1:J,k}) = \sum_{n} \int p(\tilde{y}|n_{ijk}, p_{ik}, y_{i,1:j,k}) p(n_{ijk}, p_{ik}|y_{i,1:j,k}) dp$$

with posterior samples

$$(1/S) \sum_{s} p(\tilde{y}|n_{ijk}^{(s)}, p_{ik}^{(s)}, y_{i,1:J,k})$$

The models' mean FGM & variance of FGM are compared against the observed values in Figures .

In the density graphs found in Figures , for all models, the black lines represent the observed data, while the blue lines show the 100 replicated datasets from the posterior predictive distribution. The poor matching between observed and replicated is likely due to the small sample size (3) in the lebron vs GSW example.

5 Discussion

5.1 Issues & Improvements

Model 3 and Model 1 with mean and median performed poorly, perhaps testing on different players.

we found that the mena and variance weren't always equal

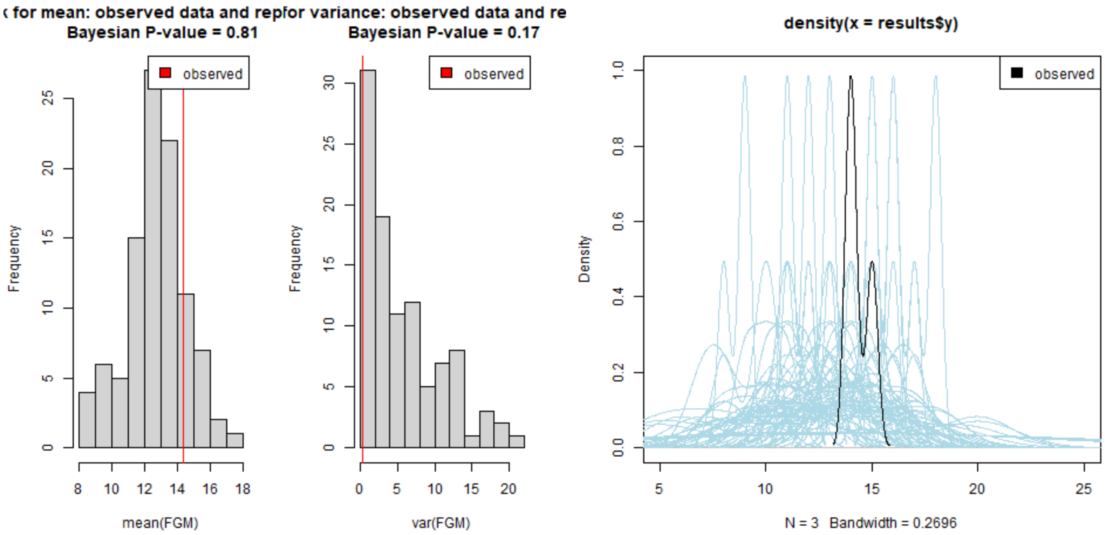

Figure 9: 4.3.3

5.2 Conclusions

[*Summarize performance of each model, discuss practical takeaways]
[Highlight limitations and possible extensions]*

6 Appendix

6.1 Mathematical Derivations

6.1.1 Model 1

Given the model below, we make the following derivations.

$$\begin{aligned} y &\sim Binom(n_i, p_{ik}) \\ n_i &= \{Mean, Median, Max\} \\ p_{ik} &\sim c_k * beta(\alpha_i, \beta_i) \end{aligned}$$

6.1.1.1 Likelihood and Posterior The likelihood is:

$$\prod_j \text{Binomial}(y_{ijk} \mid n_i, p_{ik}) = \prod_j \binom{n_i}{y_{ijk}} (p_i c_k)^{y_{ijk}} (1 - p_i c_k)^{n_i - y_{ijk}}$$

Letting $c_k = \exp[\gamma(DRTG_k - \overline{DRTG})]$, we express the conditional posterior for p_i as:

$$p(p_i|...) \propto \int_0^1 p_i^{a_i - 1 + \sum_j y_{ijk}} (1 - p_i)^{b_i - 1 + \sum_j (n_i - y_{ijk})} dp_i$$

which is proportional to the kernel of a Beta distribution. Thus, the posterior for p_i is:

$$p_i \mid y \sim \text{Beta}\left(a_i + \sum_j y_{ijk},\ b_i + \sum_j (n_i - y_{ijk})\right)$$

6.1.2 Model 2

Given the model below, we make the following derivations.

$$\begin{aligned} y &\sim Binom(n_i, p_{ik}) \\ n_i &\sim poisson(\lambda_i) \\ p(\lambda_i) &\sim \lambda_i^{-1/2} \text{Jeffrey's prior} \\ p_{ik} &\sim c_k * beta(\alpha_i, \beta_i) \end{aligned}$$

6.1.2.1 Posterior Derivations and Full Conditionals The full joint posterior is:

$$p(\lambda_i, p_i, \{n_{ijk}\} \mid \{y_{ijk}\}) \propto p(\{y_{ijk}\} \mid \{n_{ijk}\}, p_i)\, p(\{n_{ijk}\} \mid \lambda_i)\, p(p_i)\, p(\lambda_i)$$

We now derive the full conditional distributions for each parameter.

6.1.2.2 Posterior for p_i: This is derived from the Binomial likelihood and the Beta prior on p_i:

$$\begin{aligned} p(p_i \mid \cdots) &\propto \prod_j \binom{n_{ijk}}{y_{ijk}} (p_i c_k)^{y_{ijk}} (1 - p_i c_k)^{n_{ijk} - y_{ijk}} \cdot p(p_i) \\ &\propto \text{Beta}\left(a_i + \sum_j y_{ijk},\ b_i + \sum_j (n_{ijk} - y_{ijk}) \right) \end{aligned}$$

6.1.2.3 Posterior for n_{ijk}: We derive this from the Binomial likelihood and Poisson prior on n_{ijk}:

$$\begin{aligned} p(n_{ijk} \mid \cdots) &\propto \prod_j \binom{n_{ijk}}{y_{ijk}} p_{ik}^{y_{ijk}} (1 - p_{ik})^{n_{ijk} - y_{ijk}} \cdot \frac{\lambda_i^{n_{ijk}} e^{-\lambda_i}}{n_{ijk}!} \\ &\propto \prod_j \frac{((1 - p_{ik})\lambda_i)^{n_{ijk}}}{(n_{ijk} - y_{ijk})!} \end{aligned}$$

This simplified form results from algebraic cancellation of common terms across the Binomial and Poisson components.

6.1.2.4 Posterior for λ_i: This is derived from the Poisson likelihood on n_{ijk} and the Jeffreys prior:

$$\begin{aligned} p(\lambda_i \mid \cdots) &\propto \prod_j \frac{\lambda_i^{n_{ijk}} e^{-\lambda_i}}{n_{ijk}!} \cdot \lambda_i^{-1/2} \\ &\propto \lambda_i^{\sum_j n_{ijk} - 1/2} e^{-\lambda_i J_k} \\ &\sim \text{Gamma}\left(\sum_j n_{ijk} - \tfrac{1}{2},\ J_k \right) \end{aligned}$$

where we let J_k denote the number of games player i played against opponent k.

6.1.2.5 Marginal Likelihood (up to proportionality) The marginal likelihood under Model 2 can be expressed as:

$$\begin{aligned} p(y \mid M_2) = \sum_{\{n\}} \int_{\lambda_i} \int_{p_i} & \prod_j \binom{n_{ijk}}{y_{ijk}} p_{ik}^{y_{ijk}} (1 - p_{ik})^{n_{ijk} - y_{ijk}} \\ & \cdot \frac{\Gamma(a_i)\Gamma(b_i)}{\Gamma(a_i + b_i)} p_i^{a_i - 1} (1 - p_i)^{b_i - 1} \\ & \cdot \prod_j \frac{\lambda_i^{n_{ijk}} e^{-\lambda_i}}{n_{ijk}!} \cdot \lambda_i^{-1/2} \, d\lambda_i \, dp_i \end{aligned}$$

Evaluating this integral analytically is infeasible in practice, so we shall approximate through the use of R.

6.1.2.6 Marginal Likelihood (up to proportionality) The marginal likelihood under Model 1 can be expressed as:

$$
\begin{aligned}
p(y \mid M_1) = \sum_{\{n\}} \int_{p_i} \prod_j \binom{n_{ijk}}{y_{ijk}} p_{ik}^{y_{ijk}} (1 - p_{ik})^{n_{ijk} - y_{ijk}} \\
\cdot \frac{\Gamma(a_i)\Gamma(b_i)}{\Gamma(a_i + b_i)} p_i^{a_i - 1} (1 - p_i)^{b_i - 1} dp_i
\end{aligned}
$$

Evaluating this integral analytically is infeasible in practice, so we shall approximate through the use of R.

6.1.3 Model 3

Given the model below, we make the following derivations.

$$
\begin{aligned}
y &\sim Binom(n_i, p_{ik}) \\
n_i &\sim NegBinom(r_i, \theta_i) \\
p(r_i, \theta_i) &= \sqrt{\frac{r_i}{\theta_i^2 (1 - \theta_i)}} \\
p_{ik} &\sim c_k * beta(\alpha_i, \beta_i)
\end{aligned}
$$

6.1.3.1 Full Conditional for p_i Given the Binomial likelihood and Beta prior, the conditional distribution of p_i is:

$$
\begin{aligned}
p(p_i \mid \cdots) &\propto \prod_{j,k} \binom{n_{ijk}}{y_{ijk}} (p_i c_k)^{y_{ijk}} (1 - p_i c_k)^{n_{ijk} - y_{ijk}} \cdot p(p_i) \\
&\propto \text{Beta}\left(a_i + \sum_{j,k} y_{ijk},\ b_i + \sum_{j,k} (n_{ijk} - y_{ijk}) \right)
\end{aligned}
$$

6.1.3.2 Full Conditional for n_{ijk} We derive this from the Binomial likelihood and the Negative Binomial prior:

$$
\begin{aligned}
n_{ijk} \mid \cdots &\propto p(y_{ijk} \mid p_{ik}, n_{ijk})\, p(n_{ijk} \mid r_i, \theta_i) \\
&\propto \prod_j \binom{n_i}{y_{ijk}} {p_{ik}}^{y_{ijk}} (1 - p_{ik})^{n_i - yijk} \binom{n_{ijk} + r_i - 1}{n_{ijk}} (1 - \theta_i)^{n_{ijk}} \theta_i^{r_i} \\
&\propto (\prod_j \frac{(n_{ijk} + r_i - 1)!}{(n_{ijk} - y_{ijk})!})((1 - \theta_i)(1 - p_{ik}))^{\sum_j n_{ijk}}
\end{aligned}
$$

Where we compute the log:

$$
\propto \ln\left(\prod_{j,k} (n_{ijk} + r_i - 1)! \right) - \ln\left(\prod_{j,k} (n_{ijk} - y_{ijk})! \right) + \ln\left(\prod_{j,k} \left[(1 - \theta_i)(1 - p_{ik}) \right]^{n_{ijk}} \right)
$$

This form is used to compute the log full conditional for use in a Metropolis-Hastings sampler.

6.1.3.3 Full Conditional for r_i The full conditional for r_i arises from the Negative Binomial likelihood and the prior $p(r_i, \theta_i)$:

$$\begin{aligned} p(r_i \mid \cdots) &\propto \prod_{j,k} \binom{n_{ijk} + r_i - 1}{n_{ijk}} \theta_i^{r_i} (1-\theta_i)^{n_{ijk}} \cdot \sqrt{\frac{r_i}{\theta_i^2(1-\theta_i)}} \\ &\propto \prod_{j,k} \frac{(n_{ijk}+r_i-1)!}{(r_i-1)!\, n_{ijk}!} \cdot \theta_i^{r_i} \cdot r_i^{1/2} \end{aligned}$$

Again, implementation is done using a Metropolis step on the log scale. The log unnormalized conditional is:

$$\log p(r_i \mid \cdots) = \sum_{j,k} \log\left(\frac{(n_{ijk}+r_i-1)!}{(r_i-1)!}\right) + r_i \log(\theta_i) + \tfrac{1}{2}\log(r_i)$$

6.1.3.4 Full Conditional for θ_i We derive this from the Negative Binomial likelihood and the Jeffreys prior:

$$\begin{aligned} p(\theta_i \mid \cdots) &\propto \prod_{j,k} \binom{n_{ijk} + r_i - 1}{n_{ijk}} (1-\theta_i)^{n_{ijk}} \theta_i^{r_i} \cdot \sqrt{\frac{r_i}{\theta_i^2(1-\theta_i)}} \\ &\propto (1-\theta_i)^{\sum_{j,k} n_{ijk}} \cdot \theta_i^{r_i} \cdot \theta_i^{-1} (1-\theta_i)^{-1/2} \\ &\propto (1-\theta_i)^{\sum_{j,k} n_{ijk} + 1/2 - 1} \cdot \theta_i^{r_i - 1} \end{aligned}$$

Thus, the full conditional distribution is:

$$\theta_i \mid \cdots \sim \text{Beta}\left(r_i, \sum_{j,k} n_{ijk} + \tfrac{1}{2}\right)$$

6.1.3.5 Marginal Likelihood (up to proportionality) The marginal likelihood under Model 3 can be expressed as:

$$\begin{aligned} p(y \mid M_2) = \sum_{\{n\}} \sum_{\{r\}} \int_{p_i} \int_{\theta_i} \prod_j & \binom{n_{ijk}}{y_{ijk}} p_{ik}^{y_{ijk}} (1-p_{ik})^{n_{ijk} - y_{ijk}} \\ & \cdot \frac{\Gamma(a_i)\Gamma(b_i)}{\Gamma(a_i + b_i)} p_i^{a_i - 1} (1-p_i)^{b_i - 1} \\ & \cdot \prod_j \frac{\lambda_i^{n_{ijk}} e^{-\lambda_i}}{n_{ijk}!} \cdot \lambda_i^{-1/2} \\ & \cdot \binom{n_{ijk} + r_i - 1}{n_{ijk}} (1-\theta_i)^{n_{ijk}} \theta_i^{r_i} \\ & \cdot \sqrt{r_i / \theta_i^2 (1-\theta_i)} d\theta_i \, dp_i \end{aligned}$$

Evaluating this integral analytically is infeasible in practice, so we shall approximate through the use of R.

6.2 Posterior Predictive Simulation:

6.2.1 Model 1

To generate predictions for future field goals made under Model 1, we use the posterior predictive distribution:

$$p(\tilde{y}_{ijk} \mid y) = \int p(\tilde{y}_{ijk} \mid n_i, p_i c_k) \cdot p(p_i \mid y) \, dp_i$$

Since n_i is fixed and $p_i \mid y \sim \text{Beta}(a_i^*, b_i^*)$, we draw posterior predictive samples by:

1. Drawing $p_i^{(s)} \sim \text{Beta}(a_i^*, b_i^*)$
2. Computing $p_{ik}^{(s)} = p_i^{(s)} \cdot c_k$
3. Drawing $\tilde{y}_{ijk}^{(s)} \sim \text{Binomial}(n_i, p_{ik}^{(s)})$

This results in samples from $p(\tilde{y}_{ijk} \mid y)$, allowing prediction intervals and summary statistics.

We simulate posterior predictive distributions for a future game against the Golden State Warriors. This involves adjusting posterior draws of LeBron's baseline shooting percentage p_i using the opponent's defensive rating:

$$p_{\text{GSW}}^{(s)} = p_i^{(s)} \cdot \exp\left[\gamma(\text{DRTG}_{\text{GSW}} - \overline{\text{DRTG}})\right], \quad \gamma = 0.01$$

We use $\gamma = 0.01$ throughout the simulations, as established earlier in the model specification.

We then draw predicted field goals made (FGM) from a Binomial distribution:

$$\text{FGM}^{(s)} \sim \text{Binomial}(n_i, p_{\text{GSW}}^{(s)})$$

We repeat this process for three fixed values of shot attempts: LeBron's mean (18), median (18), and maximum (27) FGA across the season. These simulations isolate how usage affects the distribution of expected scoring outcomes, holding shooting efficiency constant.

The resulting posterior predictive distributions are shown below. Each histogram reflects uncertainty in both LeBron's shooting ability and the variability induced by matchup effects. The red vertical line marks the posterior predictive mean for each scenario.

These simulations illustrate how assumptions about shot volume influence predictive outcomes, even when the estimated FG% is held constant. Notably, the location and spread of each distribution shifts with n, reflecting how higher attempt counts increase the range and potential variability in FGM.

6.2.2 Model 2

The posterior predictive distribution under Model 2 accounts for uncertainty in both p_i and n_{ijk}:

$$p(\tilde{y}_{ijk} \mid y) = \iint p(\tilde{y}_{ijk} \mid \tilde{n}_{ijk}, p_i c_k) \cdot p(\tilde{n}_{ijk} \mid \lambda_i) \cdot p(p_i, \lambda_i \mid y) \, d\tilde{n}_{ijk} \, dp_i \, d\lambda_i$$

This is approximated by:

1. Drawing $p_i^{(s)} \sim \text{Beta}(a_i^*, b_i^*)$
2. Drawing $\lambda_i^{(s)} \sim \text{Gamma}(\alpha, \beta)$ from its posterior

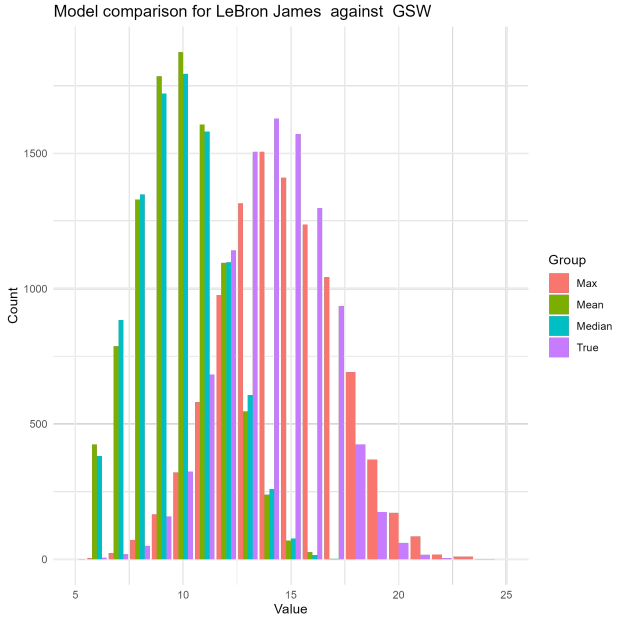

Figure 10: 3.3.1

3. Drawing $\tilde{n}_{ijk}^{(s)} \sim \text{Poisson}(\lambda_i^{(s)})$
4. Computing $p_{ik}^{(s)} = p_i^{(s)} \cdot c_k$
5. Drawing $\tilde{y}_{ijk}^{(s)} \sim \text{Binomial}(\tilde{n}_{ijk}^{(s)}, p_{ik}^{(s)})$

These steps simulate $\tilde{y}_{ijk} \mid y$ under full parameter uncertainty.

To generate posterior predictive draws under Model 2, we simulate future field goals made (FGM) for LeBron James in a game against the Golden State Warriors (GSW). Unlike Model 1, which assumes a fixed number of field goal attempts, Model 2 treats the shot attempts n_{ijk} as Poisson-distributed with player-specific rate λ_i. We account for both uncertainty in shot attempts and matchup-adjusted shooting efficiency.

Posterior samples for p_i, λ_i, and n_{ijk} are obtained from the MCMC output of `mcmc_model_2()`. For each posterior iteration $s = 1, \ldots, S$, we compute:

1. The matchup-adjusted shooting probability against GSW:

$$p_{\text{GSW}}^{(s)} = p_i^{(s)} \cdot \exp\left[\gamma(\text{DRTG}_{\text{GSW}} - \overline{\text{DRTG}})\right]$$

2. A new Poisson draw for the number of shot attempts:

$$n_{\text{new}}^{(s)} \sim \text{Poisson}(\lambda_i^{(s)})$$

3. A Binomial draw for field goals made in the simulated game:

$$y_{\text{new}}^{(s)} \sim \text{Binomial}(n_{\text{new}}^{(s)}, p_{\text{GSW}}^{(s)})$$

This process captures both the game-to-game variability in LeBron's shot volume and the impact of opposing defense on his shooting efficiency. The posterior predictive distribution of y_{new} reflects realistic uncertainty for future performance.

In our implementation, we set the tuning parameter $\gamma = 0.01$, as chosen in prior model evaluation. The opponent's defensive rating is taken from our centered DRTG proxy, and simulations use the full posterior from 10,000 MCMC iterations.

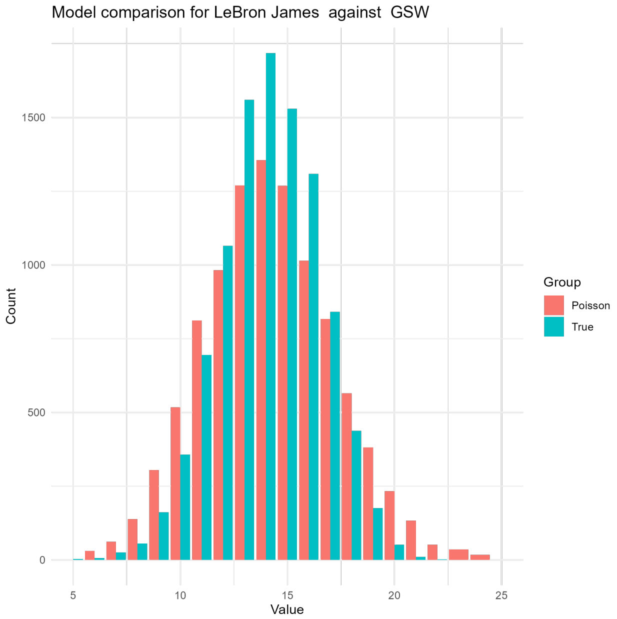

Figure 11: 3.3.2

6.2.3 Model 3

The posterior predictive distribution under Model 3 incorporates overdispersion in n_{ijk}:

$$p(\tilde{y}_{ijk} \mid y) = \iiint p(\tilde{y}_{ijk} \mid \tilde{n}_{ijk}, p_i c_k) \cdot p(\tilde{n}_{ijk} \mid r_i, \theta_i) \cdot p(p_i, r_i, \theta_i \mid y) \, d\tilde{n}_{ijk} \, dp_i \, dr_i \, d\theta_i$$

We approximate this by:

1. Drawing $p_i^{(s)} \sim \text{Beta}(a_i^*, b_i^*)$
2. Drawing $r_i^{(s)}$, $\theta_i^{(s)}$ from their posterior samples
3. Drawing $\tilde{n}_{ijk}^{(s)} \sim \text{NegBin}(r_i^{(s)}, \theta_i^{(s)})$
4. Computing $p_{ik}^{(s)} = p_i^{(s)} \cdot c_k$
5. Drawing $\tilde{y}_{ijk}^{(s)} \sim \text{Binomial}(\tilde{n}_{ijk}^{(s)}, p_{ik}^{(s)})$

These simulated draws reflect posterior predictive uncertainty in all model components.

Under Model 3, both the number of shot attempts n_{ijk} and the player's shooting efficiency p_{ik} are treated as random, with the number of attempts modeled as Negative Binomial. This model allows for overdispersion in shot volume and incorporates uncertainty from the MCMC-estimated parameters r_i, θ_i, p_i. We simulate LeBron's field goals made (FGM) in a future game against the Golden State Warriors using the following steps:

For each posterior iteration $s = 1, \ldots, S$, we:

1. Adjust LeBron’s shooting efficiency for opponent strength:

$$p_{\text{GSW}}^{(s)} = p_i^{(s)} \cdot \exp\left[\gamma(\text{DRTG}_{\text{GSW}} - \overline{\text{DRTG}})\right]$$

2. Draw number of shot attempts from the Negative Binomial distribution parameterized by posterior samples:

$$n_{\text{new}}^{(s)} \sim \text{NegBin}(r_i^{(s)}, \theta_i^{(s)})$$

3. Simulate field goals made using the adjusted success probability:

$$y_{\text{new}}^{(s)} \sim \text{Binomial}(n_{\text{new}}^{(s)}, p_{\text{GSW}}^{(s)})$$

The resulting posterior predictive distribution of y_{new} captures uncertainty in shot volume (via n), shooting efficiency (via p), and opponent strength (via centered DRTG).

As in previous sections, we use $\gamma = 0.01$ and draw opponent defensive strength from the centered DRTG proxy for Golden State. Simulations are based on 100,000 iterations from the Metropolis-within-Gibbs sampler described in Section 3.2.3.

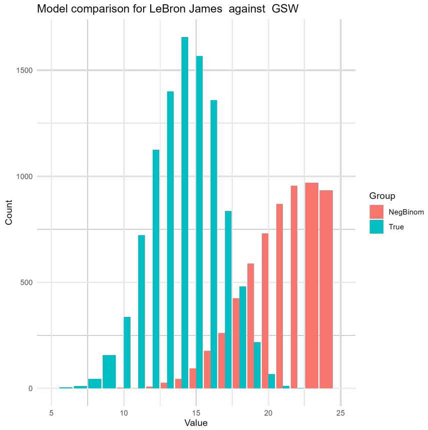

Figure 12: 3.3.3

6.3 Model Comparison Derivations

Model 1 seems to draw almost all of the BF weight, as it gives a very close approximation at both the center, and on the tails of the distribution

$$p(y \mid M_1) = \frac{\Gamma(a_i)\Gamma(b_i)}{\Gamma(a_i + b_i)} \sum_{n_i} \prod_j \binom{n_i}{y_{ijk}} \cdot \frac{\Gamma\left(\sum_j y_{ijk} + a_i - 1\right)\Gamma\left(\sum_j (n_i - y_{ijk}) + b_i - 1\right)}{\Gamma\left(\sum_j n_i + a_i + b_i - 2\right)}$$

$$\hat{\Pi}_{M_1} = P(y|M_1) / \sum_{M^* \in M} P(y|M^*) \overset{\text{R approximate}}{\rightarrow} 0.9913269$$

Model 2 is given very little weight as it’s model includes a large amount of variance.

$$p(y \mid M_2) = \frac{\Gamma(a_i)\Gamma(b_i)}{\Gamma(a_i + b_i)} \sum_{n_i} \prod_j \binom{n_i}{y_{ijk}} \cdot \left(\frac{1}{n_{ijk}!} \Gamma(n_{ijk} + 1/2) \right) \cdot \frac{\Gamma\left(\sum_j y_{ijk} + a_i - 1\right) \Gamma\left(\sum_j (n_i - y_{ijk}) + b_i - 1\right)}{\Gamma\left(\sum_j n_i + a_i + b_i - 2\right)}$$

$$\hat{\Pi}_{M_2} = P(y|M_2) / \sum_{M^* \in M} P(y|M^*) \overset{\text{R approximate}}{\rightarrow} 0.00867307$$

Model 3 is given a value very close to 0. The model's had difficulty converging for all the parameters, and had even higher variance. This is all reflected in the Bayes Factor

$$\begin{aligned} p(y \mid M_3) &= \frac{\Gamma(a_i)\Gamma(b_i)}{\Gamma(a_i + b_i)} \sum_{n_i} \prod_j \binom{n_i}{y_{ijk}} \times \frac{\Gamma\left(\sum_j y_{ijk} + a_i - 1\right) \Gamma\left(\sum_j (n_i - y_{ijk}) + b_i - 1\right)}{\Gamma\left(\sum_j n_i + a_i + b_i - 2\right)} \\ &\times \frac{\sqrt{\pi} 2^{1 - n_{ijk}}}{5} \left(\frac{3 - 2n_{ijk}\Gamma(n_{ijk} + 1)}{\Gamma(n_{ijk} - 1/2)} + \frac{4(n_{ijk} - 3)\Gamma(2n_{ijk} + 1)}{\Gamma(2n_{ijk} - 1/2)} \right) \\ \hat{\Pi}_{M_1} &= P(y|M_1) / \sum_{M^* \in M} P(y|M^*) \overset{\text{R approximate}}{\rightarrow} 0.2713688e - 09 \end{aligned}$$

6.4 A.1 Data Tidying Code

```
# Data cleaning and filtering steps here
original_tbl <- read.csv("./NBA-BoxScores-2023-2024.csv") |>
  mutate(
    START_POSITION = na_if(START_POSITION, "") |> factor(),
    COMMENT = na_if(COMMENT, "") |> factor(),
    MIN = na_if(MIN, ""),
    MIN = str_replace(MIN, "([0-9]+)\\.[0-9]+:", "\\1:")
  )

starting_dat <- original_tbl |> filter(!is.na(START_POSITION))

team_points <- original_tbl |>
  filter(!is.na(PTS)) |>
  group_by(GAME_ID, TEAM_ID) |>
  summarize(TeamPoints = sum(PTS), .groups = "drop")

team_vs_opponent <- team_points |>
  inner_join(team_points, by = "GAME_ID", suffix = c("", ".opp")) |>
  filter(TEAM_ID != TEAM_ID.opp) |>
  rename(OPP_TEAM_ID = TEAM_ID.opp, OpponentPoints = TeamPoints.opp)

drtg <- team_vs_opponent |>
  group_by(TEAM_ID) |>
  summarize(DRTG_proxy = mean(OpponentPoints), .groups = "drop")

game_team_pairs <- original_tbl |> select(GAME_ID, TEAM_ID) |> distinct()

opponent_map <- game_team_pairs |>
  inner_join(game_team_pairs, by = "GAME_ID") |>
  filter(TEAM_ID.x != TEAM_ID.y) |>
  rename(TEAM_ID = TEAM_ID.x, OPP_TEAM_ID = TEAM_ID.y) |>
  left_join(drtg |> rename(OPP_TEAM_ID = TEAM_ID, OPP_DRTG = DRTG_proxy), by = "OPP_TEAM_ID")

mean_drtg <- mean(drtg$DRTG_proxy)

starting_dat <- starting_dat |>
  left_join(opponent_map, by = c("GAME_ID", "TEAM_ID")) |>
  mutate(centered_OPP_DRTG = OPP_DRTG - mean_drtg)
```

6.5 A.3 Method of Moments Code for Deriving Beta Prior

```
# Empirical FG%
fg_pct <- lebron_dat$FGM / lebron_dat$FGA
mean_fg <- mean(fg_pct, na.rm = TRUE)
var_fg <- var(fg_pct, na.rm = TRUE)

# Method of moments estimation for Beta(a,b)
alpha_est <- mean_fg * ((mean_fg * (1 - mean_fg) / var_fg) - 1)
beta_est  <- (1 - mean_fg) * ((mean_fg * (1 - mean_fg) / var_fg) - 1)

# Plot empirical FG% with fitted Beta prior
ggplot(data.frame(FG_PCT = fg_pct), aes(x = FG_PCT)) +
  geom_histogram(aes(y = ..density..), bins = 20, fill = "steelblue", alpha = 0.7) +
  stat_function(fun = dbeta, args = list(shape1 = alpha_est, shape2 = beta_est),
                color = "darkred", size = 1.2, linetype = "dashed") +
  labs(
    title = sprintf("LeBron FG%% vs Fitted Beta(%.1f, %.1f)", alpha_est, beta_est),
    x = "Field Goal Percentage", y = "Density"
  ) +
  theme_minimal()
```

6.6 A.2 MCMC Algorithms

[Clean up code, allow different players to be explored other than LeBron]

```
# Custom samplers for Models 1, 2, 3

# Model 1: Fixed n_i

# Metropolis-Hastings sampler for p_i
log_q = function(theta, y, n) {
  if (theta < 0 || theta > 1) return(-Inf)
  sum(dbinom(y, size = n, prob = theta, log = TRUE)) + dbeta(theta, alpha_est, beta_est, log = TRUE)
}

MH_beta_binom = function(current, prop_sd, n_vec, y_vec, n_iter = 1000) {
  samps = numeric(n_iter)
  for (i in 1:n_iter) {
    proposed = rnorm(1, current, prop_sd)
    logr = log_q(proposed, y_vec, n_vec) - log_q(current, y_vec, n_vec)
    if (log(runif(1)) < logr) current = proposed
    samps[i] = current
  }
  return(samps)
}

set.seed(5440)
lebron_dat <- starting_dat |> filter(PLAYER_ID == 2544)
Y <- lebron_dat$FGM
N_vec <- lebron_dat$FGA

chains <- lapply(c(0.3, 0.5, 0.7), function(init) MH_beta_binom(init, 0.05, N_vec, Y, n_iter = 1000))

# Model 2: Poisson n_i

# Poisson sampler for n_i
log_n_con = function(p, lambda, n, y) {
  if (all(is.nan(log( ((1-p)*lambda)^sum(n) ) - sum(log((factorial(n-y)))) ))){
    return(rep(-Inf,length(y)))
  }else{
    log( ((1-p)*lambda)^sum(n) ) - sum(log((factorial(n-y))))
  }
}
mcmc_model_2 = function(data, player_id, opp_team_id, n_iter=5000,
                        init_lambda = c(), init_n = c(), gamma=0.01) {
  # Gather true data
  player_dat = data[data$PLAYER_ID == player_id, ]
  player_dat = player_dat[player_dat$OPP_TEAM_ID == opp_team_id, ]
  y = player_dat$FGM
  true_n = player_dat$FGA
  def_factor<- exp(gamma*(data$centered_OPP_DRTG[1]))
  if(length(init_lambda) ==0) {
    init_lambda = mean(player_dat$FGA)
  }
  if(length(init_n) ==0) {
    init_n = mean(player_dat$FGA)
  }


  big_N<- length(y)
  lambda<- init_lambda
  n<- rep(init_n,big_N)

  # setting up lists/matrices for returning
  p_matrix<- matrix(NA, nrow=n_iter, ncol=big_N)
  lambda_list<- rep(lambda, n_iter)
  n_matrix<- matrix(NA, nrow=n_iter, ncol=big_N)
  y_new_list <- rep(NA, n_iter)
  n_new_list <- rep(NA, n_iter)

  for (i in 1:n_iter) {
    # sample p
    p_unscaled<- rbeta(big_N, 5 + sum(y), 5 + sum(n-y))
    p<- p_unscaled*def_factor

    # sample lambda
    lambda<- rgamma(1,shape=sum(n)-1/2,rate=big_N)

    # sample n
    n_prop<- rnorm(big_N, n, 1) # the third 1 is a tuning parameter
    logr<- log_n_con(p, lambda, n_prop, y)-log_n_con(p, lambda, n, y)
    for (j in 1:length(logr)) {
      if (is.finite(logr[j]) && log(runif(1))<logr[j]) {
        n[j]<- n_prop[j]
      }
    }

    # generate new values
    n_new = rpois(1, lambda)
    y_new <- rbinom(1, size = n_new, prob = mean(p))

    # save values
    p_matrix[i,] = p
    n_matrix[i,] = n
    lambda_list[i] = lambda
    n_new_list[i] = n_new
    y_new_list[i] =  y_new
  }
  return(data.frame(iteration=1:n_iter,
                    parameter=rep(c(paste("n[",1:big_N,"]", sep=""), "lambda", "n_new", "y_new", paste(
                    value=c(as.numeric(n_matrix),lambda_list,n_new_list,y_new_list, as.numeric(p_matrix
}
# running mcmc
n_iter<- 10000
MCMC_model_2 = mcmc_model_2(data = starting_dat,
                            player_id = 2544, #LeBron
                            opp_team_id = 1610612744, #GSW
                            n_iter=10000,
                            gamma=0.01)

# Model 3: Negative Binomial n_i
log_n_con = function(n, r, theta, y,p) {
  dummy<- sum(log((factorial(n+r-1)))) - sum(log((factorial(n-y)))) + log((1-theta)*((1-p)^sum(n)))
  if (all(is.nan(dummy))){
    return(rep(-Inf,length(n)))
  }else{
    return(dummy)
  }
}
log_r_con = function(n, r, theta, y) {
  dummy<- sum(log((factorial(n+r-1)))) - sum(log((factorial(r-1)))) + log(theta^sum(r)) + log(r^(length
  if (all(is.nan(dummy))){
    return(rep(-Inf,length(r)))
  }else{
    return(dummy)
  }
}
mcmc_model_3 = function(data, player_id, opp_team_id, gamma = 0.01,
                        n_iter=5000, init_r, init_theta, init_n, init_p,
                        prop_r_sd = 3.5, prop_n_sd = 3.5) {
  # Gather true data
  player_dat = data[data$PLAYER_ID == player_id, ]
  player_dat = player_dat[player_dat$OPP_TEAM_ID == opp_team_id, ]
  y = player_dat$FGM
  true_n = player_dat$FGA
  def_factor<- exp(gamma*(data$centered_OPP_DRTG[1]))

  # Start code
  big_N<- length(y)
  r<- init_r
  theta<- init_theta
  n<- rep(init_n,big_N)
  p<- init_p

  # setting up lists/matrices for returning
  r_list<- rep(NA, n_iter)
  theta_list<- rep(NA, n_iter)
  p_matrix<- matrix(NA, nrow=n_iter, ncol=big_N)
  n_matrix<- matrix(NA, nrow=n_iter, ncol=big_N)

  for (i in 1:n_iter) {
    # sample p
    p_unscaled<- rbeta(big_N, 5 + sum(y), 5 + sum(true_n-y))
    p<- p_unscaled*def_factor

    # sample theta
    theta<- rbeta(1,sum(n)+1/2,r-1)

    # sample r
    r_prop<- rnorm(1, r, prop_r_sd) # the third 1 is a tuning parameter
    logr<- log_r_con(n, r_prop, theta, y)-log_r_con(n, r, theta, y)
    if (is.finite(logr)) {
      if (log(runif(1))<logr) {
        r<- r_prop
      }
    }

    # sample n
    n_prop<- rnorm(big_N, n, prop_n_sd) # the third 1 is a tuning parameter
    logr<- log_n_con(n_prop, r, theta, y,p)-log_n_con(n, r, theta, y,p)
    for (j in 1:length(logr)) {
      if (is.finite(logr[j])) {
        if (log(runif(1))<logr[j]) {
          n[j]<- n_prop[j]
        }
      }
    }
    # save values
    r_list[i] = r
    theta_list[i] = theta
    p_matrix[i,] = p
    n_matrix[i,] = n
  }
  # return(data.frame(iteration=1:n_iter,
  #                   r = r_list,
  #                   theta = theta_list,
  #                   p1 = p_matrix[,1],
  #                   p2 = p_matrix[,2],
  #                   p3 = p_matrix[,3],
  #                   n1 = n_matrix[,1],
  #                   n2 = n_matrix[,2],
  #                   n3 = n_matrix[,3]))
  return(data.frame(iteration=1:n_iter,
                    parameter=rep(c("r","theta",paste("p[",1:big_N,"]", sep=""),paste("n[",1:big_N,"]",
                    value=c(r_list,theta_list,as.numeric(p_matrix),as.numeric(n_matrix))))
}
```

7 References

- Kaggle dataset: NBA Boxscore - Season 2023 / 2024 by Alberto Filosa
- Project GitHub repo: https://github.com/ArgentCode/Stat5440Project

[Include any relevant textbooks or class notes referenced]