Post01: Rank NBA players by PCA, and analyze the results using PCA visualization package: factoextra

Sicheng Ouyang

2017/10/29

Introduction:

- In stat133 lecture, we've learned how to conduct a PCA (Principle Component Analysis) to develop a ranking system for objects with multidimensional variables. As a basketball fan, in this post, I want to illustrate how to extract the data from the internet and develop a reliable ranking system for NBA players in 16-17 season. Also, I will introduce some new statisics other than those we have used in HW2 to evaluate the performance of a basketball player. After that, I will rank the NBA players and find out the most important factors to determine a good player by visualizing and analysing the PCA result using a new package called *"factoextra"*.

NBA Players

---

1) First at all, we need to load some needed packages

```
#load pakages
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

---

2) Data preparation

In the data preparation stage, we need to consider: “What variables we need to use to evaluate the NBA players?”

- We need some basic statistics which can directly reflect players’ performance on the court, such as total points a player got, total number of assists or steals he made.
- Other than the basic stats, we need some advanced statistic to show players’ contribution to the team, such as PER(Player Efficiency Rating), WS(Win Shares), TS%(True Shooting Percentage)…

Here is the brief introduction of some of the advanced stats:

1. PER(Player Efficiency Rating): PER is the index to measure a player’s per-minute productivity. It adds up all the positive contributions a player makes to his team, while subtracting the negative ones in a statistical point value system.
2. WS(Win Shares): Win shares estimate an individual player’s contribution to their team’s win total. It equals offensive winshares + deffensive winshares.
3. TS%(True Shooting Percentage) True shooting percentage is a statistic that measures a player’s efficiency at shooting the ball. It is intended to more accurately calculate a player’s shooting than field goal percentage, free throw percentage, and three-point field goal percentage taken individually. Two and three-point field goals and free throws are all considered in its calculation.

The formula for TS% is : $$TS\% = \frac{PTS}{2(FGA + (0.44 \times FTA))}$$

For more details about advanced stats, please go to:

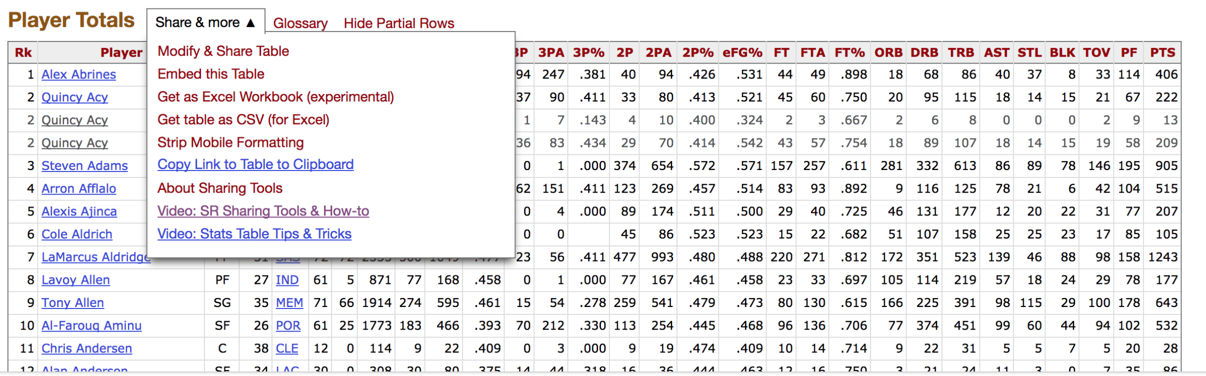
http://bleacherreport.com/articles/1813902-advanced-nba-stats-for-dummies-how-to-understand-the-new-hoops-math and https://www.basketball-reference.com/about/glossary.html

---

3) Data gathering:

After deciding what variable we need to use, we need to extract the data set from internet. Here I suggest a website called “Sports Reference”.

- Here’s the Link: https://www.sports-reference.com/
- We can find all kinds of data for many different sports, and we can store the data as csv file for later analyze. For example, when I go to the basic stat page for NBA players in 16-17, I can click on the “Share & More” button for editing or saving the data.



reference

- Here’s also some tips for you to download the file: https://www.youtube.com/watch?v=MWapXbaWs_U&feature=youtu.be and https://www.youtube.com/watch?v=JkDLV0roT14&feature=youtu.be

- I hope all the sports lovers can learn how to discover the data they want and doing their own research on R, after reading this post.

4)Importing data:

Import basic data for players:

```
# importing basic data for players
dat1 <- data.frame(read.csv('/Users/cosy/stat133/stat133-hws-fall17/post01/data/basic-stats.csv', stringsAsFactors
= FALSE))
str(dat1)
```

```
## 'data.frame':    486 obs. of  29 variables:
##  $ Player : chr  "Alex Abrines" "Quincy Acy" "Steven Adams" "Arron Afflalo" ...
##  $ Pos    : chr  "SG" "PF" "C" "SG" ...
##  $ Age    : int  23 26 23 31 28 28 31 27 35 26 ...
##  $ Tm     : chr  "OKC" "BRK" "OKC" "SAC" ...
##  $ G      : int  68 32 80 61 39 62 72 61 71 61 ...
##  $ GS     : int  6 1 80 45 15 0 72 5 66 25 ...
##  $ MP     : int  1055 510 2389 1580 584 531 2335 871 1914 1773 ...
##  $ FG     : int  134 65 374 185 89 45 500 77 274 183 ...
##  $ FGA    : int  341 153 655 420 178 86 1049 168 595 466 ...
##  $ FG_PER : num  0.393 0.425 0.571 0.44 0.5 0.523 0.477 0.458 0.461 0.393 ...
##  $ P3     : int  94 36 0 62 0 0 23 0 15 70 ...
##  $ P3A    : int  247 83 1 151 4 0 56 1 54 212 ...
##  $ P3_PER : num  0.381 0.434 0 0.411 0 NA 0.411 0 0.278 0.33 ...
##  $ P2     : int  40 29 374 123 89 45 477 77 259 113 ...
##  $ P2A    : int  94 70 654 269 174 86 993 167 541 254 ...
##  $ P2_PER : num  0.426 0.414 0.572 0.457 0.511 0.523 0.48 0.461 0.479 0.445 ...
##  $ eFG_PER: num  0.531 0.542 0.571 0.514 0.5 0.523 0.488 0.458 0.473 0.468 ...
##  $ FT     : int  44 43 157 83 29 15 220 23 80 96 ...
##  $ FTA    : int  49 57 257 93 40 22 271 33 130 136 ...
##  $ FT_PER : num  0.898 0.754 0.611 0.892 0.725 0.682 0.812 0.697 0.615 0.706 ...
##  $ ORB    : int  18 18 281 9 46 51 172 105 166 77 ...
##  $ DRB    : int  68 89 332 116 131 107 351 114 225 374 ...
##  $ TRB    : int  86 107 613 125 177 158 523 219 391 451 ...
##  $ AST    : int  40 18 86 78 12 25 139 57 98 99 ...
##  $ STL    : int  37 14 89 21 20 25 46 18 115 60 ...
##  $ BLK    : int  8 15 78 6 22 23 88 24 29 44 ...
##  $ TOV    : int  33 19 146 42 31 17 98 29 100 94 ...
##  $ PF     : int  114 58 195 104 77 85 158 78 178 102 ...
##  $ PTS    : int  406 209 905 515 207 105 1243 177 643 532 ...
```

```
head(dat1,10)
```

```
##               Player Pos Age  Tm  G GS   MP  FG  FGA FG_PER P3 P3A P3_PER
## 1       Alex Abrines  SG  23 OKC 68  6 1055 134  341  0.393 94 247  0.381
## 2         Quincy Acy  PF  26 BRK 32  1  510  65  153  0.425 36  83  0.434
## 3       Steven Adams   C  23 OKC 80 80 2389 374  655  0.571  0   1  0.000
## 4      Arron Afflalo  SG  31 SAC 61 45 1580 185  420  0.440 62 151  0.411
## 5      Alexis Ajinca   C  28 NOP 39 15  584  89  178  0.500  0   4  0.000
## 6       Cole Aldrich   C  28 MIN 62  0  531  45   86  0.523  0   0     NA
## 7  LaMarcus Aldridge  PF  31 SAS 72 72 2335 500 1049  0.477 23  56  0.411
## 8        Lavoy Allen  PF  27 IND 61  5  871  77  168  0.458  0   1  0.000
## 9         Tony Allen  SG  35 MEM 71 66 1914 274  595  0.461 15  54  0.278
## 10   Al-Farouq Aminu  SF  26 POR 61 25 1773 183  466  0.393 70 212  0.330
##     P2 P2A P2_PER eFG_PER  FT FTA FT_PER ORB DRB TRB AST STL BLK TOV  PF
## 1   40  94  0.426   0.531  44  49  0.898  18  68  86  40  37   8  33 114
## 2   29  70  0.414   0.542  43  57  0.754  18  89 107  18  14  15  19  58
## 3  374 654  0.572   0.571 157 257  0.611 281 332 613  86  89  78 146 195
## 4  123 269  0.457   0.514  83  93  0.892   9 116 125  78  21   6  42 104
## 5   89 174  0.511   0.500  29  40  0.725  46 131 177  12  20  22  31  77
## 6   45  86  0.523   0.523  15  22  0.682  51 107 158  25  25  23  17  85
## 7  477 993  0.480   0.488 220 271  0.812 172 351 523 139  46  88  98 158
## 8   77 167  0.461   0.458  23  33  0.697 105 114 219  57  18  24  29  78
## 9  259 541  0.479   0.473  80 130  0.615 166 225 391  98 115  29 100 178
## 10 113 254  0.445   0.468  96 136  0.706  77 374 451  99  60  44  94 102
##     PTS
## 1   406
## 2   209
## 3   905
## 4   515
## 5   207
## 6   105
## 7  1243
## 8   177
## 9   643
## 10  532
```

Pick the basic variables

```
#Pick the needed basic variables
dat11<-select(dat1,Player,Tm,G,GS,MP,FGA,FG_PER,P3,P3A,P3_PER,P2,P2A,P2_PER,FT,FTA,FT_PER,ORB,DRB,TRB,AST,STL,BLK,
TOV,PTS)
head(dat11,10)
```

```
##               Player  Tm  G GS   MP  FGA FG_PER P3 P3A P3_PER  P2 P2A
## 1       Alex Abrines OKC 68  6 1055  341  0.393 94 247  0.381  40  94
## 2         Quincy Acy BRK 32  1  510  153  0.425 36  83  0.434  29  70
## 3       Steven Adams OKC 80 80 2389  655  0.571  0   1  0.000 374 654
## 4      Arron Afflalo SAC 61 45 1580  420  0.440 62 151  0.411 123 269
## 5      Alexis Ajinca NOP 39 15  584  178  0.500  0   4  0.000  89 174
## 6       Cole Aldrich MIN 62  0  531   86  0.523  0   0     NA  45  86
## 7  LaMarcus Aldridge SAS 72 72 2335 1049  0.477 23  56  0.411 477 993
## 8        Lavoy Allen IND 61  5  871  168  0.458  0   1  0.000  77 167
## 9         Tony Allen MEM 71 66 1914  595  0.461 15  54  0.278 259 541
## 10   Al-Farouq Aminu POR 61 25 1773  466  0.393 70 212  0.330 113 254
##    P2_PER  FT FTA FT_PER ORB DRB TRB AST STL BLK TOV  PTS
## 1   0.426  44  49  0.898  18  68  86  40  37   8  33  406
## 2   0.414  43  57  0.754  18  89 107  18  14  15  19  209
## 3   0.572 157 257  0.611 281 332 613  86  89  78 146  905
## 4   0.457  83  93  0.892   9 116 125  78  21   6  42  515
## 5   0.511  29  40  0.725  46 131 177  12  20  22  31  207
## 6   0.523  15  22  0.682  51 107 158  25  25  23  17  105
## 7   0.480 220 271  0.812 172 351 523 139  46  88  98 1243
## 8   0.461  23  33  0.697 105 114 219  57  18  24  29  177
## 9   0.479  80 130  0.615 166 225 391  98 115  29 100  643
## 10  0.445  96 136  0.706  77 374 451  99  60  44  94  532
```

Import advanced data

```
# import advanced data
dat2 <- data.frame(read.csv('/Users/cosy/stat133/stat133-hws-fall17/post01/data/player-adv-stats.csv',stringsAsFac
tors = FALSE))
str(dat2)
```

```
## 'data.frame':    486 obs. of  26 variables:
##  $ Player  : chr  "Alex Abrines" "Quincy Acy" "Steven Adams" "Arron Afflalo" ...
##  $ Pos     : chr  "SG" "PF" "C" "SG" ...
##  $ Age     : int  23 26 23 31 28 28 31 27 35 26 ...
##  $ Tm      : chr  "OKC" "BRK" "OKC" "SAC" ...
##  $ G       : int  68 32 80 61 39 62 72 61 71 61 ...
##  $ MP      : int  1055 510 2389 1580 584 531 2335 871 1914 1773 ...
##  $ PER     : num  10.1 13.1 16.5 8.9 12.9 12.7 18.6 11.6 13.3 11.3 ...
##  $ TS_PER  : num  0.56 0.587 0.589 0.559 0.529 0.549 0.532 0.485 0.493 0.506 ...
##  $ P3Ar    : num  0.724 0.542 0.002 0.36 0.022 0 0.053 0.006 0.091 0.455 ...
##  $ FTr     : num  0.144 0.373 0.392 0.221 0.225 0.256 0.258 0.196 0.218 0.292 ...
##  $ ORB_RATE: num  1.9 3.8 13 0.7 8.3 11 8.5 13.7 9.6 4.8 ...
##  $ DRB_RATE: num  7.1 18.2 15.4 8.4 23.8 23.9 16.6 14.5 13.8 23.5 ...
##  $ TRB_RATE: num  4.5 11.1 14.2 4.6 16 17.4 12.7 14.1 11.7 14.1 ...
##  $ AST_RATE: num  5.5 5.4 5.4 7.4 3.1 6.4 9.9 9.1 8.4 7.9 ...
##  $ STL_RATE: num  1.7 1.3 1.8 0.7 1.7 2.4 1 1 3.1 1.7 ...
##  $ BLK_RATE: num  0.6 2.2 2.6 0.3 3.1 3.7 3 2.4 1.4 2 ...
##  $ TOV_RATE: num  8.3 9.6 16 8.4 13.7 15.1 7.7 13.7 13.3 15.2 ...
##  $ USG_RATE: num  15.9 16.5 16.2 14.4 17.2 9.4 24.5 10.9 17.9 15.4 ...
##  $ OWS     : num  1.2 0.6 3.3 1.2 0 0.6 3.5 0.9 0.2 -0.1 ...
##  $ DWS     : num  0.9 0.5 3.1 0.2 0.9 0.7 3.7 0.8 2.9 2 ...
##  $ WS      : num  2.1 1.1 6.5 1.4 1 1.3 7.2 1.7 3.1 1.9 ...
##  $ WS.48   : num  0.096 0.102 0.13 0.043 0.08 0.116 0.149 0.093 0.077 0.051 ...
##  $ OBPM    : num  -0.3 -1.1 -0.7 -1.4 -5.1 -2 -0.3 -1.5 -1.8 -2.3 ...
##  $ DBPM    : num  -2.2 -0.7 1.2 -2.1 1 2.6 1.3 1.3 2.4 1.2 ...
##  $ BPM     : num  -2.5 -1.8 0.6 -3.5 -4.1 0.6 1 -0.3 0.6 -1.1 ...
##  $ VORP    : num  -0.1 0 1.5 -0.6 -0.3 0.4 1.8 0.4 1.3 0.4 ...
```

```r
head(dat2,10)
```

```
##                Player Pos Age  Tm  G   MP  PER TS_PER  P3Ar   FTr ORB_RATE
## 1        Alex Abrines  SG  23 OKC 68 1055 10.1  0.560 0.724 0.144      1.9
## 2          Quincy Acy  PF  26 BRK 32  510 13.1  0.587 0.542 0.373      3.8
## 3        Steven Adams   C  23 OKC 80 2389 16.5  0.589 0.002 0.392     13.0
## 4       Arron Afflalo  SG  31 SAC 61 1580  8.9  0.559 0.360 0.221      0.7
## 5       Alexis Ajinca   C  28 NOP 39  584 12.9  0.529 0.022 0.225      8.3
## 6        Cole Aldrich   C  28 MIN 62  531 12.7  0.549 0.000 0.256     11.0
## 7  LaMarcus Aldridge  PF  31 SAS 72 2335 18.6  0.532 0.053 0.258      8.5
## 8         Lavoy Allen  PF  27 IND 61  871 11.6  0.485 0.006 0.196     13.7
## 9          Tony Allen  SG  35 MEM 71 1914 13.3  0.493 0.091 0.218      9.6
## 10   Al-Farouq Aminu  SF  26 POR 61 1773 11.3  0.506 0.455 0.292      4.8
##    DRB_RATE TRB_RATE AST_RATE STL_RATE BLK_RATE TOV_RATE USG_RATE  OWS DWS
## 1       7.1      4.5      5.5      1.7      0.6      8.3     15.9  1.2 0.9
## 2      18.2     11.1      5.4      1.3      2.2      9.6     16.5  0.6 0.5
## 3      15.4     14.2      5.4      1.8      2.6     16.0     16.2  3.3 3.1
## 4       8.4      4.6      7.4      0.7      0.3      8.4     14.4  1.2 0.2
## 5      23.8     16.0      3.1      1.7      3.1     13.7     17.2  0.0 0.9
## 6      23.9     17.4      6.4      2.4      3.7     15.1      9.4  0.6 0.7
## 7      16.6     12.7      9.9      1.0      3.0      7.7     24.5  3.5 3.7
## 8      14.5     14.1      9.1      1.0      2.4     13.7     10.9  0.9 0.8
## 9      13.8     11.7      8.4      3.1      1.4     13.3     17.9  0.2 2.9
## 10     23.5     14.1      7.9      1.7      2.0     15.2     15.4 -0.1 2.0
##     WS WS.48 OBPM DBPM  BPM VORP
## 1  2.1 0.096 -0.3 -2.2 -2.5 -0.1
## 2  1.1 0.102 -1.1 -0.7 -1.8  0.0
## 3  6.5 0.130 -0.7  1.2  0.6  1.5
## 4  1.4 0.043 -1.4 -2.1 -3.5 -0.6
## 5  1.0 0.080 -5.1  1.0 -4.1 -0.3
## 6  1.3 0.116 -2.0  2.6  0.6  0.4
## 7  7.2 0.149 -0.3  1.3  1.0  1.8
## 8  1.7 0.093 -1.5  1.3 -0.3  0.4
## 9  3.1 0.077 -1.8  2.4  0.6  1.3
## 10 1.9 0.051 -2.3  1.2 -1.1  0.4
```

Pick advanced variables.

```r
#Pick the need advanced variables.
dat22 <- select(dat2,Player,PER,TS_PER,OWS,DWS,WS)
head(dat22,10)
```

```
##                Player  PER TS_PER  OWS DWS  WS
## 1        Alex Abrines 10.1  0.560  1.2 0.9 2.1
## 2          Quincy Acy 13.1  0.587  0.6 0.5 1.1
## 3        Steven Adams 16.5  0.589  3.3 3.1 6.5
## 4       Arron Afflalo  8.9  0.559  1.2 0.2 1.4
## 5       Alexis Ajinca 12.9  0.529  0.0 0.9 1.0
## 6        Cole Aldrich 12.7  0.549  0.6 0.7 1.3
## 7  LaMarcus Aldridge 18.6  0.532  3.5 3.7 7.2
## 8         Lavoy Allen 11.6  0.485  0.9 0.8 1.7
## 9          Tony Allen 13.3  0.493  0.2 2.9 3.1
## 10   Al-Farouq Aminu 11.3  0.506 -0.1 2.0 1.9
```

Merge two data frame

```
mer_dat <- merge(dat11, dat22)
head(mer_dat,10)
```

```
##              Player  Tm  G GS   MP FGA FG_PER P3 P3A P3_PER  P2 P2A P2_PER
## 1      A.J. Hammons DAL 22  0  163  42  0.405  5  10  0.500  12  32  0.375
## 2      Aaron Brooks IND 65  0  894 300  0.403 48 128  0.375  73 172  0.424
## 3      Aaron Gordon ORL 80 72 2298 865  0.454 77 267  0.288 316 598  0.528
## 4    Aaron Harrison CHO  5  0   17   4  0.000  0   2  0.000   0   2  0.000
## 5     Adreian Payne MIN 18  0  135  54  0.426  3  15  0.200  20  39  0.513
## 6        Al Horford BOS 68 68 2193 801  0.473 86 242  0.355 293 559  0.524
## 7      Al Jefferson IND 66  1  931 471  0.499  0   1  0.000 235 470  0.500
## 8   Al-Farouq Aminu POR 61 25 1773 466  0.393 70 212  0.330 113 254  0.445
## 9     Alan Anderson LAC 30  0  308  80  0.375 14  44  0.318  16  36  0.444
## 10    Alan Williams PHO 47  0  708 267  0.517  0   1  0.000 138 266  0.519
##       FT FTA FT_PER ORB DRB TRB AST STL BLK TOV  PTS  PER TS_PER  OWS DWS
## 1      9  20  0.450   8  28  36   4   1  13  10   48  8.4  0.472 -0.2 0.2
## 2     32  40  0.800  18  51  69 125  25   9  66  322  9.5  0.507 -0.2 0.5
## 3    156 217  0.719 116 289 405 150  65  40  89 1019 14.5  0.530  2.0 1.7
## 4      1   2  0.500   0   3   3   3   0   0   0    1 -2.2  0.102 -0.1 0.0
## 5     14  19  0.737   9  24  33   7   8   7   8   63 14.4  0.505  0.0 0.2
## 6    108 135  0.800  95 370 465 337  52  86 115  952 17.7  0.553  3.6 2.7
## 7     65  85  0.765  75 203 278  57  19  16  33  535 18.9  0.526  1.2 1.1
## 8     96 136  0.706  77 374 451  99  60  44  94  532 11.3  0.506 -0.1 2.0
## 9     12  16  0.750   3  21  24  11   3   0   7   86  5.0  0.494  0.0 0.1
## 10    70 112  0.625  94 198 292  23  27  32  37  346 19.5  0.547  1.1 0.9
##        WS
## 1     0.0
## 2     0.3
## 3     3.7
## 4    -0.1
## 5     0.2
## 6     6.3
## 7     2.3
## 8     1.9
## 9     0.1
## 10    2.1
```

*Last but not least, we need one more variable, which is the number of wins the players' team got in the season. It's important for evaluating a player because a good player not only needs to have a good performance on the court, but also needs to make his team better and lead his team to win as many games as possible.

Import the number of wins for each team

```
dat_win<-data.frame(read.csv('/Users/cosy/stat133/stat133-hws-fall17/post01/data/team_win.csv'))
dat_win
```

```
##    Team Win
## 1   GSW  67
## 2   SAS  61
## 3   HOU  55
## 4   BOS  53
## 5   CLE  51
## 6   LAC  51
## 7   TOR  51
## 8   UTA  51
## 9   WAS  49
## 10  OKC  47
## 11  ATL  43
## 12  MEM  43
## 13  IND  42
## 14  MIL  42
## 15  CHI  41
## 16  POR  41
## 17  MIA  41
## 18  DEN  40
## 19  DET  37
## 20  CHO  36
## 21  NOP  34
## 22  DAL  33
## 23  SAC  32
## 24  MIN  31
## 25  NYK  31
## 26  ORL  29
## 27  PHI  28
## 28  LAL  26
## 29  PHO  24
## 30  BRK  20
```

In order to add the number of win for players' team in the big data frame, we need to write a for loop:

```r
#add a win col in mer_dat and write a for loop to input the number of wins for each player
mer_dat$Win<-0

for(i in 1:486){
        if(mer_dat[i,"Tm"]=='GSW'){
                mer_dat[i,"Win"]<-67
        }else if(mer_dat[i,"Tm"]=='SAS'){
                mer_dat[i,"Win"]<-61
        }else if(mer_dat[i,"Tm"]=='HOU'){
                mer_dat[i,"Win"]<-55
        }else if(mer_dat[i,"Tm"]=='BOS'){
                mer_dat[i,"Win"]<-53
        }else if(mer_dat[i,"Tm"]=='CLE'){
                mer_dat[i,"Win"]<-51
        }else if(mer_dat[i,"Tm"]=='LAC'){
                mer_dat[i,"Win"]<-51
        }else if(mer_dat[i,"Tm"]=='TOR'){
                mer_dat[i,"Win"]<-51
        }else if(mer_dat[i,"Tm"]=='UTA'){
                mer_dat[i,"Win"]<-51
        }else if(mer_dat[i,"Tm"]=='WAS'){
                mer_dat[i,"Win"]<-49
        }else if(mer_dat[i,"Tm"]=='OKC'){
                mer_dat[i,"Win"]<-47
        }else if(mer_dat[i,"Tm"]=='ATL'){
                mer_dat[i,"Win"]<-43
        }else if(mer_dat[i,"Tm"]=='MEM'){
                mer_dat[i,"Win"]<-43
        }else if(mer_dat[i,"Tm"]=='IND'){
                mer_dat[i,"Win"]<-42
        }else if(mer_dat[i,"Tm"]=='MIL'){
                mer_dat[i,"Win"]<-42
        }else if(mer_dat[i,"Tm"]=='CHI'){
                mer_dat[i,"Win"]<-41
        }else if(mer_dat[i,"Tm"]=='POR'){
                mer_dat[i,"Win"]<-41
        }else if(mer_dat[i,"Tm"]=='MIA'){
                mer_dat[i,"Win"]<-41
        }else if(mer_dat[i,"Tm"]=='DEN'){
                mer_dat[i,"Win"]<-40
        }else if(mer_dat[i,"Tm"]=='DET'){
                mer_dat[i,"Win"]<-37
        }else if(mer_dat[i,"Tm"]=='CHO'){
                mer_dat[i,"Win"]<-36
        }else if(mer_dat[i,"Tm"]=='NOP'){
                mer_dat[i,"Win"]<-34
        }else if(mer_dat[i,"Tm"]=='DAL'){
                mer_dat[i,"Win"]<-33
        }else if(mer_dat[i,"Tm"]=='SAC'){
                mer_dat[i,"Win"]<-32
        }else if(mer_dat[i,"Tm"]=='MIN'){
                mer_dat[i,"Win"]<-31
        }else if(mer_dat[i,"Tm"]=='NYK'){
                mer_dat[i,"Win"]<-31
        }else if(mer_dat[i,"Tm"]=='ORL'){
                mer_dat[i,"Win"]<-29
        }else if(mer_dat[i,"Tm"]=='PHI'){
                mer_dat[i,"Win"]<-28
        }else if(mer_dat[i,"Tm"]=='LAL'){
                mer_dat[i,"Win"]<-26
        }else if(mer_dat[i,"Tm"]=='PHO'){
                mer_dat[i,"Win"]<-24
        }else if(mer_dat[i,"Tm"]=='BRK'){
                mer_dat[i,"Win"]<-20
        }
        i=i+1
}
```

If a player doesn’t have some specific stats (eg: Some players don’t shoot 3-pointer at all), we need to replace it with 0.

```r
mer_dat[is.na(mer_dat)] <- 0
```

Those players who didn’t play more than 40 games in a season are not taken into account in this reseach.

```r
mer_dat <- filter(mer_dat, G>40)
```

5)Let’s start doing Principle Components Analysis(PCA)

First, pick variables

```
pca_dat <- data.frame(select(
        mer_dat,
        G,
        GS,
        MP,
        FGA,
        FG_PER,
        P3,
        P3A,
        P3_PER,
        P2,
        P2A,
        P2_PER,
        FT,
        FTA,
        FT_PER,
        ORB,
        DRB,
        TRB,
        AST,
        STL,
        BLK,
        TOV,
        PTS,
        PER,
        TS_PER,
        OWS,
        DWS,
        WS,
        Win
        ), row.names = mer_dat$Player)
```

Here is the data dictionary for pca_dat:

column	description
G	total games played
GS	games played as a starter
MP	minute played
FGA	field goals attempts
FG_PER	field goals percentage
P3	three points field goals made
P3A	three points field goals attempts
P3_PER	three points field goals percentage
P2	two points field goals made
P2A	two points field goals attempts
P2_PER	two points field goals percentage
FT	free throw made
FTA	free throw attempts
FT_PER	free throw percentage
ORB	offensive rebounds
DRB	deffensive rebounds
TRB	totoal rebounds
PTS	total points
PER	player efficiency rating
TS_PER	true shooting percentage
OWS	offensive winshares
DWS	deffensive winshares
WS	winshares
Win	number of wins the player’s team had

Perform a principal components analysis (PCA) on the specified variables

```
pca <- prcomp(pca_dat, scale. = TRUE)
```

Now, instead of using the original way we've learned in the lecture to draw graphs and analyze the PCA result, I want to introduce a more convenient way to do it using a package called *"factoextra"*.

- Brief introduction: **Factoextra** is an R package to extract and visualize the output of exploratory multivariate data analyses, including: Principal Component Analysis (PCA), Multiple Correspondence Analysis (MCA), and Multiple Factor Analysis (MFA).
- Why should we use this?

1. The R package factoextra has flexible and easy-to-use methods to extract quickly, in a human-readable standard data format, the analysis results from the different packages mentioned above.
2. It produces a ggplot2-based elegant data visualization with less typing.
3. It also contains many functions facilitating clustering analysis and visualization.

- For more information: https://cran.r-project.org/web/packages/factoextra/factoextra.pdf and https://rdrr.io/cran/factoextra/man/fviz_pca.html
- You can find all the functions you can use in the *"factoextra" package in the above links.

Install and load factoextra

```
install.packages("factoextra")
library("factoextra")
```

*In order to see how many variances each principle component capture, we can extract eigenvalues/variances by calling "get_eig()" function.

Extract eigenvalues/variances

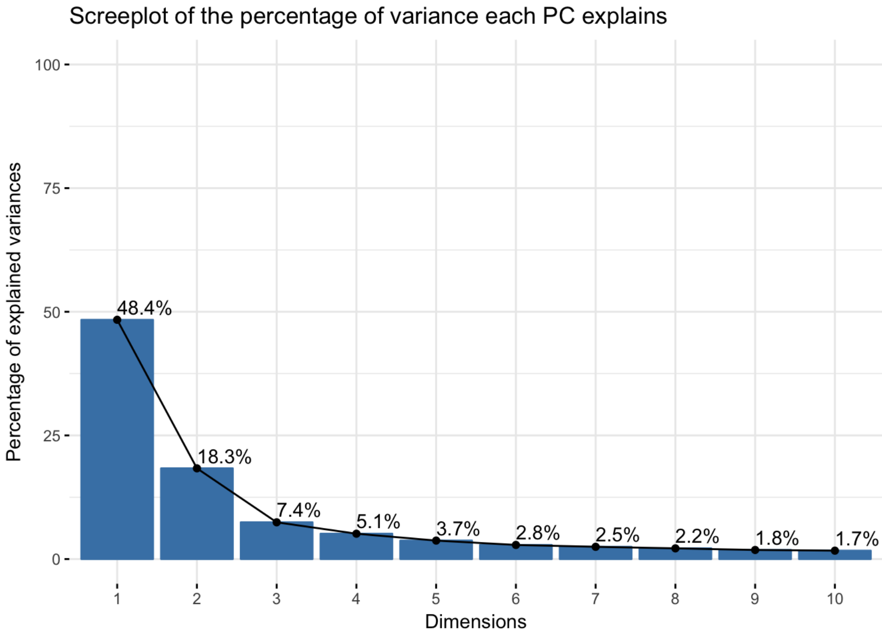
```
# Extract eigenvalues/variances
get_eig(pca)
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  1.354428e+01     4.837243e+01                    48.37243
## Dim.2  5.135653e+00     1.834162e+01                    66.71405
## Dim.3  2.079957e+00     7.428418e+00                    74.14247
## Dim.4  1.434263e+00     5.122368e+00                    79.26483
## Dim.5  1.045492e+00     3.733902e+00                    82.99874
## Dim.6  7.979645e-01     2.849873e+00                    85.84861
## Dim.7  6.926957e-01     2.473913e+00                    88.32252
## Dim.8  6.044340e-01     2.158693e+00                    90.48122
## Dim.9  5.137155e-01     1.834698e+00                    92.31591
## Dim.10 4.742784e-01     1.693852e+00                    94.00977
## Dim.11 3.626088e-01     1.295031e+00                    95.30480
## Dim.12 3.072831e-01     1.097440e+00                    96.40224
## Dim.13 2.803581e-01     1.001279e+00                    97.40351
## Dim.14 2.085215e-01     7.447198e-01                    98.14823
## Dim.15 1.490203e-01     5.322153e-01                    98.68045
## Dim.16 1.132207e-01     4.043598e-01                    99.08481
## Dim.17 1.005163e-01     3.589867e-01                    99.44380
## Dim.18 6.513467e-02     2.326238e-01                    99.67642
## Dim.19 3.723027e-02     1.329652e-01                    99.80939
## Dim.20 2.934652e-02     1.048090e-01                    99.91419
## Dim.21 1.328583e-02     4.744938e-02                    99.96164
## Dim.22 6.037328e-03     2.156189e-02                    99.98321
## Dim.23 3.440780e-03     1.228850e-02                    99.99549
## Dim.24 1.117650e-03     3.991609e-03                    99.99949
## Dim.25 1.439259e-04     5.140211e-04                   100.00000
## Dim.26 2.671138e-30     9.539780e-30                   100.00000
## Dim.27 8.759450e-32     3.128375e-31                   100.00000
## Dim.28 8.759450e-32     3.128375e-31                   100.00000
```

- In this case, Dim1…Dim28 stand for PC1…PC28. We can see that the most important component(PC1) almost captures half of the variance. And the first three PC can explain most of the variance.

We can draw the screeplot of the percentage of variance each PC explains by calling the function: "fviz_screeplot()""

```
#draw the screeplot of the percentage of variance each PC explains
fviz_screeplot(pca, addlabels = TRUE, ylim = c(0,100), title = "Screeplot of the percentage of variance each PC ex
plains")
```



- The x-axis for this graph is the principal components from PC1 to PC10, and the y-axis is the percentage of variance each PC explains. From this graph, we can easily see that the first 3 components explain most of the variances. So we decide to use these 3 PCs to rank NBA players. It should be convincing enough to use PC1, PC2 and PC3 as news variables to rank the NBA players because these three variable explain over 74% of the variance.

---

*As we know, each PC is a linear combination of all different variables, which can capture as many as variances as possible. In order to figure out what's the weight of each variable, we can extract the results for variables easily by calling "get_pca_var()" function.

```
#extract the results for variables
var <- get_pca_var(pca)
var
```

```
## Principal Component Analysis Results for variables
##  ==================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

- Noted that Coordinates for the variables is the same as the weights for the variables for each PC(which is the linear combination of variables).

Get weight for each variable for the first 3 PC

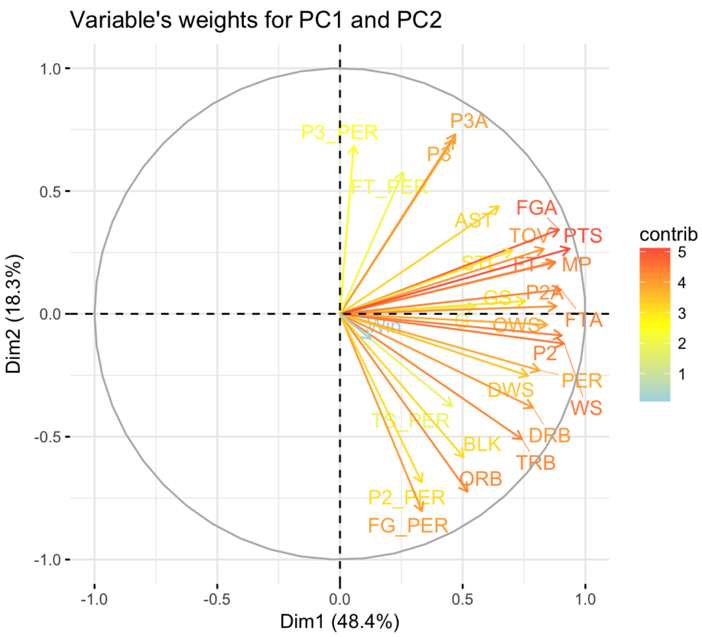
```
#Weight of variables
var$coord[,1:3]
```

```
##              Dim.1       Dim.2       Dim.3
## G       0.55441511  0.02903521 -0.13823633
## GS      0.75588549  0.05249235 -0.17383549
## MP      0.87688705  0.20887194 -0.16378226
## FGA     0.89267030  0.34314974 -0.07590817
## FG_PER 0.33418008 -0.80315558  0.36349350
## P3      0.46170142  0.70918160  0.29343132
## P3A     0.46984150  0.73100957  0.21755169
## P3_PER 0.05625096  0.68454180  0.21532060
## P2      0.90474777 -0.08804522 -0.15045197
## P2A     0.88260851  0.03113931 -0.21921742
## P2_PER 0.33384597 -0.68623056  0.44335376
## FT      0.87707269  0.21189748  0.02228110
## FTA     0.90045340  0.09892285 -0.03087405
## FT_PER 0.25727059  0.57674944  0.23370597
## ORB     0.52007346 -0.72384396 -0.24591404
## DRB     0.78659759 -0.38182104 -0.24373254
## TRB     0.74166177 -0.50998969 -0.25677823
## AST     0.64797994  0.43765127 -0.06101851
## STL     0.70268966  0.25800534 -0.16884059
## BLK     0.50438215 -0.58358935 -0.19898445
## TOV     0.83109715  0.26468251 -0.17861840
## PTS     0.93585784  0.26516387  0.02476767
## PER     0.81341323 -0.22988207  0.27290420
## TS_PER 0.45700476 -0.37529790  0.71831589
## OWS     0.84487871 -0.04425031  0.39639222
## DWS     0.76617932 -0.25155926 -0.14089444
## WS      0.91379581 -0.12114048  0.25566854
## Win     0.12435187 -0.10128374  0.48973658
```

- From this table, we can see what's the weight for each variable in different principal components. By looking at the table, we may have an idea about what variables contribute the most to each principal component.

We can also draw a graph for variables, showing their weights for PC1 and PC2 by calling the function: "fviz_pca_var()"

```
#draw a graph for variables
fviz_pca_var(pca, col.var="contrib",
             gradient.cols = c("light blue","yellow","tomato"),
             repel = TRUE, # Avoid text overlapping
             title = "Variable's weights for PC1 and PC2"
             )
```



- The color of each variable is related to its contribution to the principal component. If the color of the variable is closer to red, then it contributes more to the principle component comparing with the other variables. From this graph, we can see that PTS(total points), FTA(field goal attempt), and WS(winshare) are three of the variables that contribute the most to PC1 and PC2. On the other hand, the number of wins of the player's team contributes relatively less.

---

*In order to have a better understanding of what variables contribute the most to each principal component, and figure out what does each principle component mainly represent, we can check"var$contrib"

Get contribution of variables

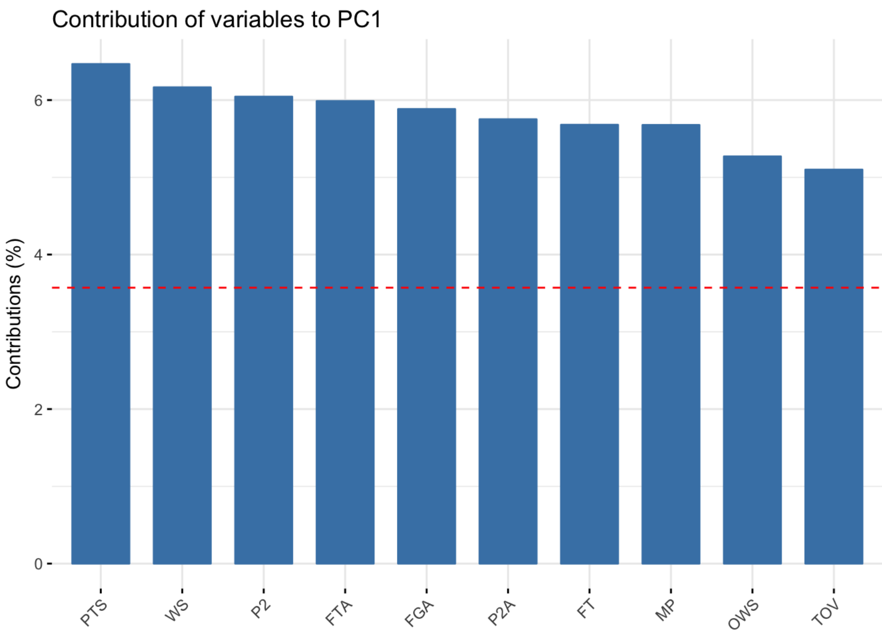
```
var$contrib[,1:3]
```

```
##            Dim.1      Dim.2       Dim.3
## G       2.26941635  0.01641550  0.91873458
## GS      4.21848081  0.05365329  1.45285587
## MP      5.67716315  0.84950221  1.28967234
## FGA     5.88337088  2.29282901  0.27702735
## FG_PER  0.82452754 12.56040620  6.35241644
## P3      1.57386136  9.79307841  4.13960210
## P3A     1.62984690 10.40520018  2.27546724
## P3_PER  0.02336167  9.12439881  2.22903467
## P2      6.04364711  0.15094401  1.08828184
## P2A     5.75148881  0.01888089  2.31044584
## P2_PER  0.82287965  9.16947397  9.45031855
## FT      5.67956715  0.87429080  0.02386816
## FTA     5.98641122  0.19054499  0.04582820
## FT_PER  0.48867973  6.47707131  2.62594278
## ORB     1.99697877 10.20220916  2.90745027
## DRB     4.56824389  2.83872967  2.85609509
## TRB     4.06121377  5.06438965  3.17002039
## AST     3.10003918  3.72958663  0.17900651
## STL     3.64561827  1.29616923  1.37056409
## BLK     1.87829359  6.63161078  1.90363604
## TOV     5.09973543  1.36412699  1.53390344
## PTS     6.46641869  1.36909317  0.02949280
## PER     4.88502201  1.02899799  3.58068478
## TS_PER  1.54200402  2.74256283 24.80713412
## OWS     5.27026911  0.03812737  7.55432917
## DWS     4.33415976  1.23221057  0.95440644
## WS      6.16513206  0.28574779  3.14268065
## Win     0.11416913  0.19974860 11.53110027
```

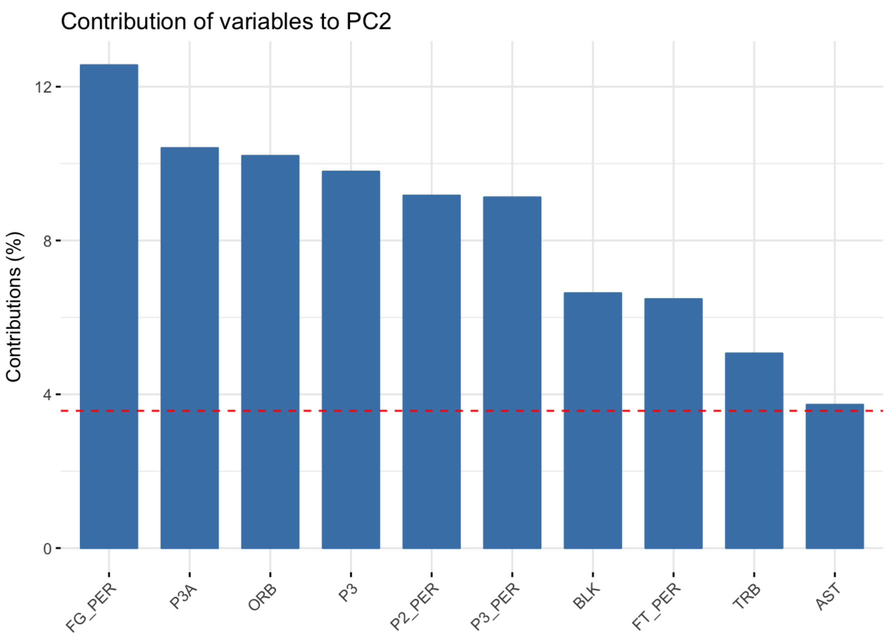
- From this table, we can see PTS, WS and P2 contribute the most to PC1.
- FG_PER and P3A contribute the most to PC2.
- TS_PER contribute the most to PC3

We can even draw a barchart by calling function: “fviz_contrib” to show the contribution of each variable more clearly

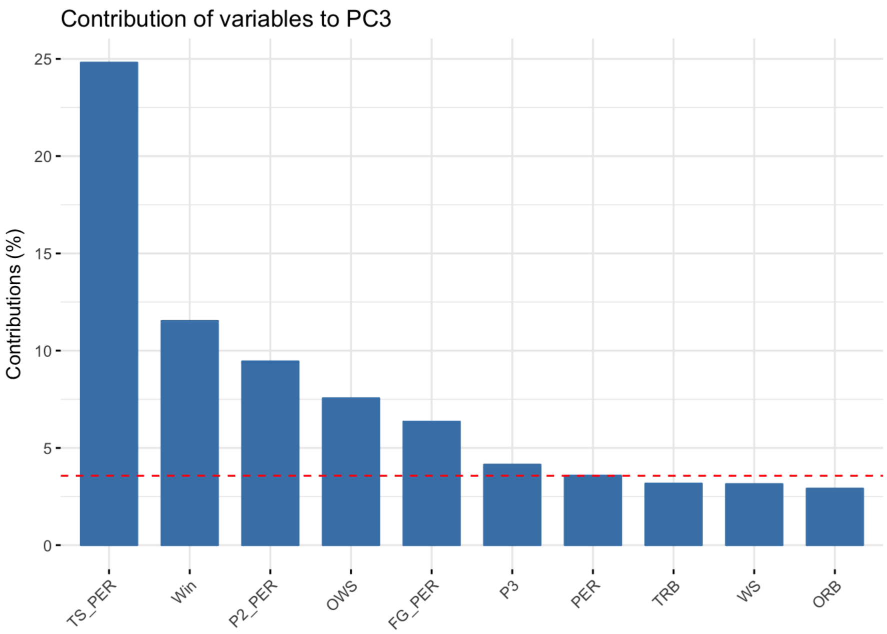
```
# Contributions of top 10 variables to PC1
fviz_contrib(pca, choice = "var", axes = 1, top = 10, title = "Contribution of variables to PC1")
```




```
# Contributions of top 10 variables to PC2
fviz_contrib(pca, choice = "var", axes = 2, top = 10, title = "Contribution of variables to PC2")
```




```r
# Contributions of top 10 variables to PC3
fviz_contrib(pca, choice = "var", axes = 3, top = 10, title = "Contribution of variables to PC3")
```



- From the bar chart, we can see
- For each graph, I display the top 10 variables that contribute the most to the Principle component.
- From the first graph, we can clearly see that PTS, WS and P2 contribute the most to PC1.
- From the second graph, it's not hard to find that FG_PER and P3A contribute the most to PC2.
- According to the third graph, TS_PER contribute the most to PC3.

Interpretation of the PCs.

- Noted that we've reduced the 28 variables to only 3 new variables, which are PC1, PC2 and PC3. And we need to give an interpretation of these 3 new variables and explain what they really mean in terms of ranking the NBA player.
- Interpretation of P1: From the graph we drew in the last part, we found that total points(PTS), winshare(WS) contribute the most to PC1. In addition, in the top 10 variables that contribute to PC1, the variables of different kinds of field goal attempts(FTA,FGA,P2A) are very common. That shows that an NBA player with very high ranking must score many points and make a lot of contribution to the team's win. As we know, high score is always supported by high field goal attempts. However, only the leader or the star in the team is allowed to have such a high field goal attempts. Therefore, we can conclude that the most important factor(PC1) to evaluate an NBA player is whether the player is in the dominant position in the team and his scoring ability.
- Interpretation of P2: According the graph for P2, FG_PER and P3A contribute the most to PC2. Among the top 10 variables, the variables for all kinds of shooting percentage are very common. That means players who have high shooting tendency(especially three pointers) and very high shoot percentage will perform better on the court. Therefore, the second most important factor(PC2) to evaluate an NBA player is their shooting skills(especially 3 pointers).
- Interpretation of P3: From the third graph, we can see that the true shooting percentage(TS_PER) has the dominant position among the top 10 variables. That means P3 is a variable about a player's efficiency.

- In conclusion, the most important factor(PC1) to evaluate the player is whether the player is in the dominant position in the team and his scoring ability, the second factor(PC2) is his shooting skills(especially 3 pointers), and the third factor(PC3) is the player's efficiency.

After analyzing the variables, we need to extract and visualize the results for individuals.

Extract the results for individuals by calling "get_pca_ind()"

```
# Extract the results for individuals
ind <- get_pca_ind(pca)
ind
```

```
## Principal Component Analysis Results for individuals
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the individuals"
## 2 "$cos2"    "Cos2 for the individuals"
## 3 "$contrib" "contributions of the individuals"
```

- Noted that the "Coordinates for the individuals" is the same as the score for the individuals for each PC.

We can check the player's score for each PC using "ind$coord"

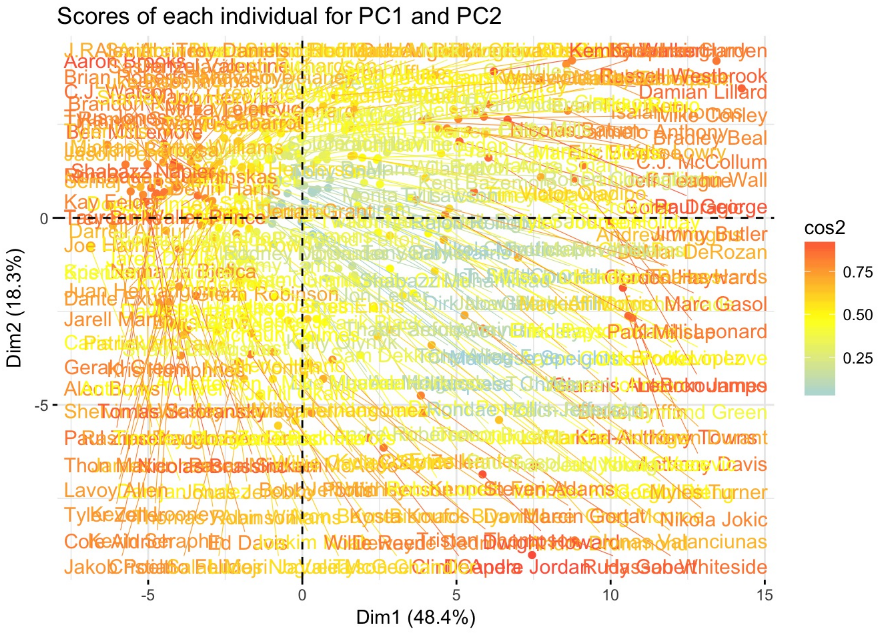
```
# Display the scores of first 10 players for the first 5 PCs.
ind$coord[1:10,1:5]
```

```
##                       Dim.1        Dim.2       Dim.3      Dim.4      Dim.5
## Aaron Brooks    -4.16155311  1.471716411 -0.06157910 -0.2764601  0.2256605
## Aaron Gordon     1.95937718 -0.008068329 -1.33165827  0.2911474 -1.3340691
## Al Horford       2.84052848 -0.477992643  0.08977012  1.0053515  0.4386133
## Al Jefferson    -1.96132233 -2.112173045 -0.10765214 -1.7903891  0.2666631
## Al-Farouq Aminu -0.98968877  0.229129739 -1.59913382  1.0138000  0.1840487
## Alan Williams   -2.77746977 -3.022002985 -0.20219383 -2.5891892 -0.1616306
## Alec Burks      -4.53562936  0.581110838  0.29200106 -1.0792990  1.1015132
## Alex Abrines    -3.34842235  1.880205483  1.28272334  0.6977300 -0.3664025
## Alex Len         0.45576175 -2.703869172 -1.80298798 -0.1965422 -1.4809647
## Allen Crabbe    -0.06687859  1.553616057  1.64680791  1.0302806 -1.4131467
```

```
#Store the player's scores for PC1, PC2 and PC3
PC1 = ind$coord[,1]
PC2 = ind$coord[,2]
PC3 = ind$coord[,3]
```

We can visualize the the scores of each individual for PC1 and PC2 using "fviz_pca_ind()"

```
fviz_pca_ind(pca, col.ind = "cos2", # control the color of individuals using the cos2
             gradient.cols = c("light blue","yellow","tomato"),
             repel = TRUE, # Avoid text overlapping
             title = "Scores of each individual for PC1 and PC2"
             )
```

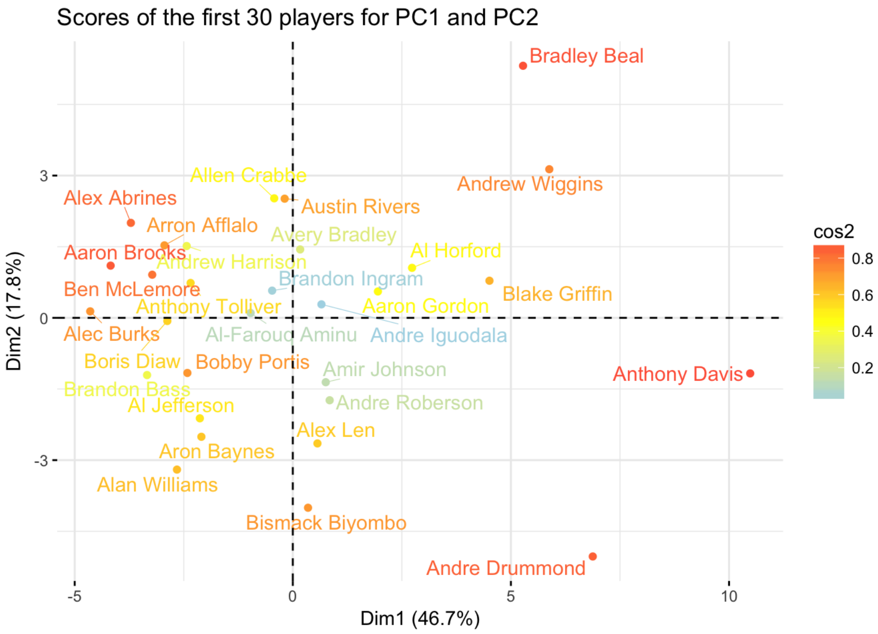


- The graph may look a little bit messy because there're too many individuals.

To show what the graph actually looks like clearly, let's pick the first 30 player to draw the graph.

```
pca30 <- prcomp(pca_dat[1:30, ], scale. = TRUE)

fviz_pca_ind(pca30, col.ind = "cos2", # control the color of individuals using the cos2
             gradient.cols = c("light blue","yellow","tomato"),
             repel = TRUE, # Avoid text overlapping
             title = "Scores of the first 30 players for PC1 and PC2"
             )
```



- In the graph, if the color of the individual is closer to red, then the PC scores of the individual is higher. In the sample of first 30 players, we can see that Anthony Davis and Andre Drummond have the highest PC scores.

Let's rescale the PC score for each individual from 0 to 100

```
#Rescale the first PC with a new scale ranging from 0 to 100
PC1 <- 100 * (PC1-min(PC1))/(max(PC1)-min(PC1))
mer_dat$PC1<-PC1
TOP_PC1<-select(arrange(mer_dat, desc(PC1))[1:20,], Player, PC1)
TOP_PC1
```

```
##                   Player       PC1
## 1      Russell Westbrook 100.00000
## 2           James Harden  96.13183
## 3          Anthony Davis  83.27104
## 4      Karl-Anthony Towns  82.60930
## 5  Giannis Antetokounmpo  81.77857
## 6           LeBron James  79.65575
## 7           Jimmy Butler  75.65554
## 8            Rudy Gobert  74.52094
## 9          Kawhi Leonard  74.31893
## 10         Stephen Curry  73.96215
## 11             John Wall  73.29970
## 12         Isaiah Thomas  73.08071
## 13         DeMar DeRozan  69.17916
## 14        Damian Lillard  69.13943
## 15      Hassan Whiteside  68.18636
## 16        DeAndre Jordan  67.82780
## 17          Kevin Durant  65.66071
## 18           Paul George  64.06864
## 19        Andre Drummond  62.76757
## 20          Kemba Walker  61.97712
```

- It shows that Russell Westbrook has the highest scaled PC1 score. It shows that Russell Westbrook is in dominant position in his team and he has the best scoring ability. And this conclusion is very closed to the reality.(Russell Westbrook scored the most points in 16-17 season)

```
#Rescale the first PC with a new scale ranging from 0 to 100
PC2 <- 100 * (PC2-min(PC2))/(max(PC2)-min(PC2))
mer_dat$PC2<-PC2
TOP_PC2<-select(arrange(mer_dat, desc(PC2))[1:20,], Player, PC2)
TOP_PC2
```

```
##                      Player       PC2
## 1             Stephen Curry 100.00000
## 2              James Harden  99.91870
## 3             Isaiah Thomas  99.21889
## 4              Kemba Walker  97.77798
## 5              Devin Booker  97.00214
## 6            Damian Lillard  96.01068
## 7               Eric Gordon  95.14479
## 8         Russell Westbrook  94.37070
## 9           Wesley Matthews  92.92457
## 10              J.J. Redick  91.23079
## 11             Kyrie Irving  91.14342
## 12         D'Angelo Russell  90.83936
## 13            Klay Thompson  90.10642
## 14       Matthew Dellavedova  90.00286
## 15 Kentavious Caldwell-Pope  90.00241
## 16            Jamal Crawford  89.92404
## 17              Troy Daniels  89.88083
## 18                 John Wall  89.86448
## 19              Bradley Beal  89.78339
## 20             C.J. McCollum  88.64849
```

- It shows that Stephen Curry has the highest scaled PC2 score. It shows that Stephen Curry has the best shooting skills(especially 3 pointers), which has proven to be true as well.

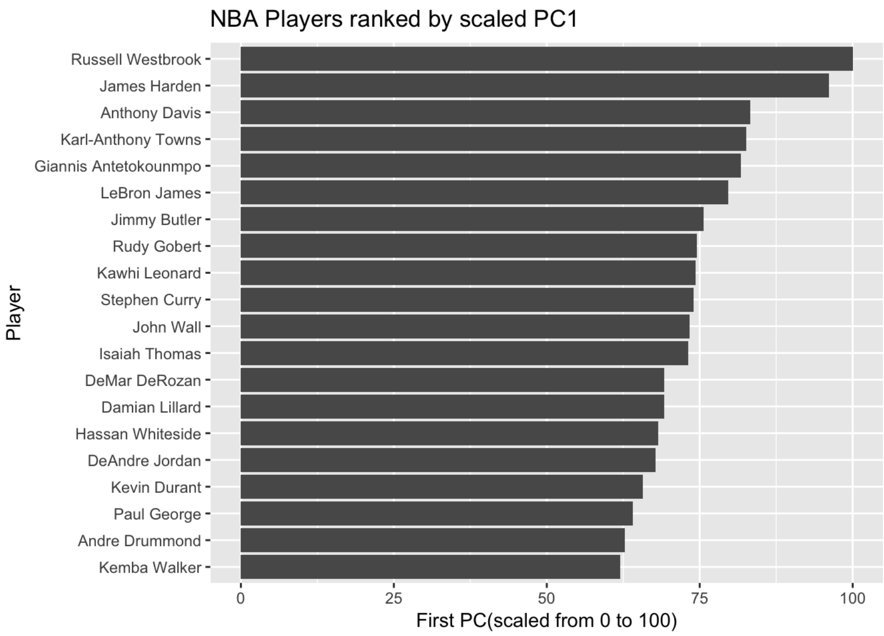
```
PC3 <- 100 * (PC3-min(PC3))/(max(PC3)-min(PC3))
mer_dat$PC3<-PC3
TOP_PC3<-select(arrange(mer_dat, desc(PC3))[1:20,], Player, PC3)
TOP_PC3
```

```
##              Player       PC3
## 1     Isaiah Thomas 100.00000
## 2      Brandon Bass  98.79618
## 3     Stephen Curry  97.17118
## 4  Montrezl Harrell  95.76740
## 5    Lucas Nogueira  94.50545
## 6      Kevin Durant  94.45969
## 7       James Jones  91.19191
## 8      JaVale McGee  90.60446
## 9        Kyle Lowry  90.57497
## 10   Andre Iguodala  90.21551
## 11    Davis Bertans  89.32184
## 12       Chris Paul  87.58565
## 13      George Hill  84.71516
## 14    Channing Frye  84.50914
## 15     Bradley Beal  83.28156
## 16     Nene Hilario  83.07415
## 17    Klay Thompson  82.68648
## 18        Ian Clark  82.17786
## 19      Gary Harris  81.90189
## 20      Mike Conley  81.35160
```

- It shows that Isaiah Thomas has the highest scaled PC3 score. That means he is the player who has the best efficiency in the NBA.

Let’s draw a barchart of NBA Players ranked by scaled PC1

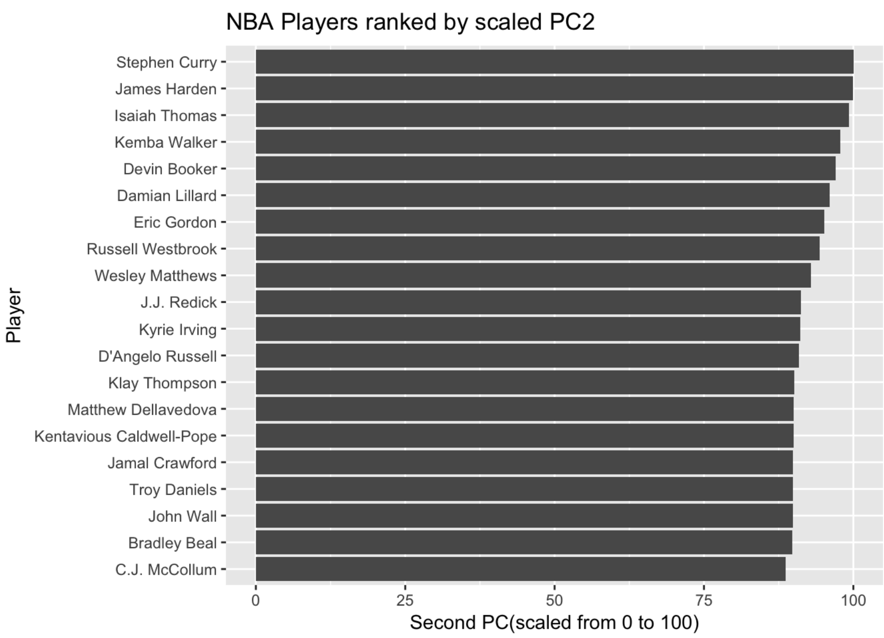
```
#Produce a barchart of top 20 NBA Players ranked by scaled PC1
ggplot(data = TOP_PC1) +
  geom_bar(aes(x=reorder(Player, PC1), y=PC1),stat='identity') +
  ylab("First PC(scaled from 0 to 100)") +
  xlab("Player") +
  ggtitle(label = "NBA Players ranked by scaled PC1", subtitle = NULL) +
  coord_flip()
```



- From this barchart, we can see that Russell Westbrook is in Rank #1, James Harden is in #2 and Anthony Davis is in #3. It means that they are very good at scoring and they all have very high field attempts because they are in the dominant position in their team. We can see that most of the players in top20 are the leader of the team. It seems like the result is very accurate and closed to the reality.

Produce a barchart of NBA Players ranked by scaled PC2

```
#Produce a barchart of top 20 NBA Players ranked by scaled PC2
ggplot(data = TOP_PC2) +
  geom_bar(aes(x=reorder(Player, PC2), y=PC2),stat='identity') +
  ylab("Second PC(scaled from 0 to 100)") +
  xlab("Player") +
  ggtitle(label = "NBA Players ranked by scaled PC2", subtitle = NULL) +
  coord_flip()
```

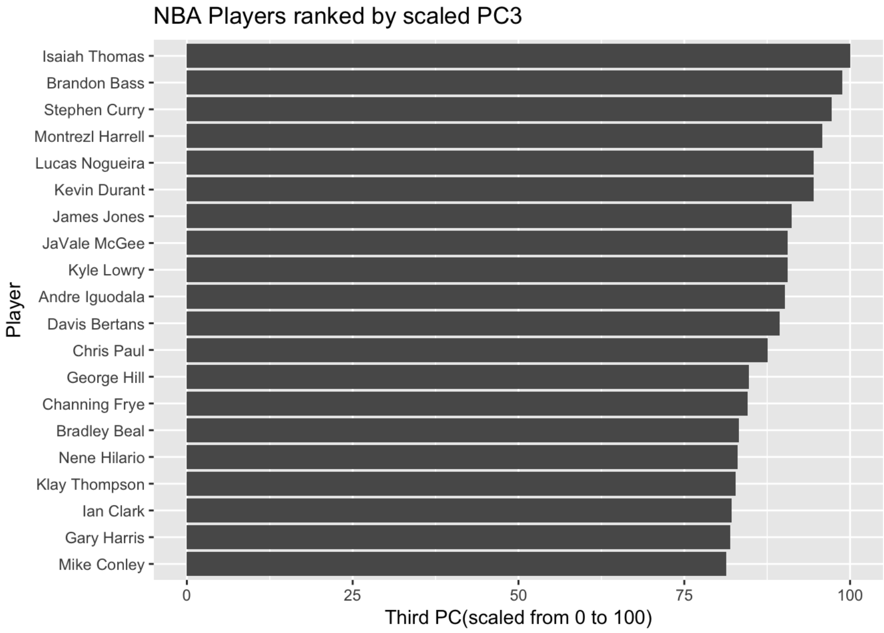


- From this barchart, we can see that Stephen Curry is in Rank #1, James Harden is in #2, and Isaiah Thomas is in #3. It means that they are very good at shooting(especially 3 pointers). However, this result is not as accurate as the last one because PC2 capture less variance comparing to PC1. Since PC2 focus more on player’s shooting skills, we can see many great three-point shooters are in top20, even though they may not be the best player in the league, such as Devin Booker, Eric Gordon, Wesley Matthews, JJ Redick and D’Angelo

Russell.

Produce a barchart of NBA Players ranked by scaled PC3

```
#Produce a barchart of top 20 NBA Players ranked by scaled PC3
ggplot(data = TOP_PC3) +
  geom_bar(aes(x=reorder(Player, PC3), y=PC3),stat='identity') +
  ylab("Third PC(scaled from 0 to 100)") +
  xlab("Player") +
  ggtitle(label = "NBA Players ranked by scaled PC3", subtitle = NULL) +
  coord_flip()
```

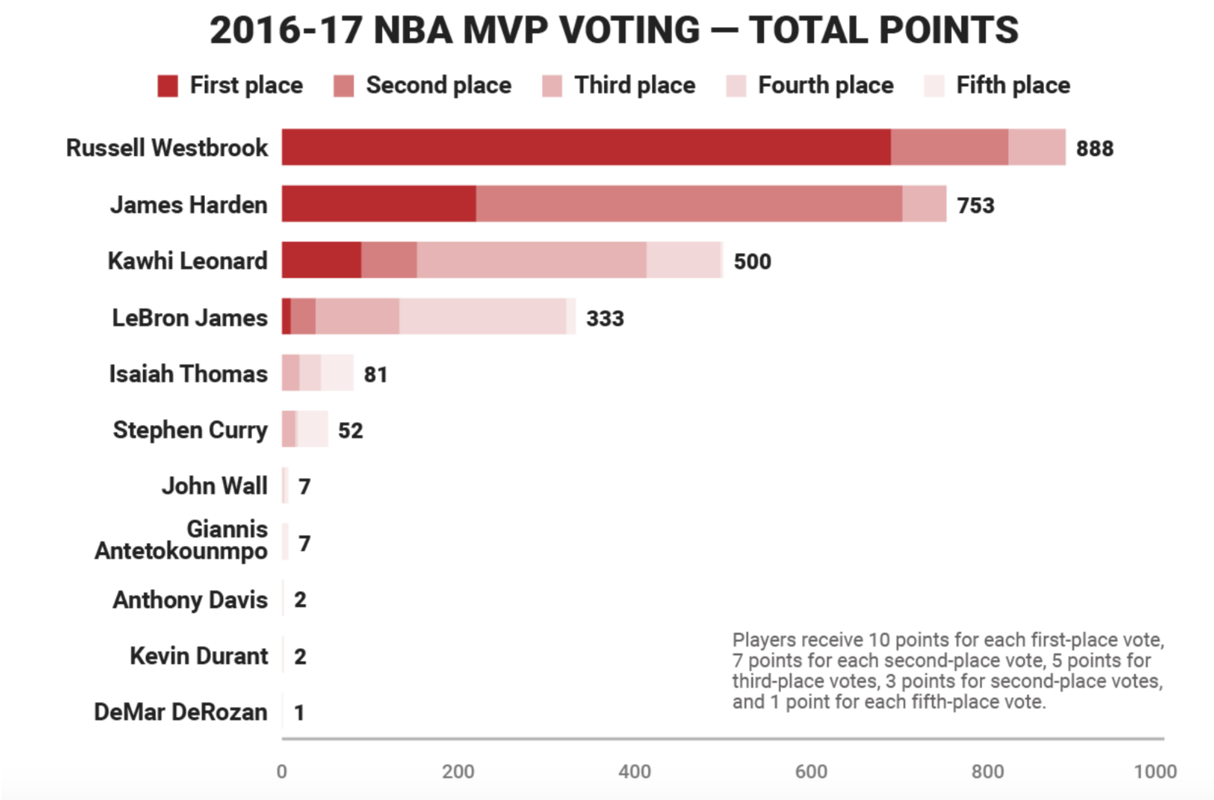


- From this barchart, we can see that Isaiah Thomas is in Rank #1, Brandon Bass is in #2 and Stephen Curry is in #3. It means that they have very high efficiency. However, this result is less accurate than the last two because PC3 explain less variance than PC1 and PC2. Since PC3 mainly focus on the efficiency, some players with very high TS_PER(true shooting percentage) are in the top20, even though they are actually not that good, such as Brandon Bass, Montrezl Harrell and Javale McGee.

Finally, let's compare the results of PCA with the actual mvp voting result in reality for NBA 16-17 season, and the player ranking given by ESPN.

*Since PC1 explains the most variance, we use the ranking result by PC1 to compare.

This is the MVP voting result:



MVP Voting Result

- The source is from http://www.businessinsider.com/nba-mvp-voting-results-2017-6
- From the actual MVP voting result, we can see that the rank#1 player is Russell Westbrook and #2 player is James Harden. The result from PC1 ranking also shows that Russell Westbrook is the rank#1 player and James Harden is the rank#2 player. And the top15 players in the ranking by PC1 contains all the top10 players in the actual MVP voting. That means the ranking result by PC1 is very accurate.

This is the players ranking given by ESPN:

Rank	Player
1	LeBron James
2	Kevin Durant
3	Kawhi Leonard
4	Stephen Curry
5	Russell Westbrook
6	Anthony Davis
7	Chris Paul
8	James Harden
9	Giannis Antetokounmpo
10	Draymond Green

- The source is from http://www.espn.com/nba/story/_/page/nbarank110/nbarank-players-1-10
- Although the top 3 player in this rank is not the same as the results we got by PC1, these two results are still pretty similar to each other. The top15 players in the PC1 ranking contains almost all the players in this rank except for Chris Paul.

In conclusion: Ranking of the NBA players by PC1 is very accurate and is closed to the actual rank for NBA players in season 16-17.

- And three of the most important factors to evaluate NBA players are:

1. whether the player is in the dominant position in the team and his scoring ability
2. A player’s shooting skills(especially 3 pointers)
3. A player’s efficiency.

Here’s a quick summary of what we have done in this post (Take-home message):

1. I showed how to find and store the data set that we want to analyze. The recommended website for us to find sports-related data is https://www.sports-reference.com/. And we also learned how to save the data.
2. I introduced some advanced statistics to evaluate an NBA player’s performance.
3. I performed PCA to rank the players after doing some data frame manipulations.
4. After that, I extracted and visualized the result of the PCA using “factoextra” package. It’s a very convenient package for us to visualize and analyze the PCA results.
5. We concluded that 3 of the most important factors to determine a good player are: 1. whether the player is in the dominant position in the team and his scoring ability 2. A player’s shooting skills 3. A player’s efficiency.
6. At last, I compared the rank of players by PC1 with the MVP voting result and the ranking given by ESPN. And we found that it is very accurate and closed to the actual rank for NBA players in season 16-17.

Thanks for reading!

References:

- Advanced stats: http://bleacherreport.com/articles/1813902-advanced-nba-stats-for-dummies-how-to-understand-the-new-hoops-math
- Advanced stats: https://www.basketball-reference.com/about/glossary.html
- Data source: https://www.sports-reference.com/
- Tips for saving data: https://www.youtube.com/watch?v=MWapXbaWs_U&feature=youtu.be
- Tips for saving data: https://www.youtube.com/watch?v=JkDLV0roT14&feature=youtu.be
- PCA example: http://www.cs.odu.edu/~mukka/cs795sum10dm/Lecturenotes/Day4/0605252.pdf
- Factoextra package: https://cran.r-project.org/web/packages/factoextra/factoextra.pdf
- Factoextra package: https://rdrr.io/cran/factoextra/man/fviz_pca.html
- MVP voting result: http://www.businessinsider.com/nba-mvp-voting-results-2017-6
- ESPN ranking http://www.espn.com/nba/story/_/page/nbarank110/nbarank-players-1-10