The value of a basketball player

J.A.Hartigan Yale University
September 2013

1 Money

In 2012-2013, the top salary in the National Baseball Association was 30 million dollars for Kobe Bryant, Los Angeles Lakers; the top ten player each receive more than 18 million dollars. The starting salary for a rookie player was 0.5 million dollars.

The EWA(Estimated Wins Added) is a statistic developed by John Hollinger that puts the price of a win at 1.6 million dollars, and estimates for each player how many wins he adds to a team. His salary is regarded as good for the team if the number of wins added multiplied by 1.6 million exceeds his salary. See

http://www.sonicscentral.com/statsite.html

as a source for many different theoretical approaches to basketball. Many of them are based on "play by play" statistics, the basketball events that occur during each "matchup" between a particular 5 players on one team with a particular 5 players on the other team. The play by play data is available up to the end of the 2012 season at

http://basketballvalue.com/downloads.php

We develop a value measure predicting each player's contribution to the point difference between his team's score and the opponent's team score, after adjusting for home court advantage and possession advantage. This statistic is routinely produced for evaluating players, for example, at http://www.82games.com/1112/1112NYK1.HTM.

We copy and paste the following partial table for 6 March 2012. The production scores are for a 48 minute game. For example, Lin played 32% of the time in the games up till 6 March 2012. His Net48 is 7.8 more points for his team than for the opponents team per 48 minutes played.

Player	Minutes	Net48
Anthony	51%	+0.7
Novak	22%	+14.5
Jeffries	32%	+1.1
Lin	31%	+7.8
Shumpert	50%	+0.5
Fields	64%	+3.7

2 Estimating player value

We assume that each player on the floor makes an additive contribution to the difference in team scores in each minute of play. There are five players from each team, so the away team – away team score is estimated by the sum of 5 players values from the away team less the sum of 5 players values from the home team. In addition, a term is added to the minute score difference for the away team in possession of the ball, and for the home court advantage of the home team.

There are many substitutions, so to determine these player values, we need to know the player on the floor at any time. The data will come with one row of data for each interval of time between substitutions in which the same sets of 5 players are on the floor from both teams. The estimation problem is then a simple (but voluminous) regression calculation in which the score difference is predicted by a sum of terms of player values, possession values and home court values; there are 525 players, but only 10 of them appear in any given time segment.

The matchup data is assembled in section 7, and saved in the file "data/201213matchups.csv"

Start Data analysis from here.

```
m<-read.csv(file="data/201213matchups.csv",as.is=T)
head(m, 3)
```

```
    gameId times ahscore aposs           a1           a2
1 400277721   372       2  .off Emeka Okafor Trevor Ariza
2 400277721   137      -4  .def Emeka Okafor Trevor Ariza
3 400277721    60      -3  .off   Jan Vesely Trevor Ariza
            a3            a4          a5 hposs
1 Bradley Beal Trevor Booker A.J. Price  .def
2 Bradley Beal Trevor Booker A.J. Price  .off
3 Bradley Beal Trevor Booker A.J. Price  .def
                h1            h2           h3           h4
1 Anderson Varejao    Alonzo Gee Kyrie Irving Dion Waiters
2       C.J. Miles    Alonzo Gee Kyrie Irving Tyler Zeller
3       C.J. Miles Daniel Gibson Kyrie Irving Tyler Zeller
                h5             ateam           hteam
```

```
1 Tristan Thompson Washington Wizards Cleveland Cavaliers
2 Tristan Thompson Washington Wizards Cleveland Cavaliers
3 Tristan Thompson Washington Wizards Cleveland Cavaliers
```

```
dim(m)
```

```
[1] 20235    17
```

3 Regression to estimate player value

Define iterative regression, to be repeated in various ways. The algorithm does the regression one team at a time. The best fit of the first team to is carried out with least squares on the per second score in each time interval, weighted by the total number of seconds in the time interval. The variables used are the 5 team players in that time interval, the beginning possession, and the home court advantage. The residuals from the fit are used in the next least squares fit for the second team. The iterative calculation is repeated 10 times for the 30 teams.

```
iR <- iterateRegress(m)
```

```
it: 1 sqres: 5963035.646
it: 2 sqres: 5944757.88
it: 3 sqres: 5943853.923
it: 4 sqres: 5943713.473
it: 5 sqres: 5943682.52
it: 6 sqres: 5943675.036
it: 7 sqres: 5943673.096
it: 8 sqres: 5943672.49
it: 9 sqres: 5943672.21
it: 10 sqres: 5943672.01
it: 11 sqres: 5943671.829
it: 12 sqres: 5943671.653
it: 13 sqres: 5943671.478
it: 14 sqres: 5943671.303
it: 15 sqres: 5943671.128
it: 16 sqres: 5943670.954
it: 17 sqres: 5943670.779
it: 18 sqres: 5943670.605
it: 19 sqres: 5943670.43
it: 20 sqres: 5943670.256
```

```
s1 <- ShowTeam(iR$est,16)
```

We will compare this later with a smoothed estimate.

4 Prior adjustment for infrequent players

Do analysis by teams with a prior for each player effect corresponding to zero score difference in 5-g games, whenever the player played in g <5 games. This prior compresses the scores of those infrequent players towards zero.

```
mfake <- m
est <- initializeEst(m)
```

We need to know the estimated times to do the adjustment:

```
est$time <- iR$est$time
for (t in 1:length(est$team))
 est$est[[t]] <- (rep(0, length(est$teams[[t]])))

# add fake data for each player, 5-g games with 0 scores
i <- dim(m)[1]
for (t in 1:length(est$team)){
 for ( p in 2:length(est$teams[[t]]) ){
   if( est$time[[t]][p] < 5){
     i <- i+1
     mfake[i, ] <- mfake[i-1, ]
     mfake$ateam[i] <- est$team[t]
     mfake$hteam[i] <- ""
# a game has 60*48 = 2880 seconds
     mfake$times[i] <- (5-est$time[[t]][p]) * 2880
     mfake$ahscore[i] <- 0
     mfake[i, 4:15] <- ""
     mfake[i, 9] <- est$teams[[t]][p]
   }
 }
}
iR <- iterateRegress(mfake)
```

```
it: 1 sqres: 5997591.415
it: 2 sqres: 5982294.941
it: 3 sqres: 5981700.565
it: 4 sqres: 5981593.254
it: 5 sqres: 5981540.117
```

```
it: 6 sqres: 5981496.34
it: 7 sqres: 5981454.762
it: 8 sqres: 5981413.998
it: 9 sqres: 5981373.766
it: 10 sqres: 5981334.004
it: 11 sqres: 5981294.695
it: 12 sqres: 5981255.833
it: 13 sqres: 5981217.41
it: 14 sqres: 5981179.424
it: 15 sqres: 5981141.868
it: 16 sqres: 5981104.738
it: 17 sqres: 5981068.028
it: 18 sqres: 5981031.735
it: 19 sqres: 5980995.854
it: 20 sqres: 5980960.379
```

```
s2 <- ShowTeam(iR$est,16)
names(s2) <- c("pdiff48", "pse", "pgames")
print(iR$est$team[16])
```

```
[1] "Miami Heat"
```

The difference between estimates with and without priors:

```
print(cbind(s1,s2))
```

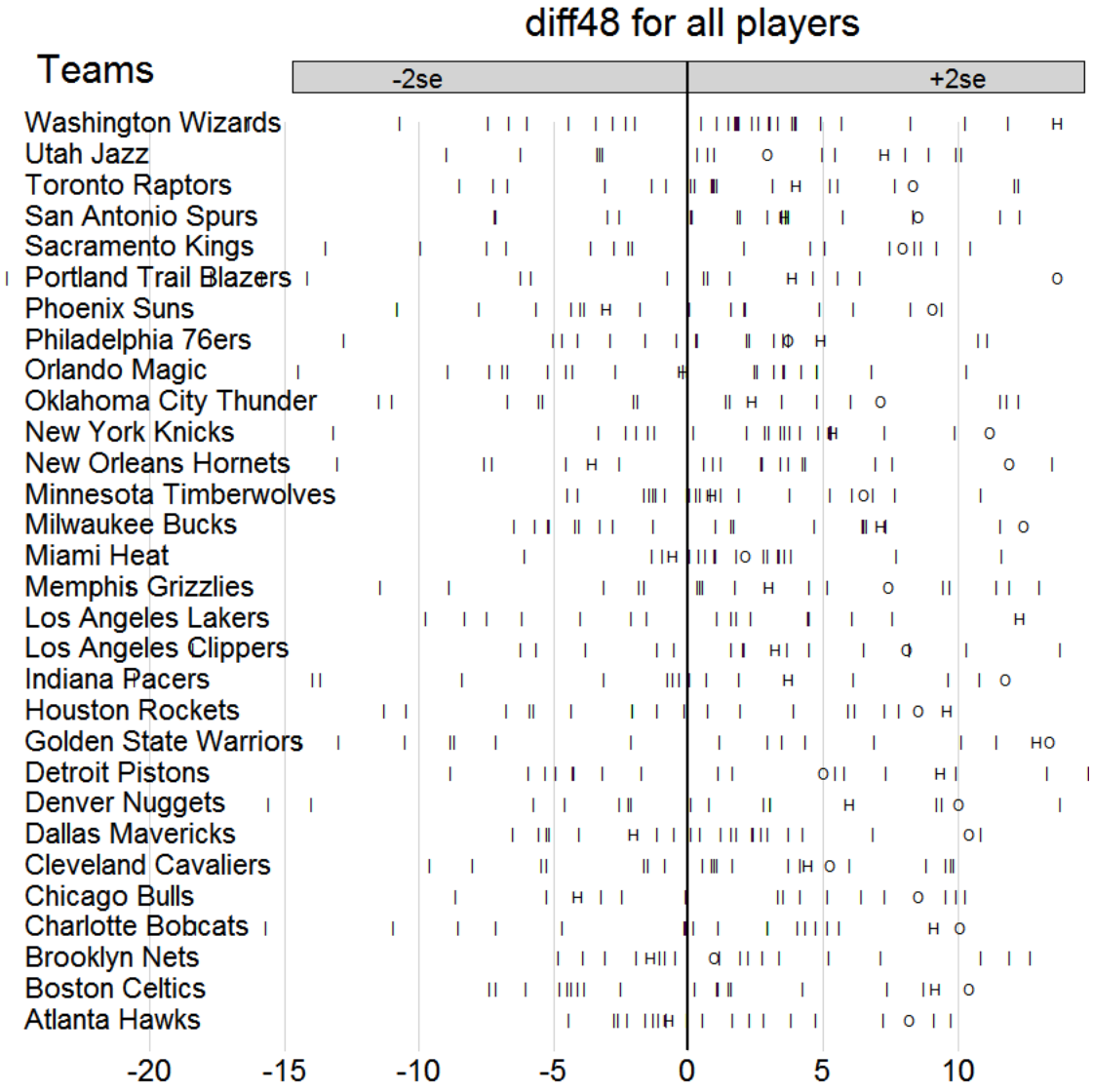
```
                diff48 se games pdiff48 pse pgames
..home court        -1  5    26      -1   5     26
.off                 3  5    26       2   5     26
Chris Andersen       6 10     7       8  10      7
Chris Bosh           1  8    31       2   7     31
Dwyane Wade          4  6    29       3   6     29
James Jones          9 15     2       3   8      5
Jarvis Varnado      -8 28     1       1   8      5
Joel Anthony        -1 10     6       1  10      6
Josh Harrellson    -78 48     0       0   8      5
Juwan Howard        18 27     0       4   8      5
LeBron James        14  7    37      12   7     37
Mario Chalmers      19  9    27      16   9     27
Mike Miller          2  7    11       3   6     11
Norris Cole          7  9    18       4   8     18
```

Patrick Patterson	-37	35	0	1	8	5
Rashard Lewis	-1	7	9	-1	7	9
Ray Allen	1	6	23	0	6	23
Shane Battier	-3	6	21	-1	6	21
Terrel Harris	59	44	0	3	8	5
Udonis Haslem	-8	7	19	-6	7	19

Note that the estimates for frequent players are affected only a little, and the wildly varying estimates for the infrequent players, (less than 5 games), are set close to zero. It tends to reduce the positive effects of the star players like LeBron James, but we don't know these effects very accurately based just on one season.

5 Player estimates displayed

```
tiff("pictures/diff48.dif", w=900, h=900)

Grid(c(seq(-25, 10, 5), 12), c(0, 30), at=c(0, -22),
  ylab=" diff48 for all players/Teams",cex=2.5)

se <- mean(unlist(iR$est$se))
rect(-2*se, 31, 2*se, 32, col="light grey", xpd=T)
lines(c(0,0), c(0,32), lwd=2)

text(10, 31.5, "+2se", cex=1.7, xpd=T)
text(-10, 31.5, "-2se", cex=1.7, xpd=T)

# plot points for each team
# allow text outside plot region
par(xpd=T)
for(i in 1:30){
 points(iR$est$est[[i]][-c(1,2)],
 rep(i,length(iR$est$est[[i]])-2), pch="|")

 points(iR$est$est[[i]][1], i, pch="H")
 points(iR$est$est[[i]][2], i, pch="O")

 text( -25, i, iR$est$team[i], cex=1.8, pos=4)
}

dev.off()
```




```
mean(unlist(iR$est$se))
```

```
[1] 7.339993953
```

There are relatively few values greater than 0, many less than 0.. If you were testing for statistical significance, none of the players would be judged to have a statistically significant effect on the score. The possession effects (o), and home court effects(H) are nearly all positive.

6 Detailed estimates

We show only estimates that exceed sqrt(2) the standard error, which corresponds to the Akaike criterion for model fitting:

```
teams<- iR$est$team
for(t in 1:30){
cat(" \n")
cat(teams[t], "\n")

print(ShowTeam(iR$est,t, threshold = sqrt(2)))

}
```

```
Atlanta Hawks
             diff48 se games
.off              8  5    27
Kyle Korver       9  6    30

Boston Celtics
              diff48 se games
..home court       9  5    25
.off              10  5    24
Jeff Green         9  6    25
Paul Pierce       18  6    32

Brooklyn Nets
                diff48 se games
Andray Blatche      13  8    19
Gerald Wallace      11  6    27

Charlotte Bobcats
               diff48 se games
..home court        9  5    26
.off               10  5    26
Ben Gordon         -8  5    18
Byron Mullens     -16  6    18
Hakim Warrick     -18  8     7

Chicago Bulls
            diff48 se games
```

.off	9	5	25
Taj Gibson	10	6	16

Cleveland Cavaliers

	diff48	se	games
Kyrie Irving	10	7	28
Shaun Livingston	10	7	14

Dallas Mavericks

	diff48	se	games
.off	10	5	25
Dirk Nowitzki	11	6	21
Vince Carter	16	6	25

Denver Nuggets

	diff48	se	games
.off	10	5	26
Andre Iguodala	14	6	38
Andre Miller	9	5	27
JaVale McGee	-14	7	18
Timofey Mozgov	-16	9	5
Ty Lawson	9	5	33

Detroit Pistons

	diff48	se	games
..home court	9	5	23
Austin Daye	13	9	5
Khris Middleton	15	10	5

Golden State Warriors

	diff48	se	games
..home court	13	5	25
.off	13	5	24
Andrew Bogut	-14	8	10
Carl Landry	-13	7	21
Festus Ezeli	-10	7	15
Klay Thompson	10	7	38
Richard Jefferson	16	9	6
Stephen Curry	11	7	37

Houston Rockets

	diff48	se	games
..home court	10	5	22
.off	9	5	23
Omer Asik	16	7	32

Indiana Pacers

	diff48	se	games
.off	12	5	24
David West	28	13	29
Ian Mahinmi	-21	12	15
Tyler Hansbrough	19	12	17

Los Angeles Clippers

	diff48	se	games
.off	8	5	27
Chris Paul	14	9	31
Ronny Turiaf	-18	10	6

Los Angeles Lakers

	diff48	se	games
..home court	12	5	28
.off	16	5	29

Memphis Grizzlies

	diff48	se	games
Chris Johnson	12	8	5
Darrell Arthur	-21	7	12
Jerryd Bayless	11	7	21
Marreese Speights	-22	9	7
Mike Conley	16	7	35
Quincy Pondexter	10	6	15
Tayshaun Prince	13	8	15
Wayne Ellington	17	8	8
Zach Randolph	-11	7	33

Miami Heat

	diff48	se	games
LeBron James	12	7	37
Mario Chalmers	16	9	27

Milwaukee Bucks

```
             diff48 se games
.off             13  5    24
J.J. Redick      12  7     9

Minnesota Timberwolves
[1] diff48 se     games
<0 rows> (or 0-length row.names)

New Orleans Hornets
                 diff48 se games
.off                 12  5    22
Greivis Vasquez      14  9    34
Xavier Henry        -13  7     7

New York Knicks
             diff48 se games
.off             11  5    24
James White     -13  9     6

Oklahoma City Thunder
                  diff48 se games
Hasheem Thabeet       12  8    10
Kendrick Perkins      12  8    26
Kevin Durant          12  7    39
Nick Collison         18  8    20

Orlando Magic
               diff48 se games
.off               16  5    24
Jameer Nelson      10  6    26
Moe Harkless      -14  6    21

Philadelphia 76ers
                diff48 se games
Royal Ivey         -13  7     8
Spencer Hawes       11  6    28
Thaddeus Young      11  6    35

Phoenix Suns
                 diff48 se games
.off                 9  5    22
```

Goran Dragic	20	9	31
Marcin Gortat	-11	7	24
Markieff Morris	-11	7	21

Portland Trail Blazers

	diff48	se	games
.off	14	5	25
Damian Lillard	25	8	41
J.J. Hickson	-25	8	31
Joel Freeland	-16	10	5
Luke Babbitt	-14	8	7
Meyers Leonard	-18	8	14
Ronnie Price	19	10	6

Sacramento Kings

	diff48	se	games
..home court	16	5	24
.off	8	5	24
James Johnson	-10	6	11
John Salmons	9	6	31
Thomas Robinson	-13	8	10

San Antonio Spurs

	diff48	se	games
.off	9	5	25
Tiago Splitter	12	7	25
Tim Duncan	12	7	26

Toronto Raptors

	diff48	se	games
.off	8	5	24
Amir Johnson	12	7	28
Quincy Acy	12	8	5

Utah Jazz

	diff48	se	games
Alec Burks	10	7	13
Gordon Hayward	10	6	27

Washington Wizards

	diff48	se	games

```
..home court        14   5    26
.off                19   5    23
Garrett Temple      10   6    16
John Wall           12   7    21
Kevin Seraphin     -11   6    30
Trevor Booker      -16   6    15
```

I found these estimates surprising. LeBron James of the Miami Dolphins widely regarded as the best player in the league, but these results suggest many other players are better, including Mario Chalmers on his own team. Still, none of the diff48's are very accurately known.

There are substantial effects in most teams for home court advantage and offense. For example, The Washington Wizards play 14 points better per game at home than away, and play offense better than defense by 19 points per game.

7 Assembling matchup data

7.1 Find the team names for 2012-2013 season:

The team names are located at:
http://espn.go.com/nba/teams

```
teams <- scan("http://espn.go.com/nba/teams",
what="", sep="\n")
```

Pick the team names out of the source file to match an expression beginning with team/-/name and ending in characters -,/, letters, or 7 or 6:

```
teamnames <- regmatches(teams,
regexpr("team/_/name/[-/a-z76]*", teams))
teamnames <- substring(teamnames, 13,
nchar(teamnames))
```

The last name is a repeat:

```
teamnames <- teamnames[-31]
```

7.2 Find the recap ID's for all games

Find the URLS for the different teams' schedules:

```
snames <- teamnames
```

Capitalize the first letter:

```
uctn <- toupper(substr(teamnames,1,1))
head(uctn)
```

```
[1] "B" "B" "N" "P" "T" "G"
```

```
substr(snames,1,1) <- uctn
head(snames)
```

```
[1] "Bos/boston-celtics"        "Bkn/brooklyn-nets"
[3] "Ny/new-york-knicks"        "Phi/philadelphia-76ers"
[5] "Tor/toronto-raptors"       "Gs/golden-state-warriors"
```

Add seasontype/2/ to the names:

```
ssnames <- sub("/", "/seasontype/2/", snames)
head(ssnames)
```

```
[1] "Bos/seasontype/2/boston-celtics"
[2] "Bkn/seasontype/2/brooklyn-nets"
[3] "Ny/seasontype/2/new-york-knicks"
[4] "Phi/seasontype/2/philadelphia-76ers"
[5] "Tor/seasontype/2/toronto-raptors"
[6] "Gs/seasontype/2/golden-state-warriors"
```

Paste the espn header to the expanded team name:

```
sssnames <- paste(
"http://espn.go.com/nba/team/schedule/_/name/",
ssnames, sep="")
head(sssnames, 2)
```

```
[1]
"http://espn.go.com/nba/team/schedule/_/name/Bos/seasontype
```

```
/2/boston-celtics"
[2]
"http://espn.go.com/nba/team/schedule/_/name/Bkn/seasontype
/2/brooklyn-nets"
```

Extract gameIds for all games:

```
gameId <- NULL
for( team in 1:length(sssnames)){
sched <- scan(sssnames[team], what="", sep="\n",
              quiet=T)

# Locate the single line in the source page containing
the gameId:
recap <- grep("recap", sched)

# Find the positions of all those gameId's:
 pos <- unlist(gregexpr("[0-9]{9}", sched[recap]))
 if( team==1) head(pos)

#Loop through the positions to get the ID's:
u <- rep("", length(pos))
for(i in 1:length(pos)) u[i] <- substr(sched[recap],
                                pos[i], pos[i]+8)
if(team==1) head(u)
gameId <- c(gameId, u)
}
gameId <- sort(unique(gameId))
length(gameId)
```

```
[1] 0
```

We sort the unique gameIds (each gameId appears twice because each scheduled game appears in two teams list of games).
The gameIds go from 400277721 to 400278950 in order, corresponding to order of dates of the games, with two additional mysterious gameIds at 400431053 and 400440940, which indeed are regular season games.

You may view the play by plays for the first game at:
http://espn.go.com/nba/playbyplay?gameId=400277721

7.3 Collecting Play by Play records

A single paragraph is extracted from the source file of the play by play for each game.. The collected lines are saved for further processing. It takes about 30 minutes, so you don't want to do this more than once.

```
if (FALSE){

root <- "http://espn.go.com/nba/playbyplay?gameId="

for (id in gameId){
  x <- scan(paste(root, id, "&period=0", sep=""),
            what="", sep="\n", quiet=T)

# possible play by play not available
if(length(grep("Play-By-Play not available", x))== 0
){
    x <- scan(paste(root, id, "&period=0", sep=""),
              what="", sep="\n", quiet=T)
    start <- grep("Complete Play-By-Play", x)
    end  <- grep("End of Game |End of the 4th
Quarter|End of Fourth Quarter", x)

    if(length(end)>0){
      end <- max(end)
      use <- start:end
      pbp <- c(pbp, x[use])
    }
  }

  teams <- Matches("[.A-Za-z67 ]+ - Play By Play - ",
                   x[5])
  teams <- gsub(" - Play By Play - ", "", teams)
  teams <- paste(id, teams, sep="/")
  teams <- gsub(" vs. ", "/", teams)
  gameteams <- c(gameteams, teams)
}
gameteams <- unlist(strsplit(gameteams, "/"))
gameteams<-data.frame(t(matrix(gameteams,nrow=3)))
```

```
names(gameteams) <- c("gameId", "ateam", "hteam")
cat(pbp, file="data/201213play-by-play", sep="\n")
head(pbp, 3)

write.csv(gameteams,
      file="data/201213gameteams.csv",row.names=F)
head(gameteams)
}
```

7.4 Express play by plays as data frame

```
pbp <-
scan("data/201213play-by-play",what="q",sep="\n")
length(pbp)
```

```
[1] 531797
```

```
head(pbp, 3)
```

```
[1] "<h2>Complete Play-By-Play</h2><A
HREF=\"/nba/playbyplay?gameId=400277721&period=1\">1</A> |
<A HREF=\"/nba/playbyplay?gameId=400277721&period=2\">2</A>
| <A
HREF=\"/nba/playbyplay?gameId=400277721&period=3\">3</A> |
<A HREF=\"/nba/playbyplay?gameId=400277721&period=4\">4</A>
| All<div class=\"mod-container mod-no-header-footer mod-open
mod-open-gamepack mod-box\"><div
class=\"mod-content\"><table border=\"1\" width=\"100%\"
class=\"mod-data\"><thead><tr bgcolor=\"#555555\"><td
colspan=4 style=\"text-align:center; padding:0;\"><div
class=\"mod-header\" style=\"border:none;\"><h4
style=\"text-align:center\">1st Quarter
Summary</h4></div></td></tr><tr align=\"center\"><th
width=\"5%\" style=\"text-align:center;\">TIME</b></th><th
width=\"40%\"
style=\"text-align:left;\">WASHINGTON</b></th><th
width=\"5%\" style=\"text-align:center;\">SCORE</b></th><th
width=\"40%\"
style=\"text-align:left;\">CLEVELAND</th></tr></thead><tr
class=\"even\"><td valign=top width=50
style=\"text-align:center;\">12:00</td><td
valign=\"top\"> </td><td valign=\"top\"
style=\"text-align:center;\" NOWRAP>0-0</td><td
valign=\"top\" style=\"text-align:left;\">Emeka Okafor vs.
Tristan Thompson (Trevor Booker gains possession)</td></tr>"
[2] "<tr class=\"odd\"><td valign=top width=50
style=\"text-align:center;\">11:42</td><td valign=\"top\"
style=\"text-align:left;\"><B>Trevor Ariza makes  three point
jumper  (A.J. Price assists )</B></td><td valign=\"top\"
```

```
style=\"text-align:center;\" NOWRAP>3-0</td><td
valign=\"top\"> </td></tr>"
[3] "<tr class=\"even\"><td valign=top width=50
style=\"text-align:center;\">11:18</td><td
valign=\"top\"> </td><td valign=\"top\"
style=\"text-align:center;\" NOWRAP>3-0</td><td
valign=\"top\" style=\"text-align:left;\">Emeka Okafor blocks
Alonzo Gee's  layup</td></tr>"
```

Find GameId:

```
gameId <- Matches("[0-9]{9}", pbp)
length(gameId)
```

```
[1] 1224
```

Find scores:
Eliminate entries without scores:

```
noscores <- grepl("[0-9]+-[0-9]+", pbp)
pbp <- pbp[noscores]
scores <- Matches("[0-9]+-[0-9]+", pbp)
length(scores)
```

```
[1] 511040
```

```
length(pbp)
```

```
[1] 511040
```

Find times:

```
times <- Matches("[0-9]+:[0-9]{2}", pbp)
length(times)
```

```
[1] 511040
```

Make times numerical in seconds:

```
t <- as.numeric(unlist(strsplit(times, ":")))
times <- 60 * t[seq(1, length(t), 2)] +  t[seq(2,
length(t), 2)]
```

Find Possessions:
Specify possess and gameId (NOWRAP) appears after action for away actions; the expression describes an action with at least two words:

```
expr <- ">[ ]?[ ]?[-.':,\\(\\)a-zA-Z0-9]+ [-
.':,\\(\\)a-zA-Z0-9]+<"
possess <- rep(NA, length(pbp))
away <- grepl(paste(expr,".*NOWRAP", sep=""),  pbp)
home <- !away & grepl(expr, pbp)
possess[away] <- "AWAY"
possess[home] <- "HOME"
possess[grepl("Complete Play-By-Play", pbp)] <-
gameId
head(possess)
```

```
[1] "400277721" "AWAY"      "HOME"      "AWAY"
[5] "AWAY"      "HOME"
```

Put gameId into every entry:
Find where the gameId's are located:

```
startgame <- grep("Complete Play-By-Play", pbp)
length(startgame)
```

```
[1] 1224
```

```
startgame <- c(startgame, length(times)+1)
Game <- rep("", length(times))
```

Loop through each startgame and copy gameId into full game:

```
for(i in 1:(length(startgame)-1))
Game[startgame[i]:(startgame[i+1]-1)] <-
possess[startgame[i]]
possess[startgame[i]] <- "NA"
head(Game,3)
```

```
[1] "400277721" "400277721" "400277721"
```

```
tail(Game,3)
```

```
[1] "400440940" "400440940" "400440940"
```

Find actions after fixing awkward phrases:

```
actions <- rep("", length(pbp))
pbp <- gsub("Complete
Play|1st|2nd|3rd|4th|Quarter|Summary", "", pbp)
pbp <- gsub("Double technical|off foul|Jr. |Al-", "",
pbp)
pbp <- gsub("shot clock|game violation|delay of","",
pbp)
pbp <- gsub("Ed Davis makes free throw 2 of 2. ", "",
pbp)
```

Get rid of team names:

```
pbp <- gsub ("[A-Z]{3,}", "", pbp)
```

Handle a circumflex and player with more than two names:

```
pbp <- gsub("Jose Juan Barea", "Jose Barea", pbp)
pbp <- gsub("Nando de Colo", "Nando Colo", pbp)
pbp <- gsub("Nenê", "Nene Nene", pbp)
pbp <- gsub("Metta World Peace", "Metta World", pbp)
pbp <- gsub("Luc Richard Mbah a Moute", "Luc Moute",
pbp)
```

Lots of funny things can happen in the actions, discovered by seeing when a proposed regular expression for the action fails:

```
wactions <- grepl(expr, pbp)
```

These bad actions will be deleted from the list later:

```
actions[!wactions] <- ""
actions[wactions] <- Matches(expr, pbp[wactions])
actions <- gsub("<|>", "", actions)
length(actions)
```

```
[1] 511040
```

```
head(actions)
```

```
[1] "Emeka Okafor vs. Tristan Thompson (Trevor Booker gains
possession)"
[2] "Trevor Ariza makes  three point jumper  (A.J. Price
assists )"
[3] "Emeka Okafor blocks Alonzo Gee's  layup"
[4] "Trevor Ariza defensive rebound"
[5] "Emeka Okafor makes layup (Trevor Ariza assists )"
[6] "Kyrie Irving makes  three point jumper  (Tristan Thompson
assists )"
```

Make data frame with the 5 variables:

```
pbpExtract <- data.frame(gameId = Game, times=times,
              scores=scores, possess=possess,
              actions=actions,
stringsAsFactors=FALSE)
head(pbpExtract)
```

```
    gameId times scores   possess
1 400277721   720    0-0 400277721
2 400277721   702    3-0      AWAY
3 400277721   678    3-0      HOME
4 400277721   675    3-0      AWAY
5 400277721   668    5-0      AWAY
6 400277721   648    5-3      HOME

actions
1  Emeka Okafor vs. Tristan Thompson (Trevor Booker gains
possession)
2       Trevor Ariza makes  three point jumper  (A.J. Price
assists )
3                           Emeka Okafor blocks Alonzo
Gee's  layup
4                                    Trevor Ariza
defensive rebound
5                   Emeka Okafor makes layup (Trevor Ariza
assists )
```

```
6 Kyrie Irving makes  three point jumper  (Tristan Thompson
assists )

dim(pbpExtract)

[1] 511040      5

pbpExtract <- pbpExtract[actions != "",]
pbpExtract <- na.omit(pbpExtract)
```

Modify actions to consist of player names involved in action:
First handle team actions and blocks and various non player actions:

```
pbpExtract <-
pbpExtract[!grepl("team|vs.", pbpExtract$actions),]
actions <- pbpExtract$actions
actions <- gsub("[-'.a-zA-Z]+ [-'.a-zA-Z]+ blocks ",
"", actions)
actions[grep("technical foul", actions)] <- ""
actions <- gsub("'s", "", actions)
dim(pbpExtract)

[1] 486542      5

length(actions)

[1] 486542

actions[grep("Sec Inbound", actions)] <- ""
# change some particular enter errors
actions[pbpExtract$times== 363 &
pbpExtract$gameId=="400277723"] <-"Metta World
enters the game for Antawn Jamison"
length(actions)

[1] 486542
```

Handle non-enter actions:

```
enters <- grepl(" enters the game for ", actions)
pexpr <- "[A-Z][-'.a-zA-Z]+ [A-Z][-'.a-zA-Z]+"
```

```
bad <- !grepl(pexpr, actions)
actions[bad] <- ""
actions[!(enters|bad)] <-
                Matches(pexpr,
actions[!(enters|bad)])
length(actions)
```

```
[1] 486542
```

Treat enter the game actions differently:

```
actions[enters] <-
gsub("[ ]+enters the game for[ ]+", "/",
actions[enters])
eexpr <- "[-'.a-zA-Z]+ [-'.a-zA-Z]+/[-'.a-zA-Z]+
[-'.a-zA-Z]+"
badenter <- !grepl(eexpr, actions)
actions[enters & badenter] <- ""
be <- enters &!badenter
actions[be] <- Matches(eexpr, actions[be])
pbpExtract$actions <- actions
length(actions)
```

```
[1] 486542
```

```
dim(pbpExtract)
```

```
[1] 486542      5
```

Remove missing and "" values:

```
pbpExtract <- na.omit(pbpExtract)
pbpExtract <- pbpExtract[actions !="", ]
pbpExtract[pbpExtract$gameId=="400277723",
][140:145, ]
```

```
       gameId times scores possess
1005 400277723   384  37-37    HOME
1006 400277723   363  37-37    HOME
1007 400277723   356  39-37    AWAY
1008 400277723   326  39-39    HOME
1009 400277723   313  39-39    AWAY
```

```
1010 400277723   311  39-39    HOME
                        actions
1005                   Pau Gasol
1006 Metta World/Antawn Jamison
1007          Rodrigue Beaubois
1008                 Kobe Bryant
1009                 Jae Crowder
1010                 Metta World

write.csv(pbpExtract, "data/1213pbpExtract.csv",
row.names=F)
```

7.5 Parse pbp actions to get matchups

```
Extract <- read.csv( "data/1213pbpExtract.csv",
as.is=T)
Extract[Extract$gameId=="400277723", ][140:145, ]
```

```
      gameId times scores possess
958 400277723   384  37-37    HOME
959 400277723   363  37-37    HOME
960 400277723   356  39-37    AWAY
961 400277723   326  39-39    HOME
962 400277723   313  39-39    AWAY
963 400277723   311  39-39    HOME
                       actions
958                  Pau Gasol
959 Metta World/Antawn Jamison
960          Rodrigue Beaubois
961                Kobe Bryant
962                Jae Crowder
963                Metta World
```

Find matchups in each game applying the function gamematchup. The starters are found for each quarter, and then a new quintet of players is discovered after each enter action. Many times the enter actions are contradictory, and we do not know the quintet. The times and score information is used to get the change in score and the change in time for each quintet-quintet matchup.

```
gid <- unique(Extract$gameId)
length(gid)
```

```
[1] 1224
```

```
system.time({

 for( g in gid) {
   gm <- gamematchup(Extract[Extract$gameId==g,])
   if(g == gid[1]) m <- gm
   if(g != gid[1]) m <- rbind(m, gm)
```

```
 }
})
```

```
   user  system elapsed
 151.01    0.34  151.74
```

```
print(dim(m))
```

```
[1] 34646    15
```

```
head(m[,1:6])
```

```
     gameId times ahscore aposs           a1
1  400277721   372       2  .off Emeka Okafor
3  400277721   137      -4  .def Emeka Okafor
5  400277721    60      -3  .off   Jan Vesely
8  400277721   151      -2  .off   Jan Vesely
10 400277721   147      -2  .off   Jan Vesely
11 400277721    72       1  .def   Jan Vesely
                 a2
1     Trevor Ariza
3     Trevor Ariza
5     Trevor Ariza
8  Martell Webster
10   Jannero Pargo
11   Jannero Pargo
```

Add team names to matchups.

```
gt <- read.csv("data/201213gameteams.csv",  as.is=T)
head(gt)
```

```
    gameId               ateam               hteam
1 400277721 Washington Wizards Cleveland Cavaliers
2 400277722      Boston Celtics          Miami Heat
3 400277723   Dallas Mavericks  Los Angeles Lakers
4 400277724     Indiana Pacers     Toronto Raptors
5 400277725     Denver Nuggets  Philadelphia 76ers
6 400277726    Houston Rockets     Detroit Pistons
```

```
mrow <- dim(m)[1]
m$ateam <- rep("", mrow)
m$hteam <- rep("", mrow)
```

Run over all games and insert teams corresponding to the gameId:

```
gid <- unique(m$gameId)
for (g in gid){
m$ateam[m$gameId == g] <- gt$ateam[gt$gameId==g]
m$hteam[m$gameId == g] <- gt$hteam[gt$gameId==g]
}
head(m)
```

```
     gameId times ahscore aposs           a1
1  400277721   372       2  .off Emeka Okafor
3  400277721   137      -4  .def Emeka Okafor
5  400277721    60      -3  .off   Jan Vesely
8  400277721   151      -2  .off   Jan Vesely
10 400277721   147      -2  .off   Jan Vesely
11 400277721    72       1  .def   Jan Vesely
                a2              a3              a4
1     Trevor Ariza    Bradley Beal   Trevor Booker
3     Trevor Ariza    Bradley Beal   Trevor Booker
5     Trevor Ariza    Bradley Beal   Trevor Booker
8  Martell Webster Jordan Crawford   Trevor Booker
10   Jannero Pargo Martell Webster Chris Singleton
11   Jannero Pargo Martell Webster Chris Singleton
               a5 hposs               h1            h2
1       A.J. Price  .def Anderson Varejao    Alonzo Gee
3       A.J. Price  .off       C.J. Miles    Alonzo Gee
5       A.J. Price  .def       C.J. Miles Daniel Gibson
8       A.J. Price  .def       C.J. Miles Daniel Gibson
10 Jordan Crawford  .def     Tyler Zeller Daniel Gibson
11 Jordan Crawford  .off Anderson Varejao Daniel Gibson
            h3           h4              h5
1  Kyrie Irving Dion Waiters Tristan Thompson
3  Kyrie Irving Tyler Zeller Tristan Thompson
5  Kyrie Irving Tyler Zeller Tristan Thompson
8  Kyrie Irving Tyler Zeller      Luke Walton
10   C.J. Miles Donald Sloan      Luke Walton
```

```
11   C.J. Miles Donald Sloan      Luke Walton
                ateam                hteam
1  Washington Wizards Cleveland Cavaliers
3  Washington Wizards Cleveland Cavaliers
5  Washington Wizards Cleveland Cavaliers
8  Washington Wizards Cleveland Cavaliers
10 Washington Wizards Cleveland Cavaliers
11 Washington Wizards Cleveland Cavaliers
```

Compute Seconds of time covered by valid and invalid matchups:

```
sum(m$times[is.na(m[,5])])
```

```
[1] 1383068
```

```
sum(m$times[!is.na(m[,5])])
```

```
[1] 2142052
```

So we lose nearly 40% of the matchups due to lack of accurate reporting of substitutions.

```
mm <- na.omit(m)
dim(mm)
```

```
[1] 20235    17
```

```
write.csv(mm, file="data/201213matchups.csv",
row.names=F)
```

11 Functions

```
Grid <- function(xticks, yticks, ylab="",
      at=(min(xticks)+ mean(xticks))/2, cex=2.5){
# background for plot using grid of light grey lines

# initialise plot space
par(mar=c(3,3,6,2))
plot(1, 1,  xlim=range(xticks),
           ylim = range(yticks),
           xlab="", ylab="", axes=F, pch="")

# use only interior values of tick ranges in plots
usey <- rep( T, length(yticks) )
usey[c( 1, length(yticks) )] <- F
usex <- rep( T, length(xticks) )
usex[c( 1, length(xticks) )] <- F

# grey lines in both directions
for ( row in yticks[usey] )
lines(range(xticks), c(row, row), col="light grey")
for ( col in xticks[usex] )
lines(c(col, col), range(yticks), col="light grey")

# put ylab on left top, using / to split long
expressions
ylabs <- unlist(strsplit(ylab,"/"))

# identify tick marks on both axes
if (length(yticks) > 2)
text(pos=2, rep(min(xticks), length(yticks)-2 ),
     yticks[usey], yticks[usey], cex=2, xpd=T)
if (length(xticks)>2)
text(pos=1, xticks[usex],  rep(min(yticks),
    length(xticks)-2), xticks[usex], cex=2, xpd=T)

lylabs <- min(5, length(ylabs))
if(lylabs > 0)
mtext(ylabs, side=3,line = (5/lylabs)*(lylabs-1):0,
      at = at, cex=cex)
```

```
# restore margin spacing
par(mar=c(5, 4, 4, 2))
invisible()
}

except <- function(x,y){
if(length(y)==0) return(x)
return (x[ ! (x %in% y)])
}



starters <- function(x){

# find starters given history of plays and
substitutions

# initialise values
enters <- grep("/", x)
le <- length(enters)
bench <- rep("", 0)
start <- rep("",0)
mm <- rep(NA, 5)

if (length(enters)==0)  return(unique(x))

# consider changes at substitution
for(ent in 1:length(enters)){

# identify two player in substitution
player <- unlist(strsplit(x[ enters[ent] ], "/") )

# handle substitution on the first play
if (enters[ent] == 1) {

start <- player[2]
  bench <- player[1]
}
```

```
if (enters[ent] > 1) {
# handle the first substitution
  if (ent==1){
    if(player[1] %in% start) return(mm)
    start <-
unique(c(player[2],x[1:(enters[ent]-1)]))
    bench <- player[1]
}

# handle later substitutions
  if (ent > 1){
# add a player to bench unless he already starts
    bench<-unique( c( bench,
except(player[1],start)))

# add a player to start unless already bench

start<-unique(c(start,except(player[2],bench)))

# add intervening player to start unless bench
    if ( enters[ent-1] < enters[ent]-1)
      start <- unique( c(start ,
      except(x[(enters[ent-1]+1) : (enters[ent]-1)],
             bench)) )
  }
}

if(length(start)>5) return(mm)
}

#  add player to start if any after last enter
if ( length(x) >  max(enters) )
start <- unique( c(start ,
except(x[(max(enters)+1) : length(x)], bench)) )

# give up if not 5 starters
if(length(start) !=5) return(mm)
return(start)
}
```

```
matchups <- function(x, allstarters){

# find matchups given history of plays and subs
enters <- grep("/", x)

# initialize array
le <- length(enters)
options(stringsAsFactors = FALSE)
m <- data.frame( matrix(NA, nrow= le +2, ncol=10) )
names(m) <-c( paste("a", 1:5, sep=""),
            paste("h", 1:5, sep="") )

# return missing for bad starters
if (sum( is.na(allstarters) ) > 0) return(m)
if(length(allstarters) != 10) return(m)
```

```
# now do  matchups
m[1, ] <- allstarters
for(ent in 1:length(enters)){
player <- unlist(strsplit(x[ enters[ent] ], "/") )

# the player going out must be in previous 10
# the player coming in must not be in previous 10
if(!(player[2] %in% unlist(m[ent,]))|
   player[1] %in% unlist(m[ent,]) ) return(m)
m[ent+1, ] <- m[ent, ]
m[ent+1, which(unlist(m[ent,]) == player[2])] <-
player[1]
}
m[ent+2,] <- m[ent+1, ]
return(m)
}
```

```
matchup <- function(Extract){

# combine matchups with other game data
am <-
starters(Extract$actions[Extract$possess=="AWAY"])
hm <-
starters(Extract$actions[Extract$possess=="HOME"])
m <- matchups(Extract$actions, c(am, hm))

# get variables at enter times
enters <- grep("/", Extract$actions)
le <- length(enters)
```

```
mgameId <-rep(Extract$gameId[1], le+2)
la <- length(Extract$actions)
if(le ==0) {
mtimes <- c(720, 0)
mscores <- c("0-0",  Extract$scores[la])
mpossess <- c(Extract$possess[1],
Extract$possess[la])
}

# take some care with initial and final quintets
if(le > 0){
mtimes <- c(720, Extract$times[enters], 0)
mscores <- c("0-0",
Extract$scores[enters],Extract$scores[la])

penters <- enters+1
if(penters[le]> la)  penters[le] <- la
mpossess <- c(Extract$possess[1],
Extract$possess[penters], Extract$possess[la])
}

# define data frame with additional variables
ma <- data.frame(matrix("", le+2, 15))
names(ma) <- c("gameId", "times", "score", "aposs",
names(m)[1:5], "hposs", names(m)[6:10])

# specify variable values
ma[, 1] <- mgameId
ma[, 2] <- mtimes
ma[, 3] <- mscores
ma[, 4] <- ".off"
ma[mpossess=="HOME", 4] <- ".def"
ma[, 10] <- ".off"
ma[mpossess=="AWAY", 10] <- ".def"
ma[, c(5:9, 11:15)] <- m
return(ma)
}
```

```
gamematchup <- function(Extract){

# do matchups for all quarters
```

```
ends <- which ( diff(Extract$times) > 0 )
ends <- c(0, ends, dim(Extract)[1])

# handle overtime
if(length(ends) > 5) ends <- ends[1,5]

# bind together the 4 quarters

for( i in 1:(length(ends)-1) ){

mi <- matchup(Extract[(ends[i]+1): ends[i+1],])
if(i == 1) gm <- mi
if(i > 1)  gm <- rbind(gm, mi)
}

# getting starting scores from previous quarter
st <- which(gm$times==720)

gm$score[st[-1]] <- gm$score[st[-1]-1]

# diff times and scores
gm$times <- c(-diff(gm$times), 0)
scores <- unlist(strsplit(gm$score, "-"))
scores <- as.numeric(scores)
score1 <- scores[seq(1, length(scores), 2)]
score2 <- scores[seq(2, length(scores), 2)]

#away-home score, differenced
gm$score <- c(diff(score1-score2), 0)
names(gm) <- gsub("score", "ahscore", names(gm))
gm <- gm[gm$times > 0,]

return(gm)
}
```

```
initializeEst <- function(m){

# Initialize  estimates, se , times played for each
player
est <- list()
team <- sort(unique(m$ateam))
est$team <- team

# tricky first two names are possession and home court
for (t in 1:length(team)){

# Define list of players for each team
# avoid first .def player to avoid singularity in reg
est$teams[[t]] <-
sort(unique(unlist(m[m$ateam==team[t],][,4:9])))
est$teams[[t]] <- est$teams[[t]][est$teams[[t]] != ""
&

est$teams[[t]]!=".def"]

#add in term for home court advantage
est$teams[[t]] <- c("..home court",est$teams[[t]])
est$est[[t]] <- (rep(0, length(est$teams[[t]])))
est$time[[t]] <- est$est[[t]]
est$se[[t]] <- est$est[[t]]
}

return(est)
}
```

```
iterateRegress <- function(m, iter=20){

# iterative regression, one team at a time, to estimate
player value
est <- initializeEst(m)
team <- est$team

# update estimates and differences
sqres <- rep(0,iter)
for (it in 1:iter){
for ( t in 1:length(team) ){

# find the home and away games for this team
  lp <- length(est$teams[[t]])
  if(lp >0){
    htid <- m$hteam == team[t]
    atid <- m$ateam == team[t]
    lht <- sum(htid)
    lat <- sum(atid)
    ht <- m[htid,]
    at <- m[atid,]
    pdiff <- c(at$ahscore, -ht$ahscore)
    ptime <- c(at$times,  ht$times)/(48*60)
    playmat <- matrix(0, lht + lat, lp)
    playmat[(lat+1):(lat + lht), 1] <- 1
    est$time[[t]][1] <- sum(ht$times) /(48*60)

# go through the players to fix prediction matrix
    for (p in 2:lp){
```

```
# set the predictor vector 1 for segments with player
p
      pl <- est$teams[[t]][p]
      huse <- pl==ht[,10]|pl==ht[,11]|pl==ht[,12]|
              pl==ht[,13]|pl==ht[,14]|pl==ht[,15]
      ause <- pl==at[,5]|pl==at[,6]|pl==at[,7]|
              pl==at[,8]|pl==at[,9]|pl==at[,4]
      playmat[c(ause, huse), p] <- 1

      est$time[[t]][p] <- sum(ptime[c(ause, huse)])
    }

# use no constant term in the regression for each player
contribution
    blm <- lm(pdiff/ptime ~ 0 + playmat,weight=ptime)

# update estimate for player and for ahscore res
    res <- blm$res * ptime
    m$ahscore[atid] <- res[1:lat]
    m$ahscore[htid] <- -res[lat +(1:lht)]
    est$est[[t]] <- est$est[[t]] + blm$coef
    est$se[[t]] <- summary(blm)$coef[,2]

}
}
sqres[it] <- sum(m$ahscore^2/(m$times/(48*60)))
cat("it:" , it, "sqres:", sqres[it], "\n")

# make estimates time average to zero, to cancel
indeterminacy
tsum <-0
ttime <- 0
for ( t in 1:length(est$team)){
  if(length(est$est[[t]]) > 0){
    tsum <- sum(est$est[[t]][-c(1,2)] *
est$time[[t]][-c(1,2)])
    ttime <- sum(est$time[[t]][-c(1,2)])
  }
}
tave <- sum(tsum)/sum(ttime)
```

```
for ( t in 1:length(est$team)){
  if(length(est$est[[t]]) > 0)
  est$est[[t]][-c(1,2)] <- est$est[[t]][-c(1,2)] -
tave
}
}
return(list(est=est, sqres=sqres))
}
```

```
ShowTeam <- function(est, t, threshold =0){

# show the estimates for a team
nt <- length(est$est[[t]])
show <- data.frame(matrix(NA, nrow=nt, ncol=3) )
colnames(show) <- c("diff48","se","games")

# handle empty teams
if( nt==0) {
show <- data.frame(matrix(NA, nrow=1, ncol=3) )
colnames(show) <- c("diff48","se","games")
rownames(show) <- est$teams[[t]]
return(show)
}

teamname <- rep(est$team[t], nt)
rownames(show) <- sort(unlist(est$teams[[t]]))
show[, 1] <- round(unlist(est$est[t]))
show[, 2] <- round(unlist(est$se[t]))
show[, 3] <- round(unlist(est$time[t]))
show <- show[abs(show$diff48) >= threshold * show$se,
]
return(show)
```

```
}
```