

NBADataScrapingComplete.R

danielpeslherbe

2020-08-13

```
##NBA Data Collection

options(warn = -1)
##install.packages('rvest')
library(rvest)

## Loading required package: xml2

##install.packages('tidyverse')
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2      v purrr  0.3.4
## v tibble  3.0.3      v dplyr  1.0.1
## v tidyr   1.1.1      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()      masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()         masks stats::lag()
## x purrr::pluck()       masks rvest::pluck()

##devtools::install_github('abresler/nbastatR')
library(nbastatR)
##install.packages('readxl')
library(readxl)
##install.packages('VIM')
library(VIM)

## Loading required package: colorspace
## Loading required package: grid
## VIM is ready to use.
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
## The following object is masked from 'package:datasets':
##
##     sleep

#WORKING
##Get Player Salary Data from 1999-2000 to 2019-2020 from ESPN
```

```

DataSalaries <- c()
for (i in 2000:2020) {
  for (j in 1:19) {
    url <- paste0('http://www.espn.com/nba/salaries/_/year/', i, '/page/', j)
    webpage <- read_html(url)
    names <- webpage %>%
      html_nodes('.evenrow td:nth-child(2) , .oddrow td:nth-child(2)') %>%
      html_text()
    positions <- c(',', PG', ', G', ', SG', ', F', ', SF', ', PF', ', C')
    for(k in 1:length(positions)){
      names <- str_remove_all(names, positions[k])
    }
    contracts <- webpage %>%
      html_nodes('.evenrow td:nth-child(4) , .oddrow td:nth-child(4)') %>%
      html_text()
    contracts <- sub(' ', '', contracts)
    contracts <- str_remove_all(contracts, ',')
    if(i == 2000){
      DataSalaries <- rbind(DataSalaries, cbind(rep(paste0((i-1), '-'), '00'), length(names)), names,
    }
    if(i > 2000 && i < 2010){
      DataSalaries <- rbind(DataSalaries, cbind(rep(paste0((i-1), '-0'), (i-2000)), length(names)), na
    }
    if(i >= 2010){
      DataSalaries <- rbind(DataSalaries, cbind(rep(paste0((i-1), '-'), (i-2000)), length(names)), na
    }
  }
}
DataSalaries[,3] <- as.numeric(DataSalaries[,3])
DataSalaries <- as.data.frame(DataSalaries)
names(DataSalaries)[1] <- 'Season'
DataSalaries[,3] <- as.numeric(DataSalaries[,3])

##Note that this data does not seem to be complete
##There is data available on hoopshype.com
##Note that the hoopshype functions from nbastatR don't
##give us past player salary information
##Using WebScaper from WebScaper.io, we compile in .xlsx form
##data scraped from the hoopshype website
DataSal <- c()
for (i in 1990:1998) {
  temp <- read_excel(paste0('~/.Library/Mobile Documents/com~apple-CloudDocs/NBA Players Project/hoopshy
datapoints <- (dim(temp)[1]-1)/2
namestemp <- c(temp[c(2:(datapoints+1)),3])
namestemp <- namestemp[[1]]
contractstemp <- c(temp[c((datapoints+2):dim(temp)[1]),4])
contractstemp <- as.numeric(contractstemp[[1]])
##rename columns
temp <- cbind(rep(paste0(i, '-9', (i-1989)), length(namestemp)), namestemp, contractstemp)
DataSal <- rbind(DataSal, temp)
}

## New names:
## * '' -> ...2

```

```

## * '' -> ...3
## * '' -> ...4

## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4

for (i in 1999:2008) {
  temp <- read_excel(paste0('~/.Library/Mobile Documents/com~apple-CloudDocs/NBA Players Project/hoopshy
datapoints <- (dim(temp)[1]-1)/2
namestemp <- c(temp[c(2:(datapoints+1)),3])
namestemp <- namestemp[[1]]
contractstemp <- c(temp[c((datapoints+2):dim(temp)[1]),4])
contractstemp <- as.numeric(contractstemp[[1]])
##rename columns
temp <- cbind(rep(paste0(i, '-0', (i-1999)), length(namestemp)), namestemp, contractstemp)
DataSal <- rbind(DataSal, temp)
}

## New names:
## * '' -> ...2
## * '' -> ...3
## * '' -> ...4
## New names:
## * '' -> ...2
## * '' -> ...3

```

```

## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4

```

```

for (i in 2009:2018) {
  temp <- read_excel(paste0('~/.Library/Mobile Documents/com~apple-CloudDocs/NBA Players Project/hoopshy
datapoints <- (dim(temp)[1]-1)/2
namestemp <- c(temp[c(2:(datapoints+1)),3])
namestemp <- namestemp[[1]]
contractstemp <- c(temp[c((datapoints+2):dim(temp)[1]),4])
contractstemp <- as.numeric(contractstemp[[1]])
  ##rename columns
  temp <- cbind(rep(paste0(i, '-1', (i-2009)), length(namestemp)), namestemp, contractstemp)
  DataSal <- rbind(DataSal, temp)
}

```

```

## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:
## * ' -> ...2
## * ' -> ...3
## * ' -> ...4
## New names:

```



```

## parsed http://www.basketball-reference.com/leagues/NBA_2012_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2013_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2014_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2015_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2016_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2017_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2018_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2019_advanced.html
## parsed http://www.basketball-reference.com/leagues/NBA_2020_advanced.html
## PerGame
## Advanced

## Joining, by = c("slugSeason", "yearSeason", "isSeasonCurrent", "slugPlayerSeason", "slugPlayerBREF",
## # A tibble: 13,563 x 67
##   slugSeason namePlayer groupPosition yearSeason slugPosition isSeasonCurrent
##   <chr>      <chr>      <chr>          <dbl> <chr>          <lgl>
## 1 1990-91    Alaa Abde~ F            1991 PF            FALSE
## 2 1990-91    Mahmoud A~ G            1991 PG            FALSE
## 3 1990-91    Mark Acres C            1991 C              FALSE
## 4 1990-91    Michael A~ G            1991 PG            FALSE
## 5 1990-91    Mark Agui~ F            1991 SF            FALSE
## 6 1990-91    Danny Ain~ G            1991 SG            FALSE
## 7 1990-91    Mark Alar~ F            1991 PF            FALSE
## 8 1990-91    Steve Alf~ G            1991 PG            FALSE
## 9 1990-91    Greg Ande~ F            1991 PF            FALSE
## 10 1990-91   Nick Ande~ G            1991 SG            FALSE
## # ... with 13,553 more rows, and 61 more variables: slugPlayerSeason <chr>,
## #   slugPlayerBREF <chr>, agePlayer <dbl>, slugTeamBREF <chr>,
## #   countGames <dbl>, countGamesStarted <dbl>, pctFG <dbl>, pctFG3 <dbl>,
## #   pctFG2 <dbl>, pctEFG <dbl>, pctFT <dbl>, isHOFPlayer <lgl>,
## #   slugTeamsBREF <chr>, idPlayerNBA <dbl>, urlPlayerThumbnail <chr>,
## #   urlPlayerHeadshot <chr>, urlPlayerPhoto <chr>, urlPlayerStats <chr>,
## #   urlPlayerActionPhoto <chr>, minutesPerGame <dbl>, fgmPerGame <dbl>,
## #   fgaPerGame <dbl>, fg3mPerGame <dbl>, fg3aPerGame <dbl>, fg2mPerGame <dbl>,
## #   fg2aPerGame <dbl>, ftmPerGame <dbl>, ftaPerGame <dbl>, orbPerGame <dbl>,
## #   drbPerGame <dbl>, trbPerGame <dbl>, astPerGame <dbl>, stlPerGame <dbl>,
## #   blkPerGame <dbl>, tovPerGame <dbl>, pfPerGame <dbl>, ptsPerGame <dbl>,
## #   countTeamsPlayerSeasonPerGame <dbl>, minutes <dbl>, ratioPER <dbl>,
## #   pctTrueShooting <dbl>, pct3PRate <dbl>, pctFTRate <dbl>, pctORB <dbl>,
## #   pctDRB <dbl>, pctTRB <dbl>, pctAST <dbl>, pctSTL <dbl>, pctBLK <dbl>,
## #   pctTOV <dbl>, pctUSG <dbl>, ratioOWS <dbl>, ratioDWS <dbl>, ratioWS <dbl>,
## #   ratioWSPer48 <dbl>, ratioOBPM <dbl>, ratioDBPM <dbl>, ratioBPM <dbl>,
## #   ratioVORP <dbl>, countTeamsPlayerSeason <dbl>, urlPlayerBREF <chr>

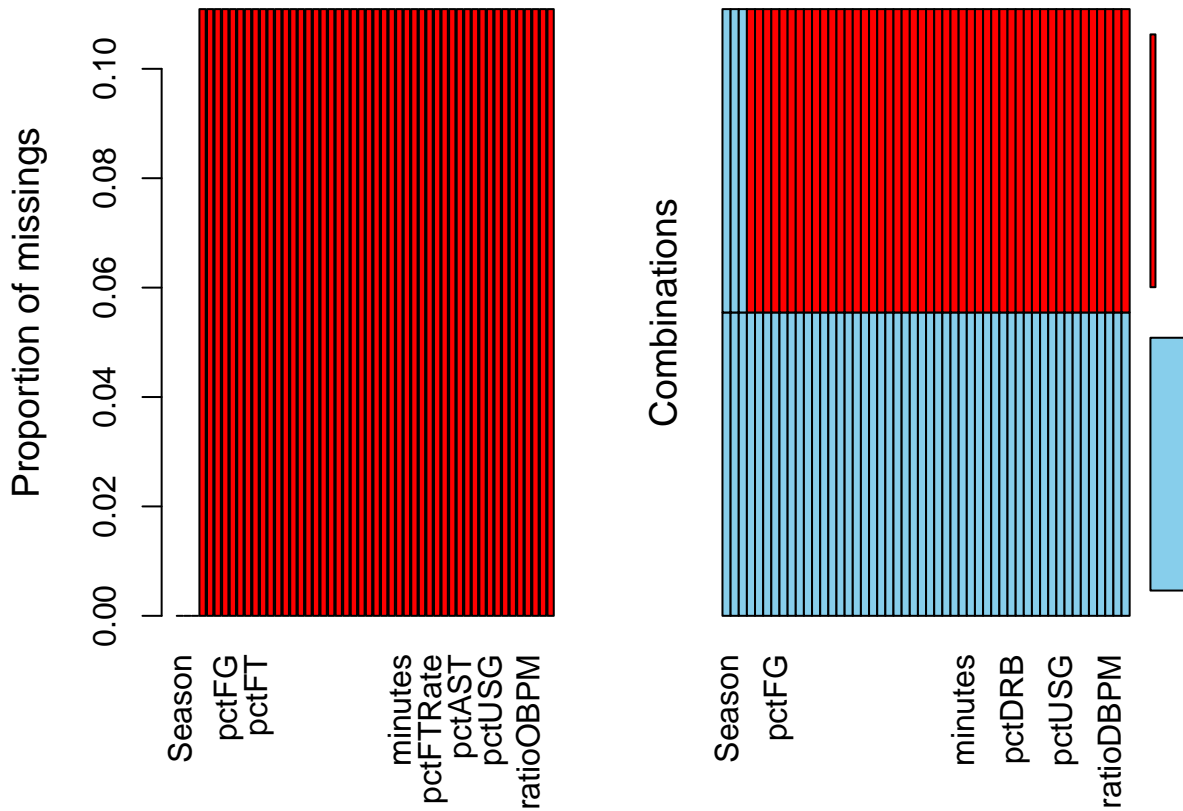
DataperGame <- dataBREFPlayerPerGame[,-c(2:5,7,9,17:24,43)]
DataAdvanced <- dataBREFPlayerAdvanced[,-c(2:5,7,9,11:18,40)]

##Now we have all the data about players and their salaries
##We now have to create the dataset with all the information together

DataStats <- bind_cols(DataperGame, DataAdvanced[,-c(1:4)])
names(DataStats)[1] <- 'Season'
names(DataStats)[2] <- 'names'

```

```
DataSet <- left_join(DataSal, DataStats, by = c('Season' = 'Season', 'names' = 'names'))
aggr(DataSet)
```



```
na.index <- which(is.na(DataSet[,4]))
na.player.names <- unique(DataSet[na.index,2])
for (i in 1:dim(DataStats)[1]) {
  if(DataStats[i,2] == 'Lafayette Lever'){
    DataStats[i,2] <- 'Fat Lever'
  }
  if(DataStats[i,2] == 'Dan Schayes'){
    DataStats[i,2] <- 'Danny Schayes'
  }
  if(DataStats[i,2] == 'David Greenwood'){
    DataStats[i,2] <- 'Dave Greenwood'
  }
  if(DataStats[i,2] == 'B.J. Armstrong'){
    DataStats[i,2] <- 'BJ Armstrong'
  }
  if(DataStats[i,2] == 'Eddielee Wilkins'){
    DataStats[i,2] <- 'Eddie Lee Wilkins'
  }
  if(DataStats[i,2] == 'Clifford Robinson'){
    DataStats[i,2] <- 'Cliff Robinson'
  }
  if(DataStats[i,2] == 'Steven Smith'){
```



```

    DataStats[i,2] <- 'Steve Smith'
  }
  if(DataStats[i,2] == 'Ike Austin'){
    DataStats[i,2] <- 'Isaac Austin'
  }
  if(DataStats[i,2] == 'A.J. Wynder'){
    DataStats[i,2] <- 'AJ Wynder'
  }
  if(DataStats[i,2] == 'Clar. Weatherspoon'){
    DataStats[i,2] <- 'Clarence Weatherspoon'
  }
  if(DataStats[i,2] == 'P.J. Brown'){
    DataStats[i,2] <- 'PJ Brown'
  }
  if(DataStats[i,2] == 'Derrick Alston Sr'){
    DataStats[i,2] <- 'Derrick Alston'
  }
  if(DataStats[i,2] == 'B.J. Tyler'){
    DataStats[i,2] <- 'BJ Tyler'
  }
  if(DataStats[i,2] == 'Horacio Llamas'){
    DataStats[i,2] <- 'Horacio Llamas Grey'
  }
  if(DataStats[i,2] == 'Rasho Nesterovic'){
    DataStats[i,2] <- 'Radoslav Nesterovic'
  }
  if(DataStats[i,2] == 'Jeff Sheppard'){
    DataStats[i,2] <- 'Jeffrey Sheppard'
  }
  if(DataStats[i,2] == 'Peja Stojakovic'){
    DataStats[i,2] <- 'Predrag Stojakovic'
  }
  if(DataStats[i,2] == 'Hedo Turkoglu'){
    DataStats[i,2] <- 'Hidayet Turkoglu'
  }
  if(DataStats[i,2] == 'Slava Medvedenko'){
    DataStats[i,2] <- 'Stanislav Medvedenko'
  }
  if(DataStats[i,2] == 'Wang Zhi-zhi'){
    DataStats[i,2] <- 'Wang Zhizhi'
  }
  if(DataStats[i,2] == 'Norman Richardson'){
    DataStats[i,2] <- 'Norm Richardson'
  }
  if(DataStats[i,2] == 'Nene'){
    DataStats[i,2] <- 'Nenê'
  }
  if(DataStats[i,2] == "Amar'e Stoudemire"){
    DataStats[i,2] <- 'Amare Stoudemire'
  }
  if(DataStats[i,2] == 'Flip Murray'){
    DataStats[i,2] <- 'Ronald Murray'
  }
}

```

```

if(DataStats[i,2] == 'Roger Mason Jr.'){
  DataStats[i,2] <- 'Roger Mason'
}
if(DataStats[i,2] == 'J.R. Bremer'){
  DataStats[i,2] <- 'JR Bremer'
}
if(DataStats[i,2] == 'T.J. Ford'){
  DataStats[i,2] <- 'TJ Ford'
}
if(DataStats[i,2] == 'Michael Sweetney'){
  DataStats[i,2] <- 'Mike Sweetney'
}
if(DataStats[i,2] == 'Sasha Pavlovic'){
  DataStats[i,2] <- 'Aleksandar Pavlovic'
}
if(DataStats[i,2] == 'Mo Williams'){
  DataStats[i,2] <- 'Maurice Williams'
}
if(DataStats[i,2] == 'Ibrahim Kutluay'){
  DataStats[i,2] <- 'Ibo Kutluay'
}
if(DataStats[i,2] == 'DJ Mbenga'){
  DataStats[i,2] <- 'Didier Ilunga-Mbenga'
}
if(DataStats[i,2] == 'Ha Ha'){
  DataStats[i,2] <- 'Ha Seung-Jin'
}
if(DataStats[i,2] == 'Lou Williams'){
  DataStats[i,2] <- 'Louis Williams'
}
if(DataStats[i,2] == 'Boniface Ndong'){
  DataStats[i,2] <- "Boniface N'Dong"
}
if(DataStats[i,2] == 'John Lucas III'){
  DataStats[i,2] <- 'John Lucas'
}
if(DataStats[i,2] == 'P.J. Tucker'){
  DataStats[i,2] <- 'PJ Tucker'
}
if(DataStats[i,2] == 'J.J. Barea'){
  DataStats[i,2] <- 'Jose Juan Barea'
}
if(DataStats[i,2] == 'D.J. Strawberry'){
  DataStats[i,2] <- 'DJ Strawberry'
}
if(DataStats[i,2] == 'C.J. Watson'){
  DataStats[i,2] <- 'CJ Watson'
}
if(DataStats[i,2] == 'O.J. Mayo'){
  DataStats[i,2] <- 'OJ Mayo'
}
if(DataStats[i,2] == 'D.J. Augustin'){
  DataStats[i,2] <- 'DJ Augustin'
}

```

```

}
if(DataStats[i,2] == 'J.R. Giddens'){
  DataStats[i,2] <- 'JR Giddens'
}
if(DataStats[i,2] == 'Sun Sun'){
  DataStats[i,2] <- 'Sun Yue'
}
if(DataStats[i,2] == 'Byron Mullens'){
  DataStats[i,2] <- 'BJ Mullens'
}
if(DataStats[i,2] == 'Lou Amundson'){
  DataStats[i,2] <- 'Louis Amundson'
}
if(DataStats[i,2] == 'Patty Mills'){
  DataStats[i,2] <- 'Patrick Mills'
}
if(DataStats[i,2] == 'Hamady Ndiaye'){
  DataStats[i,2] <- "Hamady N'Diaye"
}
if(DataStats[i,2] == 'Ish Smith'){
  DataStats[i,2] <- 'Ishmael Smith'
}
if(DataStats[i,2] == 'Marcus Morris Sr.'){
  DataStats[i,2] <- 'Marcus Morris'
}
if(DataStats[i,2] == 'Maurice Harkless'){
  DataStats[i,2] <- 'Moe Harkless'
}
if(DataStats[i,2] == 'Perry Jones III'){
  DataStats[i,2] <- 'Perry Jones'
}
if(DataStats[i,2] == 'Otto Porter Jr.'){
  DataStats[i,2] <- 'Otto Porter'
}
if(DataStats[i,2] == 'Vitor Luis Faverani'){
  DataStats[i,2] <- 'Vitor Faverani'
}
if(DataStats[i,2] == 'Dennis Schroder'){
  DataStats[i,2] <- 'Dennis Schroeder'
}
if(DataStats[i,2] == 'Tim Hardaway Jr.'){
  DataStats[i,2] <- 'Tim Hardaway Jr'
}
if(DataStats[i,2] == 'Glen Rice Jr.'){
  DataStats[i,2] <- 'Glen Rice Jr'
}
if(DataStats[i,2] == "Toure' Murry"){
  DataStats[i,2] <- 'Toure Murry'
}
if(DataStats[i,2] == 'T.J. Warren'){
  DataStats[i,2] <- 'TJ Warren'
}
if(DataStats[i,2] == 'C.J. Wilcox'){

```

```

    DataStats[i,2] <- 'CJ Wilcox'
  }
  if(DataStats[i,2] == "Johnny O'Bryant III"){
    DataStats[i,2] <- "Johnny O'Bryant"
  }
  if(DataStats[i,2] == 'James Ennis III'){
    DataStats[i,2] <- 'James Ennis'
  }
  if(DataStats[i,2] == 'Kelly Oubre Jr.'){
    DataStats[i,2] <- 'Kelly Oubre'
  }
  if(DataStats[i,2] == 'Larry Nance Jr.'){
    DataStats[i,2] <- 'Larry Nance'
  }
  if(DataStats[i,2] == 'T.J. McConnell'){
    DataStats[i,2] <- 'TJ McConnell'
  }
  if(DataStats[i,2] == 'Juancho Hernangomez'){
    DataStats[i,2] <- 'Juan Hernangomez'
  }
  if(DataStats[i,2] == 'Wade Balwin IV'){
    DataStats[i,2] <- 'Wade Baldwin'
  }
  if(DataStats[i,2] == "DeAndre' Bembry"){
    DataStats[i,2] <- 'DeAndre Bembry'
  }
  if(DataStats[i,2] == 'Timothe Luwawu-Cabarrot'){
    DataStats[i,2] <- 'Timothe Luwawu'
  }
  if(DataStats[i,2] == 'Danuel House Jr.'){
    DataStats[i,2] <- 'Danuel House'
  }
  if(DataStats[i,2] == 'Derrick Jones Jr.'){
    DataStats[i,2] <- 'Derrick Jones'
  }
  if(DataStats[i,2] == 'James Webb III'){
    DataStats[i,2] <- 'James Webb'
  }
  if(DataStats[i,2] == 'Dennis Smith Jr.'){
    DataStats[i,2] <- 'Dennis Smith'
  }
  if(DataStats[i,2] == 'D.J. Wilson'){
    DataStats[i,2] <- 'DJ Wilson'
  }
  if(DataStats[i,2] == 'Frank Mason III'){
    DataStats[i,2] <- 'Frank Mason'
  }
  if(DataStats[i,2] == 'Wes Iwundu'){
    DataStats[i,2] <- 'Wesley Iwundu'
  }
  if(DataStats[i,2] == 'Walt Lemon Jr.'){
    DataStats[i,2] <- 'Walter Lemon Jr'
  }
}

```

```

if(DataStats[i,2] == 'Derrick Walton Jr.'){
  DataStats[i,2] <- 'Derrick Walton'
}
if(DataStats[i,2] == 'Matt Williams Jr.'){
  DataStats[i,2] <- 'Matt Williams'
}
if(DataStats[i,2] == 'Vincent Hunter'){
  DataStats[i,2] <- 'Vince Hunter'
}
if(DataStats[i,2] == 'C.J. Williams'){
  DataStats[i,2] <- 'CJ Williams'
}
if(DataStats[i,2] == 'Marvin Bagley III'){
  DataStats[i,2] <- 'Marvin Bagley'
}
if(DataStats[i,2] == 'Jaren Jackson Jr.'){
  DataStats[i,2] <- 'Jaren Jackson'
}
if(DataStats[i,2] == 'Wendell Carter Jr.'){
  DataStats[i,2] <- 'Wendell Carter'
}
if(DataStats[i,2] == 'Kevin Knox II'){
  DataStats[i,2] <- 'Kevin Knox'
}
if(DataStats[i,2] == 'Michael Porter Jr.'){
  DataStats[i,2] <- 'Michael Porter'
}
if(DataStats[i,2] == 'Troy Brown Jr.'){
  DataStats[i,2] <- 'Troy Brown'
}
if(DataStats[i,2] == 'Lonnie Walker IV'){
  DataStats[i,2] <- 'Lonnie Walker'
}
if(DataStats[i,2] == 'Robert Williams III'){
  DataStats[i,2] <- 'Robert Williams'
}
if(DataStats[i,2] == 'Svi Mykhailiuk'){
  DataStats[i,2] <- 'Sviatoslav Mykhailiuk'
}
if(DataStats[i,2] == 'Melvin Frazier Jr.'){
  DataStats[i,2] <- 'Melvin Frazier'
}
if(DataStats[i,2] == "Devonte' Graham"){
  DataStats[i,2] <- 'Devonte Graham'
}
if(DataStats[i,2] == 'Gary Trent Jr.'){
  DataStats[i,2] <- 'Gary Trent Jr'
}
if(DataStats[i,2] == 'Cameron Reynolds'){
  DataStats[i,2] <- 'Cam Reynolds'
}
if(DataStats[i,2] == 'J.P. Macura'){
  DataStats[i,2] <- 'JP Macura'
}

```

```

}
if(DataStats[i,2] == 'Mitchell Creek'){
  DataStats[i,2] <- 'Mitch Creek'
}
if(DataStats[i,2] == 'P.J. Washington'){
  DataStats[i,2] <- 'PJ Washington'
}
if(DataStats[i,2] == 'Kevin Porter Jr.'){
  DataStats[i,2] <- 'Kevin Porter'
}
if(DataStats[i,2] == 'Zach Norvell Jr.'){
  DataStats[i,2] <- 'Zach Norvell'
}
if(DataStats[i,2] == 'Charles Brown Jr.'){
  DataStats[i,2] <- 'Charlie Brown'
}
if(DataStats[i,2] == 'Brian Bowen II'){
  DataStats[i,2] <- 'Brian Bowen'
}
}

##na.player.names[i] injured and out for the season
##4, 5, 6, 18, 33, 48, 49, 56, 58, 59, 62, 79, 81, 83, 97, 99,
##101, 110, 134, 167, 170, 173, 176, 187, 195, 218, 221, 225,
##311, 327, 338, 343, 361, 362, 367, 371, 380, 390, 397, 398,
##401, 415, 421, 460, 467, 468, 471, 475, 504, 505, 506, 508,
##537, 541, 546, 554, 556, 570, 573, 585, 592, 593, 595, 621,
##623, 624, 630, 648, 695, 699, 768, 808, 811, 824, 852, 853,
##854, 855, 860

##na.player.names[i] retired for the season
##7, 12, 13, 15, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 32
##36, 37, 38, 39, 41, 42, 44, 45, 46, 47, 54, 55, 57, 60, 61, 63,
##64, 66, 67, 70, 71, 78, 80, 82, 84, 85, 87, 88, 91, 92, 94, 95,
##96, 98, 100, 102, 103, 104, 105, 106, 107, 108, 109, 111, 113,
##118, 119, 120, 121, 122, 126, 127, 128, 129, 132, 133, 137, 138,
##139, 140, 141, 142, 143, 144, 145, 148, 149, 150, 151, 153, 155,
##158, 159, 161, 164, 171, 172, 174, 177, 179, 180, 184, 185, 186,
##188, 189, 190, 191, 192, 194, 198, 199, 201, 202, 208, 209, 210,
##212, 217, 219, 220, 222, 223, 224, 228, 229, 230, 231, 232, 234,
##237, 241, 242, 243, 245, 246, 247, 249, 250, 251, 253, 254, 255,
##258, 259, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271,
##272, 273, 275, 277, 278, 279, 280, 282, 283, 287, 289, 290, 291,
##295, 297, 299, 302, 304, 305, 306, 307, 309, 310, 312, 313, 314,
##315, 316, 317, 320, 321, 325, 326, 328, 336, 337, 339, 340, 341,
##342, 344, 346, 347, 348, 349, 350, 353, 355, 357, 363, 364, 365,
##366, 368, 369, 370, 372, 374, 376, 377, 379, 381, 382, 388, 389,
##391, 392, 393, 394, 395, 396, 399, 400, 402, 405, 407, 408, 409,
##411, 412, 414, 419, 420, 422, 423, 424, 425, 426, 427, 429, 430,
##431, 432, 434, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447,
##448, 449, 450, 452, 453, 454, 461, 463, 466, 469, 470, 472, 473,
##476, 478, 481, 482, 483, 484, 489, 490, 492, 496, 500, 503, 507,
##509, 510, 512, 513, 514, 515, 516, 521, 524, 525, 526, 527, 528,
##529,

```

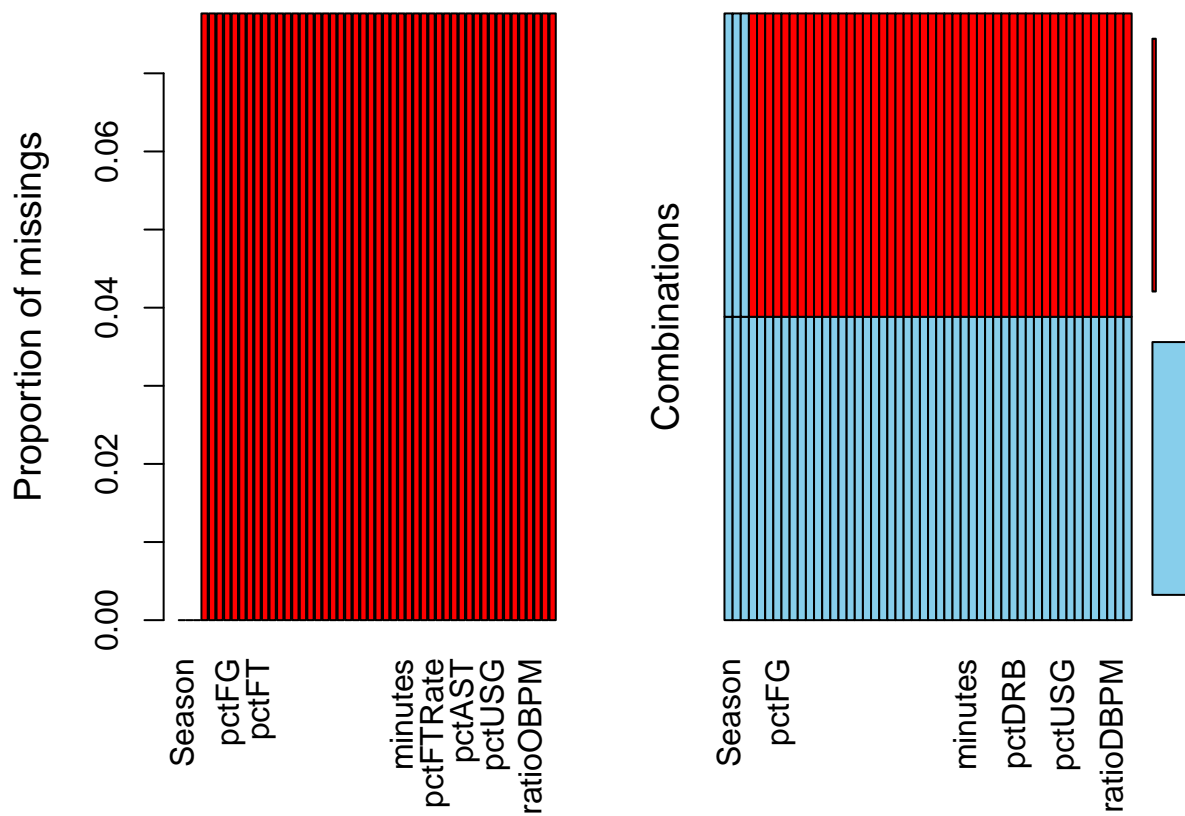
```

##531, 532, 534, 535, 536, 538, 539, 543, 547, 548, 549, 550, 551,
##552, 555, 558, 559, 560, 561, 563, 567, 569, 580, 582, 583, 584,
##587, 588, 589, 590, 591, 594, 597, 599, 601, 604, 605, 606, 607,
##609, 610, 612, 614, 615, 625, 627, 628, 629, 631, 634, 635, 638,
##642, 644, 645, 646, 647, 649, 651, 652, 653, 656, 658, 659, 664,
##665, 669, 670, 674, 675, 677, 678, 680, 683, 684, 692, 693, 696,
##697, 698, 700, 702, 703, 706, 707, 709, 710, 714, 715, 716, 719,
##721, 723, 730, 750, 752, 756, 757, 758, 759, 760, 763, 764, 765,
##767, 769, 771, 772, 773, 776, 777, 780, 781, 784, 785, 786, 798,
##799, 804, 805, 806, 807, 813, 817, 818, 821, 822, 826, 832, 833,
##837, 839, 842, 843, 844, 861, 864, 865, 871

##na.player.names[i] that did not see playing time for the season (bench or other league)
##16, 17,
##11, 23, 31, 34, 35, 50, 52, 53, 65, 68, 69, 72, 73, 74, 75, 76,
##77, 86, 89, 93, 112, 115, 116, 117, 123, 124, 125, 130, 131, 136,
##146, 147, 152, 154, 156, 157, 160, 162, 163, 165, 166, 168, 169,
##175, 178, 181, 182, 183, 193, 196, 197, 200, 204, 205, 206, 207,
##211, 213, 214, 215, 216, 226, 233, 235, 236, 238, 239, 244, 252,
##256, 257, 260, 274, 276, 281, 284, 285, 286, 288, 292, 293, 294,
##296, 298, 300, 301, 303, 308, 318, 319, 323, 324, 329, 330, 331,
##333, 334, 335, 345, 351, 352, 356, 358, 360, 378, 383, 384, 406,
##413, 417, 418, 435, 436, 437, 451, 456, 458, 462, 465, 474, 477,
##479, 480, 485, 488, 491, 493, 494, 495, 497, 498, 499, 501, 502,
##511, 518, 519, 520, 522, 528, 540, 545, 553, 557, 564, 568, 571,
##572, 575, 576, 578, 579, 581, 596, 600, 602, 603, 611, 613, 616,
##617, 618, 619, 620, 622, 637, 641, 643, 654, 657, 661, 663, 666,
##667, 668, 671, 672, 673, 681, 682, 685, 686, 688, 689, 690, 691,
##694, 701, 712, 713, 720, 722, 724, 725, 726, 727, 728, 729, 731,
##732, 733, 734, 735, 737, 738, 739, 740, 741, 742, 743, 744, 745,
##746, 747, 748, 749, 751, 753, 754, 755, 762, 770, 778, 779, 782,
##783, 788, 789, 790, 791, 792, 797, 800, 801, 802, 803, 820, 827,
##829, 835, 836, 838, 840, 845, 847, 848, 849, 850, 851, 855, 856,
##857, 858, 862, 863, 867, 868, 869, 870, 872, 873, 874, 876, 877,
##878, 879, 881, 882, 884, 885, 886, 887, 888, 889

DataSet <- left_join(DataSal, DataStats, by = c('Season' = 'Season', 'names' = 'names'))
aggr(DataSet)

```



```
invalid.seasons.index <- c(4, 5, 6, 18, 33, 48, 49, 56, 58, 59, 62, 79, 81, 83, 97, 99,
  101, 110, 134, 167, 170, 173, 176, 187, 195, 218, 221, 225,
  311, 327, 338, 343, 361, 362, 367, 371, 380, 390, 397, 398,
  401, 415, 421, 460, 467, 468, 471, 475, 504, 505, 506, 508,
  537, 541, 546, 554, 556, 570, 573, 585, 592, 593, 595, 621,
  623, 624, 630, 648, 695, 699, 768, 808, 811, 824, 852, 853,
  854, 855, 860,
  ##na.player.names[i] retired for the season
  7, 12, 13, 15, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 32,
  36, 37, 38, 39, 41, 42, 44, 45, 46, 47, 54, 55, 57, 60, 61, 63,
  64, 66, 67, 70, 71, 78, 80, 82, 84, 85, 87, 88, 91, 92, 94, 95,
  96, 98, 100, 102, 103, 104, 105, 106, 107, 108, 109, 111, 113,
  118, 119, 120, 121, 122, 126, 127, 128, 129, 132, 133, 137, 138,
  139, 140, 141, 142, 143, 144, 145, 148, 149, 150, 151, 153, 155,
  158, 159, 161, 164, 171, 172, 174, 177, 179, 180, 184, 185, 186,
  188, 189, 190, 191, 192, 194, 198, 199, 201, 202, 208, 209, 210,
  212, 217, 219, 220, 222, 223, 224, 228, 229, 230, 231, 232, 234,
  237, 241, 242, 243, 245, 246, 247, 249, 250, 251, 253, 254, 255,
  258, 259, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271,
  272, 273, 275, 277, 278, 279, 280, 282, 283, 287, 289, 290, 291,
  295, 297, 299, 302, 304, 305, 306, 307, 309, 310, 312, 313, 314,
  315, 316, 317, 320, 321, 325, 326, 328, 336, 337, 339, 340, 341,
  342, 344, 346, 347, 348, 349, 350, 353, 355, 357, 363, 364, 365,
  366, 368, 369, 370, 372, 374, 376, 377, 379, 381, 382, 388, 389,
  391, 392, 393, 394, 395, 396, 399, 400, 402, 405, 407, 408, 409,
  411, 412, 414, 419, 420, 422, 423, 424, 425, 426, 427, 429, 430,
```



```

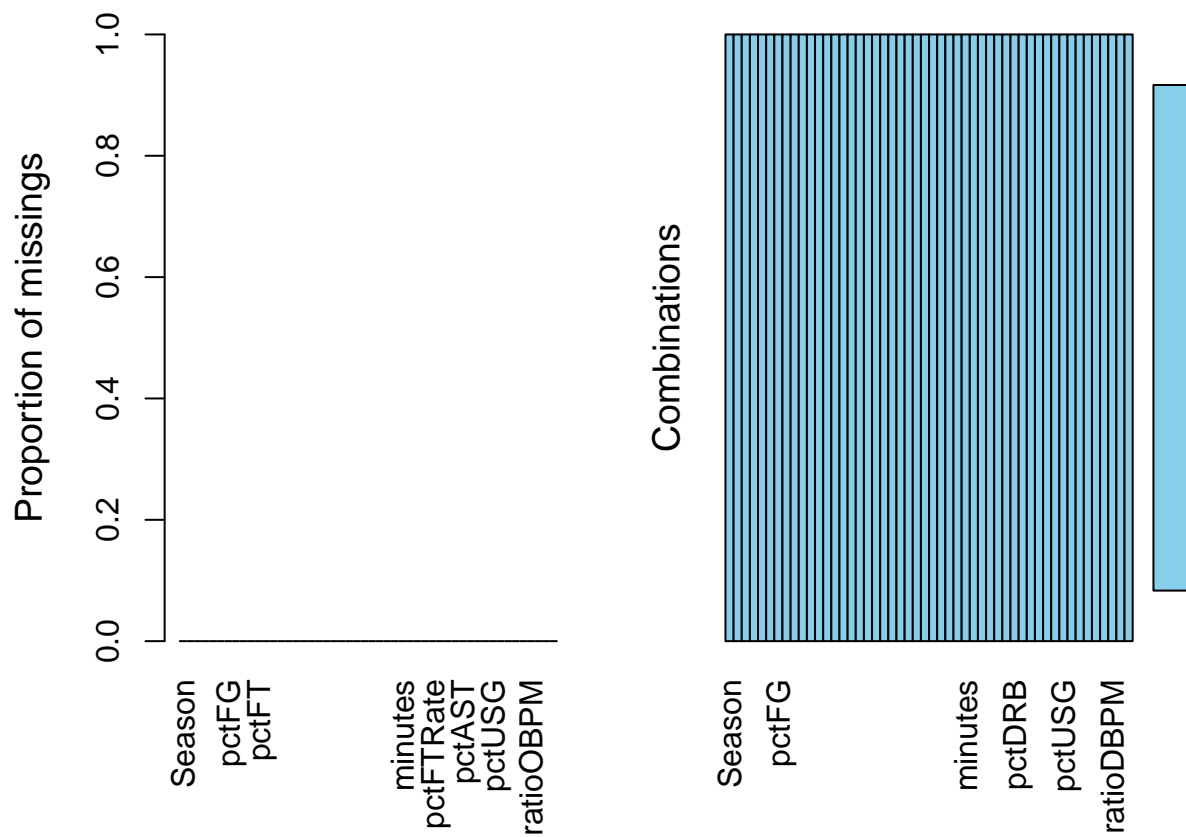
431, 432, 434, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447,
448, 449, 450, 452, 453, 454, 461, 463, 466, 469, 470, 472, 473,
476, 478, 481, 482, 483, 484, 489, 490, 492, 496, 500, 503, 507,
509, 510, 512, 513, 514, 515, 516, 521, 524, 525, 526, 527, 528,
529,
531, 532, 534, 535, 536, 538, 539, 543, 547, 548, 549, 550, 551,
552, 555, 558, 559, 560, 561, 563, 567, 569, 580, 582, 583, 584,
587, 588, 589, 590, 591, 594, 597, 599, 601, 604, 605, 606, 607,
609, 610, 612, 614, 615, 625, 627, 628, 629, 631, 634, 635, 638,
642, 644, 645, 646, 647, 649, 651, 652, 653, 656, 658, 659, 664,
665, 669, 670, 674, 675, 677, 678, 680, 683, 684, 692, 693, 696,
697, 698, 700, 702, 703, 706, 707, 709, 710, 714, 715, 716, 719,
721, 723, 730, 750, 752, 756, 757, 758, 759, 760, 763, 764, 765,
767, 769, 771, 772, 773, 776, 777, 780, 781, 784, 785, 786, 798,
799, 804, 805, 806, 807, 813, 817, 818, 821, 822, 826, 832, 833,
837, 839, 842, 843, 844, 861, 864, 865, 871,
##na.player.names[i] that did not see playing time for the season (bench or other league)
16, 17,
11, 23, 31, 34, 35, 50, 52, 53, 65, 68, 69, 72, 73, 74, 75, 76,
77, 86, 89, 93, 112, 115, 116, 117, 123, 124, 125, 130, 131, 136,
146, 147, 152, 154, 156, 157, 160, 162, 163, 165, 166, 168, 169,
175, 178, 181, 182, 183, 193, 196, 197, 200, 204, 205, 206, 207,
211, 213, 214, 215, 216, 226, 233, 235, 236, 238, 239, 244, 252,
256, 257, 260, 274, 276, 281, 284, 285, 286, 288, 292, 293, 294,
296, 298, 300, 301, 303, 308, 318, 319, 323, 324, 329, 330, 331,
333, 334, 335, 345, 351, 352, 356, 358, 360, 378, 383, 384, 406,
413, 417, 418, 435, 436, 437, 451, 456, 458, 462, 465, 474, 477,
479, 480, 485, 488, 491, 493, 494, 495, 497, 498, 499, 501, 502,
511, 518, 519, 520, 522, 528, 540, 545, 553, 557, 564, 568, 571,
572, 575, 576, 578, 579, 581, 596, 600, 602, 603, 611, 613, 616,
617, 618, 619, 620, 622, 637, 641, 643, 654, 657, 661, 663, 666,
667, 668, 671, 672, 673, 681, 682, 685, 686, 688, 689, 690, 691,
694, 701, 712, 713, 720, 722, 724, 725, 726, 727, 728, 729, 731,
732, 733, 734, 735, 737, 738, 739, 740, 741, 742, 743, 744, 745,
746, 747, 748, 749, 751, 753, 754, 755, 762, 770, 778, 779, 782,
783, 788, 789, 790, 791, 792, 797, 800, 801, 802, 803, 820, 827,
829, 835, 836, 838, 840, 845, 847, 848, 849, 850, 851, 855, 856,
857, 858, 862, 863, 867, 868, 869, 870, 872, 873, 874, 876, 877,
878, 879, 881, 882, 884, 885, 886, 887, 888, 889)

```

```

FinalFullData <- na.omit(DataSet)
aggr(FinalFullData)

```



```
write.csv(FinalFullData, file = "~/Library/Mobile Documents/com~apple~CloudDocs/NBA Players Project/Data")
```