Kevin H(Y) Shu (kshu)
36-350
15 December 2014
Final Project Report

1 Introduction

The project this group undertook examined all 548 983 play-by-play events of the 2005–06 NBA season, originally compiled by Aaron Barzilai of basketballvalue.com. The events, all of which were listed in a text file, were parsed and aggregated into 1320 data frames corresponding to each match of the season. Then, for each player, the exact time at which he was present on the court was ascertained in order to aid in calculating the primary statistic of interest, the Plus/Minus, defined as the number of points his team scores during his presence on the court minus the number of points he allows the opposing team to score while still present on the court. The reader is advised to exercise patience in the running of this group's code, due to the presence of multiple data values in excess of 100 MB in size.

Because a player only needs to be present on the court during a scoring play, he need not be the one scoring nor surrendering the points. Those with large positive Plus/Minus statistics are generally well-rounded players, but not necessarily superstars; large positive Plus/Minus statistics may be explained by frequent assists and rebounds (allowing teammates to score), or good lockdown defence against the opposing team, minimising their gains. Thus Plus/Minus can be as much of an individual statistic as it is a team statistic, and discretion should be used in interpreting it.

2 Methods

2.1 Data extraction

The text file `playbyplay.txt` contained nearly 549 000 lines, each corresponding to a particular play of the 2005–06 NBA season and containing four elements in a more or less tabular format. These four elements were then parsed into the first six variables (bulleted list below) in the initial `dats` dataframe. The next task was to determine the plays during which either team was awarded points, and fill in the last five variables (bulleted list below) accordingly. Of note was the 10 March 2006 match between the Los Angeles Lakers and San Antonio Spurs, where the San Antonio Spurs' Sean Marks' last scoring play had appeared in `playbyplay.txt` over 90 000 lines after the play denoting the game's end appeared. This error had caused an anomaly larger than the combined final score of the two teams during that match.

- `gameID`: The unique identifier for each game, consisting of the date of the match (in YYYYMMDD format), followed by the three-letter code for each participating team
- `date`: The date the match occurred, in YYYYMMDD format
- `away`: The three-letter code for the away team
- `home`: The three-letter code for the home team
- `index`: The play number with respect to a particular match, with higher indices occurring later in their matches
- `time.remaining`: The time remaining (in HH:MM:SS) at each play
- `awaypts`: The number of points scored by the away team in this play (non-cumulative differential)
- `homepts`: The number of points scored by the away team in this play (non-cumulative differential)
- `scoring.team`: The three-letter code for the team that made this scoring play; `NA` if this play did not score.
- `scoring.team.score`: The cumulative points scored by this team as of this play; 0 if this play did not score.

- `scoring.team.is.home`: Whether the scoring team is the home team (`TRUE` or `FALSE`); `NA` if this play did not score.

The next critical step was to obtain a roster of players for each of the 30 NBA teams. First, the set of lines with a team-specific play was partitioned by each team, and then, within each partition, the players' surnames were extracted; although only the surnames immediately following the team code marker (e.g. `[WAS]`) were considered, the likelihood of a team member being missed due to this method of extraction was considered exceedingly low, if not zero. The only considerable hassles at this juncture were the presence of plays where coaches would induce a "technical foul" by, for instance arguing with the referee, as well as the possible presence of nouns which are not part of a player's surname, but are basketball parlance, such as "turnover" or "layup". All in all, there were 514 unique NBA players listed during the 2005–06 season, and while some athletes had competed for multiple teams during that season, this consideration was ultimately unimportant to the calculation of individual Plus/Minus statistics.

The other step in data extraction that preceded any data analysis was the compiling of information on substitutions. With no widespread irregularities found, this step was significantly less involved than obtaining each team's roster, and was a matter of simple string parsing. The information on the substitutions was compiled into a separate data frame called `rotated`, with the following columns:

- `index`: The index/line number with respect to the source text at which this substitution occurred
- `team`: The team of the rotating players
- `subst.player`: The member that is coming *on* to the court during this play; his presence on the court begins at exactly this play.
- `replaced.player`: The member that is coming *off* the court during this play; his presence on the court ends at the play immediately preceding this.

Finally, the main data frame `dats` was split into a list, named `split.game` of 1230 data frames corresponding to each match, in order to both ease testing and compute game-by-game statistics.

2.2 Pre Data Analysis

Although the anomalies in the data extraction stage, they were not remotely as consistent as the inconsistency between game ID and play description in the following four teams' three-letter codes:

Team	**Code in Game ID**	**Code in play description**
San Antonio Spurs	SAS	SAN
Utah Jazz	UTA	UTH
Golden State Warriors	GSW	GOS
Philadelphia 76ers	PHI	PHL

The conversion from the codes used in the Game IDs to those used in the play descriptions was essential to complete what was, by far, the most computationally substantial function, `on.court.substitutions`, the task of which was, in essence, to determine for all plays of a game the (at most five) players that were present on the court.

At this stage, the group project members decided it was highly beneficial to separate this information into two data frames corresponding to the home team and away team, `psh` and `psa`, respectively; `psh` and `psa` had several dozen columns each, corresponding to indicator variables for the presence of each player on said team. First, for each team member, the indices of the plays in which he was involved in were determined, and the corresponding indices in `psh` and `psa` were filled in with 1 to indicate his presence on the court at the time.

Having completed the easier task of `on.court.substitutions`, it was now time to account for the substitutions. Each match is divided into quarters that last 12 minutes each (5 minutes during overtime); the handful of exceptions where the plays explicitly demarcating the end of periods were

badly misplaced or absent altogether, a fault of Barzilai's, were handled by hand. At the end of each quarter, teams may rotate their active (meaning present on the court) members without *any* such explicit indication of specific players given in the play-by-play file. This gives rise to the potential of some players to have *no* rotations during a specific quarter, and the following assumptions were made:

- If the teammate is involved in *any* plays during that quarter, he is deemed *present* for the entire period.
- If the teammate is involved in *no* plays during that quarter, he is deemed *absent* (is a back-burner) for the entire period.

With these considerations in mind, the previously mentioned data frame with substitution information, `rotated`, was especially pertinent to the task at hand. It was then partitioned based on the quarters, and for each partition, the sub-function `subst.fill` would finally fill in the columns of `psh` and `psa`, using the following branches:

- Player was a substitute but not replaced $\implies$: Absent from beginning of quarter up until but not including the time of substitution, and present beginning at time of substitution until end of quarter.
- Player was not a substitute but was replaced $\implies$: Present from beginning of quarter up until but not including the time of replacement, and absent beginning at time of replacement until end of quarter.
- Player was both a substitute and was replaced $\implies$:
  - If the first substitution occurred *prior to* the first replacement, the player is absent up until but not including the time of the first substitution, and present up until but not including the time of the first replacement, and toggles on and off accordingly.
  - If the first substitution occurred *after* the first replacement, the player is present up until but not including the time of the first replacement, and absent up until but not including the time of the first substitution, and toggles on and off accordingly.

The remaining anomalies not considered to be an input error on the part of Barzilai were:

- Match began later than the 36-minute mark: No matches were found to begin at or after the 24-minute mark, so in these cases, there would only be three quarters in non-overtime.
- Match went overtime: Since the number of overtime minutes was quite varied from match to match, the ends (and hence number) of overtime periods were found using a different regular expression (`"End Period"`). This was the only substantive difference from the four (or three) quarters during normal time.

Having finally built `psh` and `psa`, the end result was, for the sake of convenience later on, to include them separately in a list, the return type of `on.court.substitutions`. This function was later called on the list `split.game` to produce a list, called `games.filled` with $2 \times 1230 = 2460$ data frames, each an augmented version (the extra columns being presence-on-court indicator variables for each team member) of `dats` corresponding to the away or home team for each game. **An excerpt of a constituent data frame is given below**:

2.3 Analysis

After determining for each player which times they were on the court, it became possible to calculate his seasonal Plus/Minus statistic, defined as the number of points his team scores during his presence minus the number of points the opposing team scores during his presence, and the total time he spent on the court for the season. Given the aforementioned extraordinarily large data frame `games.filled`, the function `calculateStats` essentially compiled the information contained therein: for each match, and for each team member, said member's match total Plus/Minus statistic and time (in minutes) spent on the court was calculated and reported in a single data frame, `final.stats`.

3 Peer Assessment

Evan and I had begun the coding in earnest on the night of Sunday, 30 November, and additionally worked together the nights of the 2nd and 4th of December, albeit with significant hurdles due to a shortfall in foresight on how to best acquire the roster for each team. I believe that during this formative three-night window he and I formed complementary roles: out of the three of us, he is by far the basketball guru, while I addressed a nontrivial amount of coding issues on his end. The foresight on building team rosters that was badly needed could have come before 5 December had Emmie joined us on one of those three nights, especially the latter two.

Emmie's late entry may be understandable in the context of her busy last week of classes, with two assignments in 36-401, not to mention other courses. Even on Friday the 5th, the first day she was available to meet with Evan and me, she told me that she could not find the e-mail from Prof. Shalizi with our TXT file, and fell even further behind due to a sorority event that evening. Thus, given the hurdle Evan and I faced over compiling the team rosters, I cite her late entry as the foremost or 2nd main reason the three of us had to pull an all-nighter the night of Saturday the 6th. However, her catch-up work, especially her ability to independently write the Analysis functions independently of my work on `on.court.substitutions`, during the afternoon of the following day, Saturday the 6th, is to be commended.

Overall, I would summarise the final contributions as follows:

- me: Entirety of the code section entitled `#SUBSTITUTIONS#` (ending with the creation of the data frame `rotated`); vast majority of code chunk dominated by `on.court.substitutions`; improved `parse.time`, `group.index`, `play.time.length`.
- Evan: Outer two columns of the poster and proofread the middle; he is the ultimate author of the code for the initial data extraction from the first three out of four columns in the TXT; together with Emmie, wrote the functions to calculate the Plus/Minus statistic and the initial code to calculate the total amount of time each player spent on the court.

- Emmie: Middle column of the poster; together with Evan, wrote the functions to calculate the Plus/Minus statistic and the initial code to calculate the total amount of time each player spent on the court. Most of the comments on the code are hers.