2005-08-16

Using Box-Scores to Determine a Position's Contribution to Winning Basketball Games

Garritt L. Page
Brigham Young University - Provo

USING BOX-SCORES TO DETERMINE A POSITION'S CONTRIBUTION

TO WINNING BASKETBALL GAMES

by

Garritt L. Page

A Project submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Statistics

Brigham Young University

August 2005

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a Project submitted by

Garritt L. Page

This Project has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Dr. Gilbert W. Fellingham, Chair
Date	Dr. C. Shane Reese
Date	Dr. Bruce J. Collings

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the Project of Garritt L. Page in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Dr. Gilbert W. Fellingham
Chair, Graduate Committee

Accepted for the Department

G. Bruce Schaalje
Graduate Coordinator

Accepted for the College

G. Rex Bryce
Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

USING BOX-SCORES TO DETERMINE A POSITION'S CONTRIBUTION TO WINNING BASKETBALL GAMES

Garritt L. Page

Department of Statistics

Master of Science

Basketball is a sport that has become increasingly popular world-wide. At the professional level it is a game in which each of the five positions has a specific responsibility that requires unique skills. It seems likely that it would be valuable for coaches to know which skills for each position are most conducive to winning. Knowing which skills to develop for each position could help coaches optimize each player's ability by customizing practice to contain drills that develop the most important skills for each position that would in turn improve the teams overall ability. Through the use of Bayesian hierarchical modeling and NBA box-score performance categories, this project will determine how each position needs to perform in order for their team to be successful.

Acknowledgements

I would like to thank Dr. Fellingham and Dr. Reese for their guidance and suggestions that ultimately made this project a successful learning experience. Also I would like to thank all those who participated in countless brainstorming discussions from which many good ideas were generated. Finally I would like to thank Elizabeth, my wife, who bravely endured the Provo experience.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

Basketball is a sport that has become increasingly popular world-wide. At the professional level it is a game in which each of the five positions has a specific responsibility that requires unique skills. In order for a team to be successful the team members should recognize their roles and combine them so as to play as a single unit (Bray and Brawley 2002). It seems likely that it would be valuable for coaches to know which skills for each position are most conducive to winning. Knowing which skills to develop for each position could help coaches optimize each player's ability by customizing practice to contain drills that develop the most important skills for each position, which would in turn improve the team's overall ability.

The purpose of this project is to determine which skills a particular basketball player needs at his position to help his team be successful. For example, under certain offensive systems, a point guard's main objective would be to direct the team's offense, which often requires good passing and ball handling skills. A coach could then include drills to develop the skills relevant to being a point guard.

It seems reasonable that some abilities have varying importance according to position. For example, a turnover from a guard could possibly be more detrimental to the outcome of the game than a turnover from the center position. A turnover from a guard can occur on the perimeter, which leads to a fast break opportunity for the opposing team. A turnover from a center generally occurs near the basket where the offense can make the transition to defense faster. Also, it seems plausible that defensive rebounds from the center position are more important to the outcome of the game than a defensive rebound from the shooting guard. If a center allows the opposition an offensive rebound it usually leads to an easy basket attempt, whereas a guard allowing an offensive rebound usually results in resetting the offense, which gives the defense time to reestablish itself.

Each coach has a different style of play or coaching philosophy which can affect a player's contribution. For example, Larry Brown of the Detroit Pistons is vocal about his dislike of point guards that try to score more often than pass. Under Coach Brown's system a high scoring point guard is not desirable. The relative abilities of the other players on the team also should be considered. A team with a dominant center may not need as much point production from other positions.

The results of NBA games are summarized in a box-score (see Table 1.1). In these box-scores the final score of the game, where the game was played, and each player's totals for thirteen performance categories are given. These thirteen categories are field goals made, field goal attempts, free throws made, free throw attempts, offensive rebounds, defensive rebounds, total rebounds, assists, personal fouls, steals, turnovers, and points. Through USA Today's web site, *http://www.usatoday.com/*, we were able to obtain box scores for approximately 18,000 NBA regular season games from the 1996-1997 season to the 2003-2004 season.

LA LAKERS (78) AT UTAH (104)
LA LAKERS

PLAYER	POS	MIN	FGM	FGA	FTM	FTA	REBOUNDS OFF	REBOUNDS DEF	REBOUNDS TOT	AST	PF	ST	TO	PTS
======	==	===	===	===	===	===	===	===	===	===	==	==	==	===
C BUTLER	F	24	1	8	2	2	1	2	3	0	2	0	0	4
L ODOM	F	35	3	10	8	10	2	7	9	1	5	1	3	14
C MIHM	C	24	3	8	0	2	2	3	5	0	4	0	3	6
C ATKINS	G	19	0	2	2	2	0	1	1	1	3	0	2	2
K BRYANT	G	41	9	21	16	20	1	3	4	1	5	1	2	38
T BROWN		23	0	3	1	2	1	2	3	2	0	0	0	1
B COOK		17	1	5	1	2	0	2	2	0	4	0	0	3
J JONES		15	1	2	0	0	1	1	2	1	4	0	0	2
B GRANT		10	0	1	1	2	1	1	2	0	3	0	2	1
L WALTON		12	1	3	0	0	1	0	1	0	1	0	0	3
K RUSH		17	0	4	0	0	3	0	3	0	0	2	1	0
S VUJACIC		3	1	1	2	2	0	2	2	1	1	0	1	4
TOTALS		240	20	68	33	44	13	24	7	7	32	4	14	78

UTAH

PLAYER	POS	MIN	FGM	FGA	FTM	FTA	REBOUNDS OFF	REBOUNDS DEF	REBOUNDS TOT	AST	PF	ST	TO	PTS
======	==	===	===	===	===	===	===	===	===	===	==	==	==	===
A KIRILENKO	F	38	5	6	6	7	2	4	6	1	2	1	5	16
C BOOZER	F	34	10	13	7	10	5	6	11	3	1	0	0	27
J COLLINS	C	10	0	1	0	0	0	1	1	0	4	0	0	0
K MCLEOD	G	29	2	7	2	2	0	0	0	8	5	1	1	6
G GIRICEK	G	12	4	9	3	3	0	1	1	0	4	0	2	11
M OKUR		20	0	4	2	2	1	4	5	4	1	1	0	2
R BELL		25	4	8	2	2	0	3	3	0	5	0	1	10
M HARPRING		28	9	11	3	3	2	5	7	4	4	1	2	23
H EISLEY		19	2	7	0	0	0	0	0	3	1	0	0	4
C BORCHARDT		14	2	3	1	2	1	5	6	1	2	0	1	5
K SNYDER		9	0	3	0	0	0	3	3	0	2	0	0	0
K HUMPHRIES		2	0	3	0	0	0	0	0	0	0	0	0	0
TOTALS		240	38	75	26	31	11	32	43	24	31	4	12	104

Table 1.1: Typical NBA Box-Score

In these box-scores only the players that started had their position identified. Therefore, our analysis will be constrained to include only players that started the game. In addition, these box-scores make no distinction between a point guard and a shooting guard or between a small forward and a power forward. Since point guards and shooting guards usually have vastly different roles within the framework of the team we wanted to be able to distinguish between the two. So based on our personal recollection and using the internet as a further resource we separate the guards into point guards and shooting guards and the forwards into small forwards and power forwards.

We will approach this problem from a hierarchical Bayesian point of view and will model the difference in the points scored, by the two competing teams, as a function of the difference of nine performance categories found in the box-score.

Chapter 2

Literature Review

The following literature review can be divided into three sections. The first section is a general review of Bayesian methods. The second is a review of papers written in a sports statistics framework. The third is a review of Bayesian regression models.

2.1 Bayesian Statistics

The Reverend Thomas Bayes is generally given credit for discovering Bayes' Theorem because of a paper that was published after his death (Bayes 1764). The theorem, for two simple events A and B, states:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}. \tag{2.1}$$

Stigler (1983) addresses the issue that Bayes might have erroneously been given credit for first discovering the theorem. Hartley (1749) reveals that a friend of his found a solution for the inverse probability problem (that is, inverting the conditionality in probability). Hartley then provides Bayes' Theorem and an explanation of the importance of the discovery. Since Hartley's book was published

12 years before Thomas Bayes' death, Stigler concludes that it is likely that Bayes was not the first person who discovered the theorem that carries his name.

Bayes' paper still gives a good example of the use of Bayes' Theorem to estimate an unknown parameter. The transition from using simple events A and B to estimating parameters using Bayes' Theorem is done by simply substituting a vector of parameters θ for B and a vector of data y for A. Therefore, for parameter estimation Bayes' Theorem becomes:

$$\pi(\theta|y) = \frac{f(y|\theta)\pi(\theta)}{f(y)} \tag{2.2}$$

$$= \frac{f(y|\theta)\pi(\theta)}{\int f(y|\theta)\pi(\theta)\, d\theta}, \tag{2.3}$$

where $\pi(\theta|y)$ is called the posterior distribution of the parameters given the data, $f(y|\theta)$ is the likelihood (sometimes referred to as the sampling density or the distribution of the data given the parameters), $\pi(\theta)$ is the prior distribution of the parameters, and $f(y)$ is the marginal density of the data. The denominators from equation (2.2) and (2.3) are equal by the law of total probability and are often referred to as the normalizing constant.

Bayes constructed a binomial sampling density with an unknown probability of success (θ). He first estimated θ by rolling a ball on a billiard table and recording its horizontal coordinate, constrained to be between 0 and 1. He argued that a priori each horizontal coordinate is just as likely to be the resting place for the ball. Then a second ball was rolled n times and a success occured if the ball came to rest to the left of the first ball thrown. The random variable X was the number of successes. Thus the sampling density of X would be $\binom{n}{p}\theta^p(1-\theta)^{n-p}$. Bayes was ultimately interested in finding the probability of θ being between two values given p successes. Since $P(a < \theta < b \cap X = p) = \int_b^a \binom{n}{p}\theta^p(1-\theta)^{n-p}d\theta$ and $P(X = p) = \int_0^1 \binom{n}{p}\theta^p(1-\theta)^{n-p}d\theta$, we can use Bayes' Theorem to show:

$$P(a < \theta < b|X = p) = \frac{\int_b^a \binom{n}{p}\theta^p(1-\theta)^{n-p}d\theta}{\int_0^1 \binom{n}{p}\theta^p(1-\theta)^{n-p}d\theta}.$$

Here Bayes assumed a uniform distribution for θ so $\pi(\theta) = 1$. He justified the analogy and uniform assumption in a scholium to his 1764 paper. (For a complete discussion on Bayes' analogy and details on his probability propositions and controversies surrounding his 9th scholium see Stigler (1982) and Schafer (1982).)

Laplace independently discovered Bayes' Theorem in the late 18$^{\text{th}}$ century (Stigler 1986). He applied Bayes' Theorem to estimate parameters in a binomial and hypergeometric setting. He assumed the probability of success in the binomial setting to be uniform, similar to Bayes, but did not give any justification for the assumption. In the hypergeometric case, he assumed that sampling from one of two possible urns, say A and B, is equally likely and then showed that Bayes' Theorem gives the correct result.

Even though Bayes' Theorem was discovered in the 18$^{\text{th}}$ Century, as a modeling technique it was not widely used until the late 20$^{\text{th}}$ Century because of the difficulty of implementing even basic models. Consider the case of modeling a phenomena with a normal likelihood where the mean and variance are unknown and are given a normal and an inverse gamma as prior distributions. Even in this simple case obtaining the normalizing constant requires evaluating the non-trivial integral found in the denominator of equation(2.3). Because of the difficulty of implementation, Bayesian methods were not practical for most of the 20^{th} Century.

The impracticality of Bayesian methods was remedied in the mid 20^{th} Century unbeknownst to statisticians. Metropolis et al. (1953) created an algorithm implementing Markov Chain Monte Carlo (MCMC) and an acceptance/rejection rule that would generate draws from a target distribution under certain conditions. The first step is to choose a starting point for the parameter vector which will be noted as θ^0. The choice of starting value is not particularly important so long as the selection is not extremely strange (for example, choosing a negative value as the starting value for a variance parameter.) The second step is to draw a candidate parameter vector θ^{cand} from some symmetric distribution, where a symmetric distribution is on in which $f(\theta_a|\theta_b) = f(\theta_b|\theta_a)$ for all θ_a and θ_b. Then the ratio of the densities,

$$r = \frac{g(\theta^{cand}|y)}{g(\theta^{t-1}|y)} \tag{2.4}$$

is calculated, where $g(\cdot)$ is the numerator from equation(2.3) and is often referred to as the un-normalized posterior distribution. Finally, set

$$\theta^t = \begin{cases} \theta^{cand} & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise.} \end{cases}$$

Then repeat the algorithm beginning with the second step.

Hastings (1970) was able to generalize Metropolis' algorithm by relaxing the requirement that the distribution from which θ^{cand} is drawn has to be symmetric.

The ratio r then becomes

$$r = \frac{g(\theta^{cand}|y)/p(\theta^{cand}|\theta^{t-1})}{g(\theta^{t-1}|y)/p(\theta^{t-1}|\theta^{cand})}. \tag{2.5}$$

This method of posterior estimation is known as the Metropolis-Hastings algorithm.

Geman and Geman (1984) developed a special case of the Metropolis-Hastings algorithm called Gibbs sampling (see Gelman et al. 2004, chap. 11), which is useful in many multidimensional statistical models. In this algorithm the complete conditionals (i.e. conditional distributions) are used. They are defined as:

$$[\theta_j] = (\theta_j|\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_k). \tag{2.6}$$

Thus the complete conditional for θ_j is the value of θ_j conditioned on the other k parameters in the model. The Gibbs sampler can be implemented if a complete conditional is in the form of a distribution from which values can be easily drawn. If this is the case then the algorithm cycles through all of the complete conditionals in one iteration, therefore there are k steps for each iteration. Each complete conditional is updated on the latest values of the other complete conditionals which are at iteration t if they have already been updated, otherwise they are at iteration t-1.

Inference using iteratively simulated draws, like MCMC, requires some diagnostic checks to insure convergence to the target distribution. Kass et al. (1998) talk about different ways to check for convergence. They discuss the use of $\hat{R}$ which is the "...estimated posterior variance of the parameter, based on the mixture of all the simulated sequences, divided by the average of the variances with in each sequence." They suggest running different chains by using different starting points until $\hat{R}$ is close to 1 for all parameters.

Two other areas of concern are the number of iterations needed to fully explore the target distribution, and which iterates within the initial transient stage, often referred to as the burn, should be discarded. Raftery and Lewis (1992) determine the burn length and how many iterations are needed in order to estimate a quantile of the posterior distribution within some pre-specified range and a probability that the quantile lies in the pre-specified range. They implement this strategy in their *gibbsit* software, in which they give the necessary iterations and burn to obtain a desired posterior quantile and a probability that it is within a desired range.

With the advance of computational methods, Bayesian methods are becoming more widely used and application to more complicated models is possible. Hierarchical models, which often estimate large numbers of parameters, are a natural progression in Bayesian analysis. These types of models allow us to estimate more parameters than there are data points. If the parameters come from the same population we can combine information for parameter estimation and "borrow strength" from the other parameters that follow the same distribution. Reese et al. (2001) implement a hierarchical model to estimate the fetal growth and gestation time for bowhead whales of the Bering, Chukchi, and Beaufort Seas

(BCBS) stock. They estimate the length of a fetus at birth, the growth rate, and the conception date for each whale. Solving this problem necessitated estimating $3n + 2$ parameters with only n data points, which is possible using a Bayesian hierarchical model. They then model the fetal age as a function of growth rate, conception date and length of fetus at birth.

2.2 Sports Statistics

The following is a brief review of some methods that have been used to answer questions relating to sports. The first three papers use a Bayesian approach, the next three papers involve the sport of basketball, and the final papers are general sports papers.

2.2.1 Bayesian Methods Applied to Sports Data

Berry et al. (1999) propose a method to characterize players' talents regardless of era for baseball, golf, and hockey. This is difficult to do because players' abilities are confounded with era, competition, and improvements in equipment among other things. They construct a logit additive Bayesian hierarchical model with a parameter for players ability, age, and era. Through these parameters they are able to bridge players from different eras by career overlap, take into account the competition of a particular era, and model the deterioration of skills due to aging. They use 1996 as a "benchmark" year. All evaluations of players are relative to this season. To estimate a league-wide age effect they compare each player's actual performance as they age to their estimated ability. The difficulty of a particular season is estimated by comparing a player's performance in the season with their estimated ability and estimated age effect. They found that golf players have the most longevity and skill deteriorates the most rapidly in hockey players.

Bayesian hierarchical methods are also used by Graves et al. (2003) to model permutations. The permutation of interest was the order in which auto racing drivers finish a race. First they estimate a driver's ability where θ_{ij} is the i^{th} drivers ability in the j^{th} race. Then a finishing order is determined by choosing a driver to finish in last place with a probability proportional to $\lambda_{ij} = \exp(-\theta_{ij})$. Then a second-to-last-place finisher is chosen from the remaining drivers with probability proportional to $\exp(-\theta_{ij})$. This process is repeated until only two drivers remain, from which a second-place finisher is chosen with probability $\frac{\lambda_{i1j}}{(\lambda_{i1j}+\lambda_{i2j})}$. This produces an "attrition model", from which they are able to construct a likelihood. The model closely resembles what actually happens in auto-racing, that is, drivers often drop out one by one. With this model they are able to estimate a driver's racing ability, calculate the probability of a driver winning from the current standing in which the driver is at, estimate how a driver's ability varies from track to track, and predict which drivers will be successful in the future.

In golf, deciding between a 3 iron or a 7 iron for a tee shot on a par 3 hole of length 200 yards with a green that is surrounded by water is usually made based on past performance, current conditions, and intuition. Zellner (1999) uses a Bayesian analysis on this interesting golf problem. He assigns probabilities for the number of strokes needed to complete the hole for each club, and argues that using uniform probabilities or long run frequency would be inadequate in this situation. Rather, an appropriate utility function tailored to each score is used to choose the club for which the expected score is lower.

2.2.2 Studies on the Sport of Basketball

A topic of debate in many sports is the notion of a "hot hand." This is claimed to occur when a player performs significantly above his average for an extended number of plays. Tversky and Gilovich (1989) address this topic for the sport of basketball. They argue that if a hot hand exists then there would be correlation within a sequence of shots. That is, making one shot would increase the chance of making the next shot. They perform an experiment by recording foul shots of some university athletes to get an estimate of the correlation between shots. They find that making a basket is largely independent of the outcome of the previous shot attempted.

Bishop and Gajewski (2004) entertain the idea of being able to determine if a college player would be a success in the NBA. They use principal components, logistic regression, and cross validation to predict a players draftability. Each potential draft pick is rated by their physical size and the quality of their basketball skills. The rank of a player's ability is derived from factors similar to those found in a box-score. To reduce the number of variables, they use principle components to identify linear combinations that best explain the variance in the data and use these values to obtain a physical size ranking and an ability ranking. Then they use logistic regression and cross validation to predict a player's draftability. They are successful in some predictions. For example, during the 2000-2001 season Jason Gardner from Arizona was selected as an All-American, but they predicted that he would not be a very successful player in the NBA, and in fact he was never drafted.

Every March the National Collegiate Athletic Association (NCAA) Men's Division I Basketball Tournament is held to determine the national champion. Teams are assigned a number 1 through 16 according to their ability within each of four regions. These numbers are referred to as seeds. 1 seeds are paired with 16 seeds, 2 seeds are paired with 15 seeds, and so forth. Since the tournament's current format was implemented no 16-seed has beaten a 1-seed. Stern (1998) tries to determine if a 16-seed will ever beat a 1-seed in the NCAA Men's Division I Basketball Tournament. He uses the point spread as a reasonable measure to estimate the difference in the abilities of the two teams. The difference between the actual game outcome and the point spread is found to roughly follow a Gaussian distribution. So the difference in score can be thought of as normal random variables with mean equal to the point spread and the standard deviation equal to 11. To estimate the probability of an i-seed defeating a j-seed, a logistic regression model is used to estimate the probability as a function of the difference in seeding.

He finds that a 16 seed will eventually defeat a 1-seed if the NCAA continues with its present format. In fact there is a 1 in 100 chance of it eventually happening.

2.2.3 General Sports Papers

Berry (2001) talks about the amateur drafts for the four major sports leagues: the National Football League (NFL), Major League Baseball (MLB), National Basketball Association (NBA), and National Hockey League (NHL). He is interested to see which of these four leagues does the better job in evaluating talent. To do this, a player selection is defined as a success if the player is eventually chosen as an all star. Berry models the log of the odds of a player becoming an all star as a function of the draft position. He finds that the NBA and NFL's early draft picks are more likely to become all stars and that later draft picks have a better chance of becoming an all star in the NHL and MLB. Also for the NHL and MLB, Berry incorporates the positions of the players drafted and finds that there is little difference between positions.

The quarterback ranking in the NFL is a confusing and often an inappropriate way to rate quarterbacks. White and Berry (2002) propose an alternative method to rate quarterbacks. The NFL's method does not take into account the circumstances of the game when quantifying the importance of an action. White and Berry determine how important an action is in the context of the game by assigning an expected point count to each play and fitting a tiered logistic regression model. To obtain rankings they find the value of each play and then average the point value over all plays for each quarterback. Randall Cunningaham of the Minnesota Vikings was the number one rated quarterback; this matched the NFL's quarterback rating. One big difference between the rankings is illustrated by Rob Johnson of the Buffalo Bills. The NFL rank him as the second best quarterback in 1998, where as White and Berry rank him as 22^{nd} out of 44 quarterbacks.

2.3 Bayesian Regression

Regression in general uses the conditional distribution of a response variable given an explanatory variable to make predictions and inference. It is one of the most widely used statistical tools. It assumes that the data have been drawn from a relation of the form: $y = X\beta + \epsilon$ where y is the response variable, X is a matrix of known explanatory variables, β is a vector of parameters to be estimated and ϵ is a random vector of independent identically distributed error terms. Gelman et al. (2004) dedicate chapters 14 and 15 of their book to regression models in a Bayesian framework. They say that a key difference between a Bayesian linear model and that of a frequentist perspective is setting up prior distributions on the model parameters that reflect substantive knowledge but are not so strong as to dominate the data inappropriately.

Chen and Deely (1996) use a constrained Bayesian multiple regression model to predict the size of the upcoming apple crop in New Zealand. Their explanatory variable is the tree's age, which they divide into ten different intervals, and the response variable is number of cartons of apples produced. The youngest trees are associated with β_1 and the oldest trees are associated with β_{10}. Since farmers usually don't allow poor producing trees to live they constrain the parameters of interest so that $0 \leq \beta_1 \leq \beta_2 \leq \cdots \leq \beta_{10}$. That is, the older the tree the more apples it will produce. They choose a normal likelihood and assign a normal distribution for the β's and an inverse gamma for σ^2. They use the Gibbs sampler algorithm to perform MCMC, then fit the same model using ordinary least squares (OLS). They use the posterior predictive distribution to predict the upcoming apple crop using the Bayesian model and also predict the next apple crop using OLS.. They find that all of the parameters estimated using Bayesian methods preserve the constrained parameter space, whereas a few of the parameters estimated violated this space using OLS analysis.

Smith (1973) examines the Bayesian hierarchical linear model. He shows that the Bayes estimates of the parameters in question have the structure of a weighted average between the least squares estimates and the prior mean.

Chapter 3

The Model

This chapter gives an overview of the model we will use. The chapter starts with a discussion on the data. The next section discusses the motivation behind the model, and then the hierarchical aspect of the model. Next the likelihood and prior distribution assignments are described. Then computation details, convergence checks, and model fit are examined.

3.1 Data

The box-score of the NBA is a table that displays 13 summary statistics for all players that enter a NBA game (see table 1.1). In the present study we focus on nine categories. They are: assists (*ast*), steals (*stl*), turnovers (*tno*), free throws made (*ftm*), free throw percentage (*ftp*), field goals made (*fgm*), field goal percentage (*fgp*), offensive rebounds (*orb*), and defensive rebounds (*drb*). The four categories that are not considered are minutes, total rebounds, fouls, and points. Minutes, although not explicitly in the model, are used to standardized the variables, which will be explained later in the paper. Total rebounds is a linear combination of offensive rebounds and defensive rebounds and total points is a near linear combination of field goals made and free throws made, so including them in the model introduces collinearity.

In the 1996-1997 season the NBA had 29 teams so we have $T = 29$ teams and $O = 29$ opponents. Each team has twelve players who can participate in the game but in this study we only focus on starters that play at least 45 games so we have $Pl = 131$ players. We make the cut off 45 games because to make the playoffs a team needs to usually win around 45 games. Each of the five starters is assigned a position at the beginning of a game and we assume that he plays that position throughout the game, so we have $Po = 5$ positions and they are: point guard (*pg*), shooting guard (*sg*), small forward (*sf*), power forward (*pf*), and center (*c*). Because we are only including those players that have started at least 45 games there will be some teams for which some positions will not be represented. There are box-scores for 1163 games from the 1996-1997 season so we have $G = 1163$ games.

All categories are standardized by minutes. That is, the player's totals are divided by the total number of minutes played. Each player is paired by position (i.e. point guards from competing teams are paired and shooting guards from competing teams are paired, etc.) for each game and the differences for the nine standardized box-score categories between the matched players are computed. These differences become the explanatory variables. We use differences so that we might consider the talent level of the opposition. For example, a team whose power forward scores 10 points and his opponent scores 5 points would probably be better off than a team whose power forward scores 25 points but his opponent scores 30. For the response variable we use the difference in the final score of the game. So players on the same team will always have the same observed response value for a given contest.

3.2 Model

We are interested in determining how players in each of the five positions need to perform in nine box-score categories so their team wins the game. We will use multiple regression to estimate the effect that each of these categories will have on the difference in the score of a basketball game. The regression model is:

$$y = \beta_0+\beta_1 ast+\beta_2 stl+\beta_3 tno+\beta_4 ftm+\beta_5 ftp+\beta_6 fgm+\beta_7 fgp+\beta_8 orb+\beta_9 drb \quad (3.1)$$

where y is the difference in the final score for a basketball game, β_0 is the overall intercept, β_1 is the effect that difference in assists (ast) has on the difference of the final score, β_2 is the effect for difference in steals (stl), β_3 the effect for the difference in turnovers (tno), etc.

3.2.1 The Hierarchical Model

An inadequacy in equation (3.1) is the assumption that each β is the same regardless of player or position. In addition, an assumption of independent identically distributed responses would be violated. The way in which the data are used creates dependence in the response variable for individuals on the same team, the opponent they play, and the game in which they play. We obtain a separate β estimate for each player and deal with the dependency issues by incorporating a Bayesian hierarchical model on the regression coefficients and adding an effect for the team of which the player is a member, the opponent the player's team is playing, and the game in which they are competing. Thus the model becomes

$$y_{iklm} = \gamma_m + \tau_k + \phi_l + \beta_{1i}ast + \beta_{2i}stl + \beta_{3i}tno + \beta_{4i}ftm + \beta_{5i}ftp + \beta_{6i}fgm + \beta_{7i}fgp + \beta_{8i}orb + \beta_{9i}drb, \tag{3.2}$$

with $i = 1, \ldots, Pl, k = 1, \ldots, T, l = 1, \ldots, O,$ and, $m = 1, \ldots, G$. This can also be written in summation notation as:

$$y_{iklm} = \gamma_m + \tau_k + \phi_l + \sum_{h=1}^{9} \beta_{hi} x_h,$$

where x_h is the h^{th} explanatory variable and β_{hi} is the h^{th} regression coefficient for the i^{th} player.

Each of the three parameters $\gamma, \phi,$ and τ deals with a different aspect of the result. The team effect is modeled by τ which is subscripted by k so that players coming from the same team have the same intercept. This parameter also can help account for different coaching philosophies and talent level of teams. Both of these would affect a position's role in the team framework. The opponent effect is addressed with ϕ. And γ accounts for a game effect.

We obtain an estimate of the effect the different positions have on the regression coefficients by allowing the regression coefficients for each player to come from a "position" distribution. That is, $\beta_{hi} \sim N(\mu_{\beta_{h,j}}, \sigma^2_\beta)$, where $\mu_{\beta_{h,j}}$ is the mean of the position distribution for position j and regression coefficient h. The means of the distributions that correspond to the different positions will be the estimated position effect. By letting each β be drawn from a position distribution we are able to "borrow strength" from all players of the same position to estimate an overall position effect.

3.2.2 The Likelihood

Selecting a likelihood, or sampling density, is arguably the most important step in constructing a good Bayesian model. The likelihood should follow the general shape of the phenomena being modeled. In the present problem this would be the difference of points scored in a NBA basketball game. Differences in points are symmetric around zero. This is true because each game gives a positive and negative value of the same magnitude, one to each team. NBA games are fairly competitive which means most games would have differences close to zero and large differences would not be as common. For these reasons a Gaussian distribution is a good choice for the likelihood of the Bayesian model. Thus, we believe:

$$y_{iklm} \sim N(\theta_{iklm}, \sigma^2), \tag{3.3}$$

where θ_{iklm} is equal to the right side of equation (3.2).

In this study we recognize that the difference in points can never be zero. That is, a game never ends in a tie. Thus, the normal likelihood cannot be exactly true. Nevertheless, we will proceed with this choice. This likelihood has some nice properties, one of which is the ease of interpreting its parameters, thus making the elicitation of prior distributions more straightforward.

3.2.3 Prior Distributions

This section discusses the parameters in the model and the reasons behind their prior distribution assignment. In addition, the choice for values of the hyper-parameters will be discussed.

Selecting prior distributions that reflect prior knowledge is also an important step in Bayesian modeling. These distributions should be a meaningful representation of prior belief or knowledge about a set of parameters. With a good prior distribution specification the posterior distribution can be thought of as a merging of the data collected and prior knowledge.

The regression coefficients and the intercepts can theoretically be either positive or negative, which leads to choosing a distribution that is defined for all real numbers. A priori there does not seem to be any reason why these parameters would take on large values opposed to small ones or visa-versa which leads to symmetry being a desirable characteristic. Thus, we select normal prior distributions for the slopes and intercepts and we have:

$$\beta_{1i} \sim N(\mu_{\beta_{1,j}}, \sigma^2_{\beta_1}), \beta_{2i} \sim N(\mu_{\beta_{2,j}}, \sigma^2_{\beta_2}), \ldots, \beta_{10i} \sim N(\mu_{\beta_{10,j}}, \sigma^2_{\beta_{10}}), \text{and}$$

$$\phi_l \sim N(m_\phi, \sigma^2_\phi), \tau_k \sim N(m_\tau, \sigma^2_\tau), \gamma_m \sim N(m_\gamma, \sigma^2_\gamma).$$

The mean of the distributions for the slopes will change according to the j^{th} position of the i^{th} player. So $\mu_{\beta_{1,1}}$ gives the estimate for the effect of point guard assists, $\mu_{\beta_{1,2}}$ returns an estimate for the effect of shooting guard assists, etc. We assume $\sigma^2_{\beta_h}$ remains constant over the positions. In addition to being a reasonable choice, the normal distribution gives the benefit of having complete conditionals that are in a recognizable form.

The distribution of the mean for each regression coefficient also can theoretically be any real number. Similar to the regression coefficients, a priori there does not seem to exist a reason to believe that these parameters would take on large numbers versus small numbers. Therefore, a Gaussian distribution seems like a logical choice for these parameters as well and we have:

$$\mu_{\beta_{1,j}} \sim N(m_{\mu_{\beta_1}}, s^2_{\mu_{\beta_1}}), \mu_{\beta_{2,j}} \sim N(m_{\mu_{\beta_2}}, s^2_{\mu_{\beta_2}}), \ldots, \mu_{\beta_{10,j}} \sim N(m_{\mu_{\beta_{10}}}, s^2_{\mu_{\beta 10}}).$$

We assume that the means of the distributions for each of the five different positions are drawn from the same distribution. That is, $\mu_{\beta_{1,1}}$ comes from the same distribution as the mean of $\mu_{\beta_{1,5}}$.

The overall variance of the model, σ^2, is by definition greater than or equal to zero. This necessitates a prior distribution that is always positive. The inverse gamma distribution preserves the parameter space, is very flexible in its shape, and yields closed form complete conditionals. Therefore, the inverse gamma is a logical choice for the prior on σ^2. Using the same logic leads us to assign an inverse gamma distribution to the $\sigma^2_{\beta_h}$'s along with $\sigma^2_\gamma, \sigma^2_\tau$, and σ^2_ϕ. Thus, for the variance parameters we have:

$$\sigma^2_{\beta_1} \sim IG(a_{\sigma_{\beta_1}}, b_{\sigma_{\beta_1}}), \ldots, \sigma^2_{\beta_{10}} \sim IG(a_{\sigma_{\beta_{10}}}, b_{\sigma_{\beta_{10}}}), \sigma^2_\gamma \sim IG(a_{\sigma_\gamma}, b_{\sigma_\gamma}),$$

$$\sigma_\tau^2 \sim IG(a_{\sigma_\tau}, b_{\sigma_\tau}), \sigma_\phi^2 \sim IG(a_{\sigma_\phi}, b_{\sigma_\phi}), \sigma^2 \sim IG(a_\sigma, b_\sigma).$$

3.2.3.1 Hyperparameter Values

This section details the selection and reasoning behind the choices for hyperparameter values. Values need to be determined for: $m_{\mu_{\beta_1}}, m_{\mu_{\beta_2}}, \ldots, m_{\mu_{\beta_{10}}}$, $s^2_{\mu_{\beta_1}}, s^2_{\mu_{\beta_2}}, \ldots, s^2_{\mu_{\beta_{10}}}$, $m_\gamma, m_\tau, m_\phi, a_{\sigma_{\beta_1}}, a_{\sigma_{\beta_2}}, \ldots, a_{\sigma_{\beta_{10}}}, b_{\sigma_{\beta_1}}, b_{\sigma_{\beta_2}}, \ldots, b_{\sigma_{\beta_{10}}}, a_{\sigma_\gamma}, b_{\sigma_\gamma}$, $a_{\sigma_\tau}, b_{\sigma_\tau}, a_{\sigma_\phi}, b_{\sigma_\phi}, a_\sigma$, and b_σ.

Coming up with values for the parameters of the positional distributions, or the $\mu_{\beta_{h,j}}$'s is not particularly intuitive, since each slope represents the expected change in score difference given a difference of one per minute in a particular category. Since this relationship is difficult to formalize even for experts, our prior specifications should be diffuse enough so the data eventually dominates the prior parameter specification. Since an assist and a field goal made results in two points, it seems reasonable to believe that these two categories have the largest spread of possible values. So if we come up with values that are diffuse for these two categories and assign these values to the distributions of the remaining seven box-score categories, then our goal of making these distributions diffuse will be accomplished.

A priori, the $\mu_{\beta_{h,j}}$'s, could be either positive or negative depending on the regression coefficient and the position. Hence, it seems reasonable that $m_{\mu_{\beta_1}}, m_{\mu_{\beta_2}}, \ldots, m_{\mu_{\beta_{10}}} = 0$. $s^2_{\mu_{\beta_h}}$ describes the distance from zero that the $\mu_{\beta_{h,j}}$'s could plausibly assume. We focus on $s^2_{\mu_{\beta_1}}$ which is the spread of the means that corresponds to assists.

Theoretically, since an assist is worth two points and there are 48 minutes in a game, β_{1i} could be 96. But this is the extreme case and it will likely not occur. An upper limit somewhere in the neighborhood of 60 seems like a more reasonable estimate. So we need to choose a value for $s^2_{\mu_{\beta_1}}$ to reflect this belief. If we let $s^2_{\mu_{\beta_1}} = 15^2$ then $\mu_{\beta_{1,j}}$ could plausibly be assigned values up to 60 which will in turn allow β_{1i} to take on values as large as 60. We assign the same value to the remaining $s^2_{\mu_{\beta_h}}$'s, which complies with the desire to be diffuse.

The variance of the slopes, or $\sigma^2_{\beta_h}$, measures how much each player's slope varies. Since this parameter is assumed to be inverse-gamma, its parameters are not particularly intuitive. Because of this we will use moment matching to find suitable parameter values. That is, we will choose sensible means and variances for these distributions first, and then find the parameters from the inverse gamma distribution that correspond to the chosen means and variances. Once again we first consider $\sigma^2_{\beta_1}$, which is the variance of β_{1i}, the average effect on the difference in total points given that a player out assists his positional opponent by one assist per minute. It seems that the variability of an effect within a player would not be very large. If $\mathrm{E}(\sigma^2_{\beta_1}) = 2^2$ and $\mathrm{var}(\sigma^2_{\beta_1}) = 3^2$, this will allow the standard deviation of β_{1i} to plausibly reach values above 10, which seems like a good spread. This desired mean and variance produces inverse gamma parameters of $a_{\sigma_{\beta_1}} = \frac{34}{9}$ and $b_{\sigma_{\beta_1}} = \frac{9}{100}$. Now we assign the same values to the remaining distributions to the variances of the regression coefficients and we have

$$\sigma^2_{\beta_1}, \sigma^2_{\beta_2}, \ldots \sigma^2_{\beta_{10}} \sim IG(\tfrac{34}{9}, \tfrac{9}{100}).$$

We assign $m_\gamma = m_\phi = m_\tau = 0$. This is reasonable because a these three parameters (τ_k, ϕ_l, and γ_m) can take on either positive or negative values depending on the team, opponent, and game.

It seems plausible that the effects for team and opponent (τ and ϕ) are similar in their distributional form. Therefore, the same values will be given to their hyperparameters. They are interpreted as the average difference in points for a particular team (τ) or opponent (ϕ) given that two players matched by position record the same number of assists per minute, steals per minute, turnovers per minute, and so on. σ^2_τ and σ^2_ϕ are paramters that represent the variability that exists from team to team, which is probably larger than the within player variance, but large deviations from zero still seem rather unlikely. So it would seem reasonable to find inverse gamma parameters that correspond with $\mathrm{E}(\sigma^2_\tau) = \mathrm{E}(\sigma^2_\phi) = 3^2$ and $\mathrm{var}(\sigma^2_\tau) = \mathrm{var}(\sigma^2_\phi) = 3^2$. This would allow the standard deviations of τ and ϕ to plausibly take on values up to 15, which is a rather large spread. The values of the inverse gamma distribution that correspond with the desired mean and variance are $a_{\sigma_{\beta_1}} = 11$ and $b_{\sigma_{\beta_1}} = \frac{1}{90}$. Thus we have

$$\sigma^2_\tau, \sigma^2_\phi \sim IG(11, \tfrac{1}{90}).$$

The effect γ is interpreted as the difference in points for a particular game given that the competing teams recorded the same number of assists per minute, steals per minute, turnovers per minute, and so on. The variance of this effect, which is the within game variance, is probably smaller than that of the intercepts. Once again we use moment matching to find values to the distribution of σ^2_γ. We find inverse gamma values so that $E(\sigma^2_\gamma) = (\sqrt{2})^2$ and $var(\sigma^2_\gamma) = 3^2$. These values would allow the standard deviation to plausibly reach values as high as 10, which is reasonable. Thus we choose $\sigma^2_\gamma \sim IG(\frac{22}{9}, \frac{173}{500})$.

σ^2 is the variability of the model or the error term. We think it likely that on average the standard deviation would be about 6. An inverse gamma with $E(\sigma^2) = 6^2$ and $var(\sigma^2) = (2\sqrt{5})^2$ would seem to be reasonable. Again, by moment matching, we choose $\sigma^2 \sim IG(\frac{131}{25}, \frac{3}{500})$. For a summary of hyperparameter values see table 3.1.

Table 3.1: Hyperparameter Values

Parameter	m	s^2	Parameter	a	b
$\mu_{\beta_{1j}}$	0	225	$\sigma^2_{\beta_1}$	34/9	9/100
$\mu_{\beta_{2j}}$	0	225	$\sigma^2_{\beta_2}$	34/9	9/100
$\mu_{\beta_{3j}}$	0	225	$\sigma^2_{\beta_3}$	34/9	9/100
$\mu_{\beta_{4j}}$	0	225	$\sigma^2_{\beta_4}$	34/9	9/100
$\mu_{\beta_{5j}}$	0	225	$\sigma^2_{\beta_5}$	34/9	9/100
$\mu_{\beta_{6j}}$	0	225	$\sigma^2_{\beta_6}$	34/9	9/100
$\mu_{\beta_{7j}}$	0	225	$\sigma^2_{\beta_7}$	34/9	9/100
$\mu_{\beta_{8j}}$	0	225	$\sigma^2_{\beta_8}$	34/9	9/100
$\mu_{\beta_{9j}}$	0	225	$\sigma^2_{\beta_9}$	34/9	9/100
m_τ	0		σ^2_τ	11	1/90
m_ϕ	0		σ^2_ϕ	11	1/90
m_γ	0		σ^2_γ	22/9	173/500
			σ^2	131/25	3/500

3.3 Computation

The joint posterior distribution is highly multidimensional. In order to obtain the joint posterior distribution we will perform a simulation that uses Markov Chain Monte Carlo (MCMC) techniques. Recall from chapter 2 that a complete conditional for parameter θ_j is

$$\theta_j|\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_k.$$

With our choice of the likelihood and prior distributions, we have complete conditionals for all parameters that are known and easy to sample from. Because of this, we can use the Gibbs sampling algorithm as used by Gelfand and Smith (1990) to explore the posterior space and obtain draws from the posterior distribution. The complete conditionals were coded in FORTRAN and were used to obtain 100,000 draws with a burn of 50,000 from the joint posterior distribution.

3.4 Convergence Diagnostics

Checking the convergence of Markov chains is a difficult task in models that contain a large number of parameters. Time series plots were used to assess the mixing of the chains. To check convergence, we use the criteria as explained by Raftery and Lewis (1992) and their *gibbsit* function found in the statistical software package R.

3.5 Model Checking

Any statistical model should be checked to determine its validity. In this study we will do this by way of cross validation. That is, ten randomly selected games will be omitted from the data set, then new simulated draws from the joint posterior will be obtained. Then we will use the fitted model to predict the outcome of the ten omitted games using the posterior predictive distributions.

Chapter 4

Results

This chapter gives a summary of results. The chapter starts with a discussion on convergence diagnostics and then summarizes the results of the cross validation study. Next, a summary of the position effect for the nine box-score categories is discussed and conclusions are drawn. The last section details the results of a data generation simulation to determine the reliability of the FORTRAN code.

4.1 Convergence and Cross Validation

Although there are over 2000 parameters in the hierarchical model, most are nuisance parameters and will therefore not be part of the summary. We focus our attention on the $\mu_{\beta_{h,j}}$'s which give an estimate of the effect of the nine box-score categories for the five positions.

The *gibbsit* function as proposed by Raftery and Lewis (1992) provides the quantity I which determines how many more iterations need to run before the chains in the MCMC algorithms have converged. I being less than five indicates that a chain has converged for a parameter. After thinning the chains by twenty draws, that is, keeping every twentieth simulated posterior distribution draw, all of the parameters have an I value of less than five (see appendix B) and that would indicate convergence. The time series plots demonstrate good exploration of the joint posterior space (see appendix C).

The units of the $\mu_{\beta_{h,j}}$'s are the difference in points given the difference in assists per minute, and difference in steals per minute, and so forth. Because of this the interpretation of the results are not practical (A point guard will never out rebound his positional opponent by one offensive rebound per minute). For this reason we divide the posterior draws of the $\mu_{\beta_{h,j}}$'s by 48 which is how many minutes there are in a game. This changes the interpretation of the parameters to the difference in points given the difference in assists per game, and the difference in steals per game, and so forth this interpretation is more practical. Also we divide both shooting percentage effects (free throw percentage (*ftp*) and field goal percentage (*fgp*)) by 100 so we can interpret the effects as the average difference in points given one percent increase in shooting percentage. These operations are valid because of posterior invariance.

All statistical models should be checked for validity. To determine how well the model fit the data we used cross validation. The games that were randomly selected to be removed were: 19, 124, 241, 398, 486, 668, 772, 889, 954, and 1116. We will use game 19 to demonstrate an acceptable fit. The Portland Trailblazers beat the Golden State Warriors by 18 points in game 19. There were five players

Table 4.1: Cross Validation Results

		Position									
		Team A					Team B				
Game		C	PF	PG	SF	SG	C	PF	PG	SF	SG
19	Obs	18	18	18	18	18	-	-18	-18	-18	-18
	Pred	12.5	8.29	14.8	3.10	9.5	-	-8.46	-15.0	-10.0	-2.17
124	Obs	16	16	16	16	16	-16	-16	-16	-16	-16
	Pred	4.65	9.33	5.89	5.46	12.0	-4.66	-10.7	-6.50	-7.23	-9.97
241	Obs	3	3	3	-	-	-	-3	-3	-3	-3
	Pred	-5.44	-5.56	-2.38	-	-	-	6.54	3.23	6.56	13.0
434	Obs	1	1	-	1	1	-1	-1	-1	-1	-1
	Pred	2.91	-8.29	-	0.80	3.54	-0.83	10.6	-5.49	-0.51	-3.56
486	Obs	2	2	2	-	2	- 2	-2	-2	-2	-2
	Pred	4.37	9.68	0.72	-	9.17	-5.17	-9.96	-1.17	-8.17	4.53
600	Obs	-	9	9	9	9	-	-	-9	-9	-9
	Pred	-	-1.15	-2.29	4.45	6.49	-	-	6.69	-7.46	-5.37
772	Obs	7	-	-	7	7	-7	-7	-7	-	-7
	Pred	-2.20	-	-	12.3	-3.91	1.70	-0.93	-0.42	-	4.03
889	Obs	16	16	16	-	16	-16	-16	-16	- 16	-16
	Pred	7.84	11.3	8.01	-	5.15	-7.94	-11.5	-7.96	-10.7	-5.67
954	Obs	7	-	7	7	-	-7	-7	-7	-	-7
	Pred	8.02	-	16.3	5.92	-	-11.0	-8.25	-17.8	-	-8.03
1116	Obs	3	3	3	-	3	-3	-3	-3	-3	-3
	Pred	0.21	2.53	1.86	-	-4.44	-0.16	-2.69	-1.09	5.72	3.96

from Golden State and four players from Portland that participated in the game and started more than 45 games in the season. So given how each player performed in the game we use the posterior distribution with the cross validation data to predict what the difference in total points scored will be. A prediction is considered good if a players prediction mirrors how their team did. So all Portland players that predict Portland winning and all Golden State players that predict Golden State losing are good fits.

Table 4.1 is a summary of the cross validation results, table 4.2 is a summary of percentages that demonstrate the accuracy of the cross validation pre-

Figure 4.1: Plots of the posterior predictive distribution for game 1116 of the cross validation data. The first five players are members of the New York Knicks the last four are members of the Indiana Pacers. The vertical line represents the difference in score which is -3 for New York and 3 for Indiana which is what we tried to predict.

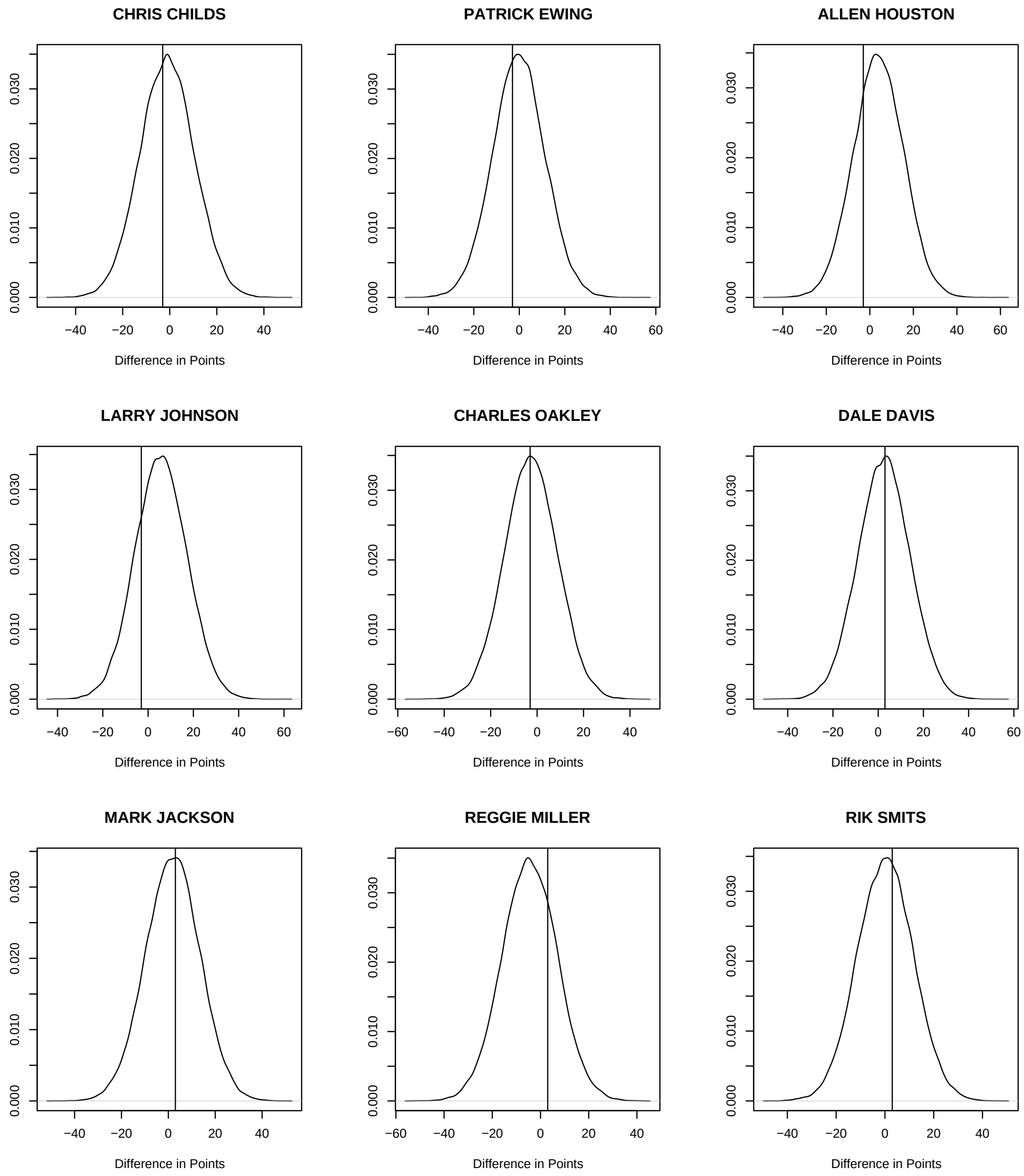


Table 4.2: Percent of correct cross validation predictions

	Position						
Game	C	PF	PG	SF	SG	total	percent
19	1/1	2/2	2/2	2/2	2/2	9/9	100%
124	2/2	2/2	2/2	2/2	2/2	10/10	100%
241	0/1	0/2	0/2	0/1	0/1	0/7	0%
434	2/2	0/2	1/1	2/2	2/2	7/9	77.8%
486	2/2	2/2	2/2	1/1	2/2	9/9	100%
600	-	0/1	0/2	2/2	2/2	4/7	57.1%
772	0/2	1/1	1/1	1/1	0/2	3/7	42.9%
889	2/2	2/2	2/2	2/2	1/1	9/9	100%
954	2/2	1/1	2/2	1/1	1/2	7/7	100%
1116	2/2	2/2	2/2	0/1	0/2	6/9	66.7%
total	13/16	12/17	14/18	12/14	13/18	64/83	77.1%
percent	81.2%	70.6%	77.8%	85.7%	72.2%	77.1%	

dictions, and figure 4.1 presents plots for the posterior predictive distributions of game 1116. Overall, the model correctly picked the winning team 77% of the time. The model performed very poorly for game 241. This game was between Chicago and Miami, two good teams in the 1996-1997 season. Miami beat Chicago by three points, but Michael Jordan dominated the shooting guard position and Scottie Pippen his small forward opponent, so it would seem like Chicago should have won. Most likely a player not represented in the data played a significant role in the outcome of the game. Jordan was so dominant in this game that the model predicted his team to win by 13 points even though they really lost by 3. Other than game 241 the model seemed to fit the data reasonably well.

Figure 4.2: Plots of posterior distributions of parameters for assists, steals, turn overs, offensive rebounds, and defensive rebounds for each position

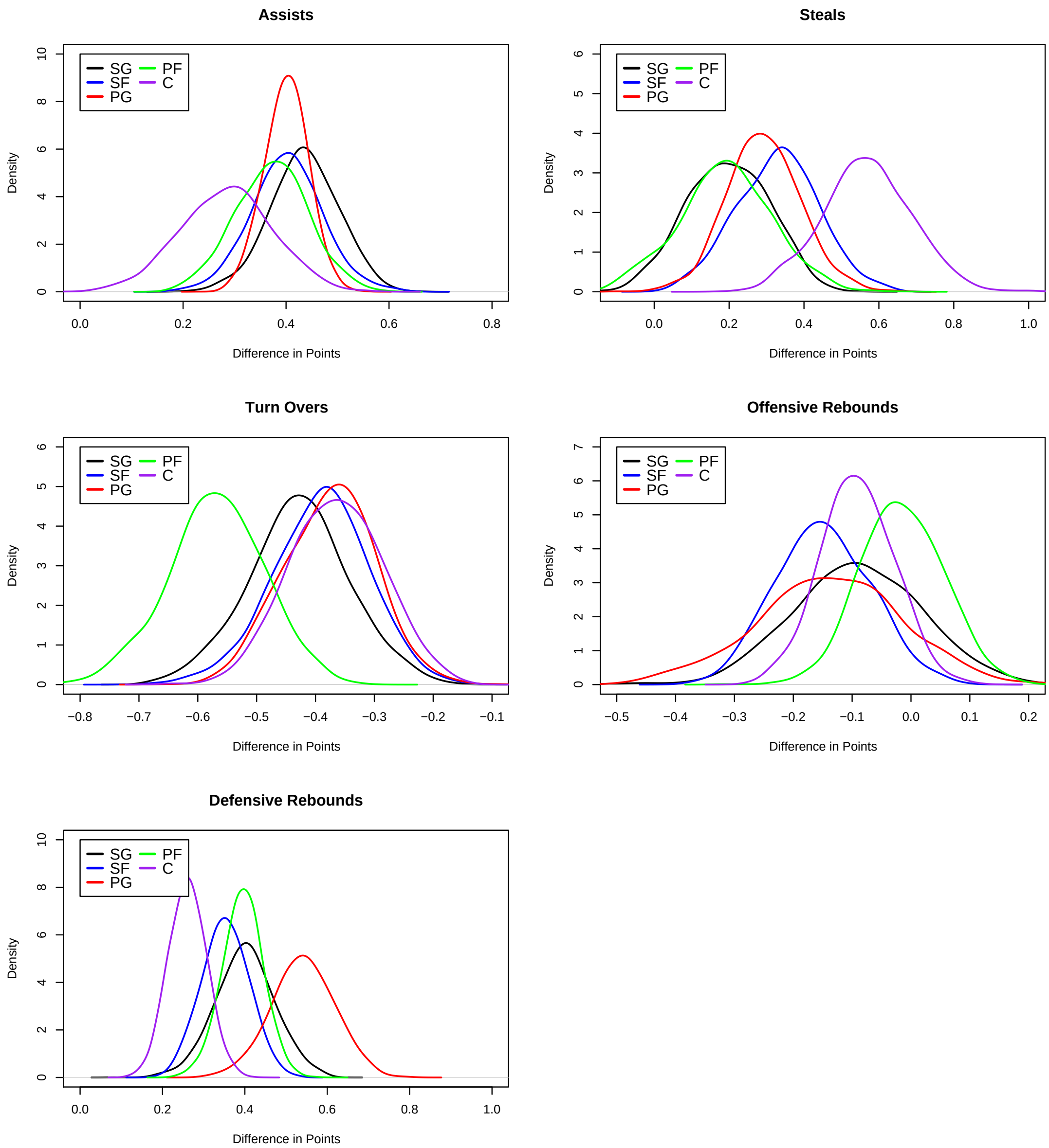


Figure 4.3: Plots of posterior distributions of parameters for free throws made, free throw percentage,field goals made, and field goal percentage for each position

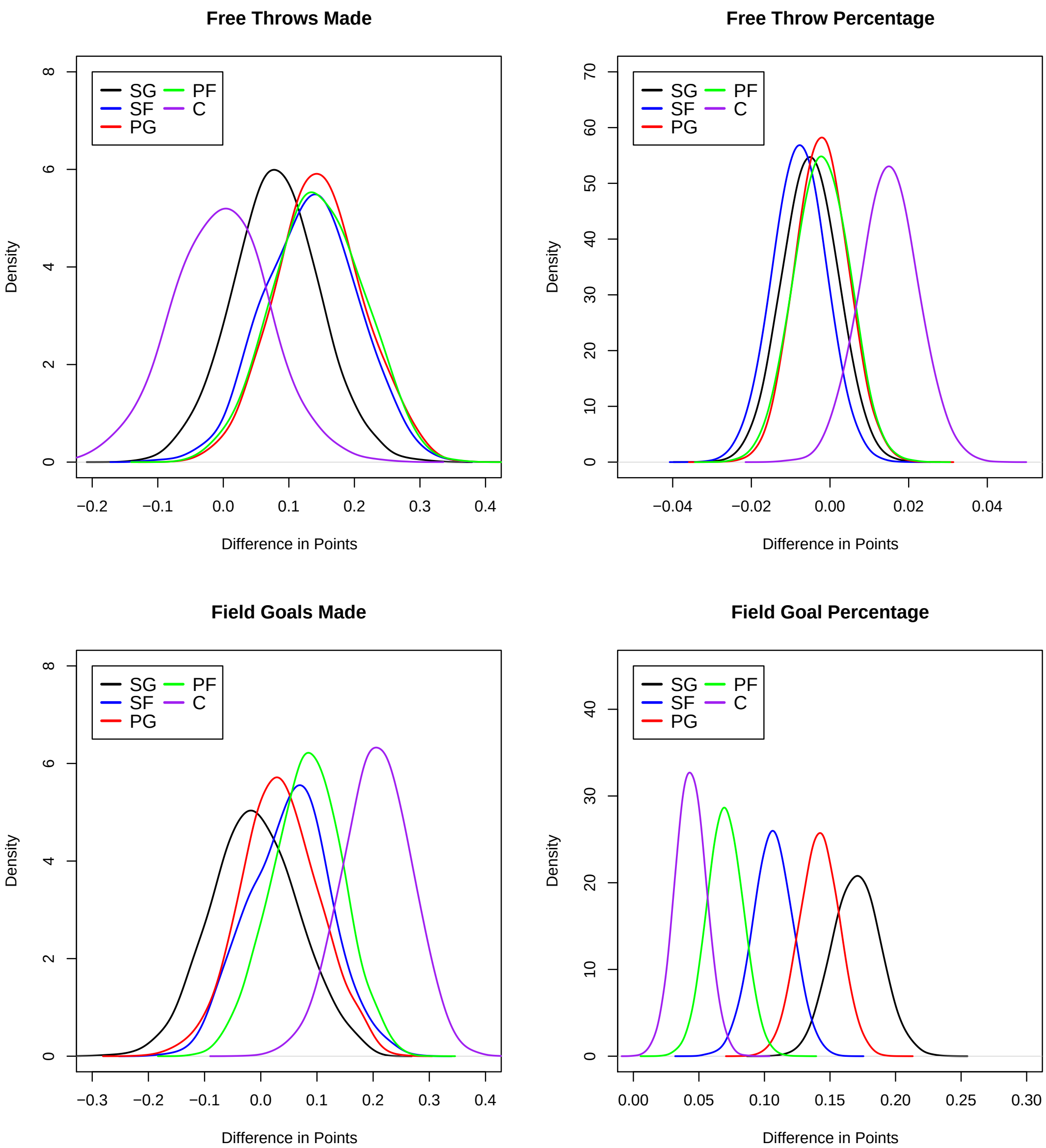


Table 4.3: Posterior means, standard deviations, and 95% highest posterior density credible intervals of parameters of assists, steals, turnovers, free throws made, free throw percentage for each position

	Assist(μ_{β_1})			
Position	Mean	Std.Dev.	2.5%LHPD	97.5%UHPD
Center	0.2836	0.0904	0.1151	0.4695
Power Forward	0.3721	0.0714	0.2269	0.5103
Small Forward	0.4013	0.0706	0.2600	0.5438
Point Guard	0.4007	0.0433	0.3135	0.4844
Shooting Guard	0.4354	0.0674	0.3032	0.5699
	Steals(μ_{β_2})			
Center	0.5689	0.1200	0.3252	0.7932
Power Forward	0.1908	0.1250	-0.0721	0.4268
Small Forward	0.3291	0.1109	0.0935	0.5316
Point Guard	0.2920	0.0998	0.0973	0.4955
Shooting Guard	0.1957	0.1118	-0.0184	0.4060
	Turn Overs(μ_{β_3})			
Center	-0.3648	0.0805	-0.5186	-0.2072
Power Forward	-0.5687	0.0827	-0.7353	-0.4053
Small Forward	-0.3912	0.0830	-0.5576	-0.2303
Point Guard	-0.3764	0.0782	-0.5345	-0.2304
Shooting Guard	-0.4281	0.0869	-0.5981	-0.2485
	Free Throws Made(μ_{β_4})			
Center	-0.0041	0.0753	-0.1578	0.1429
Power Forward	0.1457	0.0690	0.0128	0.2791
Small Forward	0.1340	0.0711	0.0009	0.2722
Point Guard	0.1468	0.0672	0.0182	0.2818
Shooting Guard	0.0804	0.0669	-0.0541	0.2121
	Free Throw Precentage(μ_{β_5})			
Center	0.0150	0.0075	0.0007	0.0301
Power Forward	-0.0021	0.0070	-0.0157	0.0115
Small Forward	-0.0078	0.0068	-0.0211	0.0059
Point Guard	-0.0020	0.0067	-0.0153	0.0108
Shooting Guard	–0.0050	0.0071	-0.0190	0.0087

4.2 Positional Summaries

Figures 4.2 and 4.3 provide density plots of the $\mu_{\beta_{h,j}}$,s. Tables 4.3 and 4.4 contain a summary of the posterior distributions to these parameters. These figures and tables reveal some interesting associations.

For all five positions, out-assisting your opponent has a very positive impact

Table 4.4: Posterior means, standard deviations, and 95% highest posterior density credible intervals of parameters for field goals made, field goal percentage, offensive rebounds, and defensive rebounds for each of the positions

	Field Goals Made(μ_{β_6})			
Position	Mean	Std.Dev.	2.5%LHPD	97.5%UHPD
Center	0.2061	0.0615	0.0871	0.3259
Power Forward	0.0820	0.0630	-0.0441	0.2043
Small Forward	0.0508	0.0723	-0.0933	0.1861
Point Guard	0.0340	0.0698	-0.0966	0.1795
Shooting Guard	-0.0105	0.0774	-0.1576	0.1453
	Field Goal Percentage(μ_{β_7})			
Center	0.0437	0.0118	0.0208	0.0672
Power Forward	0.0700	0.0136	0.0436	0.0969
Small Forward	0.1064	0.0153	0.0754	0.1360
Point Guard	0.1418	0.0153	0.1126	0.1725
Shooting Guard	0.1703	0.0185	0.1345	0.2066
	Offensive Rebounds(μ_{β_8})			
Center	-0.0911	0.0634	-0.2147	0.0351
Power Forward	-0.0152	0.0720	-0.1520	0.1279
Small Forward	-0.1492	0.0811	-0.3060	0.0059
Point Guard	-0.1391	0.1234	-0.3990	0.0933
Shooting Guard	-0.0913	0.1111	-0.3060	0.1228
	Defensive Rebounds(μ_{β_9})			
Center	0.2609	0.0469	0.1701	0.3535
Power Forward	0.3948	0.0507	0.2937	0.4944
Small Forward	0.3530	0.0580	0.2385	0.4646
Point Guard	0.5414	0.0779	0.3896	0.6974
Shooting Guard	0.4005	0.0734	0.2565	0.5481

on a basketball game. A result that is somewhat unexpected is that a shooting guard out-assisting his opponent on average is the most beneficial to the team. A result that we did expect was that a center out-assisting his opponent was the least beneficial.

Something that was not expected is how important it is for the center position to record more steals than his opponent. A center that gets one steal more than his opponent gives his team a 0.5689 points advantage on average. In fact, the steals from a center had the largest positive impact among the position box-score category combinations. This association might exist because often times steals from a center occur close to the basket and prevent a potentially easy field goal attempt.

In the introduction we hypothesized that a turnover from the point guard would be more detrimental to the team than a turnover from the center position. This did not turn out to be the case. In turns out, a point guard committing one more turnover than the opponent has the least negative effect. A power forward committing one more turnover than the opponent is the most detrimental to a team.

It is interesting that a center making more free throws than the opponent is not significant, but shooting a better free throw percentage than the opponent is. The exact opposite holds true for power forwards, small forwards, and point guards. That is, making more free throws than your opponent contributes positively to the outcome of the game, but shooting free throws at a better percentage than your opponent has no impact on the game. This could be attributed to the fact that centers are generally not as proficient at shooting free throws compared to other positions. Because of this, the center could potentially shoot a large number of free throws but make only a few, resulting in a loss of offensive production.

Having a better field goal percentage than the opposition is significant for all positions, making more field goals than the opposition is only significant for the center position. Shooting a better field goal percentage than the opposition for players that play a position that requires them to be farther from the basket has the largest effect. Both guards' field goal percentages have more influence on the outcome of a game than the other positions. This is reasonable because shooting becomes more difficult as the distance between the shooter and the basket increases. Thus having a player that shoots well at these three positions is very beneficial to a team.

A very surprising result is that of the offensive rebounds. Most people involved with basketball would agree that an offensive rebound is good, but in this study they are not significant for any position, the effect of out defensive rebounding the positional opponent has a large positive impact on the outcome of the game. The guard positions out-rebounding the opposing guards is more beneficial on average than the remaining three positions. The result that offensive rebounds are insignificant could be a result of the correlation between offensive rebounds and defensive rebounds. If one team is able to get a large number of defensive rebounds then the other team will not be able to get many offensive rebounds. So as one team's defensive rebounds goes up the other team's offensive rebounds goes down.

In summary, the odds of winning a basketball game increase if all five positions out-rebound, out-assist, and have a better field goal percentage than their positional opponent. So having drills incorporated in their practices that will improve all five positions as defensive rebounders, better shooters, and better distributors of the basketball would increase the chances of winning basketball games. In addition, if centers have drills that will improve their free throw-shooting percentage the odds of winning basketball games improve. Also, power forwards having drills that would improve their passing and dribbling skills to keep their turnovers down would improve a team's chances of winning. It appears that the small forward position needs to be proficient in all aspects of the game. If their practices incorporate drills that improve shooting percentages, speed, ball handling, strength, and decision-making the probability of winning games would increase. If the guard positions can have a better shooting percentage and out-rebound and out-assist their opponents this would be very beneficial to their team and would increase the chances of winning so effective practices for them would include drills that would improve their abilities in these areas.

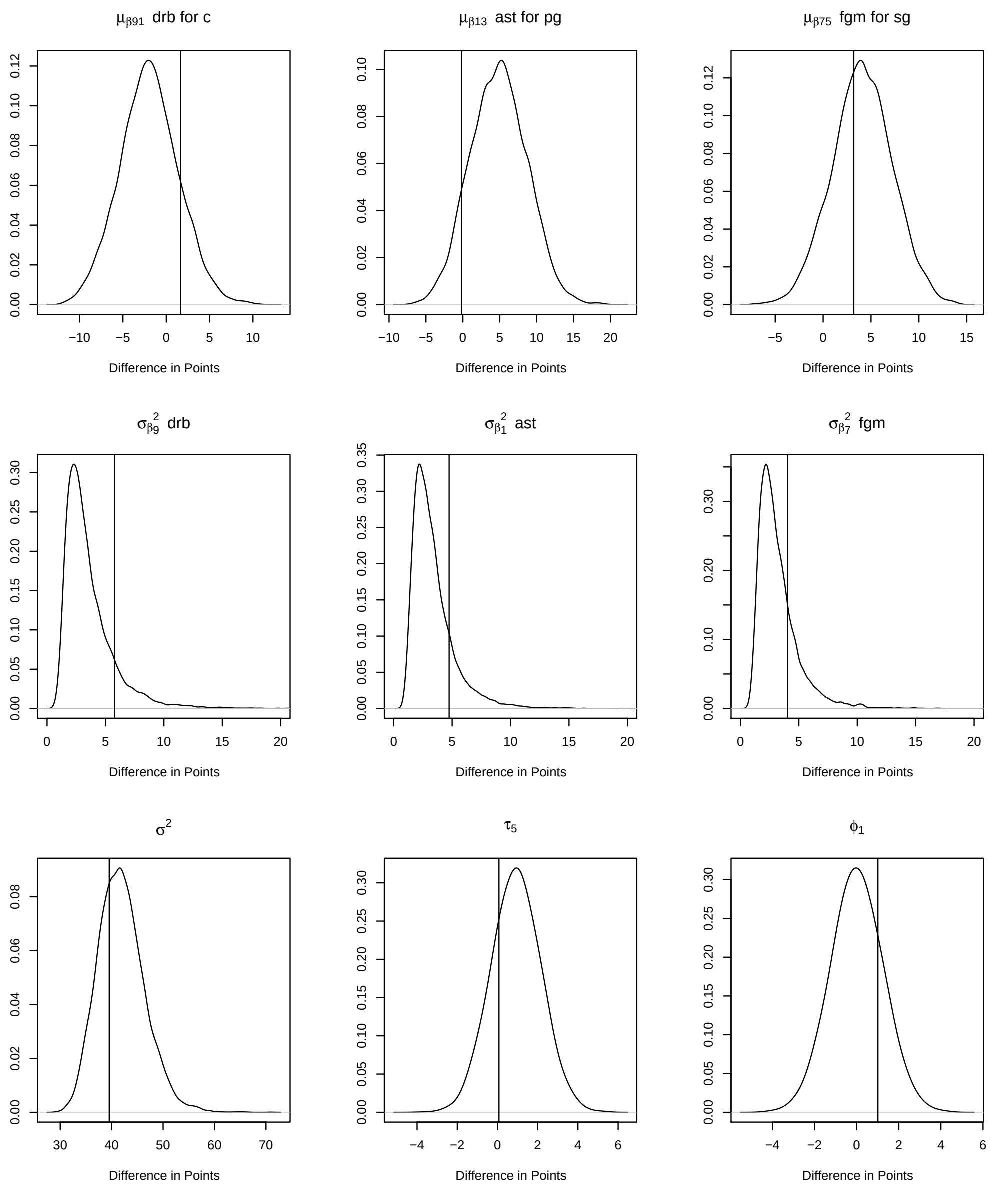
4.3 Simulation

To ensure that the computer program that implemented the Gibbs sampling algorithm was correct we carried out a simulation study. We generated some NBA box-score data using the proposed model and then we fit the model to the generated data. Since we generated the data we knew what the answers were. By comparing the estimated answers to the generated ones we could monitor the performance of the program.

To generate the data we first randomly drew a value from a normal distribution that corresponded to the positional means of the nine regression coefficients for every game that was played. Then, we drew values from an inverse gamma distribution for the variances of the nine regression coefficients, the game effect, team effect, and opponent effect. Then, a game, team, and opponent effect was generated for every game, team, and opponent using a normal distribution with mean zero and variances obtained from the inverse gamma. Next, we generated values for the regression coefficients that corresponded to the player and game being played. This was done by using the values drawn for the positional means and the corresponding variances of the nine β's as the means and variances of a normal distribution. Next, we took the generated game, team, and opponent effects along with the generated regression coefficient effects and summed these values for each player-game combination. These sums became the means to a normal distribution that will produce the difference in points. Because the players on the same team have the same difference in points, we randomly selected one player from the game and used that player's sum to obtain a random normal draw. Then the remaining players on the same team as the selected player received the same drawn value for their difference in points and all players on the opposing team received the additive reciprocal for their difference in points.

The posterior distribution values that are estimated by the computer program seem to be similar to the generated parameter values. This indicates that the program is performing correctly. Figure 4.4 shows the density plots of a few of the parameters with a vertical line indicating the mean value of the effect from the generated data.

Figure 4.4: Plots of the posterior distributions of nine selected parameters using the generated data. The mean of the parameter from generated data is represented by the vertical line



Chapter 5

Further Research and Potential Model Improvements

This chapter discusses potential problems that exist in the proposed model and some areas that could produce further research. The chapter can be divided into three sections. The first talks about the importance of players that don't start. The next section talks about the implicit assumption that interactions are negligible. The last section discusses problems with the current likelihood specification and problems that arise because of how the data are used.

5.1 Bench Players

In the present study, only players that start the game are included in the model. This fails to take into account the impact that the remaining seven players on the team can potentially have on the outcome of a game. Sometimes these players (referred to as bench players) have an important role in the offense. Thus, the association between a successful team and the box-score categories can be confounded with bench play. The bench play can be considered in two ways. One is to include all players and specify the position they play. This will take into account all players but does not give an overall bench effect. The other is to classify all players that don't start as the bench and treat the bench as a sixth position. This way we can determine how the bench players as a unit need to perform so that their team can be successful.

Also, we assumed that each player played the same position for the entire game. This, in reality, is not the case as some players played several positions during the course of a game. To remedy this assumption new data would need to be gathered that identified the position a player was playing each time his team was playing offense.

5.2 Interaction Terms

The model specified is an additive model which implies that all interactions are negligible. That is, all players play the same regardless of where they are playing, and who they are playing with and against. This assumption seems unrealistic. For example, consider the 2005 Phoenix Suns. Prior to the arrival of Steve Nash (a pass first point guard), Amare Stoudamire's point average was 20.6 points per game with a field goal percentage of 47.5%, while after Nash's arrival his point average was 26.0 points per game with a field goal percentage of 55.9%. This is a clear case of a player's performance changing relative to the players on the floor. The model that incorporates interactions would be rather complicated. Graves et al. (2003) modeled interactions between a race track and race car driver. Using an extension of their work would be a good place to start. Fitting a model that contains interaction effects would be very data intense.

5.3 Likelihood and Data Concerns

A consequence of a normal likelihood specification is that the parameter space for a normal density is violated. This is the case because the difference in total points scored is never zero. One way to remedy this would be to disregard overtime periods. That is, take data after 48 minutes regardless if the score is tied or not. Finding data for this model would be very difficult. Another approach would be to specify a new likelihood that would take into account the fact that the response variable can't be zero. In addition to the likelihood problem the way in which the data were used creates areas of concern. Using each player as an observation required some created dependencies. Instead of treating each player as an observation, each game can be treated as an observation and have a regression term for each position performance category combination. This model would still allow us to estimate positional skills and would eliminate within game dependencies of players from the same team.

5.4 Conclusions

From the results of this study basketball coaches can optimize their odds of winning basketball games by organizing practices that are customized to the needs of each position. In this way, the skills required for each position to reach its potential can be developed. Also, basketball coaches can use the results of the study to build game plans and strategies for specific teams by exploiting positional match-ups. Through developing each position's skills and building good game plans coaches can increase their chances of winning games.

Appendix A

Code Appendix

A.1 FORTRAN code

Listing A.1: **FORTRAN program to run the MCMC iterations**

```
c
c       program to produce the mcmc cross validation
c       analysis for NBA study using normal likelihood
c

        implicit none
        integer i,j,ii, jj, n, burn, out, lag, niter,op,g
        integer pl, po, tm,idindex(8845),ngames(131),sumobs
        integer DinP(8845),game(8845),team(8845),opp(8845)
        integer player(8845),pos(8845)
        real gennor,gengam,genunf,zero,one,genbet,genexp
        double precision foul(8845),ast(8845),stl(8845),tno(8845)
        double precision pts(8845),ftm(8845),ftp(8845),fgm(8845)
        double precision fgp(8845),orb(8845),drb(8845)
        double precision beta0(29), mub0, sigb0
        double precision tau(29), mut,sigt
        double precision gamma(1153), mug,sigg
        double precision beta1(131), mub1(5), sigb1
        double precision beta2(131), mub2(5), sigb2
        double precision beta3(131), mub3(5), sigb3
        double precision beta5(131), mub5(5), sigb5
        double precision beta6(131), mub6(5), sigb6
        double precision beta7(131), mub7(5), sigb7
        double precision beta8(131), mub8(5), sigb8
        double precision beta9(131), mub9(5), sigb9
        double precision beta10(131), mub10(5), sigb10
        double precision mt,st,at,bt,mg,sg,ag,bg,mb0,sb0,ab0,bb0
        double precision mb1,mb2,mb3,mb5,mb6,mb7,mb8,mb9,mb10
        double precision sb1,sb2,sb3,sb5,sb6,sb7,sb8,sb9,sb10
        double precision ab1,ab2,ab3,ab5,ab6,ab7,ab8,ab9,ab10
        double precision bb1,bb2,bb3,bb5,bb6,bb7,bb8,bb9,bb10
        double precision sigma2, asig, bsig, temp, sort(8845)
        double precision mub1pos(131),mub2pos(131),mub3pos(131)
        double precision mub5pos(131),mub6pos(131),mub7pos(131)
        double precision mub8pos(131),mub9pos(131),mub10pos(131)
        double precision position(131)

        external gennor,gengam,genunf,genexp,genbet,setall
c
c       initialize assign starting values
c
        niter = 20000
        call setall(1090,9)
        do i=1,131
           beta1(i)=0.d0
           beta2(i)=0.d0
           beta3(i)=0.d0
           beta5(i)=0.d0
           beta6(i)=0.d0
           beta7(i)=0.d0
           beta8(i)=0.d0
           beta9(i)=0.d0
           beta10(i)=0.d0
        end do

        do i= 1,29
```

```
    beta0(i)=0.d0
    tau(i)=0.d0
end do

do i=1,1153
    gamma(i)=0.d0
end do


do i = 1,5
    mub1(i) = 0.d0
    mub2(i) = 0.d0
    mub3(i) = 0.d0
    mub5(i) = 0.d0
    mub6(i) = 0.d0
    mub7(i) = 0.d0
    mub8(i) = 0.d0
    mub9(i) = 0.d0
    mub10(i) = 0.d0
end do

sigma2 = 1.d0

sigb0 = 1.d0
sigg = 1.d0
sigt = 1.d0
sigb1 = 1.d0
sigb2 = 1.d0
sigb3 = 1.d0
sigb5 = 1.d0
sigb6 = 1.d0
sigb7 = 1.d0
sigb8 = 1.d0
sigb9 = 1.d0
sigb10 = 1.d0



c
c       Hyper prior values
c
        mug = 0.d0
        mut = 0.d0
        mub0 = 0.d0

        at   = 11.0d0
        bt   = 0.0111d0

        ag   = 2.44444d0
        bg   = 0.3461538d0

        ab0 = 11.0d0
        bb0 = 0.0111d0

        mb1 = 0.0d0
        sb1 = 225.0d0
        ab1 = 3.777778d0
        bb1 = 0.09d0
        mb2 = 0.0d0
        sb2 = 225.0d0
        ab2 = 3.777778d0
        bb2 = 0.09d0
        mb3 = 0.0d0
        sb3 = 225.0d0
        ab3 = 3.777778d0
        bb3 = 0.09d0
        mb5 = 0.0d0
        sb5 = 225.0d0
        ab5 = 3.777778d0
        bb5 = 0.09d0
        mb6 = 0.0d0
        sb6 = 225.0d0
        ab6 = 3.777778d0
        bb6 = 0.09d0
        mb7 = 0.0d0
        sb7 = 225.0d0
        ab7 = 3.777778d0
        bb7 = 0.09d0
        mb8 = 0.0d0
        sb8 = 225.0d0
        ab8 = 3.777778d0
        bb8 = 0.09d0
        mb9 = 0.0d0
        sb9 = 225.0d0
        ab9 = 3.777778d0
        bb9 = 0.09d0
        mb10 = 0.0d0
```

```
      sb10 = 225.0d0
      ab10 = 3.777778d0
      bb10 = 0.09d0

      asig=5.24d0
      bsig=0.006d0



      burn=50000
      out=50000
      lag=2
c
c     read data
c
      open(1,file="nbacrossvalid.txt",status="old")

      n=8845
      pl=131
      po=5
      tm=29
      op=29
      g=1153
      do i=1,8845
       read(1,*)game(i),DinP(i),pts(i),ast(i),orb(i),drb(i),
     $          stl(i),fgm(i),fgp(i),ftm(i),ftp(i),foul(i),
     $          tno(i),opp(i),team(i),player(i),pos(i)
      end do



c
c     open output files
c

      open(2,file="beta0.txt",status="unknown")
      open(3,file="beta1.txt",status="unknown")
      open(4,file="beta2.txt",status="unknown")
      open(5,file="beta3.txt",status="unknown")
c      open(6,file="beta4.txt",status="unknown")
      open(7,file="beta5.txt",status="unknown")
      open(8,file="beta6.txt",status="unknown")
      open(9,file="beta7.txt",status="unknown")
      open(10,file="beta8.txt",status="unknown")
      open(11,file="beta9.txt",status="unknown")
      open(12,file="beta10.txt",status="unknown")
      open(13,file="mub1.txt",status="unknown")
      open(14,file="mub2.txt",status="unknown")
      open(15,file="mub3.txt",status="unknown")
c      open(16,file="mub4.txt",status="unknown")
      open(17,file="mub5.txt",status="unknown")
      open(18,file="mub6.txt",status="unknown")
      open(19,file="mub7.txt",status="unknown")
      open(20,file="mub8.txt",status="unknown")
      open(21,file="mub9.txt",status="unknown")
      open(22,file="mub10.txt",status="unknown")
      open(23,file="tau.txt",status="unknown")
      open(24,file="gamma.txt",status="unknown")
      open(25,file="sigma2.txt",status="unknown")
c      open(26,file="mut.txt",status="unknown")
      open(27,file="sigt.txt",status="unknown")
c      open(28,file="mugg.txt",status="unknown")
      open(29,file="sigg.txt",status="unknown")
c      open(30,file="mub0.txt",status="unknown")
      open(31,file="sigb0.txt",status="unknown")
      open(32,file="sigb1.txt",status="unknown")
      open(33,file="sigb2.txt",status="unknown")
      open(34,file="sigb3.txt",status="unknown")
c      open(35,file="sigb4.txt",status="unknown")
      open(36,file="sigb5.txt",status="unknown")
      open(37,file="sigb6.txt",status="unknown")
      open(38,file="sigb7.txt",status="unknown")
      open(39,file="sigb8.txt",status="unknown")
      open(40,file="sigb9.txt",status="unknown")
      open(41,file="sigb10.txt",status="unknown")



c
c     start mcmc loop
c

      do jj = 1,(burn+out)
c
c     compute new values for parameters
c

      call getsiggamma(g,ag,bg,gamma,mug,sigg)
```

```
 call getsigtau(op,at,bt,tau,mut,sigt)
 call getsigbeta0(tm,ab0,bb0,beta0,mub0,sigb0)

 call getsigbeta1(n,ab1,bb1,beta1,pl,mub1pos,sigb1)
 call getsigbeta2(n,ab2,bb2,beta2,pl,mub2pos,sigb2)
 call getsigbeta3(n,ab3,bb3,beta3,pl,mub3pos,sigb3)
 call getsigbeta5(n,ab5,bb5,beta5,pl,mub5pos,sigb5)
 call getsigbeta6(n,ab6,bb6,beta6,pl,mub6pos,sigb6)
 call getsigbeta7(n,ab7,bb7,beta7,pl,mub7pos,sigb7)
 call getsigbeta8(n,ab8,bb8,beta8,pl,mub8pos,sigb8)
 call getsigbeta9(n,ab9,bb9,beta9,pl,mub9pos,sigb9)
 call getsigbeta10(n,ab10,bb10,beta10,pl,mub10pos,sigb10)

 do ii = 1,5
 call getmubeta1(ii,n,sb1,mb1,beta1,sigb1,position,mub1(ii))
 call getmubeta2(ii,n,sb2,mb2,beta2,sigb2,position,mub2(ii))
 call getmubeta3(ii,n,sb3,mb3,beta3,sigb3,position,mub3(ii))
 call getmubeta5(ii,n,sb5,mb5,beta5,sigb5,position,mub5(ii))
 call getmubeta6(ii,n,sb6,mb6,beta6,sigb6,position,mub6(ii))
 call getmubeta7(ii,n,sb7,mb7,beta7,sigb7,position,mub7(ii))
 call getmubeta8(ii,n,sb8,mb8,beta8,sigb8,position,mub8(ii))
 call getmubeta9(ii,n,sb9,mb9,beta9,sigb9,position,mub9(ii))
 call getmubeta10(ii,n,sb10,mb10,beta10,sigb10,position,
$                mub10(ii))
 end do

 call getsigma2(i,n,player,team,opp,game,DinP,ast,
$  stl,tno,ftm,ftp,fgm,fgp,orb,drb,asig,bsig,gamma,
$  tau,beta0,beta1,beta2,beta3,beta5,beta6,beta7,
$  beta8,beta9,beta10,sigma2)

 do ii =1,1153
 call getgamma(ii,n,player,team,opp,game,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigg,mug,tau,beta0,
$  beta1,beta2,beta3,beta5,beta6,beta7,beta8,beta9,
$  beta10,gamma(ii))
 end do

 do ii = 1,29
 call gettau(ii,n,player,team,opp,game,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigt,mut,gamma,beta0,
$  beta1,beta2,beta3,beta5,beta6,beta7,beta8,beta9,
$  beta10,tau(ii))
 end do

 do ii = 1,29
 call getbeta0(ii,n,player,team,opp,game,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb0,mub0,gamma,tau,
$  beta1,beta2,beta3,beta5,beta6,beta7,beta8,beta9,
$  beta10,beta0(ii))
 end do


 do ii = 1,131
 call getbeta1(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb1,mub1,gamma,tau,
$  beta0,beta2,beta3,beta5,beta6,beta7,beta8,beta9,
$  beta10,position(ii),mub1pos(ii),beta1(ii))

 call getbeta2(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb2,mub2,gamma,tau,
$  beta0,beta1,beta3,beta5,beta6,beta7,beta8,beta9,
$  beta10,mub2pos(ii),beta2(ii))

 call getbeta3(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb3,mub3,gamma,tau,
$  beta0,beta1,beta2,beta5,beta6,beta7,beta8,beta9,
$  beta10,mub3pos(ii),beta3(ii))

 call getbeta5(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb5,mub5,gamma,tau,
$  beta0,beta1,beta2,beta3,beta6,beta7,beta8,beta9,
$  beta10,mub5pos(ii),beta5(ii))

 call getbeta6(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb6,mub6,gamma,tau,
$  beta0,beta1,beta2,beta3,beta5,beta7,beta8,beta9,
$  beta10,mub6pos(ii),beta6(ii))

 call getbeta7(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb7,mub7,gamma,tau,
$  beta0,beta1,beta2,beta3,beta5,beta6,beta8,beta9,
$  beta10,mub7pos(ii),beta7(ii))

 call getbeta8(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
$  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb8,mub8,gamma,tau,
$  beta0,beta1,beta2,beta3,beta5,beta6,beta7,beta9,
$  beta10,mub8pos(ii),beta8(ii))
```

```
      call getbeta9(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
     $  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb9,mub9,gamma,tau,
     $  beta0,beta1,beta2,beta3,beta5,beta6,beta7,beta8,
     $  beta10,mub9pos(ii),beta9(ii))

      call getbeta10(ii,n,player,team,opp,game,pos,DinP,ast,stl,tno,
     $  ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb10,mub10,gamma,tau,
     $  beta0,beta1,beta2,beta3,beta5,beta6,beta7,beta8,
     $  beta9,mub10pos(ii),beta10(ii))

      end do




c
c     write output to files
c
      if(jj .gt. burn) then

      write(2,*)(beta0(j), j=1,29)
      write(3,*)(beta1(j), j=1,131)
      write(4,*)(beta2(j), j=1,131)
      write(5,*)(beta3(j), j=1,131)
c      write(6,*)(beta4(j), j=1,131)
      write(7,*)(beta5(j), j=1,131)
      write(8,*)(beta6(j), j=1,131)
      write(9,*)(beta7(j), j=1,131)
      write(10,*)(beta8(j), j=1,131)
      write(11,*)(beta9(j), j=1,131)
      write(12,*)(beta10(j), j=1,131)
      write(13,*)(mub1(j),j=1,5)
      write(14,*)(mub2(j),j=1,5)
      write(15,*)(mub3(j),j=1,5)
c      write(16,*)(mub4(j),j=1,5)
      write(17,*)(mub5(j),j=1,5)
      write(18,*)(mub6(j),j=1,5)
      write(19,*)(mub7(j),j=1,5)
      write(20,*)(mub8(j),j=1,5)
      write(21,*)(mub9(j),j=1,5)
      write(22,*)(mub10(j),j=1,5)
      write(23,*)(tau(j),j=1,29)
      write(24,*)(gamma(j),j=1,1153)
      write(25,*)(sigma2)
c      write(26,*)(mut)
      write(27,*)(sigt)
c      write(28,*)(mug)
      write(29,*)(sigg)
c      write(30,*)(mub0)
      write(31,*)(sigb0)
      write(32,*)(sigb1)
      write(33,*)(sigb2)
      write(34,*)(sigb3)
c      write(35,*)(sigb4)
      write(36,*)(sigb5)
      write(37,*)(sigb6)
      write(38,*)(sigb7)
      write(39,*)(sigb8)
      write(40,*)(sigb9)
      write(41,*)(sigb10)
      end if
c
c     end mcmc loop
c

      end do

      stop
      end


c
c     subroutines
c

c      update sigma2
      subroutine getsigma2(i,n,player,team,game,opp,DinP,
     $  ast,stl,tno,ftm,ftp,fgm,fgp,orb,drb,asig,bsig,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta5,beta6,
     $  beta7,beta8,beta9,beta10,sigma2)
      implicit none
      integer n,i,player(8845),team(8845),DinP(8845)
      integer opp(8845), game(8845)
      double precision ast(8845),stl(8845),tno(8845),drb(8845)
      double precision ftm(8845),ftp(8845),fgm(8845)
      double precision fgp(8845),orb(8845)
      double precision beta0(29),tau(29),gamma(1153)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131)
```

```
      double precision beta8(131),beta9(131),beta10(131)
      double precision sigma2,summ,all,astar,bstar,asig,bsig
      real rastar,rbstar
      real gengam
      external gengam
        summ=0.d0
         all = 0.d0
       do i=1,8845
         all = all + 1.d0
        summ = summ + (dble(DinP(i)) -
     $   (gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                  beta1(player(i))*ast(i) +
     $                  beta2(player(i))*stl(i) +
     $                  beta3(player(i))*tno(i) +
     $                  beta5(player(i))*ftm(i) +
     $                  beta6(player(i))*ftp(i) +
     $                  beta7(player(i))*fgm(i) +
     $                  beta8(player(i))*fgp(i) +
     $                  beta9(player(i))*orb(i) +
     $                  beta10(player(i))*drb(i)))**2
       end do
      astar = (asig + (.5d0)*all)
      bstar = ((1.d0/bsig) + (.5d0)*summ)
      rastar = real(astar)
      rbstar = real(bstar)
      sigma2 = dble(gengam(rbstar,rastar))
      sigma2 = 1.d0/sigma2
      end

c     update gamma the intercept for the gamenumber being played
      subroutine getgamma(ii,n,player,team,opp,game,DinP,ast,stl,
     $   tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigg,mug,tau,beta0,
     $   beta1,beta2,beta3,beta5,beta6,beta7,beta8,beta9,
     $   beta10,gamma)
      implicit none
      integer n,i,ii,DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision drb(8845),orb(8845)
      double precision fgm(8845),fgp(8845),ftm(8845),ftp(8845)
      double precision tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131),beta8(131)
      double precision beta9(131),beta10(131)
      double precision gamma, sigma2, sigg, mug
      double precision mugamstar, sigma2gamstar
      double precision summ, sumy, sumgamma
      real zero, one
      real gennor
      external gennor
      zero = 0.0
      one = 1.0
      sumy=0.d0
      summ=0.d0
      sumgamma=0.d0
      do i = 1,8845
        if(game(i) .eq. ii) then
         summ = summ+1.d0
         sumy = sumy + dble(DinP(i))
         sumgamma = sumgamma +
     $               (tau(opp(i)) + beta0(team(i))  +
     $          beta1(player(i))*ast(i) +
     $          beta2(player(i))*stl(i) +
     $          beta3(player(i))*tno(i) +
     $          beta5(player(i))*ftm(i) +
     $          beta6(player(i))*ftp(i) +
     $          beta7(player(i))*fgm(i) +
     $          beta8(player(i))*fgp(i) +
     $          beta9(player(i))*orb(i) +
     $          beta10(player(i))*drb(i))
       end if
      end do
      mugamstar=(sigg*sumy - sigg*sumgamma + sigma2*mug)/
     $          (sigg*summ + sigma2)
      sigma2gamstar=dsqrt((sigma2*sigg)/(sigg*summ + sigma2))
      gamma = mugamstar+sigma2gamstar*dble(gennor(zero,one))
      end

c     update tau the intercept for opponent
      subroutine gettau(ii,n,player,team,opp,game,DinP,ast,stl,
     $   tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigt,mut,gamma,
     $   beta0,beta1,beta2,beta3,beta5,beta6,beta7,beta8,
     $   beta9,beta10,tau)
      implicit none
      integer n,i,ii,DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision drb(8845),orb(8845)
      double precision fgm(8845),fgp(8845),ftm(8845),ftp(8845)
```

```
      double precision gamma(1153), beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131)
      double precision beta8(131),beta9(131),beta10(131)
      double precision tau, sigma2, sigt, mut
      double precision mutaustar, sigma2taustar
      double precision summ, sumy, sumtau
      real zero, one
      real gennor
      external gennor
      zero = 0.0
      one = 1.0
      sumy=0.d0
      sumtau=0.d0
      summ=0.d0
      do i = 1,8845
       if(opp(i) .eq. ii) then
        summ = summ+1.d0
        sumy = sumy + dble(DinP(i))
        sumtau = sumtau +
     $   (gamma(game(i)) + beta0(team(i)) +
     $       beta1(player(i))*ast(i) +
     $       beta2(player(i))*stl(i) +
     $       beta3(player(i))*tno(i) +
     $       beta5(player(i))*ftm(i) +
     $       beta6(player(i))*ftp(i) +
     $       beta7(player(i))*fgm(i) +
     $       beta8(player(i))*fgp(i) +
     $       beta9(player(i))*orb(i) +
     $       beta10(player(i))*drb(i))
       end if
       end do
      mutaustar = (sigt*sumy - sigt*sumtau + sigma2*mut)/
     $                    (sigt*summ + sigma2)
      sigma2taustar = dsqrt((sigma2*sigt)/(sigt*summ+sigma2))
      tau = mutaustar + (sigma2taustar)*dble(gennor(zero,one))
      end


c     update beta0 intercept for team
      subroutine getbeta0(ii,n,player,team,opp,game,DinP,ast,stl,
     $ tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb0,mub0,gamma,
     $ tau,beta1,beta2,beta3,beta5,beta6,beta7,beta8,beta9,
     $ beta10,beta0)
      implicit none
      integer n,i,ii,DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision drb(8845),orb(8845)
      double precision fgm(8845),fgp(8845),ftm(8845),ftp(8845)
      double precision tau(29),gamma(1153)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131),beta8(131)
      double precision beta9(131),beta10(131)
      double precision beta0, sigma2, sigb0, mub0
      double precision mubeta0star, sigma2beta0star
      double precision summ, sumy, sumbeta
      real zero,one
      real gennor
      external gennor
      zero=0.0
      one=1.0
      sumy=0.d0
      sumbeta=0.d0
      summ=0.d0
       do i = 1,8845
       if (team(i) .eq. ii) then
        summ = summ + 1.d0
        sumy = sumy + dble(DinP(i))
        sumbeta = sumbeta +
     $     (gamma(game(i)) + tau(opp(i))  +
     $         beta1(player(i))*ast(i) +
     $         beta2(player(i))*stl(i) +
     $         beta3(player(i))*tno(i) +
     $         beta5(player(i))*ftm(i) +
     $         beta6(player(i))*ftp(i) +
     $         beta7(player(i))*fgm(i) +
     $         beta8(player(i))*fgp(i) +
     $         beta9(player(i))*orb(i) +
     $         beta10(player(i))*drb(i))
        end if
        end do
      mubeta0star=(sigb0*sumy - sigb0*sumbeta + sigma2*mub0)/
     $            (sigb0*summ + sigma2)
      sigma2beta0star=dsqrt((sigma2*sigb0)/(sigb0*summ+sigma2))
      beta0 = mubeta0star+sigma2beta0star*dble(gennor(zero,one))
      end

c     update beta1 assists
```

```
      subroutine getbeta1(ii,n,player,team,opp,game,pos,DinP,
     $  ast,stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb1,
     $  mub1,gamma,tau,beta0,beta2,beta3,beta5,beta6,beta7,
     $  beta8,beta9,beta10,position,mub1pos,beta1)
      implicit none
      integer n,i,ii,player(8845),team(8845),opp(8845),game(8845)
      integer pos(8845),DinP(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta2(131),beta3(131),beta5(131)
      double precision beta6(131),beta7(131),beta8(131)
      double precision beta9(131),beta10(131)
      double precision sigma2, sigb1, mub1(5)
      double precision mubeta1star, sigma2beta1star
      double precision sumya, sumbeta, suma
      double precision beta1, mub1pos,position
      real gennor, one, zero
      external gennor
      one = 1.0
      zero = 0.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
        if (player(i) .eq. ii) then
         suma = suma + ast(i)**2
         sumya = sumya + dble(DinP(i))*ast(i)
         sumbeta = sumbeta + ast(i)*(
     $    gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $        beta2(player(i))*stl(i) +
     $        beta3(player(i))*tno(i) +
     $        beta5(player(i))*ftm(i) +
     $        beta6(player(i))*ftp(i) +
     $        beta7(player(i))*fgm(i) +
     $        beta8(player(i))*fgp(i) +
     $        beta9(player(i))*orb(i) +
     $        beta10(player(i))*drb(i))
         mub1pos = mub1(pos(i))
         position=pos(i)
        end if
        end do
      mubeta1star=(sigb1*sumya-sigb1*sumbeta+sigma2*mub1pos)/
     $             (sigb1*suma + sigma2)
      sigma2beta1star=dsqrt((sigma2*sigb1)/(sigb1*suma+sigma2))
      beta1=mubeta1star+(sigma2beta1star)*dble(gennor(zero,one))
      mub1pos = mub1pos
      position=position
      end

c     update beta2 steals
      subroutine getbeta2(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb2,mub2,
     $  gamma,tau,beta0,beta1,beta3,beta5,beta6,beta7,beta8,
     $  beta9,beta10,mub2pos,beta2)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta3(131),beta5(131)
      double precision beta6(131),beta7(131),beta8(131),beta9(131)
      double precision beta10(131)
      double precision sigma2, sigb2, mub2(5)
      double precision beta2,mub2pos
      double precision mubeta2star, sigma2beta2star
      double precision sumya, sumbeta, suma
      real gennor,one,zero
      external gennor
      one=1.0
      zero=0.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
        if (player(i) .eq. ii) then
         suma = suma + (stl(i)*stl(i))
         sumya = sumya + dble(DinP(i))*stl(i)
         sumbeta = sumbeta + stl(i)*(
     $     gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                    beta1(player(i))*(ast(i)) +
     $                    beta3(player(i))*(tno(i)) +
     $                    beta5(player(i))*(ftm(i)) +
     $                    beta6(player(i))*(ftp(i)) +
     $                    beta7(player(i))*(fgm(i)) +
     $                    beta8(player(i))*(fgp(i)) +
```

```
     $                        beta9(player(i))*(orb(i)) +
     $                        beta10(player(i))*(drb(i)))
         mub2pos = mub2(pos(i))
        end if
        end do
      mubeta2star=(sigb2*sumya - sigb2*sumbeta + sigma2*mub2pos)/
     $              (sigb2*suma + sigma2)
      sigma2beta2star=dsqrt((sigma2*sigb2)/(sigb2*suma+sigma2))
      beta2=mubeta2star+sigma2beta2star*dble(gennor(zero,one))
      mub2pos=mub2pos
      end

c     update beta3 turnovers
      subroutine getbeta3(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb3,mub3,
     $  gamma,tau,beta0,beta1,beta2,beta5,beta6,beta7,beta8,
     $  beta9,beta10,mub3pos,beta3)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta5(131)
      double precision beta6(131),beta7(131),beta8(131),beta9(131)
      double precision beta10(131)
      double precision beta3,mub3pos, sigma2, sigb3, mub3(5)
      double precision mubeta3star, sigma2beta3star
      double precision sumya, sumbeta, suma
      real gennor,one,zero
      external gennor
      zero=0.0
      one=1.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
        if (player(i) .eq. ii) then
         suma = suma + (tno(i)*tno(i))
         sumya = sumya + dble(DinP(i))*tno(i)
         sumbeta = sumbeta + tno(i)*(
     $  gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                     beta1(player(i))*(ast(i)) +
     $                     beta2(player(i))*(stl(i)) +
     $                     beta5(player(i))*(ftm(i)) +
     $                     beta6(player(i))*(ftp(i)) +
     $                     beta7(player(i))*(fgm(i)) +
     $                     beta8(player(i))*(fgp(i)) +
     $                     beta9(player(i))*(orb(i)) +
     $                     beta10(player(i))*(drb(i)))
         mub3pos = mub3(pos(i))
        end if
        end do
      mubeta3star=(sigb3*sumya - sigb3*sumbeta + sigma2*mub3pos)/
     $              (sigb3*suma + sigma2)
      sigma2beta3star=dsqrt((sigma2*sigb3)/(sigb3*suma+sigma2))
      beta3=mubeta3star+sigma2beta3star*dble(gennor(zero,one))
      mub3pos = mub3pos
      end

c     update beta5 free throws made
      subroutine getbeta5(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb5,mub5,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta6,beta7,beta8,
     $  beta9,beta10,mub5pos,beta5)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta6(131),beta7(131),beta8(131),beta9(131)
      double precision beta10(131)
      double precision beta5,mub5pos, sigma2, sigb5, mub5(5)
      double precision mubeta5star, sigma2beta5star
      double precision sumya, sumbeta, suma
      real gennor,one,zero
      external gennor
      one=1.0
      zero=0.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
        if (player(i) .eq. ii) then
         suma = suma + (ftm(i)*ftm(i))
```

```
      sumya = sumya + dble(DinP(i))*ftm(i)
      sumbeta = sumbeta + ftm(i)*(
     $  gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                beta1(player(i))*ast(i) +
     $                beta2(player(i))*stl(i) +
     $                beta3(player(i))*tno(i) +
     $                beta6(player(i))*ftp(i) +
     $                beta7(player(i))*fgm(i) +
     $                beta8(player(i))*fgp(i) +
     $                beta9(player(i))*orb(i) +
     $                beta10(player(i))*drb(i))
      mub5pos = mub5(pos(i))
     end if
     end do
      mubeta5star=(sigb5*sumya - sigb5*sumbeta + sigma2*mub5pos)/
     $            (sigb5*suma + sigma2)
      sigma2beta5star=dsqrt((sigma2*sigb5)/(sigb5*suma+sigma2))
      beta5=mubeta5star+sigma2beta5star*dble(gennor(zero,one))
      mub5pos=mub5pos
      end

c     update beta6 free throw percentage
      subroutine getbeta6(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb6,mub6,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta5,beta7,beta8,
     $  beta9,beta10,mub6pos,beta6)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta7(131),beta8(131),beta9(131)
      double precision beta10(131)
      double precision beta6,mub6pos, sigma2, sigb6, mub6(5)
      double precision mubeta6star, sigma2beta6star
      double precision sumya, sumbeta, suma
      real gennor,one,zero
      external gennor
      one=1.0
      zero=0.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
        if (player(i) .eq. ii) then
      suma = suma + ftp(i)*ftp(i)
      sumya = sumya + dble(DinP(i))*ftp(i)
      sumbeta = sumbeta + ftp(i)*(
     $  gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                beta1(player(i))*ast(i) +
     $                beta2(player(i))*stl(i) +
     $                beta3(player(i))*tno(i) +
     $                beta5(player(i))*ftm(i) +
     $                beta7(player(i))*fgm(i) +
     $                beta8(player(i))*fgp(i) +
     $                beta9(player(i))*orb(i) +
     $                beta10(player(i))*drb(i))
      mub6pos = mub6(pos(i))
     end if
     end do
      mubeta6star=(sigb6*sumya - sigb6*sumbeta + sigma2*mub6pos)/
     $            (sigb6*suma + sigma2)
      sigma2beta6star=dsqrt((sigma2*sigb6)/(sigb6*suma+sigma2))
      beta6=mubeta6star+sigma2beta6star*dble(gennor(zero,one))
      mub6pos=mub6pos
      end

c     update beta7 field goals made
      subroutine getbeta7(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb7,mub7,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta5,beta6,beta8,
     $  beta9,beta10,mub7pos,beta7)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta8(131),beta9(131)
      double precision beta10(131)
      double precision beta7, mub7pos,sigma2, sigb7, mub7(5)
      double precision mubeta7star, sigma2beta7star
      double precision sumya, sumbeta, suma
      real gennor,one,zero
```

```
      external gennor
      one=1.0
      zero=0.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
         if (player(i) .eq. ii) then
          suma = suma + fgm(i)*fgm(i)
          sumya = sumya + dble(DinP(i))*fgm(i)
          sumbeta = sumbeta + fgm(i)*(
     $   gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                    beta1(player(i))*ast(i) +
     $                    beta2(player(i))*stl(i) +
     $                    beta3(player(i))*tno(i) +
     $                    beta5(player(i))*ftm(i) +
     $                    beta6(player(i))*ftp(i) +
     $                    beta8(player(i))*fgp(i) +
     $                    beta9(player(i))*orb(i) +
     $                    beta10(player(i))*drb(i))
          mub7pos = mub7(pos(i))
        end if
        end do
      mubeta7star=(sigb7*sumya - sigb7*sumbeta + sigma2*mub7pos)/
     $             (sigb7*suma + sigma2)
      sigma2beta7star=dsqrt((sigma2*sigb7)/(sigb7*suma+sigma2))
      beta7=mubeta7star+sigma2beta7star*dble(gennor(zero,one))
      mub7pos=mub7pos
      end

c     update beta8 field goal percentage
      subroutine getbeta8(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb8,mub8,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta5,beta6,beta7,
     $  beta9,beta10,mub8pos,beta8)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131),beta9(131)
      double precision beta10(131)
      double precision beta8,mub8pos, sigma2, sigb8, mub8(5)
      double precision mubeta8star, sigma2beta8star
      double precision sumya, sumbeta, suma
      real gennor,one,zero
      external gennor
      one=1.0
      zero=0.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
         if (player(i) .eq. ii) then
          suma = suma + fgp(i)*fgp(i)
          sumya = sumya + dble(DinP(i))*fgp(i)
          sumbeta = sumbeta + fgp(i)*(
     $   gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                    beta1(player(i))*ast(i) +
     $                    beta2(player(i))*stl(i) +
     $                    beta3(player(i))*tno(i) +
     $                    beta5(player(i))*ftm(i) +
     $                    beta6(player(i))*ftp(i) +
     $                    beta7(player(i))*fgm(i) +
     $                    beta9(player(i))*orb(i) +
     $                    beta10(player(i))*drb(i))
          mub8pos = mub8(pos(i))
        end if
        end do
      mubeta8star=(sigb8*sumya - sigb8*sumbeta + sigma2*mub8pos)/
     $             (sigb8*suma + sigma2)
      sigma2beta8star=dsqrt((sigma2*sigb8)/(sigb8*suma+sigma2))
      beta8=mubeta8star+sigma2beta8star*dble(gennor(zero,one))
      mub8pos=mub8pos
      end

c     update beta9 offensive rebounds
      subroutine getbeta9(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb9,mub9,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta5,beta6,beta7,
     $  beta8,beta10,mub9pos,beta9)
          implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
```

```
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131),beta8(131)
      double precision beta10(131)
      double precision beta9,mub9pos, sigma2, sigb9, mub9(5)
      double precision mubeta9star, sigma2beta9star
      double precision sumya, sumbeta, suma
      real zero,one
      real gennor
      external gennor
      zero=0.0
      one=1.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
         if (player(i) .eq. ii) then
          suma = suma + orb(i)*orb(i)
          sumya = sumya + dble(DinP(i))*orb(i)
          sumbeta = sumbeta + orb(i)*(
     $   gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                    beta1(player(i))*ast(i) +
     $                    beta2(player(i))*stl(i) +
     $                    beta3(player(i))*tno(i) +
     $                    beta5(player(i))*ftm(i) +
     $                    beta6(player(i))*ftp(i) +
     $                    beta7(player(i))*fgm(i) +
     $                    beta8(player(i))*fgp(i) +
     $                    beta10(player(i))*drb(i))
          mub9pos = mub9(pos(i))
        end if
        end do
      mubeta9star=(sigb9*sumya-sigb9*sumbeta+sigma2*mub9pos)/
     $             (sigb9*suma + sigma2)
      sigma2beta9star=dsqrt((sigma2*sigb9)/(sigb9*suma + sigma2))
      beta9= mubeta9star + sigma2beta9star*dble(gennor(zero,one))
      mub9pos=mub9pos
      end

c     update beta10 defensive rebounds
      subroutine getbeta10(ii,n,player,team,opp,game,pos,DinP,ast,
     $  stl,tno,ftm,ftp,fgm,fgp,orb,drb,sigma2,sigb10,mub10,
     $  gamma,tau,beta0,beta1,beta2,beta3,beta5,beta6,beta7,
     $  beta8,beta9,mub10pos,beta10)
      implicit none
      integer n,i,ii,pos(8845),DinP(8845)
      integer player(8845),team(8845),opp(8845),game(8845)
      double precision ast(8845),stl(8845),tno(8845)
      double precision ftm(8845),ftp(8845)
      double precision orb(8845),drb(8845),fgm(8845),fgp(8845)
      double precision gamma(1153),tau(29),beta0(29)
      double precision beta1(131),beta2(131),beta3(131)
      double precision beta5(131),beta6(131),beta7(131),beta8(131)
      double precision beta9(131)
      double precision beta10, sigma2, sigb10, mub10(5)
      double precision mubeta10star, sigma2beta10star
      double precision sumya, sumbeta, suma,mub10pos
      real zero,one
      real gennor
      external gennor
      zero=0.0
      one=1.0
      sumya=0.d0
      sumbeta=0.d0
      suma=0.d0
       do i= 1, 8845
         if (player(i) .eq. ii) then
          suma = suma + drb(i)*drb(i)
          sumya = sumya + dble(DinP(i))*drb(i)
          sumbeta = sumbeta + drb(i)*(
     $   gamma(game(i)) + tau(opp(i)) + beta0(team(i)) +
     $                    beta1(player(i))*ast(i) +
     $                    beta2(player(i))*stl(i) +
     $                    beta3(player(i))*tno(i) +
     $                    beta5(player(i))*ftm(i) +
     $                    beta6(player(i))*ftp(i) +
     $                    beta7(player(i))*fgm(i) +
     $                    beta8(player(i))*fgp(i) +
     $                    beta9(player(i))*orb(i))
          mub10pos = mub10(pos(i))
        end if
        end do
      mubeta10star=(sigb10*sumya-sigb10*sumbeta+sigma2*mub10pos)/
     $              (sigb10*suma + sigma2)
      sigma2beta10star=dsqrt((sigma2*sigb10)/(sigb10*suma+sigma2))
      beta10=mubeta10star+sigma2beta10star*dble(gennor(zero,one))
      mub10pos=mub10pos
      end
```

```
c  update mu_beta1 for the five positions
      subroutine getmubeta1(ii,n,sb1,mb1,beta1,sigb1,position,mub1)
      implicit none
      integer i,ii,n,plj
      double precision sigb1, sb1, mb1
      double precision summ,mub1,mustarb1,sigmastarb1
      double precision beta1(131),position(131)
      real gennor,zero,one
      external gennor
      zero = 0.0
      one = 1.0
      summ=0.d0
      plj=0.0
      do i = 1,131
        if(position(i) .eq. ii) then
          plj=plj + 1
         summ=summ + beta1(i)
       end if
      end do
      mustarb1 = (sb1*summ + mb1*sigb1)/(dble(plj)*sb1+sigb1)
      sigmastarb1 = dsqrt((sigb1*sb1)/(dble(plj)*sb1 + sigb1))
      mub1=mustarb1 + sigmastarb1*dble(gennor(zero,one))
      end

c update mu_beta2 for the five positions
      subroutine getmubeta2(ii,n,sb2,mb2,beta2,sigb2,position,mub2)
      implicit none
      integer  i,ii,n,plj
      double precision sigb2, sb2, mb2
      double precision summ,mub2,mustarb2,sigmastarb2
      double precision beta2(131),position(131)
      real gennor,one,zero
      external gennor
      one=1.0
      zero=0.0
      summ=0.d0
      plj=0.0
      do i = 1,131
        if(position(i) .eq. ii) then
          plj=plj + 1
         summ=summ + beta2(i)
       end if
      end do
      mustarb2 = (sb2*summ + mb2*sigb2)/(dble(plj)*sb2 + sigb2)
      sigmastarb2 = dsqrt((sigb2*sb2)/(dble(plj)*sb2 + sigb2))
      mub2 = mustarb2 + sigmastarb2*dble(gennor(zero,one))
      end

c update mu_beta3
      subroutine getmubeta3(ii,n,sb3,mb3,beta3,sigb3,position,mub3)
      implicit none
      integer i,ii,n,plj
      double precision sigb3, sb3, mb3
      double precision summ, mub3, mustarb3, sigmastarb3
      double precision beta3(131),position(131)
      real gennor, one,zero
      external gennor
      plj=0.0
      summ=0.d0
      do i = 1,131
        if(position(i) .eq. ii)then
          plj=plj+1
         summ=summ + beta3(i)
       end if
      end do
      mustarb3 = (sb3*summ + mb3*sigb3)/(dble(plj)*sb3 + sigb3)
      sigmastarb3 = dsqrt((sigb3*sb3)/(dble(plj)*sb3 + sigb3))
      mub3 = mustarb3 + sigmastarb3*dble(gennor(zero,one))
      end


c update mu_beta5
      subroutine getmubeta5(ii,n,sb5,mb5,beta5,sigb5,position,mub5)
      implicit none
      integer i,ii,n,plj
      double precision sigb5, sb5, mb5
      double precision summ, mub5, mustarb5, sigmastarb5
      double precision beta5(131),position(131)
      real gennor, one,zero
      external gennor
      plj=0.0
      summ=0.d0
      do i = 1,131
        if(position(i) .eq. ii)then
          plj=plj+1
         summ=summ + beta5(i)
       end if
```

```
      end do
      mustarb5 = (sb5*summ + mb5*sigb5)/(dble(plj)*sb5 + sigb5)
      sigmastarb5 = dsqrt((sigb5*sb5)/(dble(plj)*sb5 + sigb5))
      mub5 = mustarb5 + sigmastarb5*dble(gennor(zero,one))
      end

c update mu_beta6
      subroutine getmubeta6(ii,n,sb6,mb6,beta6,sigb6,position,mub6)
      implicit none
      integer i,ii,n,plj
      double precision sigb6, sb6, mb6
      double precision summ, mub6, mustarb6, sigmastarb6
      double precision beta6(131),position(131)
      real gennor, one,zero
      external gennor
      plj=0.0
      summ=0.d0
      do i = 1,131
       if(position(i) .eq. ii)then
        plj=plj+1
        summ=summ + beta6(i)
       end if
      end do
      mustarb6 = (sb6*summ + mb6*sigb6)/(dble(plj)*sb6 + sigb6)
      sigmastarb6 = dsqrt((sigb6*sb6)/(dble(plj)*sb6 + sigb6))
      mub6 = mustarb6 + sigmastarb6*dble(gennor(zero,one))
      end


c update mu_beta7
      subroutine getmubeta7(ii,n,sb7,mb7,beta7,sigb7,position,mub7)
      implicit none
      integer i,ii,n,plj
      double precision sigb7, sb7, mb7
      double precision summ,mub7,mustarb7,sigmastarb7
      double precision beta7(131),position(131)
      real rmsb7, rssb7
      real gennor, one,zero
      external gennor
      plj=0.0
      summ=0.d0
      do i = 1,131
       if(position(i) .eq. ii) then
        plj=plj+1
        summ=summ + beta7(i)
       end if
      end do
      mustarb7 = (sb7*summ + mb7*sigb7)/(dble(plj)*sb7 + sigb7)
      sigmastarb7 = dsqrt((sigb7*sb7)/(dble(plj)*sb7 + sigb7))
      mub7 = mustarb7 + sigmastarb7*dble(gennor(zero,one))
      end

c  update mu_beta8
      subroutine getmubeta8(ii,n,sb8,mb8,beta8,sigb8,position,mub8)
      implicit none
      integer i,ii,n,plj
      double precision sigb8, sb8, mb8
      double precision summ,mub8,mustarb8,sigmastarb8
      double precision beta8(131),position(131)
      real gennor, one,zero
      external gennor
      plj=0.0
      summ=0.d0
      do i = 1,131
       if(position(i) .eq. ii)then
        plj=plj+1
        summ=summ + beta8(i)
       end if
      end do
      mustarb8 = (sb8*summ + mb8*sigb8)/(dble(plj)*sb8 + sigb8)
      sigmastarb8 = dsqrt((sigb8*sb8)/(dble(plj)*sb8 + sigb8))
      mub8 = mustarb8 + sigmastarb8*dble(gennor(zero,one))
      end

c  update mu_beta9
      subroutine getmubeta9(ii,n,sb9,mb9,beta9,sigb9,position,mub9)
      implicit none
      integer i,ii,n,plj
      double precision sigb9, sb9, mb9
      double precision summ,mub9,mustarb9,sigmastarb9
      double precision beta9(131),position(131)
      real gennor, one,zero
      external gennor
      plj=0.0
      summ=0.d0
      do i = 1,131
       if(position(i) .eq. ii) then
        plj=plj+1
        summ=summ + beta9(i)
```

```
         end if
        end do
        mustarb9=(sb9*summ+mb9*sigb9)/(dble(plj)*sb9+sigb9)
        sigmastarb9 = dsqrt((sigb9*sb9)/(dble(plj)*sb9 + sigb9))
        mub9 = mustarb9 + sigmastarb9*dble(gennor(zero,one))
        end

c    update mu_beta10
        subroutine getmubeta10(ii,n,sb10,mb10,beta10,sigb10,
     $                        position,mub10)
        implicit none
        integer i,ii,n,plj
        double precision sigb10, sb10, mb10
        double precision summ,mub10,mustarb10,sigmastarb10
        double precision beta10(131),position(131)
        real gennor, one,zero
        external gennor
        plj=0.0
        summ=0.d0
        do i = 1,131
          if(position(i) .eq. ii)then
           plj=plj+1
           summ=summ + beta10(i)
         end if
        end do
        mustarb10 = (sb10*summ+mb10*sigb10)/(dble(plj)*sb10+sigb10)
        sigmastarb10 = dsqrt((sigb10*sb10)/(dble(plj)*sb10+sigb10))
        mub10 = mustarb10 + sigmastarb10*dble(gennor(zero,one))
        end

c    update sig_gamma
        subroutine getsiggamma(g,ag,bg,gamma,mug,sigg)
        implicit none
        integer g,i
        double precision mug, ag, bg
        double precision summ, sigg, astarg, bstarg
        double precision gamma(1153)
        real rastarg, rbstarg
        real gengam
        external gengam
        summ = 0.d0
        do i = 1,1153
         summ = summ + (gamma(i) - mug)**2
        end do
        astarg = (ag + (0.5d0)*dble(g))
        bstarg = ((1.d0/bg) + (0.5d0)*summ)
        rastarg = real(astarg)
        rbstarg = real(bstarg)
        sigg = dble(gengam(rbstarg, rastarg))
        sigg = 1.d0/sigg
        end

c    update sig_tau
        subroutine getsigtau(op,at,bt,tau,mut,sigt)
        implicit none
        integer op,i
        double precision mut, at, bt
        double precision summ, sigt, astart, bstart
        double precision tau(29)
        real rastart, rbstart
        real gengam
        external gengam
        summ = 0.d0
        do i = 1,29
         summ = summ + (tau(i) - mut)**2
        end do
        astart = (at + (.5d0)*dble(op))
        bstart = ((1.d0/bt) + (.5d0)*summ)
        rastart = real(astart)
        rbstart = real(bstart)
        sigt = dble(gengam(rbstart, rastart))
        sigt = 1.d0/sigt
        end

c    update sig_beta0
        subroutine getsigbeta0(tm,ab0,bb0,beta0,mub0,sigb0)
        implicit none
        integer tm, i
        double precision mub0, ab0, bb0
        double precision summ, sigb0, astarb0, bstarb0
        double precision beta0(29)
        real rastarb0, rbstarb0
        real gengam
        external gengam
        summ=0.d0
        do i=1,29
         summ = summ + (beta0(i) - mub0)**2
        end do
        astarb0 = (ab0 + (.5d0)*dble(tm))
```

```
      bstarb0 = ((1.d0/bb0) + (.5d0)*summ)
      rastarb0 = real(astarb0)
      rbstarb0 = real(bstarb0)
      sigb0 = dble(gengam(rbstarb0, rastarb0))
      sigb0 = 1.d0/sigb0
      end


c   update sig_beta1
      subroutine getsigbeta1(n,ab1,bb1,beta1,pl,mub1pos,sigb1)
      implicit none
      integer k,n,j,pl,i,counter
      double precision mub1(5), ab1, bb1
      double precision summ,sigb1, astarb1, bstarb1
      double precision beta1(131),mub1pos(131)
      real rastarb1, rbstarb1
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta1(i) - mub1pos(i))**2
       end do
      astarb1 = (ab1 + (.5d0)*dble(pl))
      bstarb1 = ((1.d0/bb1) + (.5d0)*summ)
      rastarb1 = real(astarb1)
      rbstarb1 = real(bstarb1)
      sigb1 = dble(gengam(rbstarb1, rastarb1))
      sigb1 = 1.d0/sigb1
      end

c   update sig_beta2
      subroutine getsigbeta2(n,ab2,bb2,beta2,pl,mub2pos,sigb2)
      implicit none
      integer n,j,pl,i
      double precision ab2, bb2
      double precision summ, sigb2, astarb2, bstarb2
      double precision beta2(131),mub2pos(131)
      real rastarb2, rbstarb2
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta2(i) - mub2pos(i))**2
      end do
      astarb2 = (ab2 + (.5d0)*dble(pl))
      bstarb2 = ((1.d0/bb2) + (.5d0)*summ)
      rastarb2 = real(astarb2)
      rbstarb2 = real(bstarb2)
      sigb2 = dble(gengam(rbstarb2, rastarb2))
      sigb2 = 1.d0/sigb2
      end

c   update sig_beta3
      subroutine getsigbeta3(n,ab3,bb3,beta3,pl,mub3pos,sigb3)
      implicit none
      integer n,j,pl,i
      double precision ab3, bb3
      double precision summ, sigb3, astarb3, bstarb3
      double precision beta3(131),mub3pos(131)
      real rastarb3, rbstarb3
      real gengam
      external gengam
      summ=0.d0
      do i=1,131
           summ = summ + (beta3(i) - mub3pos(i))**2
      end do
      astarb3 = (ab3 + (.5d0)*dble(pl))
      bstarb3 = ((1.d0/bb3) + (.5d0)*summ)
      rastarb3 = real(astarb3)
      rbstarb3 = real(bstarb3)
      sigb3 = dble(gengam(rbstarb3, rastarb3))
      sigb3 = 1.d0/sigb3
      end


c   update sig_beta5
      subroutine getsigbeta5(n,ab5,bb5,beta5,pl,mub5pos,sigb5)
          implicit none
      integer n,j,pl,i
      double precision ab5, bb5
      double precision summ, sigb5, astarb5, bstarb5
      double precision beta5(131),mub5pos(131)
      real rastarb5, rbstarb5
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta5(i) - mub5pos(i))**2
       end do
```

```
      astarb5 = (ab5 + (.5d0)*dble(pl))
      bstarb5 = ((1.d0/bb5) + (.5d0)*summ)
      rastarb5 = real(astarb5)
      rbstarb5 = real(bstarb5)
      sigb5 = dble(gengam(rbstarb5, rastarb5))
      sigb5 = 1.d0/sigb5
      end

c   update sig_beta6
      subroutine getsigbeta6(n,ab6,bb6,beta6,pl,mub6pos,sigb6)
      implicit none
      integer n,j,pl,i
      double precision ab6, bb6
      double precision summ, sigb6, astarb6, bstarb6
      double precision beta6(131),mub6pos(131)
      real rastarb6, rbstarb6
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta6(i) - mub6pos(i))**2
       end do
      astarb6 = (ab6 + (.5d0)*dble(pl))
      bstarb6 = ((1.d0/bb6) + (.5d0)*summ)
      rastarb6 = real(astarb6)
      rbstarb6 = real(bstarb6)
      sigb6 = dble(gengam(rbstarb6, rastarb6))
      sigb6 = 1.d0/sigb6
      end

c   update sig_beta7
      subroutine getsigbeta7(n,ab7,bb7,beta7,pl,mub7pos,sigb7)
      implicit none
      integer n,j,pl,i
      double precision ab7, bb7
      double precision summ, sigb7, astarb7, bstarb7
      double precision beta7(131),mub7pos(131)
      real rastarb7, rbstarb7
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta7(i) - mub7pos(i))**2
       end do
      astarb7 = (ab7 + (.5d0)*dble(pl))
      bstarb7 = ((1.d0/bb7) + (.5d0)*summ)
      rastarb7 = real(astarb7)
      rbstarb7 = real(bstarb7)
      sigb7 = dble(gengam(rbstarb7, rastarb7))
      sigb7 = 1.d0/sigb7
      end

c   update sig_beta8
      subroutine getsigbeta8(n,ab8,bb8,beta8,pl,mub8pos,sigb8)
      implicit none
      integer n,j,pl,i
      double precision ab8, bb8
      double precision summ, sigb8, astarb8, bstarb8
      double precision beta8(131),mub8pos(131)
      real rastarb8, rbstarb8
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta8(i) - mub8pos(i))**2
       end do
      astarb8 = (ab8 + (.5d0)*dble(pl))
      bstarb8 = ((1.d0/bb8) + (.5d0)*summ)
      rastarb8 = real(astarb8)
      rbstarb8 = real(bstarb8)
      sigb8 = dble(gengam(rbstarb8, rastarb8))
      sigb8 = 1.d0/sigb8
      end

c   update sig_beta9
      subroutine getsigbeta9(n,ab9,bb9,beta9,pl,mub9pos,sigb9)
      implicit none
      integer n,j,pl,i
      double precision ab9, bb9
      double precision summ, sigb9, astarb9, bstarb9
      double precision beta9(131),mub9pos(131)
      real rastarb9, rbstarb9
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta9(i)-mub9pos(i))**2
       end do
      astarb9 = (ab9 + (.5d0)*dble(pl))
```

```
      bstarb9 = ((1.d0/bb9) + (.5d0)*summ)
      rastarb9 = real(astarb9)
      rbstarb9 = real(bstarb9)
      sigb9 = dble(gengam(rbstarb9, rastarb9))
      sigb9 = 1.d0/sigb9
      end

c  update sig_beta10
      subroutine getsigbeta10(n,ab10,bb10,beta10,pl,mub10pos,sigb10)
      implicit none
      integer n,j,pl,i
      double precision ab10, bb10
      double precision summ, sigb10, astarb10, bstarb10
      double precision beta10(131),mub10pos(131)
      real rastarb10, rbstarb10
      real gengam
      external gengam
      summ=0.d0
       do i=1,131
         summ = summ + (beta10(i) - mub10pos(i))**2
       end do
      astarb10 = (ab10 + (.5d0)*dble(pl))
      bstarb10 = ((1.d0/bb10) + (.5d0)*summ)
      rastarb10 = real(astarb10)
      rbstarb10 = real(bstarb10)
      sigb10 = dble(gengam(rbstarb10, rastarb10))
      sigb10 = 1.d0/sigb10
      end
```

A.2 R code

Listing A.2: **R code to format data and summarize draws from the joint posterior distribution**

```
## read in and format data
## These are the standardized data with the gamenumber on them
## POINT GUARD
pg <- read.csv("/Users/garrittpage/mastersproject/data/pointguards/pg96_97.csv",header=TRUE)
pg$totpointdif <- pg$pointdiff
pg <- pg[,c(-21)]
pg <- stndrdzdatamaker(pg)

## SHOOTING GUARD
sg <- read.csv("/Users/garrittpage/mastersproject/data/offguards/sg96_97.csv",header=TRUE)
sg$totpointdif <- sg$pointdiff
sg <- sg[,c(-21)]
sg <- stndrdzdatamaker(sg)

## CENTERS
c <- read.csv("/Users/garrittpage/mastersproject/data/centers/c96_97.csv",header=TRUE)
c$totpointdif <- c$pointdiff
c <- c[,c(-21)]
c <- stndrdzdatamaker(c)

## SMALL FORWARDS
sf <- read.csv("/Users/garrittpage/mastersproject/data/smallforwards/sf96_97.csv",header=TRUE)
sf$totpointdif <- sf$pointdiff
sf <- sf[,c(-21)]
sf <- stndrdzdatamaker(sf)

## POWER FORWARDS
pf <- read.csv("/Users/garrittpage/mastersproject/data/powerforwards/pf96_97.csv",header=TRUE)
pf$totpointdif <- pf$pointdiff
pf <- pf[,c(-21)]
pf <- stndrdzdatamaker(pf)

## Starters by Position
nba <- rbind(pg,sg,c,pf,sf)
nba <- nba[,c(1:7,32:33,35:45)]

## subsets the data to only players that have played 45 games
k <- length(unique(nba$Player))
namevec <- unique(nba$Player)
nba2 <- as.data.frame(matrix(NA, nrow=1,ncol=20))
names(nba2) <- names(nba)
for(i in 1:k){
        if(length(nba$totpointdif[nba$Player==namevec[i]]) > 45){
                nba1 <- (subset(nba, nba$Player==namevec[i]))
                nba2 <- rbind(nba2,nba1)
                }
                cat("iteration",i,"\n")
        }
```

```
nba2 <- nba2[-1,]

nba2$Date <- as.date(as.character(nba2$date))

nba2 <- nba2[order(nba2$vsteam),]
tabjunk <- table(nba2$vsteam)
tempjunk <- rep(1:length(unique(nba2$vsteam)),times=tabjunk)
nba2$opp <- tempjunk
oppvec <- unique(nba2$vsteam)

nba2 <- nba2[order(nba2$team),]
nba2$TeamF <- match(nba2$team,unique(nba2$team))
teamvec <- unique(nba2$team)

nba2 <- nba2[order(nba2$Player),]
nba2$PlayerF <- match(nba2$Player,unique(nba2$Player))
namevec <- unique(nba2$Player)

nba2 <- nba2[order(nba2$Position),]
nba2$PositionF <- match(nba2$Position,unique(nba2$Position))
posvec <- unique(nba2$Position)

nba3 <- nba2[order(nba2$Player,nba2$team),]

nbaready <- nba3[,c(8:20,22:25)]
write(t(nbaready),"/Users/garrittpage/mastersproject/fortran_stuff/nobeta4pts/nbafortranready.
    txt",ncolumns=17)



## begin the cross validation by randomly selecting 10 games
sample(1:1163,10)

## take out randomly selecte games
nbacv <- nba3[nba3$game!=1116 & nba3$game!=954 & nba3$game!=241 &
              nba3$game!=898 & nba3$game!=486 & nba3$game!=19 &
              nba3$game!=668 & nba3$game!=398 & nba3$game!=772 &
              nba3$game!=124,]

## reassign game numbers
cvoppvec <- unique(nbacv[order(nbacv$vsteam),]$vsteam)
cvteamvec <- unique(nbacv[order(nbacv$team),]$team)
cvnamevec <- unique(nbacv[order(nbacv$Player),]$Player)
cvposvec <- unique(nbacv[order(nbacv$Position),]$Position)
nbacrossvalid[order(nbacv$game),]$game <- match(nbacv[order(nbacv$game),]$game, unique(nbacv[
    order(nbacv$game),]$game))

write(t(nbacv[,c(8:20,22:25)]),"/Users/garrittpage/mastersproject/fortran_stuff/nob4crossvalid
    /nbacrossvalid.txt", ncolumns=17)


##
## function that generates nba data used for model check simulation
##
datgen <- function(teamnum,gamenum){
    out <- NULL
    playermat <- matrix(1:(5*teamnum),byrow=TRUE,ncol=5)
    position <- rep(1:5,times=(2*gamenum))
    game <- rep(1:((gamenum*teamnum)/2),each=10)

    ## number of actual games played
    g <- ((gamenum*teamnum)/2)
    obs <- 5*teamnum*gamenum

    ## generate the team and player factor
    g1 <- NULL
    for(i in 1:(gamenum)){
        gg <- sample(1:teamnum,teamnum,replace=FALSE)
        g1 <- c(g1,gg)
        }
    player <- c(t(playermat[g1,]))
    team <- rep(g1,each=5)

    ## generate the opponent factor
    opp <- NULL
    for(i in 1:((teamnum*gamenum)/2)){
         oppO <- g1[(2*i-1)]
         oppE <- g1[(2*i)]
         opp <- c(opp,oppE,oppO)
        }
    opponent <- rep(opp,each=5)

    ## generate the means and variances for each slope (beta)
    sigbi <- matrix(1/rgamma(9*g,3.778,scale=0.09),nrow=9,ncol=g)
    mub1 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
    mub2 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
    mub3 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
    mub4 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
```

```
mub5 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
mub6 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
mub7 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
mub8 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)
mub9 <- matrix(rnorm(5*g,0,sqrt(225)),nrow=5,ncol=g)

mub <- rbind(mub1,mub2,mub3,mub4,mub5,mub6,mub7,mub8,mub9)

## generate all nine effect amounts for each position by game
betas <- matrix(NA,nrow=obs,ncol=9)
for(k in 0:(g-1)){
 g2 <- matrix(NA,nrow=5,ncol=9)
   for(j in 0:(9-1)){
    for(i in 1:5){
     g2[i,(j+1)] <- rnorm(1,mub[(5*j+1):(5*j+5),][i,k+1],sqrt(
                              sigbi[(j+1),(k+1)]))
         }
      }
       g2 <- rbind(g2,g2*(-1))
       betas[(10*k+1):(10*k+10),] <- g2
}

dimnames(betas)[[2]] <- c("ast","stl","tno","ftm","ftp","fgm",
                          "fgp","orb","drb")
depvar <- cbind((betas[,1])/48,(betas[,2])/48,(betas[,3])/48,
                (betas[,4])/48,(betas[,5])/100,(betas[,6])/48,
                (betas[,7])/100,(betas[,8])/48,(betas[,9])/48)

dimnames(depvar)[[2]] <- c("ast","stl","tno","ftm","ftp","fgm",
                           "fgp","orb","drb")


## generate the game (gamma) effect
sigg <- 1/rgamma(g,2.44,scale=0.3461)
gamma <- numeric(g*10)
for(i in (0:(g-1))){
    gamma[(10*i+1):(10*i+10)]<-rep(rnorm(1,0,sqrt(sigg[i+1])),10)
    }

## generate the team (tau) effect
sigt <- 1/rgamma(teamnum,11,scale=0.011111)
taumat <- matrix(NA,nrow=teamnum,ncol=5)
for(i in 1:teamnum){
    taumat[i,] <- rep(rnorm(1,0,sqrt(sigt[i])),each=5)
    }
tau <- c(t(taumat[g1,]))

## generate the opponent (phi) effect
sigp <- 1/rgamma(teamnum,11,scale=0.011111)
phimat <- matrix(NA,nrow=teamnum,ncol=5)
for(i in 1:teamnum){
  phimat[i,] <- rep(rnorm(1,0,sqrt(sigt[i])),each=5)
  }
phi <- c(t(phimat[opp,]))

## generate the overall variance (sigma2)
sigma2 <- 1/rgamma(obs,5.25,scale=0.006)

## generate difference in points
DinP <- numeric(0)
for(k in 0:(g-1)){
s <- sample(1:10,1)
bet <- numeric(0)
for(i in 1:9){
 bet[i] <- (depvar[,i][(10*k+1):(10*k+10)][s])
 summ <- sum(bet)}

 ggg1 <- ceiling(rep(rnorm(1,(tau[(10*k+1):(10*k+10)][s] +
                              phi[(10*k+1):(10*k+10)][s] +
                              gamma[(10*k+1):(10*k+10)][s] +
                              summ),sqrt(sigma2[k+1])),each=5))
 ggg1 <- c(ggg1,ggg1*(-1))
 DinP[(10*k+1):(10*k+10)] <- ggg1
}

out$data <- cbind(DinP,team,opponent,position,player,game,depvar)
out$sigg <- sigg
out$sigma2 <- sigma2
out$sigt <- sigt
out$sigp <- sigp
out$sigbi <- sigbi
out$mub <- mub
out$origdat <- data.frame(DinP,team,opponent,position,player,
                          game,betas,gamma,tau,phi)
out$numbers <- c(paste("games=",g,sep=""),
                 paste("teams=",teamnum,sep=""),
                 paste("players=",5*teamnum,sep=""),
                 paste("obs=",obs,sep=""))
out
```

```
}

## read in the appropriate posterior draws
setwd("/Users/garrittpage/mastersproject/fortran_stuff/nobeta4pts")
mub1 <- pos(1)/48
mub2 <- pos(2)/48
mub3 <- pos(3)/48
mub5 <- pos(5)/48
mub6 <- pos(6)/100
mub7 <- pos(7)/48
mub8 <- pos(8)/100
mub9 <- pos(9)/48
mub10 <- pos(10)/48

## Posterior summaries for the five positions and nine slopes
mean(mub1)
apply(mub1,2,sd)
apply(mub1,2,hpd)

mean(mub2)
apply(mub2,2,sd)
apply(mub2,2,hpd)

mean(mub3)
apply(mub3,2,sd)
apply(mub3,2,hpd)

mean(mub5)
apply(mub5,2,sd)
apply(mub5,2,hpd)

mean(mub6)
apply(mub6,2,sd)
apply(mub6,2,hpd)

mean(mub7)
apply(mub7,2,sd)
apply(mub7,2,hpd)

mean(mub8)
apply(mub8,2,sd)
apply(mub8,2,hpd)

mean(mub9)
apply(mub9,2,sd)
apply(mub9,2,hpd)

mean(mub10)
apply(mub10,2,sd)
apply(mub10,2,hpd)



## Plot of the position effect for each slope
par(mfrow=c(3,2))

## assists
plot(density(mub1[,5],adjust=2),main="Assists", xlim=c(0,.8),ylim=c(0,10),lwd=1.5,lty=1,xlab="
     Difference in Points")
lines(density(mub1[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub1[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub1[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub1[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(0,10,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.15,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

## steals
plot(density(mub2[,5],adjust=2),main="Steals", xlim=c(-0.1,1),ylim=c(0,6),lwd=1.5,lty=1, xlab=
     "Difference in Points")
lines(density(mub2[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub2[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub2[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub2[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(-0.1,6,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red",
     "green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

## turnovers
plot(density(mub3[,5],adjust=2),main="Turn Overs",xlim=c(-.8,-.1),ylim=c(0,6),lwd=1.5,lty=1,
     xlab="Difference in Points")
lines(density(mub3[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub3[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub3[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub3[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(-.8,6,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

 ## Offensive Rebounds
```

```
plot(density(mub9[,5],adjust=2),main="Offensive Rebounds", xlim=c(-.5,.2),ylim=c(0,7),lwd=1.5,
     lty=1,  xlab="Difference in Points")
lines(density(mub9[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub9[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub9[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub9[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(-.5,7,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

## Defensive Rebounds
plot(density(mub10[,5],adjust=2),main="Defensive Rebounds", xlim=c(0,1),ylim=c(0,10), lwd=1.5,
     lty=1, xlab="Difference in Points")
lines(density(mub10[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub10[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub10[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub10[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(0,10,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2, pt.bg="magenta")


par(mfrow=c(2,2))
## ftm
plot(density(mub5[,5],adjust=2),main="Free Throws Made", xlim=c(-.2,.4),ylim=c(0,8),lwd=1.5,
     lty=1, xlab="Difference in Points")
lines(density(mub5[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub5[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub5[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub5[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(-.2,8,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

## free throw percentage
plot(density(mub6[,5],adjust=2),main="Free Throw Percentage", xlim=c(-.05,.05),ylim=c(0,70),
     lwd=1.5,xlab="Difference in Points")
lines(density(mub6[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub6[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub6[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub6[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(-.05,70,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red"
     ,"green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

## field goal made
plot(density(mub7[,5],adjust=2),main="Field Goals Made", xlim=c(-.3,.4),ylim=c(0,8),lwd=1.5,
     xlab="Difference in Points")
lines(density(mub7[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub7[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub7[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub7[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(-.3,8,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")

 ## field goal percentage
plot(density(mub8[,5],adjust=2),main="Field Goal Percentage", xlim=c(0,.3),ylim=c(0,45),lwd
     =1.5, xlab="Difference in Points")
lines(density(mub8[,4],adjust=2),col="blue",lty=1,lwd=1.5)
lines(density(mub8[,3],adjust=2),col="red",lty=1,lwd=1.5)
lines(density(mub8[,2],adjust=2),col="green",lty=1,lwd=1.5)
lines(density(mub8[,1],adjust=2),col="purple",lty=1,lwd=1.5)
legend(0,45,c("SG","SF","PG","PF","C"),lty=c(1,1,1,1,1),lwd=2.25,col=c("black","blue","red","
     green","purple"),cex=1.25,ncol=2,pt.cex=1.2,pt.bg="magenta")



## Function that finds the differences for the standardized data
stndrdzdatamaker <- function(data){

        data$obsnum <- 1:nrow(data)

  difmaker <- function(var){
        perdif <- numeric(nrow(data))
        for(i in 1:length(perdif)){
                if(data$obsnum[i] %% 2 == 1)
                        perdif[i] <- (var[i] - var[i+1])
                else
                        perdif[i] <- (var[i] - var[i-1])
                        }
                        perdif
                }



        data$sscoredif <- difmaker(data$spoints)
        data$sastdif <- difmaker(data$sassist)
        data$sorbdif <- difmaker(data$sorb)
        data$sdrbdif <- difmaker(data$sdrb)
        data$sstealdif <- difmaker(data$ssteals)
        data$sfgmdif <- difmaker(data$sfgm)
        data$fgperdif <- difmaker(data$fgpercent)
```

```
        data$sftmdif <- difmaker(data$sftm)
        data$ftperdif <- difmaker(data$ftpercent)
        data$sfouldif <- difmaker(data$sfouls)
        data$stodif <- difmaker(data$sturnovers)

        data
}

## function to scans and formats FORTRAN output
formatbeta <- function(parameter,numparms,columnnames=NA){
          call <- paste(parameter,".txt",sep="")
          data <- scan(call)
          data <- matrix(data,byrow=T,ncol=numparms)
          data <- data.frame(data)
          names(data) <- columnnames
          data}


## Function that gives values for the inversegamma distribution
igamma.ab <- function(m,sd){
        out <- NULL
        v <- sd^2
        a <- (m^2/v) +2
        b <- v/(m^3+m*v)
        out$a <- a
        out$b <- b
        out}


## Function that creates the hpd's of the parameters
hpd <- function(parms){
       up <- sort(parms)[(ceiling(.95*length(parms))+1):length(parms)]
       low <- sort(parms)[1:(floor(.05*length(parms)))]
       diff <- up-low
       pick <- (1:length(diff))[diff==min(diff)]
       lower <- low[pick]
       upper <- up[pick]
       out <- c(lower,upper)
       out}

## these are the posterior predictive distributions for the data
   generation simulation
tau <- formatbeta("beta0",29,teamvec)
phi <- formatbeta("tau",29,teamvec)
b1 <- formatbeta("beta1",131,namevec)
b2 <- formatbeta("beta2"131,namevec)
b3 <- formatbeta("beta3"131,namevec)
b5 <- formatbeta("beta5"131,namevec)
b6 <- formatbeta("beta6"131,namevec)
b7 <- formatbeta("beta7"131,namevec)
b8 <- formatbeta("beta8"131,namevec)
b9 <- formatbeta("beta9"131,namevec)
b10 <- formatbeta("beta10"131,namevec)

junk2 <- c(11,14,20,17,18,15,16,12,13)

sigma2 <- onedim("sigma2")

## Game 19 posterior predictive distributions
## Golden State Warriors
## Mark Price
pricepredmean <- tau[,9] + phi[,22] + cbind(b1[,95],b2[,95],b3[,95],b5[,95],b6[,95],b7[,95],b8
     [,95],b9[,95],b10[,95])%*%t(nba3[nba3$game==19 & nba3$Player=="PRICE",junk2])
pricepostpred <- rnorm(50000,pricepredmean,sqrt(sigma2)[,1])
pricepostmean <- mean(pricepostpred)
priceprobwin <- sum(pricepostpred > 0)/50000
write.table(pricepostpred,"/Users/garrittpage/mastersproject/postpred/pricepostpred19")


## Joe Smith
jsmithpredmean <- tau[,9] + phi[,22] + cbind(b1[,57],b2[,57],b3[,57],b5[,57],b6[,57],b7[,57],
     b8[,57],b9[,57],b10[,57])%*%t(nba3[nba3$game==19 & nba3$Player=="J_SMITH",junk2])
jsmithpostpred <- rnorm(50000,jsmithpredmean,sqrt(sigma2)[,1])
jsmithpostmean <- mean(jsmithpostpred)
jsmithprobwin <- sum(jsmithpostpred > 0)/50000
write.table(jsmithpostpred,"/Users/garrittpage/mastersproject/postpred/jsmithpostpred19")

## Chris Mullin
mullinpredmean <- tau[,9] + phi[,22] + cbind(b1[,80],b2[,80],b3[,80],b5[,80],b6[,80],b7[,80],
     b8[,80],b9[,80],b10[,80])%*%t(nba3[nba3$game==19 & nba3$Player=="MULLIN",junk2])
mullinpostpred <- rnorm(50000,mullinpredmean,sqrt(sigma2)[,1])
mullinpostmean <- mean(mullinpostpred)
mullinprobwin <- sum(mullinpostpred > 0)/50000
write.table(mullinpostpred,"/Users/garrittpage/mastersproject/postpred/mullinpostpred19")

## Latrell Sprewell
sprewellpredmean <- tau[,9] + phi[,22] + cbind(b1[,117],b2[,117],b3[,117],b5[,117],b6[,117],b7
     [,117],b8[,117],b9[,117],b10[,117])%*%t(nba3[nba3$game==19 & nba3$Player=="SPREWELL",junk2
     ])
```

```
sprewellpostpred <- rnorm(50000,sprewellpredmean,sqrt(sigma2)[,1])
sprewellpostmean <- mean(sprewellpostpred)
sprewellprobwin <- sum(sprewellpostpred > 0)/50000
write.table(sprewellpostpred,"/Users/garrittpage/mastersproject/postpred/sprewellpostpred19")


## Portland Trailblazers
## Cliff_ROBINSON
crobinsonpredmean <- tau[,22] + phi[,9] + cbind(b1[,21],b2[,21],b3[,21],b5[,21],b6[,21],b7
    [,21],b8[,21],b9[,21],b10[,21])%*%t(nba3[nba3$game==19 & nba3$Player==" Cliff_ROBINSON",
    junk2])
crobinsonpostpred <- rnorm(50000,crobinsonpredmean,sqrt(sigma2)[,1])
crobinsonpostmean <- mean(crobinsonpostpred)
crobinsonprobwin <- sum(crobinsonpostpred > 0)/50000
write.table(crobinsonpostpred,"/Users/garrittpage/mastersproject/postpred/crobinsonpostpred19"
    )

## K_ANDERSON
kandersonpredmean <- tau[,22] + phi[,9] + cbind(b1[,60],b2[,60],b3[,60],b5[,60],b6[,60],b7
    [,60],b8[,60],b9[,60],b10[,60])%*%t(nba3[nba3$game==19 & nba3$Player=="K_ANDERSON",junk2])
kandersonpostpred <- rnorm(50000,kandersonpredmean,sqrt(sigma2)[,1])
kandersonpostmean <- mean(kandersonpostpred)
kandersonprobwin <- sum(kandersonpostpred > 0)/50000
write.table(kandersonpostpred,"/Users/garrittpage/mastersproject/postpred/kandersonpostpred19"
    )

## RIDER
riderpredmean <- tau[,22] + phi[,9] + cbind(b1[,100],b2[,100],b3[,100],b5[,100],b6[,100],b7
    [,100],b8[,100],b9[,100],b10[,100])%*%t(nba3[nba3$game==19 & nba3$Player=="RIDER",junk2])
riderpostpred <- rnorm(50000,riderpredmean,sqrt(sigma2)[,1])
riderpostmean <- mean(riderpostpred)
riderprobwin <- sum(riderpostpred > 0)/50000
write.table(riderpostpred,"/Users/garrittpage/mastersproject/postpred/riderpostpred19")

## R_WALLACE
rwallacepredmean <- tau[,22] + phi[,9] + cbind(b1[,109],b2[,109],b3[,109],b5[,109],b6[,109],b7
    [,109],b8[,109],b9[,109],b10[,109])%*%t(nba3[nba3$game==19 & nba3$Player=="R_WALLACE",
    junk2])
rwallacepostpred <- rnorm(50000,rwallacepredmean,sqrt(sigma2)[,1])
rwallacepostmean <- mean(rwallacepostpred)
rwallaceprobwin <- sum(rwallacepostpred > 0)/50000
write.table(rwallacepostpred,"/Users/garrittpage/mastersproject/postpred/rwallacepostpred19")

## SABONIS
sabonispredmean <- tau[,22] + phi[,9] + cbind(b1[,110],b2[,110],b3[,110],b5[,110],b6[,110],b7
    [,110],b8[,110],b9[,110],b10[,110])%*%t(nba3[nba3$game==19 & nba3$Player=="SABONIS",junk2
    ])
sabonispostpred <- rnorm(50000,sabonispredmean,sqrt(sigma2)[,1])
sabonispostmean <- mean(sabonispostpred)
sabonisprobwin <- sum(sabonispostpred > 0)/50000
write.table(sabonispostpred,"/Users/garrittpage/mastersproject/postpred/sabonispostpred19")

par(mfrow=c(3,3))
plot(density(pricepostpred),main="MARK_PRICE",xlab=" Difference_in_Points",ylab="")
plot(density(jsmithpostpred),main="JOE_SMITH",xlab=" Difference_in_Points",ylab="")
plot(density(mullinpostpred),main="CHRIS_MULLIN",xlab=" Difference_in_Points",ylab="")
plot(density(sprewellpostpred),main="LATRELL_SPREWELL",xlab=" Difference_in_Points",ylab="")
plot(density(crobinsonpostpred),main="CLIFF_ROBINSON",xlab=" Difference_in_Points",ylab="")
plot(density(kandersonpostpred),main="KENNY_ANDERSON",xlab=" Difference_in_Points",ylab="")
plot(density(riderpostpred),main="ISAIAH_RIDER",xlab=" Difference_in_Points",ylab="")
plot(density(rwallacepostpred),main="RASHEED_WALLACE",xlab=" Difference_in_Points",ylab="")
plot(density(sabonispostpred),main="ARVYDIS_SABONIS",xlab=" Difference_in_Points",ylab="")
means19 <- c(pricepostmean,jsmithpostmean,mullinpostmean,sprewellpostmean,crobinsonpostmean,
    kandersonpostmean,riderpostmean,rwallacepostmean,sabonispostmean)

######GAME 124 Posterior predictive
######
## UTAH JAZZ
## HORNACEK
hornpredmean <- tau[,27] + phi[,13] + cbind(b1[,47],b2[,47],b3[,47],b5[,47],b6[,47],b7[,47],b8
    [,47],b9[,47],b10[,47])%*%t(nba3[nba3$game==124 & nba3$Player=="HORNACEK",junk2])
hornpostpred <- rnorm(50000,hornpredmean,sqrt(sigma2)[,1])
hornpostmean <- mean(hornpostpred)
hornprobwin <- sum(hornpostpred > 0)/50000
write.table(sabonispostpred,"/Users/garrittpage/mastersproject/postpred/sabonispostpred19")

## MALONE
malonepredmean <- tau[,27] + phi[,13] + cbind(b1[,69],b2[,69],b3[,69],b5[,69],b6[,69],b7[,69],
    b8[,69],b9[,69],b10[,69])%*%t(nba3[nba3$game==124 & nba3$Player=="MALONE",junk2])
malonepostpred <- rnorm(50000,malonepredmean,sqrt(sigma2)[,1])
malonepostmean <- mean(malonepostpred)
maloneprobwin <- sum(malonepostpred > 0)/50000

## OSTERTAG
tagpredmean <- tau[,27] + phi[,13] + cbind(b1[,89],b2[,89],b3[,89],b5[,89],b6[,89],b7[,89],b8
    [,89],b9[,89],b10[,89])%*%t(nba3[nba3$game==124 & nba3$Player=="OSTERTAG",junk2])
tagpostpred <- rnorm(50000,tagpredmean,sqrt(sigma2)[,1])
tagpostmean <- mean(tagpostpred)
tagprobwin <- sum(tagpostpred > 0)/50000
```

```r
## RUSSELL
russpredmean <- tau[,27] + phi[,13] + cbind(b1[,104],b2[,104],b3[,104],b5[,104],b6[,104],b7
      [,104],b8[,104],b9[,104],b10[,104])%*%t(nba3[nba3$game==124 & nba3$Player=="RUSSELL",junk2
      ])
russpostpred <- rnorm(50000,russpredmean,sqrt(sigma2)[,1])
russpostmean <- mean(russpostpred)
russprobwin <- sum(russpostpred > 0)/50000

## STOCKTON
stockpredmean <- tau[,27] + phi[,13] + cbind(b1[,120],b2[,120],b3[,120],b5[,120],b6[,120],b7
      [,120],b8[,120],b9[,120],b10[,120])%*%t(nba3[nba3$game==124 & nba3$Player=="STOCKTON",
      junk2])
stockpostpred <- rnorm(50000,stockpredmean,sqrt(sigma2)[,1])
stockpostmean <- mean(stockpostpred)
stockprobwin <- sum(stockpostpred > 0)/50000

### LA Lakers
## CAMPBELL
campbellpredmean <- tau[,13] + phi[,27] + cbind(b1[,14],b2[,14],b3[,14],b5[,14],b6[,14],b7
      [,14],b8[,14],b9[,14],b10[,14])%*%t(nba3[nba3$Player=="CAMPBELL",junk2])
campbellpostpred <- rnorm(50000,campbellpredmean,sqrt(sigma2)[,1])
campbellpostmean <- mean(campbellpostpred)
campbellprobwin <- sum(campbellpostpred > 0)/50000

## E_JONES
ejonespredmean <- tau[,13] + phi[,27] + cbind(b1[,34],b2[,34],b3[,34],b5[,34],b6[,34],b7[,34],
      b8[,34],b9[,34],b10[,34])%*%t(nba3[nba3$game==124 & nba3$Player=="E_JONES",junk2])
ejonespostpred <- rnorm(50000,ejonespredmean,sqrt(sigma2)[,1])
ejonespostmean <- mean(ejonespostpred)
ejonesprobwin <- sum(ejonespostpred > 0)/50000

## ONEAL
onealpredmean <- tau[,13] + phi[,27] + cbind(b1[,88],b2[,88],b3[,88],b5[,88],b6[,88],b7[,88],
      b8[,88],b9[,88],b10[,88])%*%t(nba3[nba3$game==124 & nba3$Player=="ONEAL",junk2])
onealpostpred <- rnorm(50000,onealpredmean,sqrt(sigma2)[,1])
onealpostmean <- mean(onealpostpred)
onealprobwin <- sum(onealpostpred > 0)/50000

## SCOTT
scottpredmean <- tau[,13] + phi[,27] + cbind(b1[,112],b2[,112],b3[,112],b5[,112],b6[,112],b7
      [,112],b8[,112],b9[,112],b10[,112])%*%t(nba3[nba3$game==124 & nba3$Player=="SCOTT",junk2])
scottpostpred <- rnorm(50000,scottpredmean,sqrt(sigma2)[,1])
scottpostmean <- mean(scottpostpred)
scottprobwin <- sum(scottpostpred > 0)/50000

## VAN_EXEL
exelpredmean <- tau[,13] + phi[,27] + cbind(b1[,126],b2[,126],b3[,126],b5[,126],b6[,126],b7
      [,126],b8[,126],b9[,126],b10[,126])%*%t(nba3[nba3$game==124 & nba3$Player=="VAN_EXEL",
      junk2])
exelpostpred <- rnorm(50000,exelpredmean,sqrt(sigma2)[,1])
exelpostmean <- mean(exelpostpred)
exelprobwin <- sum(exelpostpred > 0)/50000

quartz()
par(mfrow=c(3,4))
plot(density(hornpostpred),main="JEFF_HORNACEK",xlab="Difference_in_Points",ylab="")
plot(density(malonepostpred),main="KARL_MALONE",xlab="Difference_in_Points",ylab="")
plot(density(tagpostpred),main="GREG_OSTERTAG",xlab="Difference_in_Points",ylab="")
plot(density(russpostpred),main="BRYON_RUSSELL",xlab="Difference_in_Points",ylab="")
plot(density(stockpostpred),main="JOHN_STOCKTON",xlab="Difference_in_Points",ylab="")
plot(density(campbellpostpred),main="ELDEN_CAMPBELL",xlab="Difference_in_Points",ylab="")
plot(density(ejonespostpred),main="EDDIE_JONES",xlab="Difference_in_Points",ylab="")
plot(density(onealpostpred),main="SAQ_ONEAL",xlab="Difference_in_Points",ylab="")
plot(density(scottpostpred),main="BYRON_SCOTT",xlab="Difference_in_Points",ylab="")
plot(density(exelpostpred),main="NICK_VAN_EXEL",xlab="Difference_in_Points",ylab="")

means124 <- c(hornpostmean,malonepostmean,tagpostmean,russpostmean,stockpostmean,
      campbellpostmean,ejonespostmean,onealpostmean,scottpostmean,exelpostmean)


######GAME 241 Posterior predictive
######
## CHICAGO BULLS
## JORDAN
jordanpredmean <- tau[,4] + phi[,14] + cbind(b1[,55],b2[,55],b3[,55],b5[,55],b6[,55],b7[,55],
      b8[,55],b9[,55],b10[,55])%*%t(nba3[nba3$game==241 & nba3$Player=="JORDAN",junk2])
jordanpostpred <- rnorm(50000,jordanpredmean,sqrt(sigma2)[,1])
jordanpostmean <- mean(jordanpostpred)
jordanprobwin <- sum(jordanpostpred > 0)/50000

## PIPPEN
pippredmean <- tau[,4] + phi[,14] + cbind(b1[,93],b2[,93],b3[,93],b5[,93],b6[,93],b7[,93],b8
      [,93],b9[,93],b10[,93])%*%t(nba3[nba3$game==241 & nba3$Player=="PIPPEN",junk2])
pippostpred <- rnorm(50000,pippredmean,sqrt(sigma2)[,1])
pippostmean <- mean(pippostpred)
pipprobwin <- sum(pippostpred > 0)/50000

## RODMAN
```

```
rodpredmean <- tau[,4] + phi[,14] + cbind(b1[,101],b2[,101],b3[,101],b5[,101],b6[,101],b7
      [,101],b8[,101],b9[,101],b10[,101])%*%t(nba3[nba3$game==241 & nba3$Player=="RODMAN",junk2
      ])
rodpostpred <- rnorm(50000,rodpredmean,sqrt(sigma2)[,1])
rodpostmean <- mean(rodpostpred)
rodprobwin <- sum(rodpostpred > 0)/50000

## R_HARPER
rharperpredmean <- tau[,4] + phi[,14] + cbind(b1[,106],b2[,106],b3[,106],b5[,106],b6[,106],b7
      [,106],b8[,106],b9[,106],b10[,106])%*%t(nba3[nba3$game==241 & nba3$Player=="R_HARPER",
      junk2])
rharperpostpred <- rnorm(50000,rharperpredmean,sqrt(sigma2)[,1])
rharperpostmean <- mean(rharperpostpred)
rharperprobwin <- sum(rharperpostpred > 0)/50000


## MIAMI
## MOURNING
mournpredmean <- tau[,14] + phi[,4] + cbind(b1[,79],b2[,79],b3[,79],b5[,79],b6[,79],b7[,79],b8
      [,79],b9[,79],b10[,79])%*%t(nba3[nba3$game==241 & nba3$Player=="MOURNING",junk2])
mournpostpred <- rnorm(50000,mournpredmean,sqrt(sigma2)[,1])
mournpostmean <- mean(mournpostpred)
mournprobwin <- sum(mournpostpred > 0)/50000

## P_BROWN
pbrownpredmean <- tau[,14] + phi[,4] + cbind(b1[,96],b2[,96],b3[,96],b5[,96],b6[,96],b7[,96],
      b8[,96],b9[,96],b10[,96])%*%t(nba3[nba3$game==241 & nba3$Player=="P_BROWN",junk2])
pbrownpostpred <- rnorm(50000,pbrownpredmean,sqrt(sigma2)[,1])
pbrownpostmean <- mean(pbrownpostpred)
pbrownprobwin <- sum(pbrownpostpred > 0)/50000

## T_HARDAWAY
thardawaypredmean <- tau[,14] + phi[,4] + cbind(b1[,125],b2[,125],b3[,125],b5[,125],b6[,125],
      b7[,125],b8[,125],b9[,125],b10[,125])%*%t(nba3[nba3$game==241 & nba3$Player=="T_HARDAWAY",
      junk2])
thardawaypostpred <- rnorm(50000,thardawaypredmean,sqrt(sigma2)[,1])
thardawaypostmean <- mean(thardawaypostpred)
thardawayprobwin <- sum(thardawaypostpred > 0)/50000

quartz()
par(mfrow=c(2,4))
plot(density(jordanpostpred),main="MICHEAL_JORDAN",xlab="Difference_in_Points",ylab="")
plot(density(pippostpred),main="SCOTTIE_PIPPEN",xlab="Difference_in_Points",ylab="")
plot(density(rodpostpred),main="DENNIS_RODMAN",xlab="Difference_in_Points",ylab="")
plot(density(rharperpostpred),main="RON_HARPER",xlab="Difference_in_Points",ylab="")
plot(density(mournpostpred),main="ALONZO_MOURNING",xlab="Difference_in_Points",ylab="")
plot(density(pbrownpostpred),main="PJ_BROWN",xlab="Difference_in_Points",ylab="")
plot(density(thardawaypostpred),main="TIM_HARDAWAY",xlab="Difference_in_Points",ylab="")

means241 <- c(jordanpostmean,pippostmean,rodpostmean,rharperpostmean,mournpostmean,
      pbrownpostmean,thardawaypostmean)

######GAME 434 Posterior predictive
######
## PHILADELPHIA 76ERS
## COLEMAN
colemanpredmean <- tau[,20] + phi[,23] + cbind(b1[,18],b2[,18],b3[,18],b5[,18],b6[,18],b7
      [,18],b8[,18],b9[,18],b10[,18])%*%t(nba3[nba3$game==434 & nba3$Player=="COLEMAN",junk2])
colemanpostpred <- rnorm(50000,colemanpredmean,sqrt(sigma2)[,1])
colemanpostmean <- mean(colemanpostpred)
colemanprobwin <- sum(colemanpostpred > 0)/50000

## IVERSON
iversonpredmean <- tau[,20] + phi[,23] + cbind(b1[,53],b2[,53],b3[,53],b5[,53],b6[,53],b7
      [,53],b8[,53],b9[,53],b10[,53])%*%t(nba3[nba3$game==434 & nba3$Player=="IVERSON",junk2])
iversonpostpred <- rnorm(50000,iversonpredmean,sqrt(sigma2)[,1])
iversonpostmean <- mean(iversonpostpred)
iversonprobwin <- sum(iversonpostpred > 0)/50000

## STACKHOUSE
stackpredmean <- tau[,20] + phi[,23] + cbind(b1[,118],b2[,118],b3[,118],b5[,118],b6[,118],b7
      [,118],b8[,118],b9[,118],b10[,118])%*%t(nba3[nba3$game==434 & nba3$Player=="STACKHOUSE",
      junk2])
stackpostpred <- rnorm(50000,stackpredmean,sqrt(sigma2)[,1])
stackpostmean <- mean(stackpostpred)
stackprobwin <- sum(stackpostpred > 0)/50000

## Scott_WILLIAMS
swilliamspredmean <- tau[,20] + phi[,23] + cbind(b1[,123],b2[,123],b3[,123],b5[,123],b6[,123],
      b7[,123],b8[,123],b9[,123],b10[,123])%*%t(nba3[nba3$game==434 & nba3$Player=="Scott_
      WILLIAMS",junk2])
swilliamspostpred <- rnorm(50000,swilliamspredmean,sqrt(sigma2)[,1])
swilliamspostmean <- mean(swilliamspostpred)
swilliamsprobwin <- sum(swilliamspostpred > 0)/50000

## WEATHERSPOON
weatherspoonpredmean <- tau[,20] + phi[,23] + cbind(b1[,128],b2[,128],b3[,128],b5[,128],b6
      [,128],b7[,128],b8[,128],b9[,128],b10[,128])%*%t(nba3[nba3$game==434 & nba3$Player=="
      WEATHERSPOON",junk2])
```

```
weatherspoonpostpred <- rnorm(50000,weatherspoonpredmean,sqrt(sigma2)[,1])
weatherspoonpostmean <- mean(weatherspoonpostpred)
weatherspoonprobwin <- sum(weatherspoonpostpred > 0)/50000


## SACRAMENTO KINGS
## M_SMITH
msmithpredmean <- tau[,23] + phi[,20] + cbind(b1[,84],b2[,84],b3[,84],b5[,84],b6[,84],b7[,84],
      b8[,84],b9[,84],b10[,84])%*%t(nba3[nba3$game==434 & nba3$Player=="M_SMITH",junk2])
msmithpostpred <- rnorm(50000,msmithpredmean,sqrt(sigma2)[,1])
msmithpostmean <- mean(msmithpostpred)
msmithprobwin <- sum(msmithpostpred > 0)/50000

## OWENS
owenspredmean <- tau[,23] + phi[,20] + cbind(b1[,90],b2[,90],b3[,90],b5[,90],b6[,90],b7[,90],
      b8[,90],b9[,90],b10[,90])%*%t(nba3[nba3$game==434 & nba3$Player=="OWENS",junk2])
owenspostpred <- rnorm(50000,owenspredmean,sqrt(sigma2)[,1])
owenspostmean <- mean(owenspostpred)
owensprobwin <- sum(owenspostpred > 0)/50000

## POLYNICE
polypredmean <- tau[,23] + phi[,20] + cbind(b1[,94],b2[,94],b3[,94],b5[,94],b6[,94],b7[,94],b8
      [,94],b9[,94],b10[,94])%*%t(nba3[nba3$game==434 & nba3$Player=="POLYNICE",junk2])
polypostpred <- rnorm(50000,polypredmean,sqrt(sigma2)[,1])
polypostmean <- mean(polypostpred)
polyprobwin <- sum(polypostpred > 0)/50000

## RICHMOND
richpredmean <- tau[,23] + phi[,20] + cbind(b1[,99],b2[,99],b3[,99],b5[,99],b6[,99],b7[,99],b8
      [,99],b9[,99],b10[,99])%*%t(nba3[nba3$game==434 & nba3$Player=="RICHMOND",junk2])
richpostpred <- rnorm(50000,richpredmean,sqrt(sigma2)[,1])
richpostmean <- mean(richpostpred)
richprobwin <- sum(richpostpred > 0)/50000

quartz()
par(mfrow=c(3,3))
plot(density(colemanpostpred),main="DERRICK_COLEMAN",xlab="Difference_in_Points",ylab="")
plot(density(iversonpostpred),main="ALLEN_IVERSON",xlab="Difference_in_Points",ylab="")
plot(density(stackpostpred),main="JERRY_STACKHOUSE",xlab="Difference_in_Points",ylab="")
plot(density(swilliamspostpred),main="SCOTT_WILLIAMS",xlab="Difference_in_Points",ylab="")
plot(density(weatherspoonpostpred),main="CLARANCE_WEATHERSPOON",xlab="Difference_in_Points",
      ylab="")
plot(density(msmithpostpred),main="MICHEAL_SMITH",xlab="Difference_in_Points",ylab="")
plot(density(owenspostpred),main="BILLY_OWENS",xlab="Difference_in_Points",ylab="")
plot(density(polypostpred),main="OLDEN_POLYNICE",xlab="Difference_in_Points",ylab="")
plot(density(richpostpred),main="MITCH_RICHMOND",xlab="Difference_in_Points",ylab="")

means434 <- c(colemanpostmean,iversonpostmean,stackpostmean,swilliamspostmean,
      weatherspoonpostmean,msmithpostmean,owenspostmean,polypostmean,richpostmean)

######GAME 486 Posterior predictive
######
## CHARLOTTE HORNETS
## BOGUES
boguespredmean <- tau[,3] + phi[,7] + cbind(b1[,10],b2[,10],b3[,10],b5[,10],b6[,10],b7[,10],b8
      [,10],b9[,10],b10[,10])%*%t(nba3[nba3$game==486 & nba3$Player=="BOGUES",junk2])
boguespostpred <- rnorm(50000,boguespredmean,sqrt(sigma2)[,1])
boguespostmean <- mean(boguespostpred)
boguesprobwin <- sum(boguespostpred > 0)/50000

## DIVAC
divacpredmean <- tau[,3] + phi[,7] + cbind(b1[,24],b2[,24],b3[,24],b5[,24],b6[,24],b7[,24],b8
      [,24],b9[,24],b10[,24])%*%t(nba3[nba3$game==486 & nba3$Player=="DIVAC",junk2])
divacpostpred <- rnorm(50000,divacpredmean,sqrt(sigma2)[,1])
divacpostmean <- mean(divacpostpred)
divacprobwin <- sum(divacpostpred > 0)/50000

## MASON
masonpredmean <- tau[,3] + phi[,7] + cbind(b1[,73],b2[,73],b3[,73],b5[,73],b6[,73],b7[,73],b8
      [,73],b9[,73],b10[,73])%*%t(nba3[nba3$game==486 & nba3$Player=="MASON",junk2])
masonpostpred <- rnorm(50000,masonpredmean,sqrt(sigma2)[,1])
masonpostmean <- mean(masonpostpred)
masonprobwin <- sum(masonpostpred > 0)/50000

## RICE
ricepredmean <- tau[,3] + phi[,7] + cbind(b1[,98],b2[,98],b3[,98],b5[,98],b6[,98],b7[,98],b8
      [,98],b9[,98],b10[,98])%*%t(nba3[nba3$game==486 & nba3$Player=="RICE",junk2])
ricepostpred <- rnorm(50000,ricepredmean,sqrt(sigma2)[,1])
ricepostmean <- mean(ricepostpred)
riceprobwin <- sum(ricepostpred > 0)/50000


## DENVER NUGGETS
## D_ELLIS
dellispredmean <- tau[,7] + phi[,3] + cbind(b1[,29],b2[,29],b3[,29],b5[,29],b6[,29],b7[,29],b8
      [,29],b9[,29],b10[,29])%*%t(nba3[nba3$game==486 & nba3$Player=="D_ELLIS",junk2])
dellispostpred <- rnorm(50000,dellispredmean,sqrt(sigma2)[,1])
dellispostmean <- mean(dellispostpred)
dellisprobwin <- sum(dellispostpred > 0)/50000
```

```r
## JOHNSON
johnsonpredmean <- tau[,7] + phi[,3] + cbind(b1[,54],b2[,54],b3[,54],b5[,54],b6[,54],b7[,54],
      b8[,54],b9[,54],b10[,54])%*%t(nba3[nba3$game==486 & nba3$Player=="JOHNSON",junk2])
johnsonpostpred <- rnorm(50000,johnsonpredmean,sqrt(sigma2)[,1])
johnsonpostmean <- mean(johnsonpostpred)
johnsonprobwin <- sum(johnsonpostpred > 0)/50000

## L_ELLIS
lellispredmean <- tau[,7] + phi[,3] + cbind(b1[,67],b2[,67],b3[,67],b5[,67],b6[,67],b7[,67],b8
      [,67],b9[,67],b10[,67])%*%t(nba3[nba3$game==486 & nba3$Player=="L_ELLIS",junk2])
lellispostpred <- rnorm(50000,lellispredmean,sqrt(sigma2)[,1])
lellispostmean <- mean(lellispostpred)
lellisprobwin <- sum(lellispostpred > 0)/50000

## M_JACKSON
mjacksonpredmean <- tau[,7] + phi[,3] + cbind(b1[,83],b2[,83],b3[,83],b5[,83],b6[,83],b7[,83],
      b8[,83],b9[,83],b10[,83])%*%t(nba3[nba3$game==486 & nba3$Player=="M_JACKSON",junk2])
mjacksonpostpred <- rnorm(50000,mjacksonpredmean,sqrt(sigma2)[,1])
mjacksonpostmean <- mean(mjacksonpostpred)
mjacksonprobwin <- sum(mjacksonpostpred > 0)/50000

## STITH
stithpredmean <- tau[,7] + phi[,3] + cbind(b1[,119],b2[,119],b3[,119],b5[,119],b6[,119],b7
      [,119],b8[,119],b9[,119],b10[,119])%*%t(nba3[nba3$game==486 & nba3$Player=="STITH",junk2])
stithpostpred <- rnorm(50000,stithpredmean,sqrt(sigma2)[,1])
stithpostmean <- mean(stithpostpred)
stithprobwin <- sum(stithpostpred > 0)/50000

quartz()
par(mfrow=c(3,3))
plot(density(boguespostpred),main="MUGGSY_BOGUES",xlab="Difference_in_Points",ylab="")
plot(density(divacpostpred),main="VLADE_DIVAC",xlab="Difference_in_Points",ylab="")
plot(density(masonpostpred),main="ANTHONY_MASON",xlab="Difference_in_Points",ylab="")
plot(density(ricepostpred),main="GLENN_RICE",xlab="Difference_in_Points",ylab="")
plot(density(dellispostpred),main="DALE_ELLIS",xlab="Difference_in_Points",ylab="")
plot(density(johnsonpostpred),main="ERVIN_JOHNSON",xlab="Difference_in_Points",ylab="")
plot(density(lellispostpred),main="LAPHONSO_ELLIS",xlab="Difference_in_Points",ylab="")
plot(density(mjacksonpostpred),main="MARK_JACKSON",xlab="Difference_in_Points",ylab="")
plot(density(stithpostpred),main="BRYANT_STITH",xlab="Difference_in_Points",ylab="")

means486 <- c(boguespostmean,divacpostmean,masonpostmean,ricepostmean,dellispostmean,
      johnsonpostmean,lellispostmean,mjacksonpostmean,stithpostmean)



######GAME 600 Posterior predictive
######
## PORTLAND TRAILBLAZERS
## Cliff_ROBINSON
crobinson2predmean <- tau[,22] + phi[,8] + cbind(b1[,21],b2[,21],b3[,21],b5[,21],b6[,21],b7
      [,21],b8[,21],b9[,21],b10[,21])%*%t(nba3[nba3$game==600 & nba3$Player=="Cliff_ROBINSON",
      junk2])
crobinson2postpred <- rnorm(50000,crobinson2predmean,sqrt(sigma2)[,1])
crobinson2postmean <- mean(crobinson2postpred)
crobinson2probwin <- sum(crobinson2postpred > 0)/50000

## K_ANDERSON
kanderson2predmean <- tau[,22] + phi[,8] + cbind(b1[,60],b2[,60],b3[,60],b5[,60],b6[,60],b7
      [,60],b8[,60],b9[,60],b10[,60])%*%t(nba3[nba3$game==600 & nba3$Player=="K_ANDERSON",junk2
      ])
kanderson2postpred <- rnorm(50000,kanderson2predmean,sqrt(sigma2)[,1])
kanderson2postmean <- mean(kanderson2postpred)
kanderson2probwin <- sum(kanderson2postpred > 0)/50000

## RIDER
rider2predmean <- tau[,22] + phi[,8] + cbind(b1[,100],b2[,100],b3[,100],b5[,100],b6[,100],b7
      [,100],b8[,100],b9[,100],b10[,100])%*%t(nba3[nba3$game==600 & nba3$Player=="RIDER",junk2])
rider2postpred <- rnorm(50000,rider2predmean,sqrt(sigma2)[,1])
rider2postmean <- mean(rider2postpred)
rider2probwin <- sum(rider2postpred > 0)/50000

## DETROIT PISTONS
## DUMARS
dumarspredmean <- tau[,8] + phi[,22] + cbind(b1[,27],b2[,27],b3[,27],b5[,27],b6[,27],b7[,27],
      b8[,27],b9[,27],b10[,27])%*%t(nba3[nba3$game==600 & nba3$Player=="DUMARS",junk2])
dumarspostpred <- rnorm(50000,dumarspredmean,sqrt(sigma2)[,1])
dumarspostmean <- mean(dumarspostpred)
dumarsprobwin <- sum(dumarspostpred > 0)/50000

## G_HILL
ghillpredmean <- tau[,8] + phi[,22] + cbind(b1[,42],b2[,42],b3[,42],b5[,42],b6[,42],b7[,42],b8
      [,42],b9[,42],b10[,42])%*%t(nba3[nba3$game==600 & nba3$Player=="G_HILL",junk2])
ghillpostpred <- rnorm(50000,ghillpredmean,sqrt(sigma2)[,1])
ghillpostmean <- mean(ghillpostpred)
ghillprobwin <- sum(ghillpostpred > 0)/50000

## HUNTER
```

```
hunterpredmean <- tau[,8] + phi[,22] + cbind(b1[,50],b2[,50],b3[,50],b5[,50],b6[,50],b7[,50],
      b8[,50],b9[,50],b10[,50])%*%t(nba3[nba3$game==600 & nba3$Player=="HUNTER",junk2])
hunterpostpred <- rnorm(50000,hunterpredmean,sqrt(sigma2)[,1])
hunterpostmean <- mean(hunterpostpred)
hunterprobwin <- sum(hunterpostpred > 0)/50000

## THORPE
thorpepredmean <- tau[,8] + phi[,22] + cbind(b1[,124],b2[,124],b3[,124],b5[,124],b6[,124],b7
      [,124],b8[,124],b9[,124],b10[,124])%*%t(nba3[nba3$game==600 & nba3$Player=="THORPE",junk2
      ])
thorpepostpred <- rnorm(50000,thorpepredmean,sqrt(sigma2)[,1])
thorpepostmean <- mean(thorpepostpred)
thorpeprobwin <- sum(thorpepostpred > 0)/50000


quartz()
par(mfrow=c(2,4))
plot(density(crobinson2postpred),main="CLIFF_ROBINSON",xlab="Difference_in_Points",ylab="")
abline(v=crobinson2postmean)
plot(density(kanderson2postpred),main="KENNY_ANDERSON",xlab="Difference_in_Points",ylab="")
abline(v=kanderson2postmean)
plot(density(rider2postpred),main="ISAIAH_RIDER",xlab="Difference_in_Points",ylab="")
abline(v=rider2postmean)
plot(density(dumarspostpred),main="JOE_DUMARS",xlab="Difference_in_Points",ylab="")
abline(v=dumarspostmean)
plot(density(ghillpostpred),main="GRANT_HILL",xlab="Difference_in_Points",ylab="")
abline(v=ghillpostmean)
plot(density(hunterpostpred),main="LINDSAY_HUNTER",xlab="Difference_in_Points",ylab="")
abline(v=hunterpostmean)
plot(density(thorpepostpred),main="OTIS_THORPE",xlab="Difference_in_Points",ylab="")
abline(v=thorpepostmean)

means600 <- c(crobinson2postmean,kanderson2postmean,rider2postmean,dumarspostmean,
      ghillpostmean,hunterpostmean,thorpepostmean)


######GAME 772 Posterior predictive
######
## SACRAMENTO
## ABDUL_RAUF
raufpredmean <- tau[,23] + phi[,17] + cbind(b1[,1],b2[,1],b3[,1],b5[,1],b6[,1],b7[,1],b8[,1],
      b9[,1],b10[,1])%*%t(nba3[nba3$game==772 & nba3$Player=="ABDUL_RAUF",junk2])
raufpostpred <- rnorm(50000,raufpredmean,sqrt(sigma2)[,1])
raufpostmean <- mean(raufpostpred)
raufprobwin <- sum(raufpostpred > 0)/50000

## M_SMITH
msmith2predmean <- tau[,23] + phi[,17] + cbind(b1[,84],b2[,84],b3[,84],b5[,84],b6[,84],b7
      [,84],b8[,84],b9[,84],b10[,84])%*%t(nba3[nba3$game==772 & nba3$Player=="M_SMITH",junk2])
msmith2postpred <- rnorm(50000,msmith2predmean,sqrt(sigma2)[,1])
msmith2postmean <- mean(msmith2postpred)
msmith2probwin <- sum(msmith2postpred > 0)/50000

## POLYNICE
poly2predmean <- tau[,23] + phi[,17] + cbind(b1[,94],b2[,94],b3[,94],b5[,94],b6[,94],b7[,94],
      b8[,94],b9[,94],b10[,94])%*%t(nba3[nba3$game==772 & nba3$Player=="POLYNICE",junk2])
poly2postpred <- rnorm(50000,poly2predmean,sqrt(sigma2)[,1])
poly2postmean <- mean(poly2postpred)
poly2probwin <- sum(poly2postpred > 0)/50000

## RICHMOND
rich2predmean <- tau[,23] + phi[,17] + cbind(b1[,99],b2[,99],b3[,99],b5[,99],b6[,99],b7[,99],
      b8[,99],b9[,99],b10[,99])%*%t(nba3[nba3$game==772 & nba3$Player=="RICHMOND",junk2])
rich2postpred <- rnorm(50000,rich2predmean,sqrt(sigma2)[,1])
rich2postmean <- mean(rich2postpred)
rich2probwin <- sum(rich2postpred > 0)/50000

######
## NEW JERSEY
## GILL
gillpredmean <- tau[,17] + phi[,23] + cbind(b1[,39],b2[,39],b3[,39],b5[,39],b6[,39],b7[,39],b8
      [,39],b9[,39],b10[,39])%*%t(nba3[nba3$game==772 & nba3$Player=="GILL",junk2])
gillpostpred <- rnorm(50000,gillpredmean,sqrt(sigma2)[,1])
gillpostmean <- mean(gillpostpred)
gillprobwin <- sum(gillpostpred > 0)/50000

## J_JACKSON
jjacksonpredmean <- tau[,17] + phi[,23] + cbind(b1[,56],b2[,56],b3[,56],b5[,56],b6[,56],b7
      [,56],b8[,56],b9[,56],b10[,56])%*%t(nba3[nba3$game==772 & nba3$Player=="J_JACKSON",junk2])
jjacksonpostpred <- rnorm(50000,jjacksonpredmean,sqrt(sigma2)[,1])
jjacksonpostmean <- mean(jjacksonpostpred)
jjacksonprobwin <- sum(jjacksonpostpred > 0)/50000

## MONTROSS
montrosspredmean <- tau[,17] + phi[,23] + cbind(b1[,78],b2[,78],b3[,78],b5[,78],b6[,78],b7
      [,78],b8[,78],b9[,78],b10[,78])%*%t(nba3[nba3$game==772 & nba3$Player=="MONTROSS",junk2])
montrosspostpred <- rnorm(50000,montrosspredmean,sqrt(sigma2)[,1])
montrosspostmean <- mean(montrosspostpred)
montrossprobwin <- sum(montrosspostpred > 0)/50000
```

```
quartz()
par(mfrow=c(2,4))
plot(density(gillpostpred),main="KENDALL_GILL",xlab="Difference_in_Points",ylab="")
abline(v=gillpostmean)
plot(density(montrosspostpred),main="ERIC_MONTROSS",xlab="Difference_in_Points",ylab="")
abline(v=montrosspostmean)
plot(density(jjacksonpostpred),main="JIM_JACKSON",xlab="Difference_in_Points",ylab="")
abline(v=jjacksonpostmean)
plot(density(msmith2postpred),main="MICHEAL_SMITH",xlab="Difference_in_Points",ylab="")
abline(v=msmith2postmean)
plot(density(raufpostpred),main="MAHMAUD_ABDUL_RAUF",xlab="Difference_in_Points",ylab="")
abline(v=raufpostmean)
plot(density(poly2postpred),main="OLDEN_POLYNICE",xlab="Difference_in_Points",ylab="")
abline(v=poly2postmean)
plot(density(rich2postpred),main="MITCH_RICHMOND",xlab="Difference_in_Points",ylab="")
abline(v=rich2postmean)

means772 <- c(gillpostmean,montrosspostmean,jjacksonpostmean,msmith2postmean,raufpostmean,
      poly2postmean,rich2postmean)


######GAME 889  Posterior  predictive
######
## MINNESOTA
## D GARRETT
garrpredmean <- tau[,16] + phi[,25] + cbind(b1[,22],b2[,22],b3[,22],b5[,22],b6[,22],b7[,22],b8
      [,22],b9[,22],b10[,22])%*%t(nba3[nba3$game==889 & nba3$Player=="D_GARRETT",junk2])
garrpostpred <- rnorm(50000,garrpredmean,sqrt(sigma2)[,1])
garrpostmean <- mean(garrpostpred)
garrprobwin <- sum(garrpostpred > 0)/50000

## Doug_WEST
dwestpredmean <- tau[,16] + phi[,25] + cbind(b1[,31],b2[,31],b3[,31],b5[,31],b6[,31],b7[,31],
      b8[,31],b9[,31],b10[,31])%*%t(nba3[nba3$game==889 & nba3$Player=="Doug_WEST",junk2])
dwestpostpred <- rnorm(50000,dwestpredmean,sqrt(sigma2)[,1])
dwestpostmean <- mean(dwestpostpred)
dwestprobwin <- sum(dwestpostpred > 0)/50000

## GUGLIOTTA
googspredmean <- tau[,16] + phi[,25] + cbind(b1[,41],b2[,41],b3[,41],b5[,41],b6[,41],b7[,41],
      b8[,41],b9[,41],b10[,41])%*%t(nba3[nba3$game==889 & nba3$Player=="GUGLIOTTA",junk2])
googspostpred <- rnorm(50000,googspredmean,sqrt(sigma2)[,1])
googspostmean <- mean(googspostpred)
googsprobwin <- sum(googspostpred > 0)/50000

## K_GARNETT
kgpredmean <- tau[,16] + phi[,25] + cbind(b1[,61],b2[,61],b3[,61],b5[,61],b6[,61],b7[,61],b8
      [,61],b9[,61],b10[,61])%*%t(nba3[nba3$game==889 & nba3$Player=="K_GARNETT",junk2])
kgpostpred <- rnorm(50000,kgpredmean,sqrt(sigma2)[,1])
kgpostmean <- mean(kgpostpred)
kgprobwin <- sum(kgpostpred > 0)/50000

## MARBURY
marburypredmean <- tau[,16] + phi[,25] + cbind(b1[,71],b2[,71],b3[,71],b5[,71],b6[,71],b7
      [,71],b8[,71],b9[,71],b10[,71])%*%t(nba3[nba3$game==889 & nba3$Player=="MARBURY",junk2])
marburypostpred <- rnorm(50000,marburypredmean,sqrt(sigma2)[,1])
marburypostmean <- mean(marburypostpred)
marburyprobwin <- sum(marburypostpred > 0)/50000

######
## SEATTLE
## G_PAYTON
paytonpredmean <- tau[,25] + phi[,16] + cbind(b1[,43],b2[,43],b3[,43],b5[,43],b6[,43],b7[,43],
      b8[,43],b9[,43],b10[,43])%*%t(nba3[nba3$game==889 & nba3$Player=="G_PAYTON",junk2])
paytonpostpred <- rnorm(50000,paytonpredmean,sqrt(sigma2)[,1])
paytonpostmean <- mean(paytonpostpred)
paytonprobwin <- sum(paytonpostpred > 0)/50000


## HAWKINS
hawkinspredmean <- tau[,25] + phi[,16] + cbind(b1[,45],b2[,45],b3[,45],b5[,45],b6[,45],b7
      [,45],b8[,45],b9[,45],b10[,45])%*%t(nba3[nba3$game==889 & nba3$Player=="HAWKINS",junk2])
hawkinspostpred <- rnorm(50000,hawkinspredmean,sqrt(sigma2)[,1])
hawkinspostmean <- mean(hawkinspostpred)
hawkinsprobwin <- sum(hawkinspostpred > 0)/50000

## KEMP
kemppredmean <- tau[,25] + phi[,16] + cbind(b1[,58],b2[,58],b3[,58],b5[,58],b6[,58],b7[,58],b8
      [,58],b9[,58],b10[,58])%*%t(nba3[nba3$game==889 & nba3$Player=="KEMP",junk2])
kemppostpred <- rnorm(50000,kemppredmean,sqrt(sigma2)[,1])
kemppostmean <- mean(kemppostpred)
kempprobwin <- sum(kemppostpred > 0)/50000

## MCILVAINE
mcvnepredmean <- tau[,25] + phi[,16] + cbind(b1[,76],b2[,76],b3[,76],b5[,76],b6[,76],b7[,76],
      b8[,76],b9[,76],b10[,76])%*%t(nba3[nba3$game==889 & nba3$Player=="MCILVAINE",junk2])
mcvnepostpred <- rnorm(50000,mcvnepredmean,sqrt(sigma2)[,1])
```

```
mcvnepostmean <- mean(mcvnepostpred)
mcvneprobwin <- sum(mcvnepostpred > 0)/50000

quartz()
par(mfrow=c(3,3))
plot(densitygarrpostpred),main="DEAN_GARRETT",xlab="Difference_in_Points",ylab="")
abline(v=garrpostmean)
plot(density(dwestpostpred),main="DOUG_WEST",xlab="Difference_in_Points",ylab="")
abline(v=dwestpostmean)
plot(density(googspostpred),main="TOM_GUGLIOTTA",xlab="Difference_in_Points",ylab="")
abline(v=googspostmean)
plot(density(kgpostpred),main="KEVIN_GARNETT",xlab="Difference_in_Points",ylab="")
abline(v=kgpostmean)
plot(density(marburypostpred),main="STEPHON_MARBURY",xlab="Difference_in_Points",ylab="")
abline(v=marburypostmean)
plot(density(paytonpostpred),main="GARY_PAYTON",xlab="Difference_in_Points",ylab="")
abline(v=paytonpostmean)
plot(density(hawkinspostpred),main="HERSEY_HAWKINS",xlab="Difference_in_Points",ylab="")
abline(v=hawkinspostmean)
plot(density(kemppostpred),main="SHAUN_KEMP",xlab="Difference_in_Points",ylab="")
abline(v=kemppostmean)
plot(density(mcvnepostpred),main="JIM_MCILVAINE",xlab="Difference_in_Points",ylab="")
abline(v=mcvnepostmean)

means889 <- c(garrpostmean,dwestpostmean,googspostmean,kgpostmean,marburypostmean,
     paytonpostmean,hawkinspostmean,kemppostmean,mcvnepostmean)


#### GAME 954
### GOLDEN STATE
## Mark Price
price2predmean <- tau[,9] + phi[,3] + cbind(b1[,95],b2[,95],b3[,95],b5[,95],b6[,95],b7[,95],b8
     [,95],b9[,95],b10[,95])%*%t(nba3[nba3$game==954 & nba3$Player=="PRICE",junk2])
price2postpred <- rnorm(50000,price2predmean,sqrt(sigma2)[,1])
price2postmean <- mean(price2postpred)
price2probwin <- sum(price2postpred > 0)/50000

## Joe Smith
jsmith2predmean <- tau[,9] + phi[,3] + cbind(b1[,57],b2[,57],b3[,57],b5[,57],b6[,57],b7[,57],
     b8[,57],b9[,57],b10[,57])%*%t(nba3[nba3$game==954 & nba3$Player=="J_SMITH",junk2])
jsmith2postpred <- rnorm(50000,jsmith2predmean,sqrt(sigma2)[,1])
jsmith2postmean <- mean(jsmith2postpred)
jsmith2probwin <- sum(jsmith2postpred > 0)/50000

## Felton Spencer
spencerpredmean <- tau[,9] + phi[,3] + cbind(b1[,116],b2[,116],b3[,116],b5[,116],b6[,116],b7
     [,116],b8[,116],b9[,116],b10[,116])%*%t(nba3[nba3$game==954 & nba3$Player=="SPENCER",junk2
     ])
spencerpostpred <- rnorm(50000,spencerpredmean,sqrt(sigma2)[,1])
spencerpostmean <- mean(spencerpostpred)
spencerprobwin <- sum(spencerpostpred > 0)/50000

## Latrell Sprewell
sprewell2predmean <- tau[,9] + phi[,3] + cbind(b1[,117],b2[,117],b3[,117],b5[,117],b6[,117],b7
     [,117],b8[,117],b9[,117],b10[,117])%*%t(nba3[nba3$game==954 & nba3$Player=="SPREWELL",
     junk2])
sprewell2postpred <- rnorm(50000,sprewell2predmean,sqrt(sigma2)[,1])
sprewell2postmean <- mean(sprewell2postpred)
sprewell2probwin <- sum(sprewell2postpred > 0)/50000

## CHARLOTTE

## BOGUES
bogues2predmean <- tau[,3] + phi[,9] + cbind(b1[,10],b2[,10],b3[,10],b5[,10],b6[,10],b7[,10],
     b8[,10],b9[,10],b10[,10])%*%t(nba3[nba3$game==954 & nba3$Player=="BOGUES",junk2])
bogues2postpred <- rnorm(50000,bogues2predmean,sqrt(sigma2)[,1])
bogues2postmean <- mean(bogues2postpred)
bogues2probwin <- sum(bogues2postpred > 0)/50000

## DIVAC
divac2predmean <- tau[,3] + phi[,9] + cbind(b1[,24],b2[,24],b3[,24],b5[,24],b6[,24],b7[,24],b8
     [,24],b9[,24],b10[,24])%*%t(nba3[nba3$game==954 & nba3$Player=="DIVAC",junk2])
divac2postpred <- rnorm(50000,divac2predmean,sqrt(sigma2)[,1])
divac2postmean <- mean(divac2postpred)
divac2probwin <- sum(divac2postpred > 0)/50000

## RICE
rice2predmean <- tau[,3] + phi[,9] + cbind(b1[,98],b2[,98],b3[,98],b5[,98],b6[,98],b7[,98],b8
     [,98],b9[,98],b10[,98])%*%t(nba3[nba3$game==954 & nba3$Player=="RICE",junk2])
rice2postpred <- rnorm(50000,rice2predmean,sqrt(sigma2)[,1])
rice2postmean <- mean(rice2postpred)
rice2probwin <- sum(rice2postpred > 0)/50000

quartz()
par(mfrow=c(2,4))
plot(density(price2postpred),main="MARK_PRICE",xlab="Difference_in_Points",ylab="")
abline(v=price2postmean)
plot(density(jsmith2postpred),main="JOE_SMITH",xlab="Difference_in_Points",ylab="")
abline(v=jsmith2postmean)
```

```
plot(density(spencerpostpred),main="FELTON_SPENCER",xlab="Difference_in_Points",ylab="")
abline(v=spencerpostmean)
plot(density(sprewell2postpred),main="LATRELL_SPREWELL",xlab="Difference_in_Points",ylab="")
abline(v=sprewell2postmean)
plot(density(bogues2postpred),main="MUGGSY_BOGUES",xlab="Difference_in_Points",ylab="")
abline(v=bogues2postmean)
plot(density(divac2postpred),main="VLADE_DIVAC",xlab="Difference_in_Points",ylab="")
abline(v=divac2postmean)
plot(density(rice2postpred),main="GLENN_RICE",xlab="Difference_in_Points",ylab="")
abline(v=rice2postmean)

means954 <- c(price2postmean,jsmith2postmean,spencerpostmean,sprewell2postmean,bogues2postmean
    ,divac2postmean,rice2postmean)


#### GAME 1116
### NEW YORK
## CHILDS
childspredmean <- tau[,18] + phi[,11] + cbind(b1[,16],b2[,16],b3[,16],b5[,16],b6[,16],b7[,16],
    b8[,16],b9[,16],b10[,16])%*%t(nba3[nba3$game==1116 & nba3$Player=="CHILDS",junk2])
childspostpred <- rnorm(50000,childspredmean,sqrt(sigma2)[,1])
childspostmean <- mean(childspostpred)
childsprobwin <- sum(childspostpred > 0)/50000

## EWING
ewingpredmean <- tau[,18] + phi[,11] + cbind(b1[,33],b2[,33],b3[,33],b5[,33],b6[,33],b7[,33],
    b8[,33],b9[,33],b10[,33])%*%t(nba3[nba3$game==1116 & nba3$Player=="EWING",junk2])
ewingpostpred <- rnorm(50000,ewingpredmean,sqrt(sigma2)[,1])
ewingpostmean <- mean(ewingpostpred)
ewingprobwin <- sum(ewingpostpred > 0)/50000

## HOUSTON
houstonpredmean <- tau[,18] + phi[,11] + cbind(b1[,48],b2[,48],b3[,48],b5[,48],b6[,48],b7
    [,48],b8[,48],b9[,48],b10[,48])%*%t(nba3[nba3$game==1116 & nba3$Player=="HOUSTON",junk2])
houstonpostpred <- rnorm(50000,houstonpredmean,sqrt(sigma2)[,1])
houstonpostmean <- mean(houstonpostpred)
houstonprobwin <- sum(houstonpostpred > 0)/50000

## L_JOHNSON
ljpredmean <- tau[,18] + phi[,11] + cbind(b1[,68],b2[,68],b3[,68],b5[,68],b6[,68],b7[,68],b8
    [,68],b9[,68],b10[,68])%*%t(nba3[nba3$game==1116 & nba3$Player=="L_JOHNSON",junk2])
ljpostpred <- rnorm(50000,ljpredmean,sqrt(sigma2)[,1])
ljpostmean <- mean(ljpostpred)
ljprobwin <- sum(ljpostpred > 0)/50000

## OAKLEY
oakpredmean <- tau[,18] + phi[,11] + cbind(b1[,86],b2[,86],b3[,86],b5[,86],b6[,86],b7[,86],b8
    [,86],b9[,86],b10[,86])%*%t(nba3[nba3$game==1116 & nba3$Player=="OAKLEY",junk2])
oakpostpred <- rnorm(50000,oakpredmean,sqrt(sigma2)[,1])
oakpostmean <- mean(oakpostpred)
oakprobwin <- sum(oakpostpred > 0)/50000


###INDIANA
## D_DAVIS
ddavispredmean <- tau[,11] + phi[,18] + cbind(b1[,28],b2[,28],b3[,28],b5[,28],b6[,28],b7[,28],
    b8[,28],b9[,28],b10[,28])%*%t(nba3[nba3$game==1116 & nba3$Player=="D_DAVIS",junk2])
ddavispostpred <- rnorm(50000,ddavispredmean,sqrt(sigma2)[,1])
ddavispostmean <- mean(ddavispostpred)
ddavisprobwin <- sum(ddavispostpred > 0)/50000

## M_JACKSON
mjackson2predmean <- tau[,11] + phi[,18] + cbind(b1[,83],b2[,83],b3[,83],b5[,83],b6[,83],b7
    [,83],b8[,83],b9[,83],b10[,83])%*%t(nba3[nba3$game==1116 & nba3$Player=="M_JACKSON",junk2
    ])
mjackson2postpred <- rnorm(50000,mjackson2predmean,sqrt(sigma2)[,1])
mjackson2postmean <- mean(mjackson2postpred)
mjackson2probwin <- sum(mjackson2postpred > 0)/50000

## R_MILLER
rmillerpredmean <- tau[,11] + phi[,18] + cbind(b1[,107],b2[,107],b3[,107],b5[,107],b6[,107],b7
    [,107],b8[,107],b9[,107],b10[,107])%*%t(nba3[nba3$game==1116 & nba3$Player=="R_MILLER",
    junk2])
rmillerpostpred <- rnorm(50000,rmillerpredmean,sqrt(sigma2)[,1])
rmillerpostmean <- mean(rmillerpostpred)
rmillerprobwin <- sum(rmillerpostpred > 0)/50000

## SMITS
smitspredmean <- tau[,11] + phi[,18] + cbind(b1[,115],b2[,115],b3[,115],b5[,115],b6[,115],b7
    [,115],b8[,115],b9[,115],b10[,115])%*%t(nba3[nba3$game==1116 & nba3$Player=="SMITS",junk2
    ])
smitspostpred <- rnorm(50000,smitspredmean,sqrt(sigma2)[,1])
smitspostmean <- mean(smitspostpred)
smitsprobwin <- sum(smitspostpred > 0)/50000


quartz()
par(mfrow=c(3,3))
plot(density(childspostpred),main="CHRIS_CHILDS",xlab="Difference_in_Points",ylab="")
```

```
abline(v=childspostmean)
plot(density(ewingpostpred),main="PATRICK_EWING",xlab="Difference_in_Points",ylab="")
abline(v=ewingpostmean)
plot(density(houstonpostpred),main="ALLEN_HOUSTON",xlab="Difference_in_Points",ylab="")
abline(v=houstonpostmean)
plot(density(ljpostpred),main="LARRY_JOHNSON",xlab="Difference_in_Points",ylab="")
abline(v=ljpostmean)
plot(density(oakleypostpred),main="CHARLES_OAKLEY",xlab="Difference_in_Points",ylab="")
abline(v=oakleypostmean)
plot(density(ddavispostpred),main="DALE_DAVIS",xlab="Difference_in_Points",ylab="")
abline(v=ddavispostmean)
plot(density(mjackson2postpred),main="MARK_JACKSON",xlab="Difference_in_Points",ylab="")
abline(v=mjackson2postmean)
plot(density(rmillerpostpred),main="REGGIE_MILLER",xlab="Difference_in_Points",ylab="")
abline(v=rmillerpostmean)
plot(density(smitspostpred),main="RIK_SMITS",xlab="Difference_in_Points",ylab="")
abline(v=smitspostmean)

means1116 <- c(childspostmean,ewingpostmean,houstonpostmean,ljpostmean,oakpostmean,
    ddavispostmean,mjackson2postmean,rmillerpostmean,smitspostmean)


game19 <- cbind(pricepostpred,jsmithpostpred,mullinpostpred,sprewellpostpred,crobinsonpostpred
    ,kandersonpostpred,riderpostpred,rwallacepostpred,sabonispostpred)
game124 <- cbind(hornpostpred,malonepostpred,tagpostpred,russpostpred,stockpostpred,
    campbellpostpred,ejonespostpred,onealpostpred,scottpostpred,exelpostpred)
game241 <- cbind(jordanpostpred,pippostpred,rodpostpred,rharperpostpred,mournpostpred,
    pbrownpostpred,thardawaypostpred)
game434 <- cbind(colemanpostpred,iversonpostpred,stackpostpred,swilliamspostpred,
    weatherspoonpostpred,msmithpostpred,owenspostpred,polypostpred,richpostpred)
game486 <- cbind(boguespostpred,divacpostpred,masonpostpred,ricepostpred,dellispostpred,
    johnsonpostpred,lellispostpred,mjacksonpostpred,stithpostpred)
game600 <- cbind(crobinson2postpred,kanderson2postpred,rider2postpred,dumarspostpred,
    ghillpostpred,hunterpostpred,thorpepostpred)
game772 <- cbind(gillpostpred,montrosspostpred,jjacksonpostpred,msmith2postpred,raufpostpred,
    poly2postpred,rich2postpred)
game889 <- cbind(garrpostpred,dwestpostpred,googspostpred,kgpostpred,marburypostpred,
    paytonpostpred,hawkinspostpred,kemppostpred,mcvnepostpred)
game954 <- cbind(price2postpred,jsmith2postpred,spencerpostpred,sprewell2postpred,
    bogues2postpred,divac2postpred,rice2postpred)
game1116 <- cbind(childspostpred,ewingpostpred,houstonpostpred,ljpostpred,oakpostpred,
    ddavispostpred,mjackson2postpred,rmillerpostpred,smitspostpred)

write.table(game19,"/Users/garrittpage/mastersproject/postpred/game19")
write.table(game124,"/Users/garrittpage/mastersproject/postpred/game124")
write.table(game241,"/Users/garrittpage/mastersproject/postpred/game241")
write.table(game434,"/Users/garrittpage/mastersproject/postpred/game434")
write.table(game486,"/Users/garrittpage/mastersproject/postpred/game486")
write.table(game600,"/Users/garrittpage/mastersproject/postpred/game600")
write.table(game772,"/Users/garrittpage/mastersproject/postpred/game772")
write.table(game889,"/Users/garrittpage/mastersproject/postpred/game889")
write.table(game954,"/Users/garrittpage/mastersproject/postpred/game954")
write.table(game1116,"/Users/garrittpage/mastersproject/postpred/game1116")
```

A.3 SAS code

Listing A.3: **SAS code that cleaned and formatted data**

```
** macro that gets rid of repetition in data;

%macro redata(fulldata, reduceddata);
data nba;
infile "F:/Masters Project/full data/&fulldata" dlm=',' firstobs=2 missover dsd;
informat date mmddyy10.;
format date mmddyy10.;
input team :$20. vsteam :$20. hometeam :$20. date Player :$20.
      Position$ minutes fgm fga ftm fta orb drb trb assist fouls
      steals turnovers points;
run;

Proc sort data=nba out=nbagood NODUPKEY;
 by _ALL_;
run;

proc sort data=nbagood;
by date hometeam;
run;

PROC EXPORT DATA= WORK.NBAGOOD
            OUTFILE= "F:masters project\reduced data\&reduceddata"
            DBMS=CSV REPLACE;
RUN;
```

```
%Mend redata;

%redata(players1996_1997.csv, rplayers1996_1997.csv);
%redata(players1997_1998.csv, rplayers1997_1998.csv);
%redata(players1998_1999.csv, rplayers1998_1999.csv);
%redata(players1999_2000.csv, rplayers1999_2000.csv);
%redata(players2000_2001.csv, rplayers2000_2001.csv);
%redata(players2001_2002.csv, rplayers2001_2002.csv);
%redata(players2002_2003.csv, rplayers2002_2003.csv);
%redata(players2003_2004.csv, rplayers2003_2004.csv);

%redata(team1996_1997.csv, rteam1996_1997.csv);
%redata(team1997_1998.csv, rteam1997_1998.csv);
%redata(team1998_1999.csv, rteam1998_1999.csv);
%redata(team1999_2000.csv, rteam1999_2000.csv);
%redata(team2000_2001.csv, rteam2000_2001.csv);
%redata(team2001_2002.csv, rteam2001_2002.csv);
%redata(team2002_2003.csv, rteam2002_2003.csv);
%redata(team2003_2004.csv, rteam2003_2004.csv);

** macro that separates players by their position and starters from the bench;

%macro datasplit;
%do j = 1996 %to 2003;
%let i=%eval(&j)_%eval(&j+1).csv;

data nba;
infile "f:/Masters Project/reduced data/rplayers&i" dlm=',' firstobs=2 missover dsd;
informat date mmddyy10.;
format date mmddyy10.;
input date team :$20. vsteam :$20. hometeam :$20.  Player :$20.
      Position$ minutes fgm fga ftm fta orb drb trb assist fouls
      steals turnovers points;
if position = "" then position = "B";
run;

Proc sort data=nba out=junksort;
 by date hometeam team vsteam;
run;
Proc means data=junksort noprint;
 var points;
 by date hometeam team vsteam;
 output out=junk sum=tpoints;
run;
data junk2;
 set junk;
 by date hometeam team vsteam;
 pointdiff=tpoints-lag(tpoints);
 if first.hometeam then pointdiff=.;
run;
Proc sort data=junk2;
 by descending date descending hometeam descending team descending vsteam;
run;
data junk3;
 set junk2;
 by descending date descending hometeam descending team descending vsteam;
 pointdiff2=tpoints-lag(tpoints);
 if last.hometeam then pointdiff=pointdiff2;
run;
Proc sort data=junk3;
 by date hometeam team vsteam;
run;
data nba;
 merge junksort junk3(drop=pointdiff2 _TYPE_);
 by date hometeam team vsteam;
 if fga ^= 0 then percent = fgm/fga;
 if fga = 0 then percent = 0;
run;

PROC EXPORT DATA= work.nba
            OUTFILE= "F:mastersproject\pointdiffdata\diffplayers&i"
            DBMS=CSV REPLACE;
RUN;


data allbench;
set nba;
if position="B";
run;

proc sort data=allbench;
by date hometeam;
run;

PROC EXPORT DATA= work.allbench
            OUTFILE= "F:mastersproject\allbench\allbench&i"
            DBMS=CSV REPLACE;
RUN;
```

```
proc means data=nba noprint sum;
var minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers points;
by date team vsteam hometeam status players;
output out=bench_starters sum=minutes fgm fga ftm fta orb drb trb assist fouls steals
     turnovers points;
run;

proc sort data=bench_starters;
by date hometeam;
run;

data bench starters;
set bench_starters;
if position ^= "B" then output starters;
        else output bench;
drop _FREQ_;
run;

PROC EXPORT DATA= work.bench
            OUTFILE= "F:masters project\bench\bench&i"
            DBMS=CSV REPLACE;
RUN;

PROC EXPORT DATA= work.starters
            OUTFILE= "F:masters project\starters\starters&i"
            DBMS=CSV REPLACE;
RUN;


data guards forwards centers;
set nba;
if position = "G" then output guards;
if position = "F" then output forwards;
if position = "C" then output centers;
drop _FREQ_;
run;

proc sort data=guards;
by date team vsteam hometeam position player;
run;

proc sort data=forwards;
by date team vsteam hometeam position player;
run;

proc sort data=centers;
by date team vsteam hometeam position player;
run;

proc means data=guards noprint sum;
var minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers points tpoints pointdiff
      percent;
by date team vsteam hometeam position player;
output out=guards sum=minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers points
      tpoints pointdiff percent;
run;

proc means data=forwards noprint sum;
var minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers points tpoints pointdiff
      percent;
by date team vsteam hometeam position player;
output out=forwards sum=minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers
     points tpoints pointdiff percent;
run;

proc means data=centers noprint sum;
var minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers points tpoints pointdiff
      percent;
by date team vsteam hometeam position player;
output out=centers sum=minutes fgm fga ftm fta orb drb trb assist fouls steals turnovers
     points tpoints pointdiff percent;
run;

proc sort data=guards;
by date hometeam;
run;

proc sort data=forwards;
by date hometeam;
run;

proc sort data=centers;
by date hometeam;
run;

PROC EXPORT DATA= work.guards
            OUTFILE= "F:masters project\guards\guards&i"
            DBMS=CSV REPLACE;
```

```
RUN;

PROC EXPORT DATA= work.forwards
            OUTFILE= "F:masters project\forwards\forwards&i"
            DBMS=CSV REPLACE;
RUN;

PROC EXPORT DATA= work.centers
            OUTFILE= "F:masters project\centers\centers&i"
            DBMS=CSV REPLACE;
RUN;
%end;
%mend;
options mlogic symbolgen;
%datasplit;

** separate point guards from shooting guards
data guards96_97;
infile "F:/MastersProject/guards/guards1996_1997.csv" dlm="," firstobs=2;
drop _TYPE_ _FREQ_;
informat date mmddyy10.;
format date mmddyy10.;
input date team :$20. vsteam :$20. hometeam :$20. Position$ Player :$20. _TYPE_ _FREQ_ minutes
      fgm     fga ftm fta orb drb trb assist fouls steals turnovers  points tpoints pointdiff
      percent;
sfgm = fgm/minutes;
sftm = ftm/minutes;
sorb=orb/minutes;
sdrb = drb/minutes;
sassist = assist/minutes;
sfouls = fouls/minutes;
ssteals = steals/minutes;
sturnovers = turnovers/minutes;
spoints = points/minutes;
ftpercent = ftm/fta;
if ftpercent = . then ftpercent = 0;
fgpercent = percent;
if fgpercent = . then fgpercent = 0;
if player = "A HARDAWAY" then do; player="A_HARDAWAY";position="PG";end;
if player = "ABDUL-RAUF" then do; player="ABDUL_RAUF"; position="PG";end;
if player = "ABDUL_RAUF" and date=mdy(04,20,1997) then position="SG";
if player = "ALEXANDER" then do; player = "C_ALEXANDER"; position="PG"; end;
if player = "ALLEN" and team = "MILWAUKEE" then do; player= "R_ALLEN";  position="SG"; end;
if player = "ALLEN" and team = "INDIANA" then do; player="J_ALLEN"; position="SG"; end;
if player = "J_ALLEN" and date = mdy(02,16,1997) then position="PG";
if player = "ANDERSON" and team = "PORTLAND" then do; player="K_ANDERSON"; position="PG";end;
if player = "ANDERSON" and team = "ORLANDO" then do; player="N_ANDERSON"; position="SG";end;
if player = "ANDERSON" and team = "MIAMI" then do; player="W_ANDERSON"; position="PG";end;
if player = "ANTHONY" then position="PG";
if player = "ARMSTRONG" then do; player="BJ_ARMSTRONG"; position="PG";end;
if player = "ASKINS" then position="SG";
if player = "AUGMON" then position="SG";
if player = "B EDWARDS" then do; player="B_EDWARDS"; position="SG";end;
if player = "B THOMPSON" then do; player="B_THOMPSON"; position="PG";end;
if player = "BARROS" then position="PG";
if player = "BARRY" then do; player="J_BARRY"; position="SG";end;
if player = "BEST" then position="PG";
if player = "BILLUPS" then position="PG";
if player = "BLAYLOCK" then position="PG";
if player = "BOGUES" then position="PG";
if player = "BOOKER" then do; player="M_BOOKER"; position="PG";end;
if player = "BOWEN" then position="PG";
if player = "BOYCE" then position="SG";
if player = "BRANDON" then position="PG";
if player = "BROWN" and team = "BOSTON" then do; player="D_BROWN"; position="PG"; end;
if player = "BROWN" and team = "CHICAGO" then do; player="R_BROWN"; position="PG";end;
if player = "D_BROWN" and date=mdy(01,11,1997) then position="SG";
if player = "BRYANT" then position="SG";
if player = "BUTLER" then position="SG";
if player = "CARR" then position="SG";
if player = "CASSELL" then position="PG";
if player = "CHAPMAN" then position="SG";
if player = "CHAPMAN" and date=mdy(11,17,1996) then position="PG";
if player = "CHEANEY" then position="SG";
if player = "CHILDS" then position="PG";
if player = "CHRISTIE" then position="SG";
if player = "CHRISTIE" and date=mdy(11,19,1996) then position="PG";
if player = "COLES" then position="PG";
if player = "COLES" and date=mdy(01,29,1997) then position="SG";
if player = "CURRY" then position="SG";
if player = "D ELLIS" then do; player="D_ELLIS"; position="SG";end;
if player = "D MARTIN" then do; player="D_MARTIN"; position="PG"; end;
if player = "DANILOVIC" then position="SG";
if player = "DAY" then position="SG";
if player = "DEHERE" then position="SG";
if player = "DEL NEGRO" then do; player="DEL_NEGRO"; position="SG";end;
if player = "DELK" then position="PG";
if player = "DOUGLAS" then position="PG";
if player = "DREXLER" then position="SG";
```

```
if player = "DUMAS" then position="SG";
if player = "DUMARS" then position="SG";
if player = "DUMARS" and date=mdy(11,02,1996) then position="PG";
if player = "DUMARS" and mdy(11,06,1996) <= date <= mdy(11,12,1996) then position="PG";
if player = "DUMARS" and date=mdy(11,15,1996) then position="PG";
if player = "DUMARS" and date=mdy(11,20,1996) then position="PG";
if player = "EDNEY" then position="PG";
if player = "EISLEY" then position="SG";
if player = "ELIE" then position="SG";
if player = "FINLEY" then position="SG";
if player = "FISHER" then position="PG";
if player = "FOX" then position="SG";
if player = "GILL" then position="SG";
if player = "GOLDWIRE" then position="PG";
if player = "GRANT" then position="SG";
if player = "GRAY" then position="SG";
if player = "HARDAWAY" and team="MIAMI" then do; player="T_HARDAWAY"; position="PG";end;
if player = "HARDAWAY" and team="ORLANDO" then do; player="A_HARDAWAY"; position="PG";end;
if player = "HARPER" and team="DALLAS" then do; player="D_HARPER";position="PG";end;
if player = "HARPER" and team="CHICAGO" then do; player="R_HARPER";position="PG";end;
if player = "HARRIS" then position="SG";
if player = "HAWKINS" then position="SG";
if player = "HILL" then position="SG";
if player = "HORNACEK" then position="SG";
if player = "HOUSTON" then position="SG";
if player = "HUNTER" then position="PG";
if player = "HURLEY" then position="PG";
if player = "IVERSON" then position="PG";
if player = "JACKSON" and team="DENVER" then do; player="M_JACKSON";position="PG";end;
if player = "JACKSON" and team="DALLAS" then do; player="J_JACKSON";position="SG";end;
if player = "JACKSON" and team="NEW JERSEY" then do; player="J_JACKSON";position="SG";end;
if player = "JACKSON" and team="INDIANA" then do; player="M_JACKSON";position="PG";end;
if player = "JOHNSON" and team="SAN ANTONIO" then do; player="A_JOHNSON";position="PG";end;
if player = "JOHNSON" and team="PHOENIX" then do; player="K_JOHNSON";position="PG";end;
if player = "K_JOHNSON" and mdy(02,18,1997) <= date <= mdy(03,05,1997) then position="SG";
if player = "K_JOHNSON" and mdy(03,22,1997) <= date <= mdy(04,19,1997) then position="SG";
if player = "JONES" then do; player="E_JONES";position="SG";end;
if player = "JORDAN" then position="SG";
if player = "KIDD" then position="PG";
if player = "KITTLES" then position="SG";
if player = "KITTLES" and date=mdy(11,13,1996) then position="PG";
if player = "KITTLES" and date=mdy(11,15,1996) then position="PG";
if player = "KITTLES" and date=mdy(02,20,1997) then position="PG";
if player = "KITTLES" and date=mdy(02,21,1997) then position="PG";
if player = "KITTLES" and date=mdy(03,26,1997) then position="PG";
if player = "KITTLES" and mdy(04,11,1997) <= date <= mdy(04,20,1997) then position="PG";
if player = "KUKOC" then position="SG";
if player = "KUKOC" and date=mdy(12,05,1996) then position="PG";
if player = "KUKOC" and mdy(01,14,1997) <= date <= mdy(01,19,1997) then position="PG";
if player = "LENARD" then position="SG";
if player = "M WILLIAMS" then do; player="M_WILLIAMS"; position="SG";end;
if player = "MACK" then position="SG";
if player = "MAJERLE" then position="SG";
if player = "MALONEY" then position="PG";
if player = "MARBURY" then position="PG";
if player = "MAXWELL" then position="SG";
if player = "MAYBERRY" then position="PG";
if player = "MCKIE" then position="SG";
if player = "MCKIE" and date=mdy(11,29,1996) then position="PG";
if player = "MERCER" then position="SG";
if player = "MILLER" then do; player="R_MILLER"; position="SG";end;
if player = "MINOR" then do; player="G_MINOR";position="SG";end;
if player = "MITCHELL" then do;player="S_MITCHELL";position="SG";end;
if player = "MOORE" then do; player="T_MOORE";position="SG";end;
if player = "MOTEN" then position="SG";
if player = "MULLIN" then position="SG";
if player = "NASH" then position="PG";
if player = "NASH" and date = mdy(01,07,1997) then position="SG";
if player = "NEWMAN" then position="SG";
if player = "OVERTON" then position="PG";
if player = "PACK" then position="PG";
if player = "PACK" and date=mdy(04,14,1997) then position="SG";
if player = "PACK" and mdy(04,01,1997)<= date <= mdy(04,05,1997) then position="SG";
if player = "PAYTON" then do; player="G_PAYTON";position="PG";end;
if player = "PEELER" then position="SG";
if player = "PERRY" then position="PG";
if player = "PERSON" then position="SG";
if player = "PHILLS" then position="SG";
if player = "PIERCE" then do; player="R_PIERCE"; position="SG";end;
if player = "PORTER" then position="PG";
if player = "PORTER" and date=mdy(12,05,1996) then position="SG";
if player = "PORTER" and date=mdy(04,16,1997) then position="SG";
if player = "PRICE" then position="PG";
if player = "RECASNER" then position="PG";
if player = "REEVES" then do; player="K_REEVES"; position="PG"; end;
if player = "RICE" then position="SG";
if player = "RICHARDSON" then position="PG";
if player = "RICHMOND" then position="SG";
if player = "RIDER" then position="SG";
```

```
if player = "ROBINSON" and team="LA LAKERS" then do; player="R_ROBINSON"; position="SG";end;
if player = "ROBINSON" and team="MINNESOTA" then do; player="J_ROBINSON"; position="SG";end;
if player = "ROBINSON" and team="VANCOUVER" then do; player="C_ROBINSON"; position="SG";end;
if player = "ROSE" and team="CHARLOTTE" then do; player="M_ROSE";position="SG";end;
if player = "ROSE" and team="INDIANA" then do; player="J_ROSE";position="SG";end;
if player = "J_ROSE" and date=mdy(01,21,1997) then position="PG";
if player = "J_ROSE" and date=mdy(01,28,1997) then position="PG";
if player = "J_ROSE" and date=mdy(02,06,1997) then position="PG";
if player = "SCOTT" then position="SG";
if player = "SEALY" then position="SG";
if player = "SHAW" then position="SG";
if player = "SHAW" and mdy(11,16,1996) =< date =< mdy(12,08,1996)then position="PG";
if player = "SHAW" and mdy(12,13,1996) =< date =< mdy(01,03,1997)then position="PG";
if player = "SMITH" and team="ATLANTA" then do; player="S_SMITH";position="SG";end;
if player = "SMITH" and team="CHARLOTTE" then do; player="T_SMITH";position="PG";end;
if player = "T_SMITH" and mdy(01,06,1997)<=date<=mdy(03,17,1997) then position="SG";
if player = "T_SMITH" and date=mdy(01,28,1997) then position="PG";
if player = "T_SMITH" and date=mdy(03,06,1997) then position="PG";
if player = "SMITH" and team="DENVER" then do; player="K_SMITH";position="PG";end;
if player = "SPREWELL" then position="SG";
if player = "STACKHOUSE" then position="SG";
if player = "STARKS" then position="SG";
if player = "STITH" then position="SG";
if player = "STOCKTON" then position="PG";
if player = "STOUDAMIRE" then position="PG";
if player = "STRICKLAND" and team="WASHINGTON" then do; player="R_STRICKLAND";position="PG";
    end;
if player = "STRICKLAND" and team="DALLAS" then do; player="E_STRICKLAND";position="SG";end;
if player = "SURA" then position="PG";
if player = "SURA" and mdy(11,02,1996) =< date =< mdy(11,08,1996) then position="SG";
if player = "SURA" and mdy(12,06,1996) =< date =< mdy(12,07,1996) then position="SG";
if player = "SURA" and mdy(01,28,1997) =< date =< mdy(01,29,1997) then position="SG";
if player = "SURA" and mdy(03,30,1997) =< date =< mdy(04,20,1997) then position="SG";
if player = "THOMPSON" then do; player="B_THOMPSON"; position="PG";end;
if player = "VAN EXEL" then do; player="VAN_EXEL";position="PG";end;
if player = "WALKER" then position="SG";
if player = "WALTERS" then position="PG";
if player = "WALTERS" and mdy(01,20,1997) =< date =< mdy(02,18,1997) then position="SG";
if player = "WALTERS" and date = mdy(02,15,1997) then position="PG";
if player = "WARD" then position="PG";
if player = "WATSON" then position="SG";
if player = "WESLEY" then position="PG";
if player = "WEST" then do; player = "Doug_WEST"; position="SG";end;
if player = "WHITESIDE" then position="SG";
if player = "WHITNEY" then position="PG";
if player = "WILKINS" then position="SG";
if player = "WILLIAMS" then do; player = "W_WILLIAMS"; position="SG";end;
if player = "WORKMAN" then position="PG";
run;

data pg;
set guards96_97;
if position="PG";
run;
data sg;
set guards96_97;
if position="SG";
run;

data forwards96_97;
infile "F:/Masters Project/forwards/forwards1996_1997.csv" dlm="," firstobs=2;
drop _TYPE_ _FREQ_;
informat date mmddyy10.;
format date mmddyy10.;
input date team :$20. vsteam :$20. hometeam :$20. Position$ Player :$20. _TYPE_ _FREQ_ minutes
      fgm fga ftm fta  orb drb trb assist fouls steals turnovers points tpoints pointdiff
    percent;
sfgm = fgm/minutes;
sftm = ftm/minutes;
sorb=orb/minutes;
sdrb = drb/minutes;
sassist = assist/minutes;
sfouls = fouls/minutes;
ssteals = steals/minutes;
sturnovers = turnovers/minutes;
spoints = points/minutes;
ftpercent = ftm/fta;
if ftpercent = . then ftpercent = 0;
fgpercent = percent;
if fgpercent = . then fgpercent = 0;
if player = "A DAVIS" then do; player="A_DAVIS"; position="SF";end;
if player = "A_DAVIS" and mdy(03,25,1997) =< date =< mdy(03,30,1997) then position="PF";
if player = "A_DAVIS" and date=mdy(01,11,1997) then position="PF";
if player = "A_DAVIS" and date=mdy(01,14,1997) then position="PF";
if player = "ABDUR-RAHIM" then do; player="ABDUR_RAHIM";position="SF";end;
if player = "ABDUR_RAHIM" and mdy(11,27,1996) =< date =< mdy(01,20,1997) then position="PF";
if player = "ABDUR_RAHIM" and date=mdy(03,06,1997) then position="PF";
if player = "ADDISON" then do; player="R_ADDISON"; position="SF";end;
if player = "ANDERSON" then position="SF";
```

```
if player = "ASKINS" then position="SF";
if player = "ASKINS" and date=mdy(03,21,1997) then position="PF";
if player = "ASKINS" and date=mdy(03,23,1997) then position="PF";
if player = "B EDWARDS" then do; player="B_EDWARDS"; position="SF";end;
if player = "B WILLIAMS" and team="NEW YORK" then do; player="BU_WILLIAMS";position="PF";end;
if player = "BAKER" then position="PF";
if player = "BARKLEY" then position="PF";
if player = "BLOUNT" then position="PF";
if player = "BLOUNT" and date=mdy(01,02,1997) then position="SF";
if player = "BROWN" and team = "MIAMI" then do; player="P_BROWN"; position="PF";end;
if player = "BROWN" and team = "MILWAUKEE" then do; player="C_BROWN";position="PF";end;
if player = "BRYANT" and team = "PHOENIX" then do; player="M_BRYANT";position="PF";end;
if player = "BRYANT" and team = "LA LAKERS" then do; player="K_BRYANT";position="SF";end;
if player = "BULLARD" then position="SF";
if player = "BURRELL" then position="SF";
if player = "BURTON" then position="SF";
if player = "C ROBINSON" then do; player="Cliff_ROBINSON"; position="SF"; end;
if player = "C WILLIAMSON" then do; player="C_WILLIAMSON";position="SF";end;
if player = "C_WILLIAMSON" and mdy(11,26,1996)<= date <=mdy(12,06,1996) then position="PF";
if player = "C_WILLIAMSON" and date=mdy(01,03,1997) then position="PF";
if player = "CAFFEY" then position="PF";
if player = "CAMBY" then position="PF";
if player = "CAMPBELL" then position="PF";
if player = "CEBALLOS" then position="SF";
if player = "CHAMBERS" then position="PF";
if player = "CHAPMAN" then position="SF";
if player = "CHILCUTT" then position="SF";
if player = "CHRISTIE" then position="SF";
if player = "COLEMAN" then position="PF";
if player = "CONLON" then position="PF";
if player = "CONLON" and mdy(04,04,1997) <= date <= mdy(04,18,1997) then position="SF";
if player = "CORBIN" then position="SF";
if player = "D DAVIS" then do; player="D_DAVIS"; position="PF";end;
if player = "D ELLIS" then do; player="D_ELLIS"; position="SF";end;
if player = "D SCHREMPF" then do; player="SCHREMPF";position="SF";end;
if player = "D WILKINS" then do; player="D_WILKINS";position="PF";end;
if player = "D_WILKINS" and mdy(02,14,1997) <= date <= mdy(02,28,1997) then position="SF";
if player = "D_WILKINS" and mdy(03,17,1997) <= date <= mdy(03,21,1997) then position="SF";
if player = "D_WILKINS" and mdy(04,18,1997) <= date <= mdy(04,20,1997) then position="SF";
if player = "D_WILKINS" and date=mdy(04,12,1997) then position="SF";
if player = "DANILOVIC" then position="SF";
if player = "DAVIS" then position="SF";
if player = "DAY" then position="SF";
if player = "DECLERCQ" then position="PF";
if player = "DIVAC" then position="PF";
if player = "DUDLEY" then position="PF";
if player = "ELIE" then position="SF";
if player = "ELLIOTT" then position="SF";
if player = "FERRELL" then position="SF";
if player = "FERRY" then position="PF";
if player = "FINLEY" then position="SF";
if player = "FOX" then position="SF";
if player = "FULLER" then position="SF";
if player = "GAMBLE" then position="SF";
if player = "GEIGER" then position="SF";
if player = "GILL" then position="SF";
if player = "GILLIAM" then position="PF";
if player = "GRANT" and team="SACRAMENTO" then do; player="B_GRANT"; position="PF";end;
if player = "GRANT" and team="WASHINGTON" then do; player="Ha_GRANT"; position="PF";end;
if player = "Ha_GRANT" and date=mdy(11,08,1996) then position="SF";;
if player = "GRANT" and team="ORLANDO" then do; player="Ho_GRANT"; position="PF";end;
if player = "GREEN" then position="PF";
if player = "GUGLIOTTA" then position="PF";
if player = "HAM" then position="SF";
if player = "HAMMONDS" then position="SF";
if player = "HARRINGTON" then position="PF";
if player = "HERRERA" then position="PF";
if player = "HILL" and team="DETROIT" then do; player="G_HILL";position="SF";end;
if player = "HILL" and team="CLEVELAND" then do; player="T_HILL";position="PF";end;
if player = "HORRY" then position="SF";
if player = "HORRY" and mdy(04,11,1997) <= date <= mdy(04,13,1997) then position="PF";
if player = "HORRY" and date=mdy(11,02,1996) then position="PF";
if player = "HOWARD" then position="SF";
if player = "HOWARD" and date=mdy(02,21,1997) then position="PF";
if player = "HUNTER" then position="SF";
if player = "J WILLIAMS" then do; player="JY_WILLIAMS";position="PF";end;
if player = "JAMES" then do; player="H_JAMES";position="SF";end;
if player = "JOHNSON" and team="NEW YORK" then do; player="L_JOHNSON";position="SF";end;
if player = "JOHNSON" and team="HOUSTON" then do; player="E_JOHNSON";position="SF";end;
if player = "JONES" and team="TORONTO" then do; player="P_JONES";position="PF";end;
if player = "P_JONES" and date=mdy(11,19,1996) then position="SF";
if player = "P_JONES" and date=mdy(02,05,1997) then position="SF";
if player = "P_JONES" and date=mdy(04,02,1997) then position="SF";
if player = "P_JONES" and date=mdy(04,12,1997) then position="SF";
if player = "P_JONES" and date=mdy(04,14,1997) then position="SF";
if player = "JONES" and team="LA LAKERS" then do; player="E_JONES";position="SF";end;
if player = "K GARNETT" then do; player="K_GARNETT"; position="SF";end;
if player = "KEEFE" then position="PF";
if player = "KEMP" then position="PF";
```

```
if player = "KERSEY" then position="SF";
if player = "KITTLES" then position="SF";
if player = "KNIGHT" then position="PF";
if player = "KUKOC" then position="PF";
if player = "L ELLIS" then do; player="L_ELLIS";position="SF";end;
if player = "L_ELLIS" and mdy(01,07,1997)=< date =< mdy(01,18,1997) then position="PF";
if player = "LAETTNER" then position="PF";
if player = "LANG" then do; player="An_LANG"; position="SF"; end;
if player = "LLAMAS" then position="PF";
if player = "LONG" then position="PF";
if player = "LYNCH" then position="SF";
if player = "M WILLIAMS" then do; player="M_WILLIAMS";position="SF";end;
if player = "M_WILLIAMS" and date=mdy(01,31,1997) then position="PF";
if player = "MACK" then position="SF";
if player = "MACLEAN" then position="SF";
if player = "MAJERLE" then position="SF";
if player = "MALONE" then position = "PF";
if player = "MANNING" then position="PF";
if player = "MANNING" and date=mdy(01,08,1997) then position="SF";
if player = "MANNING" and date=mdy(11,21,1996) then position="SF";
if player = "MARSHALL" then position="SF";
if player = "MASHBURN" then position="SF";
if player = "MASON" then position="PF";
if player = "MASSENBURG" then position="PF";
if player = "MAXWELL" then position="SF";
if player = "MCCARTY" then position="SF";
if player = "MCCASKILL" then position="SF";
if player = "MCCLOUD" then position="SF";
if player = "MCDANIEL" then do; player="X_MCDANIEL"; position="PF";end;
if player = "X_MCDANIEL" and date=mdy(02,11,1997) then position="SF";
if player = "MCDYESS" then position="PF";
if player = "MCKEY" then position="SF";
if player = "MCKIE" then position="SF";
if player = "MCMILLAN" then position="SF";
if player = "MEYER" then position="PF";
if player = "MILLER" then position="PF";
if player = "MILLS" and team="CLEVELAND" then do;player="C_MILLS";position="SF";end;
if player = "MILLS" and team="DETROIT" then do;player="T_MILLS";position="PF";end;
if player = "T_MILLS" and date=mdy(11,15,1996) then position="SF";
if player = "MITCHELL" then position="PF";
if player = "MITCHELL" and mdy(12,08,1996) =< date =< mdy(12,11,1996) then position="SF";
if player = "MOBLEY" then position="PF";
if player = "MORRIS" then position="SF";
if player = "MULLIN" then position="SF";
if player = "MURRAY" and team="WASHINGTON" then do; player="T_MURRAY"; position="SF";end;
if player = "MURRAY" and team="LA CLIPPERS" then do; player="L_MURRAY"; position="SF";end;
if player = "NEWMAN" then position="SF";
if player = "OAKLEY" then position="PF";
if player = "OBANNON" then position="SF";
if player = "OUTLAW" then position="SF";
if player = "OWENS" then position="SF";
if player = "OWES" then position="PF";
if player = "PERKINS" then position="PF";
if player = "PERSON" then position="SF";
if player = "PHILLS" then position="SF";
if player = "PIERCE" then position="SF";
if player = "PIPPEN" then position="SF";
if player = "POLYNICE" then position="PF";
if player = "RADJA" then position="PF";
if player = "RATLIFF" then position="PF";
if player = "REID" then position="PF";
if player = "RICE" then position="SF";
if player = "ROBINSON" and team="MILWAUKEE" then do; player="G_ROBINSON"; position="SF";end;
if player = "ROBINSON" and team="SAN ANTONIO" then do; player="D_ROBINSON"; position="PF";end;
if player = "RODMAN" then position="PF";
if player = "ROGERS" and team="LA CLIPPERS" then do; player="ROD_ROGERS";position="SF";end;
if player = "ROD_ROGERS" and mdy(04,01,1997) =< date =< mdy(04,08,1997) then position="PF";
if player = "ROGERS" and team="VANCOUVER" then do; player="ROY_ROGERS";position="PF";end;
if player = "ROGERS" and team="TORONTO" then do; player="C_ROGERS";position="SF";end;
if player = "ROSE" then position="SF";
if player = "ROYAL" then position="PF";
if player = "ROZIER" then position="PF";
if player = "RUSSELL" then position="SF";
if player = "SCHREMPF" then position="SF";
if player = "SCHREMPH" and mdy(04,03,1997) =< date =< mdy(04,05,1997) then position="PF";
if player = "SCOTT" then position="SF";
if player = "SLATER" then position="PF";
if player = "SMITH" and team="SAN ANTONIO" then do; player="C_SMITH";position="PF";end;
if player = "C_SMITH" and mdy(04,14,1997) =< date =< mdy(04,15,1997) then position="SF";
if player = "SMITH" and team="GOLDEN STATE" then do; player="J_SMITH";position="PF";end;
if player = "SMITH" and team="SACRAMENTO" then do; player="M_SMITH";position="PF";end;
if player = "STACKHOUSE" then position="SF";
if player = "STEWART" then position="PF";
if player = "STEWART" and mdy(02,28,1997) =< date =< mdy(03,28,1997) then position="SF";
if player = "STEWART" and date=mdy(04,12,1997) then position="SF";
if player = "STEWART" and date=mdy(04,15,1997) then position="SF";
if player = "STRICKLAND" then position="SF";
if player = "STRONG" then position="SF";
if player = "STRONG" and mdy(01,02,1997) =< date =< mdy(04,19,1997) then position="PF";
```

```
if player = "SURA" then position="SF";
if player = "TABAK" then position="PF";
if player = "THOMAS" then position="PF";
if player = "THORPE" then position="PF";
if player = "THORPE" and date=mdy(11,12,1996) then position="SF";
if player = "THORPE" and date=mdy(11,20,1996) then position="SF";
if player = "TISDALE" then position="PF";
if player = "TRENT" then position="PF";
if player = "TRENT" and date=mdy(11,15,1996) then position="SF";
if player = "TRENT" and date=mdy(03,25,1997) then position="SF";
if player = "VAUGHT" then position="PF";
if player = "W PERDUE" then do; player="PURDUE";position="PF";end;
if player = "WALKER" and team="BOSTON" then do; player="A_WALKER";position="PF";end;
if player = "A_WALKER" and date=mdy(11,06,1996) then position="SF";
if player = "WALKER" and team="DALLAS" then do; player="S_WALKER";position="PF";end;
if player = "WALLACE" and team="PORTLAND" then do; player="R_WALLACE";position="PF";end;
if player = "WALLACE" and team="NEW YORK" then do; player="J_WALLACE";position="SF";end;
if player = "WEATHERSPOON" then position="SF";
if player = "WEATHERSPOON" and mdy(11,02,1996) <= date <= mdy(11,05,1996) then position="PF";
if player = "WEATHERSPOON" and mdy(01,10,1997) <= date <= mdy(02,18,1997) then position="PF";
if player = "WEATHERSPOON" and mdy(03,30,1997) <= date <= mdy(04,19,1997) then position="PF";
if player = "WEBBER" then position="PF";
if player = "WEST" then position="SF";
if player = "WILLIAMS" and team="TORONTO" then do; player="W_WILLIAMS";position="SF";end;
if player = "WILLIAMS" and team="BOSTON" then do; player="E_WILLIAMS";position="SF";end;
if player = "E_WILLIAMS" and date=mdy(11,22,1996) then position="PF";
if player = "E_WILLIAMS" and mdy(12,08,1996)<= date <= mdy(02,06,1997) then position="PF";
if player = "WILLIAMS" and team="VANCOUVER" then do; player="A_WILLIAMS";position="PF";end;
if player = "WILLIS" then position="PF";
if player = "WINGATE" then position="SF";
if player = "WRIGHT" and team="TORONTO" then do; player="S_WRIGHT";position="SF";end;
if player = "WRIGHT" and team="LA CLIPPERS" then do; player="L_WRIGHT";position="SF";end;
if player = "ZIDEK" then position="PF";
run;

data sf;
set forwards96_97;
if position="SF";
run;
data pf;
set forwards96_97;
if position="PF";
run;

data centers96_97;
infile "F:/Masters Project/centers/centers1996_1997.csv" dlm="," firstobs=2;
drop _TYPE_ _FREQ_;
informat date mmddyy10.;
format date mmddyy10.;
input date team :$20. vsteam :$20. hometeam :$20. Position$ Player :$20. _TYPE_ _FREQ_ minutes
     fgm fga ftm fta orb drb trb assist fouls steals turnovers points tpoints pointdiff
    percent;
sfgm = fgm/minutes;
sftm = ftm/minutes;
sorb=orb/minutes;
sdrb = drb/minutes;
sassist = assist/minutes;
sfouls = fouls/minutes;
ssteals = steals/minutes;
sturnovers = turnovers/minutes;
spoints = points/minutes;
ftpercent = ftm/fta;
if ftpercent = . then ftpercent = 0;
fgpercent = percent;
if fgpercent = . then fgpercent = 0;
if player = "WILLIAMS" and team="PHILADELPHIA" then do;player="Scott_WILLIAMS";end;
if player = "WILLIAMS" and team="PHOENIX" then do;player="HotRod_WILLIAMS";end;
if player = "WEST" and team = "CLEVELAND" then do; player="M_WEST";end;
if player = "LANG" and team = "MILWAUKEE" then do; player="Andrew_LANG";end;
if player = "REEVES" and team = "VANCOUVER" then do; player = "B_REEVES";end;
run;



data work;
set pg sg pf sf centers96_97;
drop percent;
run;

proc sort data=work;
by date hometeam team ;
run;
data try;
do game = 1 to 1163;
do junk = 1 to 10;
output;
end;
end;
run;
```

```
data work1;
merge work try;
drop junk;
run;

data pg96_97;
set work1;
if position ="PG";
run;
data sg96_97;
set work1;
if position ="SG";
run;
data sf96_97;
set work1;
if position ="SF";
run;
data pf96_97;
set work1;
if position ="PF";
run;
data c96_97;
set work1;
if position ="C";
run;
```

Appendix B

Gibbsit Function Output

	Kthin	Nburn	Nprec	Nmin	I_RL	Kind
sigma2	1	2	593	600	0.99	1
sigg	1	4	823	600	1.38	2
sigt	1	3	654	600	1.09	2
sigb0	1	2	593	600	0.99	1
sigb1	1	3	654	600	1.09	2
sigb2	1	2	633	600	1.06	2
sigb3	1	3	654	600	1.09	2
sigb5	1	2	633	600	1.06	2
sigb6	1	2	613	600	1.02	2
sigb7	1	2	633	600	1.06	2
sigb8	1	3	676	600	1.13	2
sigb9	1	3	676	600	1.13	2
sigb10	1	3	654	600	1.09	2
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
C	1	11	1955	600	3.28	5
PF	1	10	1715	600	2.88	3
PG	1	4	823	600	1.38	2
SF	1	13	2171	600	3.64	6
SG	2	12	2270	600	3.80	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2

MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4

ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2

STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3

FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2

M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3

G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3

W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2
A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3

HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4
POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ABDUL_RAUF	1	2	593	600	0.99	1
ABDUR_RAHIM	1	5	880	600	1.48	3
ANDERSON	1	6	1163	600	1.95	3
A_HARDAWAY	1	3	676	600	1.13	2

A_JOHNSON	1	3	654	600	1.09	2
Andrew_LANG	2	10	1708	600	2.86	4
BAKER	1	5	911	600	1.53	2
BARKLEY	1	5	880	600	1.48	3
BLAYLOCK	1	3	721	600	1.21	2
BOGUES	1	3	721	600	1.21	2
BRADLEY	1	7	1206	600	2.02	4
BRANDON	1	2	633	600	1.06	2
B_REEVES	2	12	2492	600	4.17	5
CAMPBELL	1	9	1579	600	2.65	4
CHEANEY	1	6	1045	600	1.75	3
CHILDS	1	3	654	600	1.09	2
CHRISTIE	1	6	1045	600	1.75	3
COLEMAN	1	6	1083	600	1.82	3
CORBIN	1	7	1251	600	2.10	4
C_MILLS	2	8	1488	600	2.49	3
Cliff_ROBINSON	1	6	1060	600	1.78	4
D GARRETT	2	10	2144	600	3.59	4
DEL_NEGRO	2	10	1888	600	3.16	4
DIVAC	2	12	2444	600	4.09	5
DOUGLAS	1	3	654	600	1.09	2
DREXLER	1	6	1083	600	1.82	3
DUMARS	1	4	771	600	1.29	2
D_DAVIS	1	5	911	600	1.53	3
D_ELLIS	2	10	2116	600	3.54	4
D_MARTIN	1	2	613	600	1.02	2
Doug_WEST	2	10	1996	600	3.34	4
ELIE	1	5	1010	600	1.69	4
EWING	2	12	2238	600	3.75	4
E_JONES	1	5	911	600	1.53	3
E_WILLIAMS	2	10	1968	600	3.30	4
FERRY	2	8	1224	600	2.05	3
FINLEY	2	10	1758	600	2.95	3
FOX	2	10	1758	600	2.95	4
GILL	2	10	1810	600	3.03	5
GREEN	1	5	880	600	1.48	2
GUGLIOTTA	1	5	942	600	1.58	3
G_HILL	1	5	956	600	1.60	5
G_PAYTON	1	3	654	600	1.09	2
G_ROBINSON	2	12	1874	600	3.14	4
HAWKINS	1	6	1163	600	1.95	2
HERRERA	2	6	1348	600	2.26	3
HORNACEK	1	5	975	600	1.63	3
HOUSTON	1	6	1122	600	1.88	3
HOWARD	2	10	1862	600	3.12	4
HUNTER	1	2	633	600	1.06	2
Ho_GRANT	2	8	1718	600	2.88	3
HotRod_WILLIAMS	2	12	2044	600	3.43	4
IVERSON	1	2	613	600	1.02	2
JOHNSON	1	6	1163	600	1.95	3
JORDAN	2	8	1730	600	2.90	3
J_JACKSON	1	5	942	600	1.58	3
J_SMITH	2	8	1916	600	3.21	4
KEMP	1	5	975	600	1.63	2
KITTLES	1	2	633	600	1.06	2
K_ANDERSON	1	3	721	600	1.21	2
K_GARNETT	1	5	942	600	1.58	4
K_JOHNSON	1	3	721	600	1.21	2
L Wright	1	6	1122	600	1.88	3
LAETTNER	1	4	746	600	1.25	2
LENARD	2	6	1570	600	2.63	3
LONGLEY	2	12	2270	600	3.80	4
L_ELLIS	1	5	942	600	1.58	4
L_JOHNSON	2	10	1878	600	3.15	4
MALONE	1	4	851	600	1.43	2
MALONEY	1	3	698	600	1.17	2
MARBURY	1	3	654	600	1.09	2
MASHBURN	2	10	2166	600	3.63	8
MASON	1	5	942	600	1.58	2
MASSENBURG	1	6	1083	600	1.82	4
MCDYESS	1	5	911	600	1.53	3
MCILVAINE	2	10	1862	600	3.12	4
MCKEY	2	8	1286	600	2.16	4
MONTROSS	1	8	1402	600	2.35	5
MOURNING	2	14	2210	600	3.71	5
MULLIN	1	6	1045	600	1.75	5
MURESAN	2	8	1476	600	2.47	4
MUTOMBO	1	8	1349	600	2.26	3
M_JACKSON	1	3	698	600	1.17	2
M_SMITH	1	5	942	600	1.58	3
N_ANDERSON	2	10	1632	600	2.74	3
OAKLEY	2	8	1382	600	2.32	3
OLAJUWON	2	12	2224	600	3.73	4
ONEAL	2	10	2144	600	3.59	6
OSTERTAG	1	5	911	600	1.53	2
OWENS	1	8	1402	600	2.35	4
PEELER	1	4	823	600	1.38	2
PHILLS	1	5	975	600	1.63	3
PIPPEN	2	10	2166	600	3.63	4

POLYNICE	1	8	1402	600	2.35	4
PRICE	1	3	654	600	1.09	2
P_BROWN	1	5	880	600	1.48	3
P_JONES	2	8	1916	600	3.21	4
RICE	1	5	1010	600	1.69	4
RICHMOND	1	6	1045	600	1.75	3
RIDER	1	5	1010	600	1.69	3
RODMAN	2	10	1836	600	3.08	4
ROD_ROGERS	1	7	1269	600	2.13	5
ROY_ROGERS	1	4	796	600	1.33	2
RUSSELL	1	7	1206	600	2.02	5
R_ALLEN	1	5	880	600	1.48	3
R_HARPER	1	3	654	600	1.09	2
R_MILLER	1	5	1010	600	1.69	3
R_STRICKLAND	1	3	676	600	1.13	2
R_WALLACE	1	5	1010	600	1.69	3
SABONIS	1	6	1083	600	1.82	3
SCHREMPF	2	10	1784	600	2.99	5
SCOTT	1	5	1010	600	1.69	4
SEALY	2	10	1708	600	2.86	3
SEIKALY	2	10	1962	600	3.29	4
SMITS	2	10	2036	600	3.41	4
SPENCER	2	10	2012	600	3.37	5
SPREWELL	2	10	1878	600	3.15	4
STACKHOUSE	2	10	2012	600	3.37	4
STITH	1	6	1045	600	1.75	3
STOCKTON	1	3	721	600	1.21	2
STOUDAMIRE	1	3	698	600	1.17	2
S_SMITH	1	7	1251	600	2.10	5
Scott_WILLIAMS	2	8	1916	600	3.21	4
THORPE	1	6	1045	600	1.75	2
T_HARDAWAY	1	3	698	600	1.17	2
VAN_EXEL	1	3	698	600	1.17	2
VAUGHT	2	6	1158	600	1.94	2
WEATHERSPOON	2	8	1998	600	3.34	4
WEBBER	1	6	1083	600	1.82	3
WESLEY	1	4	746	600	1.25	3
W_WILLIAMS	2	10	1826	600	3.06	4
ATLANTA	1	3	654	600	1.09	2
BOSTON	1	3	676	600	1.13	2
CHARLOTTE	1	3	698	600	1.17	2
CHICAGO	1	3	676	600	1.13	2
CLEVELAND	1	2	613	600	1.02	2
DALLAS	1	2	633	600	1.06	2
DENVER	1	2	593	600	0.99	1
DETROIT	1	2	613	600	1.02	2
GOLDEN STATE	1	3	654	600	1.09	2
HOUSTON	1	6	1122	600	1.88	3
INDIANA	1	2	633	600	1.06	2
LA CLIPPERS	1	2	613	600	1.02	2
LA LAKERS	1	3	654	600	1.09	2
MIAMI	1	3	654	600	1.09	2
MILWAUKEE	1	3	654	600	1.09	2
MINNESOTA	1	2	613	600	1.02	2
NEW JERSEY	1	2	593	600	0.99	1
NEW YORK	1	2	593	600	0.99	1
ORLANDO	1	2	633	600	1.06	2
PHILADELPHIA	1	2	633	600	1.06	2
PHOENIX	1	2	613	600	1.02	2
PORTLAND	1	2	613	600	1.02	2
SACRAMENTO	1	3	654	600	1.09	2
SAN ANTONIO	1	2	613	600	1.02	2
SEATTLE	1	3	721	600	1.21	2
TORONTO	1	3	665	600	1.11	2
UTAH	1	3	676	600	1.13	2
VANCOUVER	1	2	613	600	1.02	2
WASHINGTON	1	2	633	600	1.06	2
ATLANTA	1	3	654	600	1.09	2
BOSTON	1	3	676	600	1.13	2
CHARLOTTE	1	3	698	600	1.17	2
CHICAGO	1	3	676	600	1.13	2
CLEVELAND	1	2	613	600	1.02	2
DALLAS	1	2	633	600	1.06	2
DENVER	1	2	593	600	0.99	1
DETROIT	1	2	613	600	1.02	2
GOLDEN STATE	1	3	654	600	1.09	2
HOUSTON	1	6	1122	600	1.88	3
INDIANA	1	2	633	600	1.06	2
LA CLIPPERS	1	2	613	600	1.02	2
LA LAKERS	1	3	654	600	1.09	2
MIAMI	1	3	654	600	1.09	2
MILWAUKEE	1	3	654	600	1.09	2
MINNESOTA	1	2	613	600	1.02	2
NEW JERSEY	1	2	593	600	0.99	1
NEW YORK	1	2	593	600	0.99	1
ORLANDO	1	2	633	600	1.06	2
PHILADELPHIA	1	2	633	600	1.06	2
PHOENIX	1	2	613	600	1.02	2
PORTLAND	1	2	613	600	1.02	2

SACRAMENTO	1	3	654	600	1.09	2
SAN ANTONIO	1	2	613	600	1.02	2
SEATTLE	1	3	721	600	1.21	2
TORONTO	1	3	665	600	1.11	2
UTAH	1	3	676	600	1.13	2
VANCOUVER	1	2	613	600	1.02	2
WASHINGTON	1	2	633	600	1.06	2
X1	1	2	613	600	1.02	2
X2	1	2	633	600	1.06	2
X3	1	2	593	600	0.99	1
X4	1	2	574	600	0.96	1
X5	1	3	721	600	1.21	2
X6	1	2	613	600	1.02	2
X7	1	2	593	600	0.99	1
X8	1	2	613	600	1.02	2
X9	1	2	613	600	1.02	2
X10	1	2	574	600	0.96	1
X11	1	2	593	600	0.99	1
X12	1	2	593	600	0.99	1
X13	1	2	574	600	0.96	1
X14	1	2	574	600	0.96	1
X15	1	2	593	600	0.99	1
X16	1	2	613	600	1.02	2
X17	1	2	574	600	0.96	1
X18	1	2	613	600	1.02	2
X19	1	2	593	600	0.99	1
X20	1	2	633	600	1.06	2
X21	1	2	613	600	1.02	2
X22	1	2	624	600	1.04	2
X23	1	2	593	600	0.99	1
X24	1	2	613	600	1.02	2
X25	1	2	613	600	1.02	2
X26	1	2	633	600	1.06	2
X27	1	2	613	600	1.02	2
X28	1	2	613	600	1.02	2
X29	1	2	613	600	1.02	2
X30	1	2	633	600	1.06	2
X31	1	2	613	600	1.02	2
X32	1	3	654	600	1.09	2
X33	1	2	633	600	1.06	2
X34	1	3	676	600	1.13	2
X35	1	2	613	600	1.02	2
X36	1	2	633	600	1.06	2
X37	1	3	676	600	1.13	2
X38	1	2	613	600	1.02	2
X39	1	2	593	600	0.99	1
X40	1	2	593	600	0.99	1
X41	1	2	593	600	0.99	1
X42	1	2	574	600	0.96	1
X43	1	3	654	600	1.09	2
X44	1	2	613	600	1.02	2
X45	1	2	633	600	1.06	2
X46	1	3	654	600	1.09	2
X47	1	2	593	600	0.99	1
X48	1	2	593	600	0.99	1
X49	1	2	574	600	0.96	1
X50	1	2	585	600	0.98	1
X51	1	2	613	600	1.02	2
X52	1	2	633	600	1.06	2
X53	1	2	593	600	0.99	1
X54	1	2	593	600	0.99	1
X55	1	2	593	600	0.99	1
X56	1	2	593	600	0.99	1
X57	1	2	604	600	1.01	2
X58	1	3	654	600	1.09	2
X59	1	2	574	600	0.96	1
X60	1	3	654	600	1.09	2
X61	1	2	613	600	1.02	2
X62	1	2	593	600	0.99	1
X63	1	2	593	600	0.99	1
X64	1	2	634	600	1.06	2
X65	1	2	633	600	1.06	2
X66	1	2	593	600	0.99	1
X67	1	2	574	600	0.96	1
X68	1	2	593	600	0.99	1
X69	1	2	593	600	0.99	1
X70	1	2	574	600	0.96	1
X71	1	2	633	600	1.06	2
X72	1	2	633	600	1.06	2
X73	1	2	593	600	0.99	1
X74	1	2	633	600	1.06	2
X75	1	2	633	600	1.06	2
X76	1	2	613	600	1.02	2
X77	1	2	613	600	1.02	2
X78	1	2	574	600	0.96	1
X79	1	2	633	600	1.06	2
X80	1	2	633	600	1.06	2
X81	1	2	593	600	0.99	1
X82	1	2	585	600	0.98	1

X83	1	2	633	600	1.06	2	X172	1	2	593	600	0.99	1
X84	1	2	633	600	1.06	2	X173	1	3	654	600	1.09	2
X85	1	2	593	600	0.99	1	X174	1	2	633	600	1.06	2
X86	1	2	593	600	0.99	1	X175	1	2	613	600	1.02	2
X87	1	2	593	600	0.99	1	X176	1	2	574	600	0.96	1
X88	1	2	593	600	0.99	1	X177	1	2	604	600	1.01	2
X89	1	2	613	600	1.02	2	X178	1	2	593	600	0.99	1
X90	1	2	633	600	1.06	2	X179	1	2	593	600	0.99	1
X91	1	3	654	600	1.09	2	X180	1	2	593	600	0.99	1
X92	1	2	574	600	0.96	1	X181	1	2	593	600	0.99	1
X93	1	2	613	600	1.02	2	X182	1	2	593	600	0.99	1
X94	1	2	613	600	1.02	2	X183	1	2	613	600	1.02	2
X95	1	2	574	600	0.96	1	X184	1	2	613	600	1.02	2
X96	1	2	613	600	1.02	2	X185	1	3	654	600	1.09	2
X97	1	2	613	600	1.02	2	X186	1	2	613	600	1.02	2
X98	1	2	633	600	1.06	2	X187	1	2	613	600	1.02	2
X99	1	2	566	600	0.95	1	X188	1	2	593	600	0.99	1
X100	1	2	593	600	0.99	1	X189	1	2	613	600	1.02	2
X101	1	2	574	600	0.96	1	X190	1	3	654	600	1.09	2
X102	1	2	633	600	1.06	2	X191	1	2	613	600	1.02	2
X103	1	2	613	600	1.02	2	X192	1	2	574	600	0.96	1
X104	1	2	633	600	1.06	2	X193	1	2	633	600	1.06	2
X105	1	2	593	600	0.99	1	X194	1	2	593	600	0.99	1
X106	1	2	574	600	0.96	1	X195	1	2	574	600	0.96	1
X107	1	2	613	600	1.02	2	X196	1	2	633	600	1.06	2
X108	1	2	574	600	0.96	1	X197	1	2	604	600	1.01	2
X109	1	2	593	600	0.99	1	X198	1	2	613	600	1.02	2
X110	1	2	593	600	0.99	1	X199	1	2	593	600	0.99	1
X111	1	2	613	600	1.02	2	X200	1	2	633	600	1.06	2
X112	1	2	593	600	0.99	1	X201	1	2	574	600	0.96	1
X113	1	2	613	600	1.02	2	X202	1	3	654	600	1.09	2
X114	1	2	593	600	0.99	1	X203	1	2	633	600	1.06	2
X115	1	2	633	600	1.06	2	X204	1	2	614	600	1.03	2
X116	1	3	676	600	1.13	2	X205	1	2	633	600	1.06	2
X117	1	2	593	600	0.99	1	X206	1	3	698	600	1.17	2
X118	1	3	654	600	1.09	2	X207	1	2	593	600	0.99	1
X119	1	2	574	600	0.96	1	X208	1	2	593	600	0.99	1
X120	1	2	593	600	0.99	1	X209	1	2	613	600	1.02	2
X121	1	2	633	600	1.06	2	X210	1	2	593	600	0.99	1
X122	1	2	593	600	0.99	1	X211	1	2	593	600	0.99	1
X123	1	2	574	600	0.96	1	X212	1	2	633	600	1.06	2
X124	1	2	613	600	1.02	2	X213	1	2	633	600	1.06	2
X125	1	2	633	600	1.06	2	X214	1	2	574	600	0.96	1
X126	1	2	593	600	0.99	1	X215	1	2	593	600	0.99	1
X127	1	2	593	600	0.99	1	X216	1	2	574	600	0.96	1
X128	1	2	613	600	1.02	2	X217	1	2	574	600	0.96	1
X129	1	3	654	600	1.09	2	X218	1	2	633	600	1.06	2
X130	1	3	656	600	1.10	2	X219	1	2	574	600	0.96	1
X131	1	2	633	600	1.06	2	X220	1	2	633	600	1.06	2
X132	1	2	593	600	0.99	1	X221	1	2	613	600	1.02	2
X133	1	2	574	600	0.96	1	X222	1	2	613	600	1.02	2
X134	1	2	613	600	1.02	2	X223	1	2	633	600	1.06	2
X135	1	2	633	600	1.06	2	X224	1	3	654	600	1.09	2
X136	1	2	633	600	1.06	2	X225	1	2	593	600	0.99	1
X137	1	2	593	600	0.99	1	X226	1	2	633	600	1.06	2
X138	1	2	613	600	1.02	2	X227	1	2	613	600	1.02	2
X139	1	2	574	600	0.96	1	X228	1	3	654	600	1.09	2
X140	1	3	654	600	1.09	2	X229	1	2	624	600	1.04	2
X141	1	3	654	600	1.09	2	X230	1	2	593	600	0.99	1
X142	1	2	613	600	1.02	2	X231	1	2	593	600	0.99	1
X143	1	2	574	600	0.96	1	X232	1	2	633	600	1.06	2
X144	1	2	613	600	1.02	2	X233	1	2	593	600	0.99	1
X145	1	2	613	600	1.02	2	X234	1	2	604	600	1.01	2
X146	1	2	633	600	1.06	2	X235	1	2	613	600	1.02	2
X147	1	2	593	600	0.99	1	X236	1	2	593	600	0.99	1
X148	1	2	593	600	0.99	1	X237	1	2	633	600	1.06	2
X149	1	2	613	600	1.02	2	X238	1	2	633	600	1.06	2
X150	1	3	654	600	1.09	2	X239	1	2	613	600	1.02	2
X151	1	2	613	600	1.02	2	X240	1	3	654	600	1.09	2
X152	1	2	593	600	0.99	1	X241	1	3	654	600	1.09	2
X153	1	2	593	600	0.99	1	X242	1	2	593	600	0.99	1
X154	1	2	574	600	0.96	1	X243	1	2	633	600	1.06	2
X155	1	2	613	600	1.02	2	X244	1	2	634	600	1.06	2
X156	1	2	593	600	0.99	1	X245	1	2	633	600	1.06	2
X157	1	2	613	600	1.02	2	X246	1	2	613	600	1.02	2
X158	1	2	633	600	1.06	2	X247	1	2	613	600	1.02	2
X159	1	2	593	600	0.99	1	X248	1	3	644	600	1.08	2
X160	1	2	633	600	1.06	2	X249	1	2	593	600	0.99	1
X161	1	2	633	600	1.06	2	X250	1	2	593	600	0.99	1
X162	1	2	593	600	0.99	1	X251	1	2	613	600	1.02	2
X163	1	2	593	600	0.99	1	X252	1	2	593	600	0.99	1
X164	1	2	613	600	1.02	2	X253	1	2	593	600	0.99	1
X165	1	2	613	600	1.02	2	X254	1	2	593	600	0.99	1
X166	1	2	593	600	0.99	1	X255	1	2	633	600	1.06	2
X167	1	2	593	600	0.99	1	X256	1	2	613	600	1.02	2
X168	1	2	633	600	1.06	2	X257	1	2	593	600	0.99	1
X169	1	2	613	600	1.02	2	X258	1	2	574	600	0.96	1
X170	1	2	633	600	1.06	2	X259	1	2	613	600	1.02	2
X171	1	3	676	600	1.13	2	X260	1	2	593	600	0.99	1

X261	1	2	574	600	0.96	1
X262	1	3	654	600	1.09	2
X263	1	2	593	600	0.99	1
X264	1	2	633	600	1.06	2
X265	1	3	654	600	1.09	2
X266	1	2	633	600	1.06	2
X267	1	2	574	600	0.96	1
X268	1	2	613	600	1.02	2
X269	1	2	613	600	1.02	2
X270	1	2	593	600	0.99	1
X271	1	3	654	600	1.09	2
X272	1	3	676	600	1.13	2
X273	1	2	593	600	0.99	1
X274	1	2	593	600	0.99	1
X275	1	2	613	600	1.02	2
X276	1	2	633	600	1.06	2
X277	1	2	633	600	1.06	2
X278	1	2	574	600	0.96	1
X279	1	2	593	600	0.99	1
X280	1	2	593	600	0.99	1
X281	1	2	613	600	1.02	2
X282	1	2	574	600	0.96	1
X283	1	2	593	600	0.99	1
X284	1	2	633	600	1.06	2
X285	1	2	593	600	0.99	1
X286	1	2	574	600	0.96	1
X287	1	2	613	600	1.02	2
X288	1	2	633	600	1.06	2
X289	1	2	585	600	0.98	1
X290	1	2	593	600	0.99	1
X291	1	2	593	600	0.99	1
X292	1	2	613	600	1.02	2
X293	1	3	654	600	1.09	2
X294	1	2	574	600	0.96	1
X295	1	2	593	600	0.99	1
X296	1	2	633	600	1.06	2
X297	1	2	593	600	0.99	1
X298	1	2	613	600	1.02	2
X299	1	2	593	600	0.99	1
X300	1	3	698	600	1.17	2
X301	1	2	613	600	1.02	2
X302	1	2	593	600	0.99	1
X303	1	2	574	600	0.96	1
X304	1	2	593	600	0.99	1
X305	1	2	593	600	0.99	1
X306	1	2	593	600	0.99	1
X307	1	2	633	600	1.06	2
X308	1	2	574	600	0.96	1
X309	1	2	593	600	0.99	1
X310	1	2	613	600	1.02	2
X311	1	2	593	600	0.99	1
X312	1	2	593	600	0.99	1
X313	1	2	593	600	0.99	1
X314	1	2	593	600	0.99	1
X315	1	2	613	600	1.02	2
X316	1	2	613	600	1.02	2
X317	1	2	593	600	0.99	1
X318	1	2	633	600	1.06	2
X319	1	3	676	600	1.13	2
X320	1	2	593	600	0.99	1
X321	1	3	654	600	1.09	2
X322	1	2	613	600	1.02	2
X323	1	2	613	600	1.02	2
X324	1	2	574	600	0.96	1
X325	1	2	633	600	1.06	2
X326	1	3	654	600	1.09	2
X327	1	2	613	600	1.02	2
X328	1	2	613	600	1.02	2
X329	1	2	633	600	1.06	2
X330	1	2	593	600	0.99	1
X331	1	2	593	600	0.99	1
X332	1	2	633	600	1.06	2
X333	1	2	613	600	1.02	2
X334	1	2	613	600	1.02	2
X335	1	2	613	600	1.02	2
X336	1	2	633	600	1.06	2
X337	1	2	593	600	0.99	1
X338	1	2	593	600	0.99	1
X339	1	2	593	600	0.99	1
X340	1	2	633	600	1.06	2
X341	1	2	574	600	0.96	1
X342	1	2	613	600	1.02	2
X343	1	2	593	600	0.99	1
X344	1	3	676	600	1.13	2
X345	1	2	574	600	0.96	1
X346	1	3	676	600	1.13	2
X347	1	2	613	600	1.02	2
X348	1	2	613	600	1.02	2
X349	1	2	633	600	1.06	2
X350	1	2	593	600	0.99	1
X351	1	2	593	600	0.99	1
X352	1	2	593	600	0.99	1
X353	1	2	613	600	1.02	2
X354	1	2	574	600	0.96	1
X355	1	2	613	600	1.02	2
X356	1	2	593	600	0.99	1
X357	1	2	593	600	0.99	1
X358	1	2	593	600	0.99	1
X359	1	2	633	600	1.06	2
X360	1	3	654	600	1.09	2
X361	1	2	613	600	1.02	2
X362	1	2	574	600	0.96	1
X363	1	2	613	600	1.02	2
X364	1	3	721	600	1.21	2
X365	1	2	593	600	0.99	1
X366	1	2	613	600	1.02	2
X367	1	2	593	600	0.99	1
X368	1	2	613	600	1.02	2
X369	1	2	593	600	0.99	1
X370	1	2	593	600	0.99	1
X371	1	2	574	600	0.96	1
X372	1	2	593	600	0.99	1
X373	1	3	654	600	1.09	2
X374	1	2	574	600	0.96	1
X375	1	2	613	600	1.02	2
X376	1	3	676	600	1.13	2
X377	1	2	574	600	0.96	1
X378	1	2	613	600	1.02	2
X379	1	2	593	600	0.99	1
X380	1	2	593	600	0.99	1
X381	1	2	593	600	0.99	1
X382	1	3	676	600	1.13	2
X383	1	2	633	600	1.06	2
X384	1	2	633	600	1.06	2
X385	1	2	613	600	1.02	2
X386	1	2	566	600	0.95	1
X387	1	3	654	600	1.09	2
X388	1	2	574	600	0.96	1
X389	1	2	574	600	0.96	1
X390	1	2	613	600	1.02	2
X391	1	3	676	600	1.13	2
X392	1	2	593	600	0.99	1
X393	1	2	574	600	0.96	1
X394	1	2	613	600	1.02	2
X395	1	2	574	600	0.96	1
X396	1	2	593	600	0.99	1
X397	1	3	654	600	1.09	2
X398	1	2	633	600	1.06	2
X399	1	2	613	600	1.02	2
X400	1	2	613	600	1.02	2
X401	1	2	585	600	0.98	1
X402	1	2	633	600	1.06	2
X403	1	2	633	600	1.06	2
X404	1	2	613	600	1.02	2
X405	1	2	613	600	1.02	2
X406	1	2	633	600	1.06	2
X407	1	2	593	600	0.99	1
X408	1	2	633	600	1.06	2
X409	1	3	676	600	1.13	2
X410	1	2	613	600	1.02	2
X411	1	2	574	600	0.96	1
X412	1	2	574	600	0.96	1
X413	1	2	574	600	0.96	1
X414	1	2	574	600	0.96	1
X415	1	2	574	600	0.96	1
X416	1	2	613	600	1.02	2
X417	1	2	613	600	1.02	2
X418	1	3	654	600	1.09	2
X419	1	2	574	600	0.96	1
X420	1	2	613	600	1.02	2
X421	1	2	593	600	0.99	1
X422	1	2	574	600	0.96	1
X423	1	2	633	600	1.06	2
X424	1	3	654	600	1.09	2
X425	1	3	698	600	1.17	2
X426	1	2	574	600	0.96	1
X427	1	2	593	600	0.99	1
X428	1	2	633	600	1.06	2
X429	1	2	613	600	1.02	2
X430	1	2	593	600	0.99	1
X431	1	2	633	600	1.06	2
X432	1	2	633	600	1.06	2
X433	1	2	613	600	1.02	2
X434	1	2	613	600	1.02	2
X435	1	3	676	600	1.13	2
X436	1	2	613	600	1.02	2
X437	1	2	574	600	0.96	1
X438	1	2	633	600	1.06	2

X439	1	2	633	600	1.06	2
X440	1	2	633	600	1.06	2
X441	1	2	613	600	1.02	2
X442	1	2	613	600	1.02	2
X443	1	2	613	600	1.02	2
X444	1	2	633	600	1.06	2
X445	1	2	593	600	0.99	1
X446	1	2	574	600	0.96	1
X447	1	2	633	600	1.06	2
X448	1	2	593	600	0.99	1
X449	1	2	613	600	1.02	2
X450	1	3	654	600	1.09	2
X451	1	2	613	600	1.02	2
X452	1	2	593	600	0.99	1
X453	1	2	574	600	0.96	1
X454	1	2	633	600	1.06	2
X455	1	3	698	600	1.17	2
X456	1	2	593	600	0.99	1
X457	1	2	593	600	0.99	1
X458	1	2	633	600	1.06	2
X459	1	3	654	600	1.09	2
X460	1	2	574	600	0.96	1
X461	1	2	593	600	0.99	1
X462	1	2	613	600	1.02	2
X463	1	2	593	600	0.99	1
X464	1	2	633	600	1.06	2
X465	1	2	574	600	0.96	1
X466	1	2	593	600	0.99	1
X467	1	2	604	600	1.01	2
X468	1	2	593	600	0.99	1
X469	1	2	613	600	1.02	2
X470	1	2	593	600	0.99	1
X471	1	3	654	600	1.09	2
X472	1	3	654	600	1.09	2
X473	1	2	613	600	1.02	2
X474	1	2	574	600	0.96	1
X475	1	2	633	600	1.06	2
X476	1	2	574	600	0.96	1
X477	1	2	633	600	1.06	2
X478	1	2	613	600	1.02	2
X479	1	2	613	600	1.02	2
X480	1	2	633	600	1.06	2
X481	1	2	574	600	0.96	1
X482	1	2	593	600	0.99	1
X483	1	3	676	600	1.13	2
X484	1	2	613	600	1.02	2
X485	1	2	633	600	1.06	2
X486	1	2	613	600	1.02	2
X487	1	2	574	600	0.96	1
X488	1	3	654	600	1.09	2
X489	1	2	633	600	1.06	2
X490	1	2	593	600	0.99	1
X491	1	2	574	600	0.96	1
X492	1	3	644	600	1.08	2
X493	1	2	593	600	0.99	1
X494	1	2	613	600	1.02	2
X495	1	2	574	600	0.96	1
X496	1	2	593	600	0.99	1
X497	1	2	593	600	0.99	1
X498	1	2	613	600	1.02	2
X499	1	2	593	600	0.99	1
X500	1	2	633	600	1.06	2
X501	1	2	613	600	1.02	2
X502	1	2	574	600	0.96	1
X503	1	2	574	600	0.96	1
X504	1	2	613	600	1.02	2
X505	1	2	613	600	1.02	2
X506	1	2	613	600	1.02	2
X507	1	2	574	600	0.96	1
X508	1	2	633	600	1.06	2
X509	1	2	633	600	1.06	2
X510	1	2	613	600	1.02	2
X511	1	3	654	600	1.09	2
X512	1	2	574	600	0.96	1
X513	1	3	654	600	1.09	2
X514	1	3	676	600	1.13	2
X515	1	2	633	600	1.06	2
X516	1	2	613	600	1.02	2
X517	1	2	613	600	1.02	2
X518	1	2	593	600	0.99	1
X519	1	2	593	600	0.99	1
X520	1	2	633	600	1.06	2
X521	1	2	593	600	0.99	1
X522	1	2	613	600	1.02	2
X523	1	2	574	600	0.96	1
X524	1	2	633	600	1.06	2
X525	1	2	593	600	0.99	1
X526	1	2	593	600	0.99	1
X527	1	3	654	600	1.09	2
X528	1	2	633	600	1.06	2
X529	1	2	633	600	1.06	2
X530	1	2	574	600	0.96	1
X531	1	2	613	600	1.02	2
X532	1	2	593	600	0.99	1
X533	1	2	613	600	1.02	2
X534	1	2	633	600	1.06	2
X535	1	2	613	600	1.02	2
X536	1	2	574	600	0.96	1
X537	1	2	593	600	0.99	1
X538	1	2	574	600	0.96	1
X539	1	2	613	600	1.02	2
X540	1	3	654	600	1.09	2
X541	1	2	593	600	0.99	1
X542	1	2	593	600	0.99	1
X543	1	2	613	600	1.02	2
X544	1	3	654	600	1.09	2
X545	1	2	604	600	1.01	2
X546	1	2	613	600	1.02	2
X547	1	2	613	600	1.02	2
X548	1	2	593	600	0.99	1
X549	1	2	574	600	0.96	1
X550	1	3	654	600	1.09	2
X551	1	2	633	600	1.06	2
X552	1	2	613	600	1.02	2
X553	1	2	633	600	1.06	2
X554	1	3	665	600	1.11	2
X555	1	3	654	600	1.09	2
X556	1	2	613	600	1.02	2
X557	1	2	593	600	0.99	1
X558	1	2	613	600	1.02	2
X559	1	3	654	600	1.09	2
X560	1	2	593	600	0.99	1
X561	1	3	654	600	1.09	2
X562	1	2	593	600	0.99	1
X563	1	2	574	600	0.96	1
X564	1	2	574	600	0.96	1
X565	1	2	593	600	0.99	1
X566	1	2	613	600	1.02	2
X567	1	2	593	600	0.99	1
X568	1	2	593	600	0.99	1
X569	1	2	613	600	1.02	2
X570	1	2	574	600	0.96	1
X571	1	2	593	600	0.99	1
X572	1	2	633	600	1.06	2
X573	1	2	613	600	1.02	2
X574	1	2	593	600	0.99	1
X575	1	2	593	600	0.99	1
X576	1	2	593	600	0.99	1
X577	1	2	633	600	1.06	2
X578	1	3	654	600	1.09	2
X579	1	2	593	600	0.99	1
X580	1	2	593	600	0.99	1
X581	1	2	574	600	0.96	1
X582	1	2	633	600	1.06	2
X583	1	2	593	600	0.99	1
X584	1	3	676	600	1.13	2
X585	1	2	574	600	0.96	1
X586	1	3	654	600	1.09	2
X587	1	2	574	600	0.96	1
X588	1	2	613	600	1.02	2
X589	1	2	593	600	0.99	1
X590	1	3	676	600	1.13	2
X591	1	2	574	600	0.96	1
X592	1	2	633	600	1.06	2
X593	1	2	613	600	1.02	2
X594	1	2	574	600	0.96	1
X595	1	2	613	600	1.02	2
X596	1	2	613	600	1.02	2
X597	1	3	654	600	1.09	2
X598	1	2	574	600	0.96	1
X599	1	2	613	600	1.02	2
X600	1	2	593	600	0.99	1
X601	1	2	613	600	1.02	2
X602	1	2	613	600	1.02	2
X603	1	2	633	600	1.06	2
X604	1	2	593	600	0.99	1
X605	1	2	574	600	0.96	1
X606	1	2	574	600	0.96	1
X607	1	2	633	600	1.06	2
X608	1	2	613	600	1.02	2
X609	1	3	654	600	1.09	2
X610	1	3	654	600	1.09	2
X611	1	2	593	600	0.99	1
X612	1	2	593	600	0.99	1
X613	1	3	654	600	1.09	2
X614	1	2	574	600	0.96	1
X615	1	2	613	600	1.02	2
X616	1	3	676	600	1.13	2

X617	1	2	593	600	0.99	1
X618	1	3	676	600	1.13	2
X619	1	2	633	600	1.06	2
X620	1	2	593	600	0.99	1
X621	1	2	593	600	0.99	1
X622	1	2	634	600	1.06	2
X623	1	2	613	600	1.02	2
X624	1	3	698	600	1.17	2
X625	1	2	613	600	1.02	2
X626	1	3	644	600	1.08	2
X627	1	2	613	600	1.02	2
X628	1	2	633	600	1.06	2
X629	1	2	593	600	0.99	1
X630	1	2	574	600	0.96	1
X631	1	2	613	600	1.02	2
X632	1	2	633	600	1.06	2
X633	1	2	633	600	1.06	2
X634	1	2	613	600	1.02	2
X635	1	2	633	600	1.06	2
X636	1	2	633	600	1.06	2
X637	1	2	593	600	0.99	1
X638	1	2	574	600	0.96	1
X639	1	2	633	600	1.06	2
X640	1	2	613	600	1.02	2
X641	1	2	633	600	1.06	2
X642	1	2	613	600	1.02	2
X643	1	2	593	600	0.99	1
X644	1	2	633	600	1.06	2
X645	1	2	614	600	1.03	2
X646	1	2	593	600	0.99	1
X647	1	2	593	600	0.99	1
X648	1	2	634	600	1.06	2
X649	1	2	613	600	1.02	2
X650	1	2	634	600	1.06	2
X651	1	2	613	600	1.02	2
X652	1	2	574	600	0.96	1
X653	1	2	593	600	0.99	1
X654	1	2	593	600	0.99	1
X655	1	2	574	600	0.96	1
X656	1	2	613	600	1.02	2
X657	1	2	633	600	1.06	2
X658	1	2	593	600	0.99	1
X659	1	2	613	600	1.02	2
X660	1	2	593	600	0.99	1
X661	1	2	613	600	1.02	2
X662	1	2	613	600	1.02	2
X663	1	2	574	600	0.96	1
X664	1	2	633	600	1.06	2
X665	1	2	604	600	1.01	2
X666	1	2	574	600	0.96	1
X667	1	2	593	600	0.99	1
X668	1	2	613	600	1.02	2
X669	1	2	593	600	0.99	1
X670	1	2	613	600	1.02	2
X671	1	2	593	600	0.99	1
X672	1	2	633	600	1.06	2
X673	1	2	613	600	1.02	2
X674	1	2	613	600	1.02	2
X675	1	3	698	600	1.17	2
X676	1	2	613	600	1.02	2
X677	1	2	574	600	0.96	1
X678	1	2	574	600	0.96	1
X679	1	2	593	600	0.99	1
X680	1	2	633	600	1.06	2
X681	1	2	613	600	1.02	2
X682	1	2	593	600	0.99	1
X683	1	2	633	600	1.06	2
X684	1	2	613	600	1.02	2
X685	1	2	613	600	1.02	2
X686	1	2	593	600	0.99	1
X687	1	2	613	600	1.02	2
X688	1	2	633	600	1.06	2
X689	1	2	593	600	0.99	1
X690	1	2	593	600	0.99	1
X691	1	2	613	600	1.02	2
X692	1	2	613	600	1.02	2
X693	1	2	633	600	1.06	2
X694	1	2	574	600	0.96	1
X695	1	3	676	600	1.13	2
X696	1	2	613	600	1.02	2
X697	1	2	613	600	1.02	2
X698	1	2	613	600	1.02	2
X699	1	2	593	600	0.99	1
X700	1	2	593	600	0.99	1
X701	1	2	633	600	1.06	2
X702	1	2	593	600	0.99	1
X703	1	2	613	600	1.02	2
X704	1	2	613	600	1.02	2
X705	1	2	613	600	1.02	2
X706	1	2	613	600	1.02	2
X707	1	2	574	600	0.96	1
X708	1	2	633	600	1.06	2
X709	1	2	613	600	1.02	2
X710	1	2	593	600	0.99	1
X711	1	2	593	600	0.99	1
X712	1	2	613	600	1.02	2
X713	1	3	698	600	1.17	2
X714	1	2	613	600	1.02	2
X715	1	2	593	600	0.99	1
X716	1	2	593	600	0.99	1
X717	1	2	633	600	1.06	2
X718	1	2	633	600	1.06	2
X719	1	2	613	600	1.02	2
X720	1	2	633	600	1.06	2
X721	1	3	656	600	1.10	2
X722	1	2	593	600	0.99	1
X723	1	2	604	600	1.01	2
X724	1	2	574	600	0.96	1
X725	1	2	593	600	0.99	1
X726	1	2	593	600	0.99	1
X727	1	2	574	600	0.96	1
X728	1	2	613	600	1.02	2
X729	1	2	593	600	0.99	1
X730	1	2	613	600	1.02	2
X731	1	2	593	600	0.99	1
X732	1	2	633	600	1.06	2
X733	1	3	654	600	1.09	2
X734	1	2	613	600	1.02	2
X735	1	2	613	600	1.02	2
X736	1	2	613	600	1.02	2
X737	1	2	574	600	0.96	1
X738	1	2	633	600	1.06	2
X739	1	2	593	600	0.99	1
X740	1	2	613	600	1.02	2
X741	1	3	654	600	1.09	2
X742	1	3	698	600	1.17	2
X743	1	3	676	600	1.13	2
X744	1	2	633	600	1.06	2
X745	1	2	593	600	0.99	1
X746	1	2	613	600	1.02	2
X747	1	2	613	600	1.02	2
X748	1	2	593	600	0.99	1
X749	1	2	593	600	0.99	1
X750	1	2	574	600	0.96	1
X751	1	2	593	600	0.99	1
X752	1	2	574	600	0.96	1
X753	1	2	613	600	1.02	2
X754	1	2	613	600	1.02	2
X755	1	2	593	600	0.99	1
X756	1	2	613	600	1.02	2
X757	1	2	593	600	0.99	1
X758	1	2	613	600	1.02	2
X759	1	2	593	600	0.99	1
X760	1	2	613	600	1.02	2
X761	1	2	613	600	1.02	2
X762	1	2	613	600	1.02	2
X763	1	2	593	600	0.99	1
X764	1	2	574	600	0.96	1
X765	1	2	613	600	1.02	2
X766	1	2	613	600	1.02	2
X767	1	2	613	600	1.02	2
X768	1	2	593	600	0.99	1
X769	1	2	633	600	1.06	2
X770	1	2	613	600	1.02	2
X771	1	2	613	600	1.02	2
X772	1	2	613	600	1.02	2
X773	1	2	633	600	1.06	2
X774	1	2	593	600	0.99	1
X775	1	2	633	600	1.06	2
X776	1	2	593	600	0.99	1
X777	1	2	613	600	1.02	2
X778	1	2	633	600	1.06	2
X779	1	3	654	600	1.09	2
X780	1	3	654	600	1.09	2
X781	1	2	574	600	0.96	1
X782	1	3	721	600	1.21	2
X783	1	2	593	600	0.99	1
X784	1	3	654	600	1.09	2
X785	1	2	613	600	1.02	2
X786	1	3	654	600	1.09	2
X787	1	2	593	600	0.99	1
X788	1	3	654	600	1.09	2
X789	1	2	574	600	0.96	1
X790	1	2	613	600	1.02	2
X791	1	2	574	600	0.96	1
X792	1	2	633	600	1.06	2
X793	1	3	654	600	1.09	2
X794	1	2	613	600	1.02	2

X795	1	2	613	600	1.02	2
X796	1	2	574	600	0.96	1
X797	1	2	593	600	0.99	1
X798	1	2	613	600	1.02	2
X799	1	2	574	600	0.96	1
X800	1	2	633	600	1.06	2
X801	1	2	633	600	1.06	2
X802	1	2	574	600	0.96	1
X803	1	2	593	600	0.99	1
X804	1	2	574	600	0.96	1
X805	1	2	593	600	0.99	1
X806	1	2	593	600	0.99	1
X807	1	2	593	600	0.99	1
X808	1	3	654	600	1.09	2
X809	1	3	676	600	1.13	2
X810	1	2	593	600	0.99	1
X811	1	2	633	600	1.06	2
X812	1	2	593	600	0.99	1
X813	1	3	676	600	1.13	2
X814	1	3	676	600	1.13	2
X815	1	2	613	600	1.02	2
X816	1	2	613	600	1.02	2
X817	1	2	633	600	1.06	2
X818	1	2	613	600	1.02	2
X819	1	2	593	600	0.99	1
X820	1	2	613	600	1.02	2
X821	1	3	676	600	1.13	2
X822	1	2	593	600	0.99	1
X823	1	2	593	600	0.99	1
X824	1	2	613	600	1.02	2
X825	1	2	613	600	1.02	2
X826	1	2	593	600	0.99	1
X827	1	3	654	600	1.09	2
X828	1	3	654	600	1.09	2
X829	1	2	633	600	1.06	2
X830	1	2	633	600	1.06	2
X831	1	2	613	600	1.02	2
X832	1	2	593	600	0.99	1
X833	1	2	613	600	1.02	2
X834	1	2	593	600	0.99	1
X835	1	2	634	600	1.06	2
X836	1	2	593	600	0.99	1
X837	1	2	593	600	0.99	1
X838	1	3	676	600	1.13	2
X839	1	2	613	600	1.02	2
X840	1	2	613	600	1.02	2
X841	1	2	574	600	0.96	1
X842	1	2	593	600	0.99	1
X843	1	2	613	600	1.02	2
X844	1	2	613	600	1.02	2
X845	1	2	574	600	0.96	1
X846	1	2	613	600	1.02	2
X847	1	2	574	600	0.96	1
X848	1	2	593	600	0.99	1
X849	1	2	633	600	1.06	2
X850	1	2	593	600	0.99	1
X851	1	2	593	600	0.99	1
X852	1	2	593	600	0.99	1
X853	1	2	593	600	0.99	1
X854	1	3	656	600	1.10	2
X855	1	2	593	600	0.99	1
X856	1	2	593	600	0.99	1
X857	1	2	633	600	1.06	2
X858	1	2	613	600	1.02	2
X859	1	2	574	600	0.96	1
X860	1	2	613	600	1.02	2
X861	1	2	613	600	1.02	2
X862	1	2	633	600	1.06	2
X863	1	2	613	600	1.02	2
X864	1	2	633	600	1.06	2
X865	1	2	633	600	1.06	2
X866	1	2	633	600	1.06	2
X867	1	2	593	600	0.99	1
X868	1	2	613	600	1.02	2
X869	1	2	633	600	1.06	2
X870	1	2	613	600	1.02	2
X871	1	3	654	600	1.09	2
X872	1	2	634	600	1.06	2
X873	1	2	633	600	1.06	2
X874	1	2	633	600	1.06	2
X875	1	2	593	600	0.99	1
X876	1	2	613	600	1.02	2
X877	1	2	613	600	1.02	2
X878	1	2	613	600	1.02	2
X879	1	2	593	600	0.99	1
X880	1	3	654	600	1.09	2
X881	1	2	593	600	0.99	1
X882	1	2	593	600	0.99	1
X883	1	2	593	600	0.99	1
X884	1	2	574	600	0.96	1
X885	1	3	654	600	1.09	2
X886	1	2	574	600	0.96	1
X887	1	2	574	600	0.96	1
X888	1	2	633	600	1.06	2
X889	1	2	574	600	0.96	1
X890	1	2	613	600	1.02	2
X891	1	3	656	600	1.10	2
X892	1	2	613	600	1.02	2
X893	1	2	574	600	0.96	1
X894	1	2	574	600	0.96	1
X895	1	2	593	600	0.99	1
X896	1	2	593	600	0.99	1
X897	1	2	593	600	0.99	1
X898	1	2	566	600	0.95	1
X899	1	2	613	600	1.02	2
X900	1	2	613	600	1.02	2
X901	1	2	574	600	0.96	1
X902	1	2	574	600	0.96	1
X903	1	3	676	600	1.13	2
X904	1	2	593	600	0.99	1
X905	1	2	633	600	1.06	2
X906	1	3	676	600	1.13	2
X907	1	2	613	600	1.02	2
X908	1	2	593	600	0.99	1
X909	1	2	633	600	1.06	2
X910	1	2	613	600	1.02	2
X911	1	2	574	600	0.96	1
X912	1	2	574	600	0.96	1
X913	1	2	633	600	1.06	2
X914	1	2	613	600	1.02	2
X915	1	2	613	600	1.02	2
X916	1	2	574	600	0.96	1
X917	1	3	654	600	1.09	2
X918	1	2	613	600	1.02	2
X919	1	2	574	600	0.96	1
X920	1	3	656	600	1.10	2
X921	1	2	633	600	1.06	2
X922	1	2	633	600	1.06	2
X923	1	3	676	600	1.13	2
X924	1	2	613	600	1.02	2
X925	1	2	613	600	1.02	2
X926	1	2	604	600	1.01	2
X927	1	3	654	600	1.09	2
X928	1	3	654	600	1.09	2
X929	1	2	613	600	1.02	2
X930	1	2	593	600	0.99	1
X931	1	2	613	600	1.02	2
X932	1	2	593	600	0.99	1
X933	1	2	613	600	1.02	2
X934	1	2	613	600	1.02	2
X935	1	3	721	600	1.21	2
X936	1	3	676	600	1.13	2
X937	1	2	613	600	1.02	2
X938	1	3	654	600	1.09	2
X939	1	2	593	600	0.99	1
X940	1	3	654	600	1.09	2
X941	1	2	613	600	1.02	2
X942	1	2	593	600	0.99	1
X943	1	2	613	600	1.02	2
X944	1	2	593	600	0.99	1
X945	1	2	633	600	1.06	2
X946	1	3	654	600	1.09	2
X947	1	2	633	600	1.06	2
X948	1	2	593	600	0.99	1
X949	1	2	613	600	1.02	2
X950	1	2	633	600	1.06	2
X951	1	2	593	600	0.99	1
X952	1	3	654	600	1.09	2
X953	1	2	613	600	1.02	2
X954	1	2	593	600	0.99	1
X955	1	2	574	600	0.96	1
X956	1	3	654	600	1.09	2
X957	1	2	613	600	1.02	2
X958	1	2	613	600	1.02	2
X959	1	2	593	600	0.99	1
X960	1	2	593	600	0.99	1
X961	1	2	613	600	1.02	2
X962	1	2	613	600	1.02	2
X963	1	2	633	600	1.06	2
X964	1	2	593	600	0.99	1
X965	1	2	593	600	0.99	1
X966	1	2	633	600	1.06	2
X967	1	2	593	600	0.99	1
X968	1	2	633	600	1.06	2
X969	1	2	633	600	1.06	2
X970	1	2	633	600	1.06	2
X971	1	2	633	600	1.06	2
X972	1	2	633	600	1.06	2

X973	1	2	593	600	0.99	1
X974	1	2	593	600	0.99	1
X975	1	2	633	600	1.06	2
X976	2	8	1638	600	2.74	3
X977	1	2	574	600	0.96	1
X978	1	2	574	600	0.96	1
X979	1	3	654	600	1.09	2
X980	1	2	593	600	0.99	1
X981	1	2	633	600	1.06	2
X982	1	2	613	600	1.02	2
X983	1	2	613	600	1.02	2
X984	1	2	574	600	0.96	1
X985	1	2	574	600	0.96	1
X986	1	3	654	600	1.09	2
X987	1	3	654	600	1.09	2
X988	1	2	613	600	1.02	2
X989	1	2	613	600	1.02	2
X990	1	2	633	600	1.06	2
X991	1	2	574	600	0.96	1
X992	1	3	654	600	1.09	2
X993	1	2	613	600	1.02	2
X994	1	2	613	600	1.02	2
X995	1	2	633	600	1.06	2
X996	1	2	574	600	0.96	1
X997	1	2	613	600	1.02	2
X998	1	2	633	600	1.06	2
X999	1	2	634	600	1.06	2
X1000	1	2	593	600	0.99	1
X1001	1	2	593	600	0.99	1
X1002	1	2	633	600	1.06	2
X1003	1	2	593	600	0.99	1
X1004	1	2	593	600	0.99	1
X1005	1	2	633	600	1.06	2
X1006	1	2	613	600	1.02	2
X1007	1	2	593	600	0.99	1
X1008	1	2	613	600	1.02	2
X1009	1	2	574	600	0.96	1
X1010	1	2	613	600	1.02	2
X1011	1	2	633	600	1.06	2
X1012	1	2	613	600	1.02	2
X1013	1	2	593	600	0.99	1
X1014	1	2	574	600	0.96	1
X1015	1	2	613	600	1.02	2
X1016	1	2	593	600	0.99	1
X1017	1	2	613	600	1.02	2
X1018	1	2	574	600	0.96	1
X1019	1	2	633	600	1.06	2
X1020	1	2	633	600	1.06	2
X1021	1	2	593	600	0.99	1
X1022	1	2	613	600	1.02	2
X1023	1	2	613	600	1.02	2
X1024	1	2	633	600	1.06	2
X1025	1	2	613	600	1.02	2
X1026	1	2	593	600	0.99	1
X1027	1	2	593	600	0.99	1
X1028	1	2	613	600	1.02	2
X1029	1	2	633	600	1.06	2
X1030	1	3	654	600	1.09	2
X1031	1	2	633	600	1.06	2
X1032	1	2	633	600	1.06	2
X1033	1	2	613	600	1.02	2
X1034	1	2	574	600	0.96	1
X1035	1	2	593	600	0.99	1
X1036	1	2	593	600	0.99	1
X1037	1	2	633	600	1.06	2
X1038	1	3	676	600	1.13	2
X1039	1	2	574	600	0.96	1
X1040	1	2	613	600	1.02	2
X1041	1	2	633	600	1.06	2
X1042	1	2	593	600	0.99	1
X1043	1	2	574	600	0.96	1
X1044	1	2	613	600	1.02	2
X1045	1	2	593	600	0.99	1
X1046	1	2	613	600	1.02	2
X1047	1	3	698	600	1.17	2
X1048	1	2	613	600	1.02	2
X1049	1	2	574	600	0.96	1
X1050	1	2	585	600	0.98	1
X1051	1	3	676	600	1.13	2
X1052	1	3	654	600	1.09	2
X1053	1	2	613	600	1.02	2
X1054	1	2	574	600	0.96	1
X1055	1	2	593	600	0.99	1
X1056	1	2	593	600	0.99	1
X1057	1	2	633	600	1.06	2
X1058	1	2	613	600	1.02	2
X1059	1	2	613	600	1.02	2
X1060	1	2	593	600	0.99	1
X1061	1	2	593	600	0.99	1
X1062	1	2	633	600	1.06	2
X1063	1	2	613	600	1.02	2
X1064	1	2	613	600	1.02	2
X1065	1	2	613	600	1.02	2
X1066	1	2	574	600	0.96	1
X1067	1	2	593	600	0.99	1
X1068	1	2	593	600	0.99	1
X1069	1	2	613	600	1.02	2
X1070	1	3	676	600	1.13	2
X1071	1	2	593	600	0.99	1
X1072	1	2	593	600	0.99	1
X1073	1	2	593	600	0.99	1
X1074	1	2	613	600	1.02	2
X1075	1	2	633	600	1.06	2
X1076	1	2	613	600	1.02	2
X1077	1	2	613	600	1.02	2
X1078	1	2	633	600	1.06	2
X1079	1	2	574	600	0.96	1
X1080	1	2	574	600	0.96	1
X1081	1	2	593	600	0.99	1
X1082	1	2	633	600	1.06	2
X1083	1	2	574	600	0.96	1
X1084	1	2	593	600	0.99	1
X1085	1	2	593	600	0.99	1
X1086	1	2	613	600	1.02	2
X1087	1	2	633	600	1.06	2
X1088	1	2	613	600	1.02	2
X1089	1	2	593	600	0.99	1
X1090	1	2	633	600	1.06	2
X1091	1	2	593	600	0.99	1
X1092	1	2	574	600	0.96	1
X1093	1	2	633	600	1.06	2
X1094	1	2	633	600	1.06	2
X1095	1	2	593	600	0.99	1
X1096	1	2	613	600	1.02	2
X1097	1	2	593	600	0.99	1
X1098	1	2	593	600	0.99	1
X1099	1	2	613	600	1.02	2
X1100	1	2	613	600	1.02	2
X1101	1	2	633	600	1.06	2
X1102	1	2	593	600	0.99	1
X1103	1	2	613	600	1.02	2
X1104	1	2	593	600	0.99	1
X1105	1	3	676	600	1.13	2
X1106	1	2	613	600	1.02	2
X1107	1	2	593	600	0.99	1
X1108	1	2	613	600	1.02	2
X1109	1	2	613	600	1.02	2
X1110	1	2	593	600	0.99	1
X1111	1	2	613	600	1.02	2
X1112	1	2	574	600	0.96	1
X1113	1	2	574	600	0.96	1
X1114	1	2	593	600	0.99	1
X1115	1	2	593	600	0.99	1
X1116	1	2	593	600	0.99	1
X1117	1	2	574	600	0.96	1
X1118	1	2	574	600	0.96	1
X1119	1	2	593	600	0.99	1
X1120	1	2	613	600	1.02	2
X1121	1	2	593	600	0.99	1
X1122	1	2	593	600	0.99	1
X1123	1	2	593	600	0.99	1
X1124	1	2	633	600	1.06	2
X1125	1	2	593	600	0.99	1
X1126	1	2	593	600	0.99	1
X1127	1	2	593	600	0.99	1
X1128	1	2	574	600	0.96	1
X1129	1	2	613	600	1.02	2
X1130	1	2	593	600	0.99	1
X1131	1	2	613	600	1.02	2
X1132	1	3	698	600	1.17	2
X1133	1	2	613	600	1.02	2
X1134	1	2	613	600	1.02	2
X1135	1	2	574	600	0.96	1
X1136	1	2	574	600	0.96	1
X1137	1	2	633	600	1.06	2
X1138	1	3	654	600	1.09	2
X1139	1	2	633	600	1.06	2
X1140	1	3	654	600	1.09	2
X1141	1	2	593	600	0.99	1
X1142	1	2	574	600	0.96	1
X1143	1	2	574	600	0.96	1
X1144	1	4	746	600	1.25	2
X1145	1	2	574	600	0.96	1
X1146	1	2	574	600	0.96	1
X1147	1	2	613	600	1.02	2
X1148	1	3	654	600	1.09	2
X1149	1	2	593	600	0.99	1
X1150	1	2	633	600	1.06	2

X1151	1	2	593	600	0.99	1	X1158	1	2	613	600	1.02	2
X1152	1	2	574	600	0.96	1	X1159	1	2	613	600	1.02	2
X1153	1	2	613	600	1.02	2	X1160	1	2	613	600	1.02	2
X1154	1	2	613	600	1.02	2	X1161	1	2	593	600	0.99	1
X1155	1	2	633	600	1.06	2	X1162	1	2	613	600	1.02	2
X1156	1	2	574	600	0.96	1	X1163	1	3	654	600	1.09	2
X1157	1	2	613	600	1.02	2							

Appendix C

Time Series Plots

Figure C.1: Time series plots for selected $\mu_{\beta_{h,j}}$'s

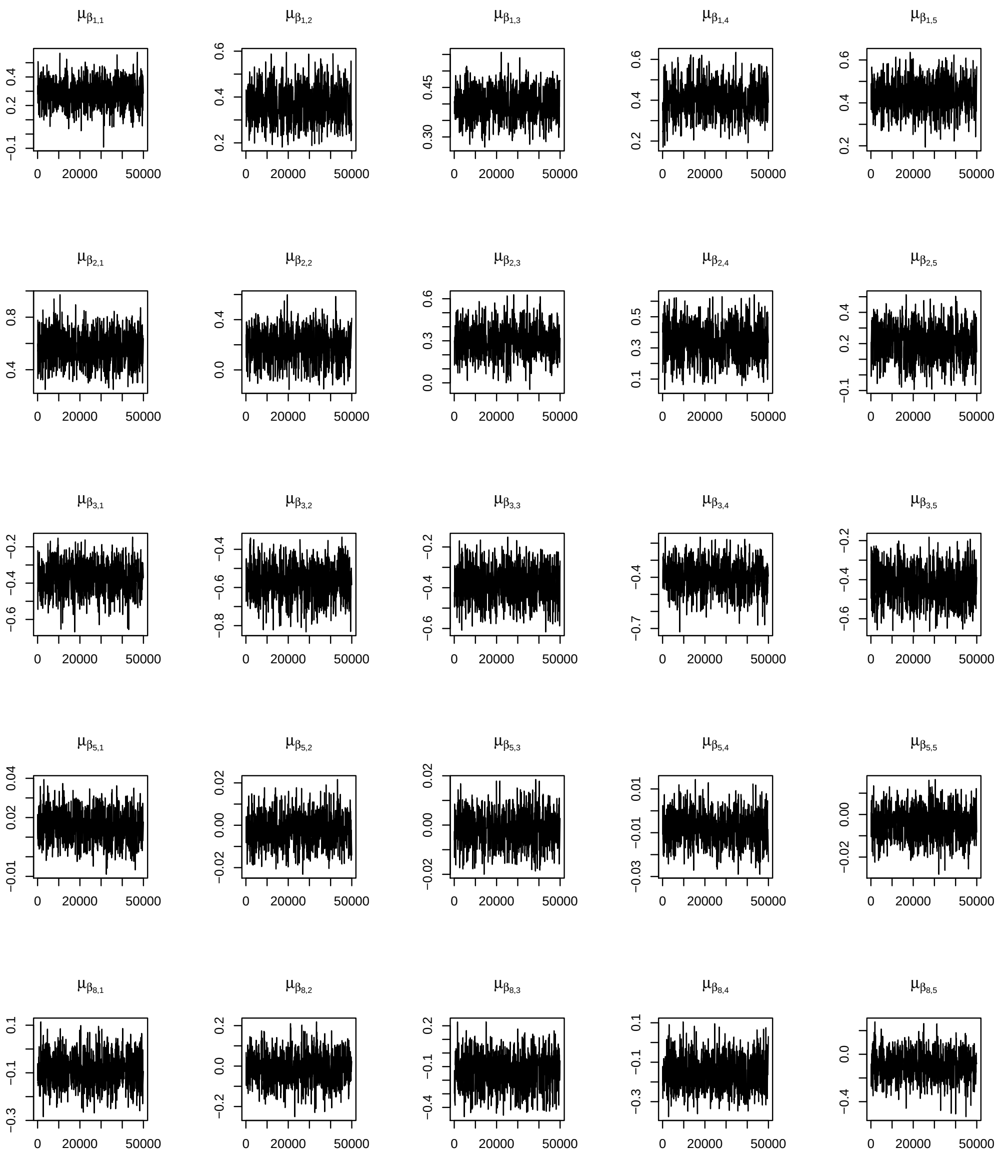
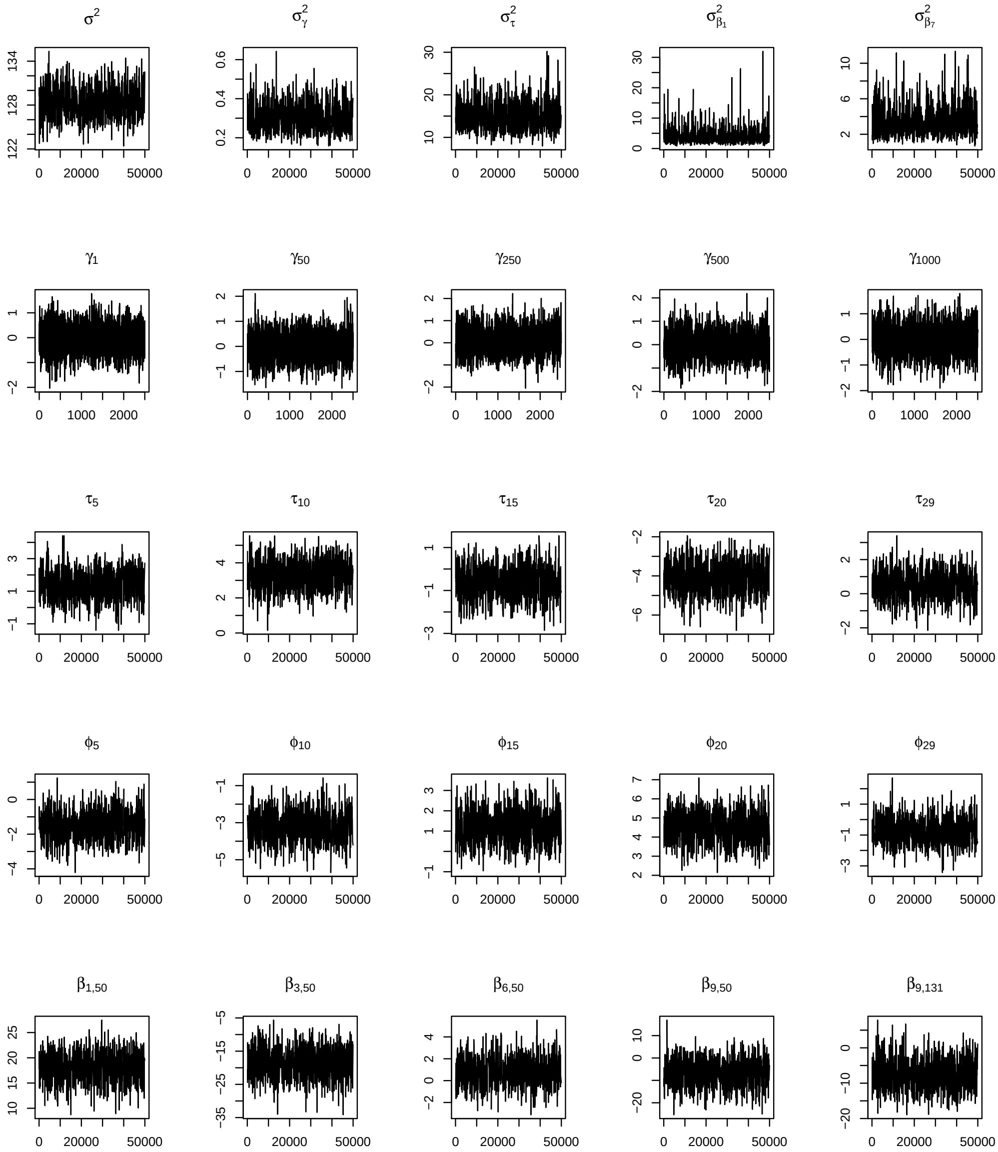


Figure C.2: Time series plots for selected variance, team, game, and regression coefficient parameters



Bibliography

Bayes, T. (1764), "An Essay Towards Solving a Problem in the Doctrine of Chances," *Philosophical Transanctions of the Royal Society of London*, 53, 370–418.

Berry, S. M. (2001), "Do You Feel a Draft in Here," *Chance*, 14, 53–57.

Berry, S. M., Reese, C. S., and Larkey, P. D. (1999), "Bridging Different Eras in Sports," *Journal of the American Statistical Association*, 94, 661–686.

Bishop, T. and Gajewski, B. J. (2004), "Drafting a Career in Sports: Determining Underclassmen College Players' Stock in the NBA Draft," *chance*, 17, 9–12.

Bray, S. R. and Brawley, L. R. (2002), "Role Efficacy, Role Clarity, and Role Performance Effectiveness," *Small Group Research*, 33, 233–253.

Chen, M.-H. and Deely, J. J. (1996), "Bayesian Analysis for a Constrained Linear Multiple Regression Problem for Prediction the New Crop of Apples," *Journal of Agricultural, Biological, and Environmental Statistics*, 1, 467–489.

Gelfand, A. and Smith, A. F. M. (1990), "Sampling-based Approaches to Calculationg Marginal Densities," *Journal of the American Statistical Association*, 85, 398–409.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis*, New York: Chapman and Hall/ CRC, 2nd ed.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and maching Intelligence*, 6, 721–741.

Graves, T., Reese, C. S., and Fitzgerald, M. (2003), "Hierarchical Models for Premutations: Analysis of Auto Racing Results," *Journal of the American Statistical Association*, 98, 282–291.

Hartley, D. (1749), *Observations on Man, his Frame, his Duty, and his Expectations*, London: Richardson.

Hastings, W. K. (1970), "Monte Carlo Methods Using Markov Chains and their Applications," *Biometrika*, 57, 97–109.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998), "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," *The American Statistician*, 52, 93–100.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953), "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1091.

Raftery, A. E. and Lewis, S. (1992), *Bayesian Statistics*, Oxford, chap. How Many Iterations in the Gibbs Sampler?, pp. 765–776.

Reese, C. S., Calvin, J. A., George, J. C., and Tarpley, R. J. (2001), "Estimation of Fetal growth and Gestation in Bowhead Whales," *Journal of the American Statistical Association*, 96, 915–938.

Schafer, G. (1982), "Bayes Two Arguements For the Rules of Conditioning," *The Annals of Statistics*, 10, 1075–1089.

Smith, A. (1973), "A General Bayesian Linear Model," *Journal of the Royal Statistical Society*, 35, 67–75.

Stern, H. S. (1998), "College Basketball Upsets: Will a 16-seed Ever Beat a 1-seed?" *Chance*, 11, 26–31.

Stigler, S. M. (1982), "Thomas Bayes's Bayesian Inference," *Journal of the Royal Statistical Society*, 145, 250–258.

— (1983), "Who Discovered Bayes's Theorem?" *The American Statistician*, 37, 290–296.

— (1986), "Laplace's 1774 Memoir on Inverse Probability," *Statistical Science*, 1, 359–363.

Tversky, A. and Gilovich, T. (1989), "The Cold Facts About the "Hot Hand" in Basketball," *Chance*, 2, 16–21.

White, C. and Berry, S. (2002), "Tiered Polychotomous Regression: Ranking NFL Quarterbacks," *The American Statistician*, 56, 10–21.

Zellner, A. (1999), "Bayesian Analysis of Golf," Tech. Rep. 904, University of California at Berkeley, Department of Agricultural and Resource Economics and Policy, http://ideas.repec.org/p/are/cudare/904.html.