Basketball Analytics with R

UConn Sports Analytics Symposium 2021

Jackson P. Lautier

10/9/2021

Workshop Outline

- Introduce the `R` package `BasketballAnalyzeR`
- Getting Data: Quick Tips
- Introduce Fundamental (Advanced) Basketball Statistics
- Statistical Case Studies (2020-2021 NBA Season)
  - (Warm-up) Shot Charts
  - Deal or No Deal: Player Similarities
  - I'm Open: Assist Networks
  - 'Tanks' For Nothing: Clustering NBA teams
  - Rack Attack: Expected Points by Shot Distance
- Summary

BasketballAnalyzeR

- Paola Zuccolotto and Marica Manisera (2020), *Basketball Data Science – with Applications in R*. Chapman and Hall/CRC. ISBN 9781138600799
- https://bdsports.unibs.it/
- https://bdsports.unibs.it/basketballanalyzer/

Figure 1: Buy Me! I'm a 'Slam Dunk'

BasketballAnalyzeR Cont.

- Includes many useful functions: `shotchart()`, `fourfactors()`, `assistnet()`, `expectedpts()`, and many more
- Includes preloaded datasets for the 2017-2018 NBA season:
  - `Obox`: GSW opponent's box scores
  - `PbP.BDP`: GSW play-by-play data
  - `PBox`: Players box score statistics
  - `Tadd`: Team Standings
  - `TBox`: Team box score statistics
- All figures and analysis in today's presentation compiled using `BasketballAnalyzer`. See associated `github` materials for code:
- https://github.com/jackson-lautier/UCSAS-Basketball-Analytics-R

Getting Data: Quick Tips

- Free Data: `nbastatR`
  - https://rdrr.io/github/abresler/nbastatR/f/README.md
  - this package used to update team and player box score data
  - code and data available on my github site
- NBA Game ID is a 10-digit code: XXXYYGGGGG,
  - XXX refers to a season prefix
    - 001 : Pre Season
    - 002 : Regular Season
    - 003 : All-Star
    - 004 : Post Season
  - YY is the season year (e.g. 14 for 2014-15),
  - GGGGG refers to the game number (1-1230 for a full 30-team regular season)
- Play-by-Play Data: https://www.bigdataball.com/
  - $30 for single season; matches format for `BasketballAnalyzeR`
  - See 2020 UCSAS presentation to replicate all displays using 2017-2018 NBA season data

Fundamental Basketball Statistics

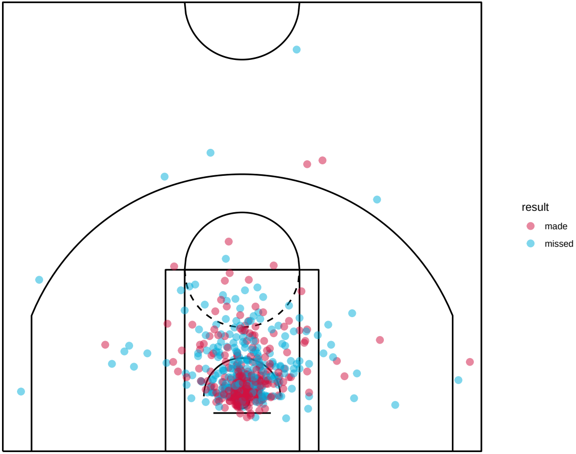
Table 2.4 (Zuccolotto and Manisera)

Factor	Offense	Defense
$eFG\%$	$\frac{(2PM)_T+1.5\times(3PM)_T}{(2PA)_T+(3PA)_T}$	$\frac{(2PM)_O+1.5\times(3PM)_O}{(2PA)_O+(3PA)_O}$
TO Ratio	$\frac{TOV_T}{POSS_T}$	$\frac{TOV_O}{POSS_O}$
$REB\%$	$\frac{OREB_T}{OREB_T+DREB_O}$	$\frac{DREB_T}{OREB_O+DREB_T}$
FT Rate	$\frac{FTM_T}{(2PA)_T+(3PA)_T}$	$\frac{FTM_O}{(2PA)_O+(3PA)_O}$

The *Four Factors* by Kubatko, J., Oliver, D., Pelton, K., and Rosenbaum, D. T. (2007). *A starting point for analyzing basketball statistics*. Journal of Quantitative Analysis in Sports, 3(3):1–22

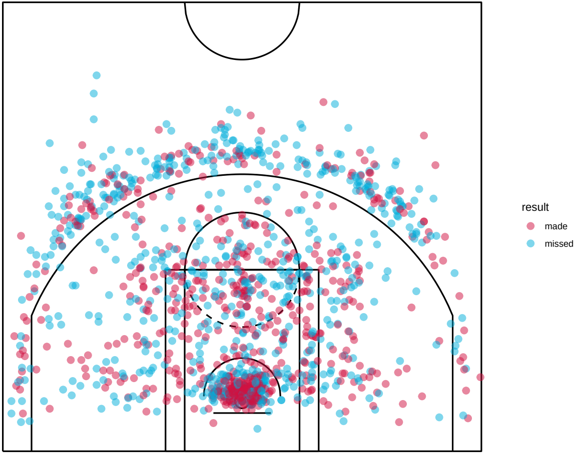
CS1: (Warm-up) Shot Charts

Guess the player (1 of 3):



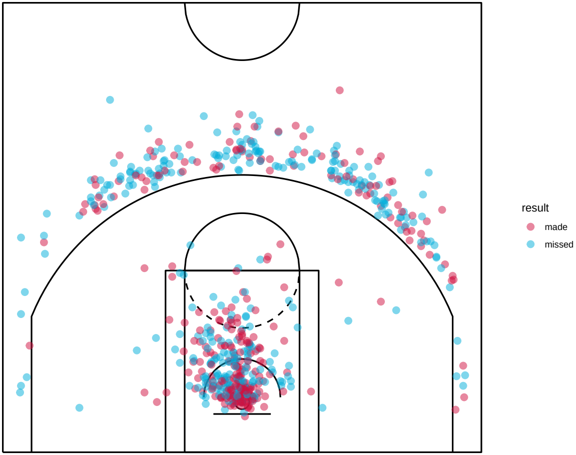
CS1: (Warm-up) Shot Charts (Cont.)

Guess the player (2 of 3):



CS1: (Warm-up) Shot Charts (Cont.)

Guess the player (3 of 3):



CS2: Deal or No Deal: Player Similarities

Multidimensional Scaling (MDS) is a "nonlinear dimensionality reduction tool that allows [us] to plot a map visualizing the level of similarity of individual cases [within] a dataset".

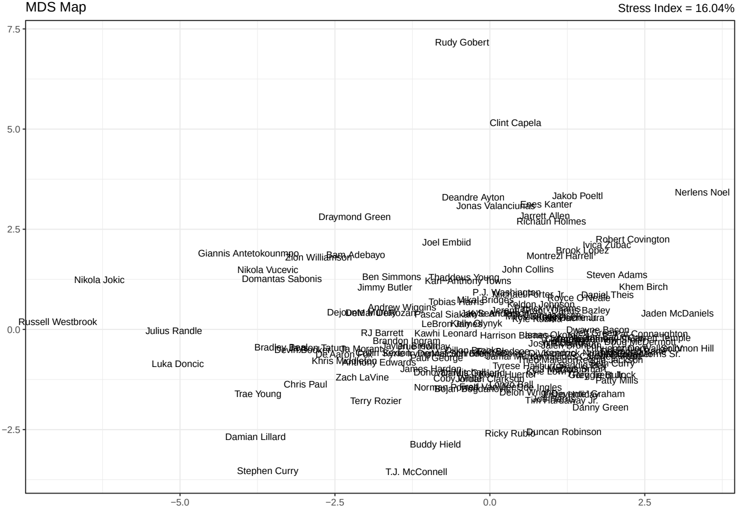
We start with a distance matrix $\mathbf{D}^p = (d_{ij})_{i,j=1,\dots,N}$ based on all p variables, $X_1, \dots, X_p$ and attempt to find $q << p$ such that $\mathbf{D}^q$ fits as closely as possible to $\mathbf{D}^p$.

A standard measure of distance is Euclidean distance,

$$d_{ij} = \sqrt{\sum_{h=1}^{p} (x_{ih} - x_{jh})^2}$$

The "Stress Index" (S) allows us to assess how close $\mathbf{D}^q$ approximates $\mathbf{D}^p$; 0.00% is a perfect fit, and we should avoid $S > 20\%$.

CS2: Deal or No Deal: Player Similarities (Cont.)



Original variable dimension (8): PTS, P3M, P2M, REB, AST, TOV, STL, BLK reduced to two dimensions. Restricted to players with over 1,500 minutes.

CS2: Deal or No Deal: Player Similarities (Cont.)

- Rudy Gobert (5YR, $205M, Avg: $41M) vs. Clint Capela (5YR, $90M, Avg: $18M)
- Luka Doncic (5YR, $207M, Avg: $41.4M) vs. Julius Randle (4YR, $117M, Avg: $29.2M)
- Buddy Hield (4YR, $94M, Avg: $23.5M) vs. Duncan Robinson (5YR, $89.9M, Avg: $17.9M)
- The NBA is a salary cap league; any “savings" can be an advantage!

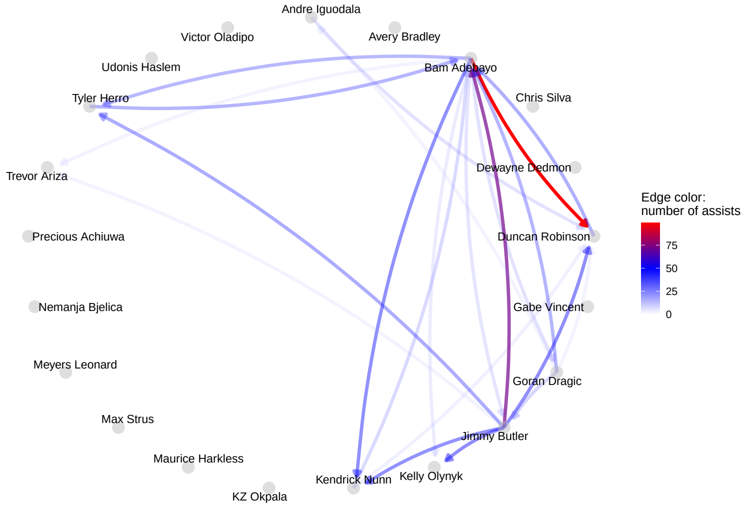
CS3: I'm Open: Assist Networks

We can employ *network analysis*, in which we construct and analyze graphs consisting of nodes related to each other by a set of attributes. This will allow us to find symmetric or asymmetric relationships between discrete objects.

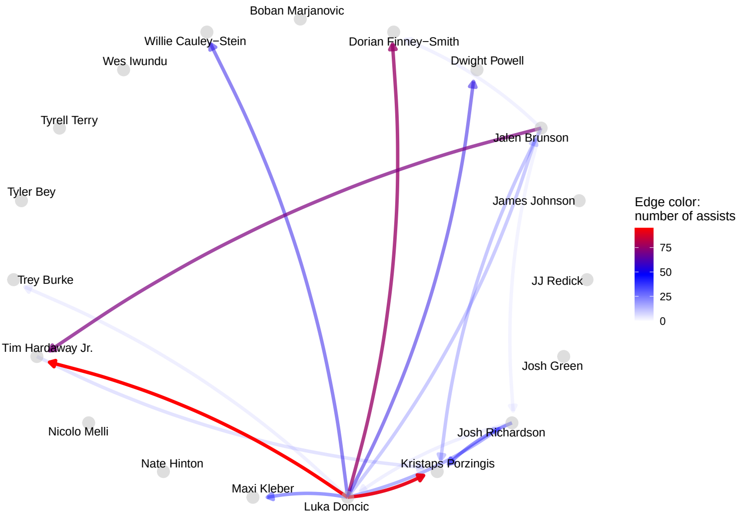
Our nodes/discrete objects will be players, and we will build an assist-network in hopes of better understanding the roles of each player within an opponent's offense.

Note: the underlying data will be "play-by-play" data.

CS3: I'm Open: Assist Networks (Cont.)



CS3: I'm Open: Assist Networks (Cont.)



CS4: ‘Tanks’ For Nothing: Clustering NBA teams

The NBA uses a weighted lottery system to determine draft selection order. The worse a team’s record from the previous season, the higher its odds at receiving a high draft pick. To take advantage of this, some teams have employed a ‘tanking strategy’, in which a team purposefully employs a weak roster in hopes of getting a high draft pick in the upcoming draft. Can we use statistics to help a team determine its strategy?

Figure 2: “The quickest way to win is to lose.” - Sam Hinkie

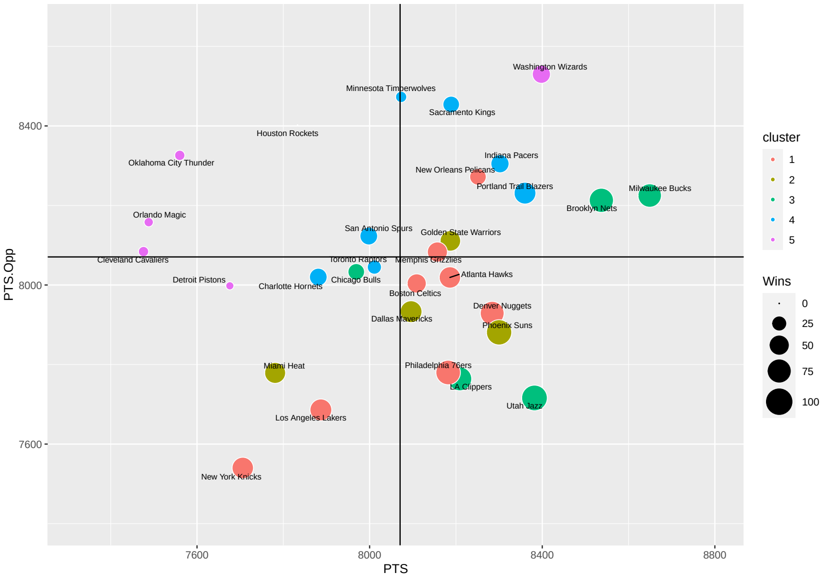
CS4: Cluster Analysis of NBA teams (Cont.)

Cluster Analysis is a classification technique used to divide individual cases into groups (clusters) such that each case within a cluster is "similar" (according to a given criterion) yet "different" from the cases in other clusters. Cluster Analysis is an *unsupervised* classification technique.

Here we employ a specific technique of Cluster Analysis, k-means clustering, to NBA teams based on the "four factors".

[see Ch. 4 of *Basketball Data Science* for details]

CS4: Cluster Analysis of NBA teams (Cont.)



CS4: Cluster Analysis of NBA teams (Cont.)

```
## [1] "Atlanta Hawks"        "Boston Celtics"       "Denver Nuggets"
## [4] "Los Angeles Lakers"   "Memphis Grizzlies"    "New Orleans Pelicans"
## [7] "New York Knicks"      "Philadelphia 76ers"
## ------------------------------------------------------------
## [1] "Dallas Mavericks"      "Golden State Warriors" "Miami Heat"
## [4] "Phoenix Suns"
## ------------------------------------------------------------
## [1] "Brooklyn Nets"   "Chicago Bulls"   "LA Clippers"     "Milwaukee Bucks"
## [5] "Utah Jazz"
## ------------------------------------------------------------
## [1] "Charlotte Hornets"      "Indiana Pacers"         "Minnesota Timberwolves"
## [4] "Portland Trail Blazers" "Sacramento Kings"       "San Antonio Spurs"
## [7] "Toronto Raptors"
## ------------------------------------------------------------
## [1] "Cleveland Cavaliers"   "Detroit Pistons"       "Houston Rockets"
## [4] "Oklahoma City Thunder" "Orlando Magic"         "Washington Wizards"
```

CS5: Rack Attack: Expected Points by Shot Distance

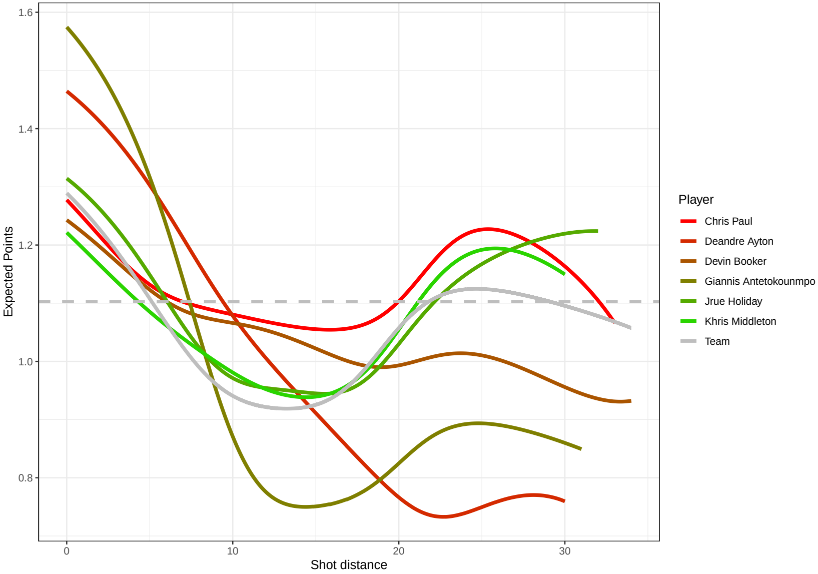
We briefly introduce the concept of *expected value*. Suppose we have a discrete random variable, X over a sample space, $\mathcal{X}$. We may define the expected value, $E(X)$, as

$$E(X) = \sum_{\mathcal{X}} x * P(X = x)$$

For example, if a player shoots 45% on 2-point FG's, his expected value per 2-point shot is

$$P(X = \text{Make}) * 2 + P(X = \text{Miss}) * 0 = (45\%)(2) + (55\%)(0) = 0.9$$

CS5: Rack Attack: Expected Points by Shot Distance (Cont.)



Summary

- `BasketballAnalyzeR`
- Getting data (very important!)
- Case studies to review
  - data visualization (shot charts)
  - dimension reduction techniques (player similarities)
  - network analysis (assist networks)
  - machine learning (clustering NBA teams)
  - law of large numbers (exp. pts by shot distance)