warmup04_Jose_Lucar.Rmd

Jose Lucar

9/19/2018

Download the Data

```
curl -O https://raw.githubusercontent.com/ucb-stat133/stat133-fall-2018/master/data/nba2018.csv
```

```
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                  Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:02 --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:03 --:--:--     0
100 94937  100 94937    0     0  24758      0  0:00:03  0:00:03 --:--:-- 24755
```

1) Import the Data in R

- **You have to explicitly specify the data-type for each column as follows:**
  - **the columns player, team, height, birth_date, country, experience, and college have to be declared as type character.**
  - **the column position has to be declared as a factor with levels 'C', 'PF', 'PG','SF', 'SG'.**
  - **the columns salary, field_goals_perc, points3_perc, points2_perc, points1_perc,and effective_field_goal_perc have to be declared as type double (or real).**
  - **the rest of the columns have to be declared as type integer.**
  - **recall that read_csv() uses the argument col_types to specify data types**

```
library(readr)
nbaData<-read_csv("nba2018.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   player = col_character(),
##   number = col_character(),
##   team = col_character(),
##   position = col_character(),
##   height = col_character(),
##   birth_date = col_character(),
##   country = col_character(),
##   experience = col_character(),
##   college = col_character(),
##   salary = col_double(),
##   field_goals_perc = col_double(),
##   points3_perc = col_double(),
##   points2_perc = col_double(),
##   effective_field_goal_perc = col_double(),
##   points1_perc = col_double()
```

```
## )
```

```
## See spec(...) for full column specifications.
```

```
nbaData<-read_csv("nba2018.csv",col_names=TRUE, col_types=cols(.default= col_integer(),player=col_chara
'SF', 'SG'))))
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 2)
```

```
## Warning: 1 parsing failure.
## row # A tibble: 1 x 5 col     row    col               expected actual          file expected   <int
```

```
str(nbaData)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    477 obs. of  38 variables:
##  $ player                   : chr  "Al Horford" "Amir Johnson" "Avery Bradley" "Demetrius Jackson" .
##  $ number                   : int  42 90 0 9 30 4 99 13 7 8 ...
##  $ team                     : chr  "BOS" "BOS" "BOS" "BOS" ...
##  $ position                 : Factor w/ 5 levels "C","PF","PG",..: 1 2 5 3 4 3 4 5 4 2 ...
##  $ height                   : chr  "6-10" "6-9" "6-2" "6-1" ...
##  $ weight                   : int  245 240 180 201 205 185 235 215 225 231 ...
##  $ birth_date               : chr  "June 3, 1986" "May 1, 1987" "November 26, 1990" "September 7, 19
##  $ country                  : chr  "do" "us" "us" "us" ...
##  $ experience               : chr  "9" "11" "6" "R" ...
##  $ college                  : chr  "University of Florida" NA "University of Texas at Austin" "Unive
##  $ salary                   : num  26540100 12000000 8269663 1450000 1410598 ...
##  $ rank                     : int  4 6 5 15 11 1 3 13 8 10 ...
##  $ age                      : int  30 29 26 22 31 27 26 21 20 29 ...
##  $ games                    : int  68 80 55 5 47 76 72 29 78 78 ...
##  $ games_started            : int  68 77 55 0 0 76 72 0 20 6 ...
##  $ minutes                  : int  2193 1608 1835 17 538 2569 2335 220 1341 1232 ...
##  $ field_goals              : int  379 213 359 3 95 682 333 25 192 114 ...
##  $ field_goals_atts         : int  801 370 775 4 232 1473 720 58 423 262 ...
##  $ field_goals_perc         : num  0.473 0.576 0.463 0.75 0.409 0.463 0.463 0.431 0.454 0.435 ...
##  $ points3                  : int  86 27 108 1 39 245 157 12 46 45 ...
##  $ points3_atts             : int  242 66 277 1 111 646 394 35 135 130 ...
##  $ points3_perc             : num  0.355 0.409 0.39 1 0.351 0.379 0.398 0.343 0.341 0.346 ...
##  $ points2                  : int  293 186 251 2 56 437 176 13 146 69 ...
##  $ points2_atts             : int  559 304 498 3 121 827 326 23 288 132 ...
##  $ points2_perc             : num  0.524 0.612 0.504 0.667 0.463 0.528 0.54 0.565 0.507 0.523 ...
##  $ effective_field_goal_perc: num  0.527 0.612 0.533 0.875 0.494 0.546 0.572 0.534 0.508 0.521 ...
##  $ points1                  : int  108 67 68 3 33 590 176 6 85 26 ...
##  $ points1_atts             : int  135 100 93 6 41 649 217 9 124 37 ...
##  $ points1_perc             : num  0.8 0.67 0.731 0.5 0.805 0.909 0.811 0.667 0.685 0.703 ...
##  $ off_rebounds             : int  95 118 65 2 16 43 48 6 45 59 ...
##  $ def_rebounds             : int  370 248 269 2 68 162 367 20 175 213 ...
##  $ total_rebounds           : int  465 366 334 4 84 205 415 26 220 272 ...
##  $ assists                  : int  337 140 122 3 33 448 155 4 64 71 ...
##  $ steals                   : int  52 51 68 0 9 70 73 10 35 25 ...
##  $ blocks                   : int  86 62 11 0 7 13 23 2 18 17 ...
##  $ turnovers                : int  115 77 88 0 25 210 80 4 68 39 ...
##  $ fouls                    : int  138 211 141 0 48 167 161 15 142 122 ...
##  $ points                   : int  952 520 894 10 262 2199 999 68 515 299 ...
##  - attr(*, "problems")=Classes 'tbl_df', 'tbl' and 'data.frame': 1 obs. of  5 variables:
##   ..$ row     : int 20
##   ..$ col     : chr "number"
```

```
##   ..$ expected: chr "no trailing characters"
##   ..$ actual  : chr "-32"
##   ..$ file    : chr "'nba2018.csv'"
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 38
##   .. ..$ player                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ number                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ team                     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ position                 :List of 3
##   .. .. ..$ levels    : chr  "C" "PF" "PG" "SF" ...
##   .. .. ..$ ordered   : logi FALSE
##   .. .. ..$ include_na: logi FALSE
##   .. .. ..- attr(*, "class")= chr  "collector_factor" "collector"
##   .. ..$ height                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ weight                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ birth_date               : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ country                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ experience               : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ college                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ salary                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ rank                     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ age                      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ games                    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ games_started            : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ minutes                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ field_goals              : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ field_goals_atts         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ field_goals_perc         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ points3                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ points3_atts             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ points3_perc             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ points2                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
```

```
##   .. ..$ points2_atts             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ points2_perc             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ effective_field_goal_perc: list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ points1                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ points1_atts             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ points1_perc             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ off_rebounds             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ def_rebounds             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ total_rebounds           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ assists                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ steals                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ blocks                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ turnovers                : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ fouls                    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ points                   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

2) Right after importing the data

- **Once you have the data in R, do a bit of preprocessing on the columns *salary and experience experience* should be of type character because of the presence of the R values that indicate *rookie players* Replace all the occurrences of "R" with 0, and then convert the entire column into *integers*. Display the** summary()** of this column.**

```
nbaData$experience[nbaData$experience=="R"]<- 0
nbaData$experience<- as.integer(nbaData$experience)
summary(nbaData$experience)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   4.000   4.662   7.000  18.000
```

- ***salary* is originally measured in dollars. Transform salary so that you have salaries in millions: e.g. 1000000 should be converted to 1. Display the summary() of this column.**

```
nbaData$salary<- nbaData$salary/100000
```

- **position should be a factor with 5 levels: 'C', 'PF', 'PG', 'SF', 'SG'. Relabel these factors using more descriptive names (see below). Display the frequencies of the relabeled factor**

with table()

```
library(plyr)
nbaData$position<-revalue(nbaData$position, c("C"= "center","PF"="power_fwd","PG"="point_guard","SF"="sm

nbaData$position
```

```
##   [1] center      power_fwd   shoot_guard point_guard small_fwd
##   [6] point_guard small_fwd   shoot_guard small_fwd   power_fwd
##  [11] power_fwd   center      shoot_guard point_guard center
##  [16] center      center      small_fwd   point_guard power_fwd
##  [21] center      shoot_guard shoot_guard small_fwd   shoot_guard
##  [26] point_guard power_fwd   shoot_guard point_guard center
##  [31] small_fwd   small_fwd   center      small_fwd   point_guard
##  [36] point_guard shoot_guard small_fwd   point_guard center
##  [41] center      point_guard center      shoot_guard small_fwd
##  [46] power_fwd   power_fwd   power_fwd   small_fwd   shoot_guard
##  [51] point_guard power_fwd   center      shoot_guard center
##  [56] center      point_guard small_fwd   center      power_fwd
##  [61] small_fwd   shoot_guard point_guard small_fwd   point_guard
##  [66] center      power_fwd   shoot_guard point_guard small_fwd
##  [71] power_fwd   shoot_guard point_guard small_fwd   center
##  [76] power_fwd   power_fwd   small_fwd   small_fwd   shoot_guard
##  [81] small_fwd   point_guard small_fwd   center      power_fwd
##  [86] shoot_guard center      small_fwd   shoot_guard point_guard
##  [91] power_fwd   center      power_fwd   shoot_guard power_fwd
##  [96] power_fwd   power_fwd   center      shoot_guard point_guard
## [101] center      small_fwd   power_fwd   small_fwd   point_guard
## [106] point_guard power_fwd   shoot_guard power_fwd   shoot_guard
## [111] center      small_fwd   power_fwd   point_guard power_fwd
## [116] shoot_guard power_fwd   point_guard center      shoot_guard
## [121] shoot_guard shoot_guard point_guard small_fwd   center
## [126] point_guard power_fwd   small_fwd   shoot_guard point_guard
## [131] center      shoot_guard point_guard center      power_fwd
## [136] power_fwd   shoot_guard small_fwd   small_fwd   power_fwd
## [141] shoot_guard point_guard center      shoot_guard center
## [146] center      center      point_guard center      shoot_guard
## [151] power_fwd   point_guard power_fwd   shoot_guard small_fwd
## [156] shoot_guard small_fwd   point_guard small_fwd   power_fwd
## [161] shoot_guard point_guard point_guard power_fwd   center
## [166] center      shoot_guard power_fwd   point_guard shoot_guard
## [171] power_fwd   small_fwd   center      shoot_guard point_guard
## [176] shoot_guard small_fwd   point_guard shoot_guard point_guard
## [181] center      shoot_guard power_fwd   center      power_fwd
## [186] center      power_fwd   small_fwd   shoot_guard shoot_guard
## [191] center      small_fwd   small_fwd   center      center
## [196] point_guard shoot_guard point_guard small_fwd   point_guard
## [201] shoot_guard power_fwd   shoot_guard shoot_guard small_fwd
## [206] center      shoot_guard center      small_fwd   power_fwd
## [211] power_fwd   shoot_guard shoot_guard center      point_guard
## [216] center      small_fwd   power_fwd   shoot_guard center
## [221] small_fwd   point_guard center      point_guard center
## [226] small_fwd   power_fwd   power_fwd   shoot_guard center
## [231] small_fwd   point_guard point_guard point_guard shoot_guard
## [236] center      small_fwd   power_fwd   power_fwd   shoot_guard
```

```
## [241] small_fwd   shoot_guard point_guard power_fwd   point_guard
## [246] center      small_fwd   center      center      power_fwd
## [251] shoot_guard power_fwd   center      power_fwd   center
## [256] shoot_guard small_fwd   shoot_guard point_guard point_guard
## [261] center      shoot_guard shoot_guard power_fwd   power_fwd
## [266] point_guard center      center      shoot_guard small_fwd
## [271] small_fwd   power_fwd   shoot_guard point_guard point_guard
## [276] center      point_guard point_guard center      center
## [281] shoot_guard point_guard point_guard shoot_guard center
## [286] center      shoot_guard power_fwd   power_fwd   small_fwd
## [291] small_fwd   small_fwd   shoot_guard power_fwd   power_fwd
## [296] power_fwd   point_guard center      center      shoot_guard
## [301] shoot_guard small_fwd   center      small_fwd   point_guard
## [306] small_fwd   shoot_guard power_fwd   point_guard power_fwd
## [311] point_guard small_fwd   center      small_fwd   small_fwd
## [316] power_fwd   point_guard shoot_guard center      point_guard
## [321] power_fwd   shoot_guard small_fwd   power_fwd   small_fwd
## [326] center      power_fwd   power_fwd   small_fwd   power_fwd
## [331] point_guard point_guard point_guard center      power_fwd
## [336] shoot_guard point_guard power_fwd   small_fwd   center
## [341] small_fwd   power_fwd   power_fwd   center      point_guard
## [346] point_guard shoot_guard shoot_guard small_fwd   point_guard
## [351] shoot_guard power_fwd   power_fwd   shoot_guard shoot_guard
## [356] point_guard power_fwd   small_fwd   small_fwd   center
## [361] small_fwd   power_fwd   power_fwd   shoot_guard point_guard
## [366] small_fwd   small_fwd   power_fwd   point_guard shoot_guard
## [371] shoot_guard point_guard center      power_fwd   power_fwd
## [376] shoot_guard center      small_fwd   center      center
## [381] shoot_guard small_fwd   center      center      power_fwd
## [386] power_fwd   center      power_fwd   shoot_guard point_guard
## [391] shoot_guard point_guard center      power_fwd   point_guard
## [396] small_fwd   small_fwd   point_guard center      power_fwd
## [401] shoot_guard point_guard power_fwd   power_fwd   center
## [406] power_fwd   point_guard power_fwd   point_guard shoot_guard
## [411] center      shoot_guard point_guard center      point_guard
## [416] shoot_guard power_fwd   shoot_guard shoot_guard shoot_guard
## [421] point_guard shoot_guard center      point_guard center
## [426] point_guard shoot_guard small_fwd   power_fwd   point_guard
## [431] small_fwd   center      power_fwd   small_fwd   shoot_guard
## [436] center      power_fwd   point_guard center      center
## [441] point_guard power_fwd   point_guard small_fwd   point_guard
## [446] shoot_guard small_fwd   small_fwd   point_guard shoot_guard
## [451] center      shoot_guard power_fwd   small_fwd   small_fwd
## [456] shoot_guard center      power_fwd   center      point_guard
## [461] center      center      shoot_guard small_fwd   shoot_guard
## [466] power_fwd   shoot_guard point_guard power_fwd   small_fwd
## [471] shoot_guard shoot_guard power_fwd   point_guard small_fwd
## [476] point_guard center
## Levels: center power_fwd point_guard small_fwd shoot_guard
```

```
table(nbaData$position)
```

```
##
##      center   power_fwd point_guard   small_fwd shoot_guard
##          97          98          96          84         102
```

3) A bit of subscripting (i.e. indexing, slicing, subsetting)

- **Use bracket notation, the dollar operator, as well as concepts of logical subsetting and indexing to calculate:**
- **How many players went to UCLA ("University of California, LosAngeles")?**

```
rows_ucla<-c(which(nbaData$college=="University of California, Los Angeles"))
data_ucla<-nbaData[rows_ucla, ]
nrow(data_ucla)
```

```
## [1] 14
```

- **How many players went to Cal ("University of California, Berkeley")?**

```
rows_ucb<-c(which(nbaData$college=="University of California"))
nbaData[rows_ucb, ]
```

```
## # A tibble: 3 x 38
##          player number  team    position height weight       birth_date
##           <chr>  <int> <chr>      <fctr>  <chr>  <int>            <chr>
## 1  Jaylen Brown      7   BOS   small_fwd    6-7    225 October 24, 1996
## 2 Ryan Anderson      3   HOU   power_fwd   6-10    240      May 6, 1988
## 3  Allen Crabbe     23   POR shoot_guard    6-6    210    April 9, 1992
## # ... with 31 more variables: country <chr>, experience <int>,
## #   college <chr>, salary <dbl>, rank <int>, age <int>, games <int>,
## #   games_started <int>, minutes <int>, field_goals <int>,
## #   field_goals_atts <int>, field_goals_perc <dbl>, points3 <int>,
## #   points3_atts <int>, points3_perc <dbl>, points2 <int>,
## #   points2_atts <int>, points2_perc <dbl>,
## #   effective_field_goal_perc <dbl>, points1 <int>, points1_atts <int>,
## #   points1_perc <dbl>, off_rebounds <int>, def_rebounds <int>,
## #   total_rebounds <int>, assists <int>, steals <int>, blocks <int>,
## #   turnovers <int>, fouls <int>, points <int>
```

```
data_berkeley<-nbaData[rows_ucb, ]
nrow(data_berkeley)
```

```
## [1] 3
```

- **What's the largest weight value?**

```
max(nbaData$weight)
```

```
## [1] 290
```

- **Who are the players with the largest weight value?**

```
which.max(nbaData$weight)
```

```
## [1] 149
```

```
heavy_player<-nbaData$player[149]
heavy_player
```

```
## [1] "Boban Marjanovic"
```

- **What's the overall average weight?**

```
mean(nbaData$weight)
```

```
## [1] 219.9119
```

- **What is the median salary of all players?**

```
median(nbaData$salary)
```

```
## [1] 30
```

- **What is the median salary of the players with 10 years of experience or more?**

```
experienced_players <-c(which(nbaData$experience>10))
data_experienced_players<-nbaData[experienced_players, ]
median(data_experienced_players$salary)
```

```
## [1] 45.61988
```

- **What is the median salary of Shooting Guards (SG) and Point Guards (PG)?**

```
rows_shoot_guard<-c(which(nbaData$position=="shoot_guard"))
rows_point_guard<-c(which(nbaData$position=="point_guard"))
rows_shoot_point_guard<- c(rows_shoot_guard,rows_point_guard)

data_guards<-nbaData[rows_shoot_point_guard, ]

median(data_guards$salary)
```

```
## [1] 27.89697
```

- **What is the median salary of Power Forwards (PF), 30 years or older, weighing 240 pounds or more?**

```
rows_PF<-c(which(nbaData$position=="power_fwd"))
data_PF<- nbaData[rows_PF, ]
rows_30plus<-c(which(data_PF$age>29))
data_PF_30plus<- data_PF[rows_30plus, ]
rows_heavy_oldy_PF<-c(which(data_PF_30plus$weight>=240))
data_heavy_oldy_PF<- data_PF_30plus[rows_heavy_oldy_PF, ]
median(data_heavy_oldy_PF$salary)
```

```
## [1] 80
```

- **Create a data frame gsw with the player name, position, height, weight, and age of Golden State Warriors (GSW). Display this data frame.**

```
GSW_rows <-c(which(nbaData$team=="GSW"))
GSW_rows
```

```
##  [1] 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261
```

```
GSW_data<-nbaData[GSW_rows, ]

gsw<-c("player", "position", "height", "weight", "age")
small_GSW_data<-GSW_data[ ,gsw]

small_GSW_data
```

```
## # A tibble: 16 x 5
##                  player    position height weight   age
##                   <chr>      <fctr>  <chr>  <int> <int>
##  1     Anderson Varejao      center   6-10    273    34
##  2       Andre Iguodala   small_fwd    6-6    215    33
##  3         Damian Jones      center    7-0    245    21
```

```
##  4           David West      center    6-9    250    36
##  5       Draymond Green   power_fwd    6-7    230    26
##  6            Ian Clark shoot_guard    6-3    175    25
##  7 James Michael McAdoo   power_fwd    6-9    230    24
##  8         JaVale McGee      center    7-0    270    29
##  9         Kevin Durant   power_fwd    6-9    240    28
## 10         Kevon Looney      center    6-9    220    20
## 11        Klay Thompson shoot_guard    6-7    215    26
## 12          Matt Barnes   small_fwd    6-7    226    36
## 13        Patrick McCaw shoot_guard    6-7    185    21
## 14     Shaun Livingston point_guard    6-7    192    31
## 15        Stephen Curry point_guard    6-3    190    28
## 16        Zaza Pachulia      center   6-11    270    32
```

4) Performance of players

- **missed_field_goals (missed field goals)**

```
missed_field_goals <- c(nbaData$field_goals_atts-nbaData$field_goals)
```

- **missed_free_throws (missed free throws)**

```
missed_free_throws<-c(nbaData$points1_atts-nbaData$points1)
```

- **rebounds (total rebounds: offensive and defensive)**

```
rebounds<- c(nbaData$total_rebounds)
```

- **mins_game (minutes per game; NOT to be used when calculating EFF)**

```
mins_game<-c(nbaData$minutes/nbaData$games)
```

- **You will have to compute the efficiency (EFF) for each player.**

```
efficiency<- c((nbaData$points + nbaData$total_rebounds + nbaData$assists + nbaData$steals + nbaData$bl
```

- **Once you have all the necessary statistics, add a column efficiency to the data frame using the formula provided above.**

```
nbaData$efficiency<- efficiency
```

- **Compute summary() statistics for efficiency**

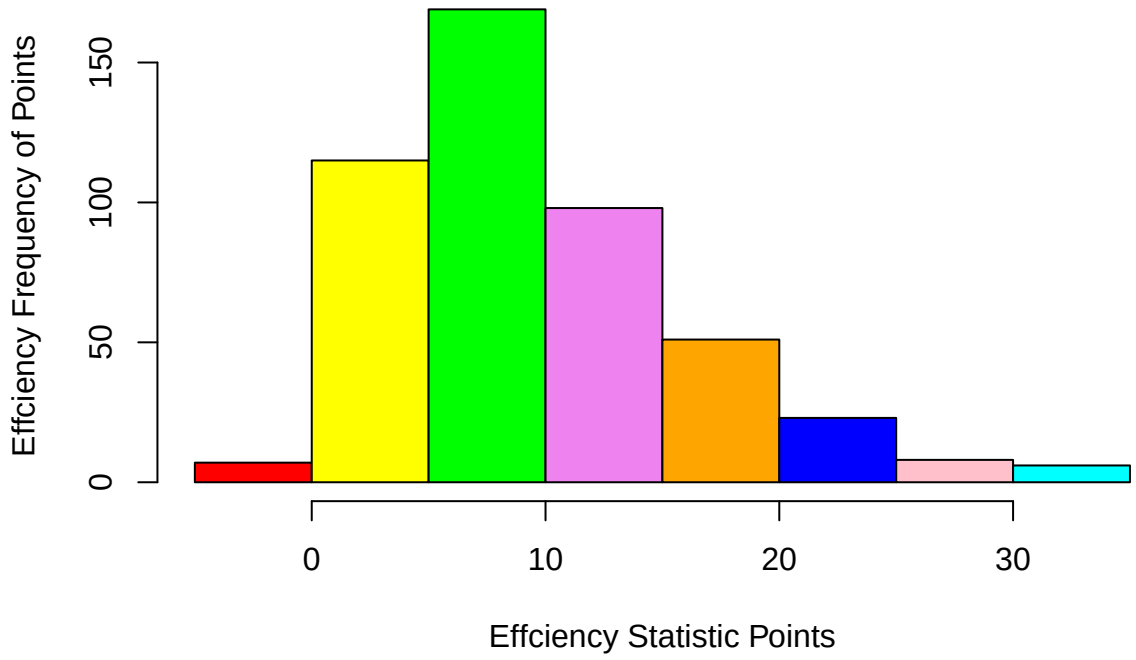
```
summary(nbaData$efficiency)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6667  5.0000  8.3472  9.5788 12.6066 33.8272
```

- **graph its histogram. Add color to the bars in hte histograms, and make sure it includes descriptive axis labels, as well as a title.**

```
colors = c("red", "yellow", "green", "violet", "orange",
"blue", "pink", "cyan")

hist(nbaData$efficiency, col = colors, main = "Histogram Efficiency of NBA Players", xlab = "Effciency
```



- **Display the player name, team, salary, and efficiency value of the top-10 players by EFF in decreasing order (display this information in a data frame).**

```
new_top10_efficiency_data<-head(nbaData[order(-efficiency),], n=10)

new_vec_eff<-c("player", "team", "salary", "efficiency")

new_top<-new_top10_efficiency_data[ ,new_vec_eff]

new_top
```

```
## # A tibble: 10 x 4
##                    player  team    salary efficiency
##                     <chr> <chr>     <dbl>      <dbl>
##  1     Russell Westbrook   OKC 265.40100   33.82716
##  2          James Harden   HOU 265.40100   32.38272
##  3         Anthony Davis   NOP 221.16750   31.14667
##  4          LeBron James   CLE 309.63450   30.95946
##  5    Karl-Anthony Towns   MIN  59.60160   30.30488
##  6          Kevin Durant   GSW 265.40100   30.19355
##  7 Giannis Antetokounmpo   MIL  29.95421   28.37500
##  8      DeMarcus Cousins   NOP 169.57900   27.88235
##  9          Jimmy Butler   CHI 175.52209   25.60526
## 10      Hassan Whiteside   MIA 221.16750   25.37662
```

- **Did you find players with a negative EFF? If yes, display their names.**
  - *Yes* :

```
rows_neg_eff<-which(nbaData$efficiency<0)
nbaData$player[rows_neg_eff]
```

```
## [1] "Gary Neal"       "Axel Toupane"    "Patricio Garino" "Ben Bentil"
```

5) Further Exploration

- **The more efficient the player is, the highest his salary**

```
new_top
```

```
## # A tibble: 10 x 4
##                    player  team    salary efficiency
##                     <chr> <chr>     <dbl>      <dbl>
##  1     Russell Westbrook   OKC 265.40100   33.82716
##  2          James Harden   HOU 265.40100   32.38272
##  3         Anthony Davis   NOP 221.16750   31.14667
##  4          LeBron James   CLE 309.63450   30.95946
##  5    Karl-Anthony Towns   MIN  59.60160   30.30488
##  6          Kevin Durant   GSW 265.40100   30.19355
##  7 Giannis Antetokounmpo   MIL  29.95421   28.37500
##  8      DeMarcus Cousins   NOP 169.57900   27.88235
##  9          Jimmy Butler   CHI 175.52209   25.60526
## 10      Hassan Whiteside   MIA 221.16750   25.37662
```

- No, neccesarily. For example LeBrom James efficiency score is 30.95946 and his salary is 309.63450 in

- **As players get older, do they tend to become more efficient?**

```
age<-nbaData$age
new_top400_age_data<-head(nbaData[order(-age,-efficiency), ], n=400)
age_efficiency<-new_top400_age_data[ ,c("age","efficiency")]
age_efficiency
```

```
## # A tibble: 400 x 2
##      age efficiency
##    <int>      <dbl>
##  1    40   9.150685
##  2    39   8.260870
##  3    39   5.378378
##  4    39   3.480000
##  5    38  15.277778
##  6    37   1.520000
##  7    36  17.203125
##  8    36   9.600000
##  9    36   9.170732
## 10    36   8.117647
## # ... with 390 more rows
```

- No, there isn't any pattern associated with the age of a player and their age. The only possible corre

- **Does the rank of a player seem to be associated with his efficiency (i.e. the more importnat the rank, the more efficient)?**

```
rank<- nbaData$rank
new_top100_rank_data<-head(nbaData[order(-efficiency,rank), ],n=100)
rank_efficiency<-new_top100_rank_data[ ,c("rank","efficiency")]
rank_efficiency
```

```
## # A tibble: 100 x 2
##     rank efficiency
```

```
##    <int>      <dbl>
##  1     1   33.82716
##  2     1   32.38272
##  3     1   31.14667
##  4     1   30.95946
##  5     2   30.30488
##  6     4   30.19355
##  7     1   28.37500
##  8    11   27.88235
##  9     1   25.60526
## 10     1   25.37662
## # ... with 90 more rows
```

```
- Only for the players with the highest efficiency scores, it looks like it shows some association betw
```

6) Comments and Reflection

- **How much time did it take to complete this HW?**
  - It took me around a week and 3 to 4 well dedicated days, days were I will only wake up to do this assigment. In addition, I'm a DSP student, so it usually take me longer than most people to complete assigments and this is why I have turn in late assidgments as one of my accomodations.
- **What things were hard, even though you saw them in class/lab?**
  - Importing data and counting specific values of columns in the data frame. the link for checking out how to import data with declared classes of columns didn't provide enough information, so I had google a lot to learn how to do it. In class with imported data, but I'm not sure how similar it was to what we were asked to do in part 1 of this assigment. Also, since there is many ways in which you can import data, there is a lot of information on how to do it which it doesn't neccesarily make it easier because I was looking for a specific way to do it.
- **What was easy(-ish) even though we haven't done it in class/lab?**
  - Calculating efficiency of players was the easiest part because it only involves mathematical calculation using data that was easy to acces from the data frame.
- **Did you need help to complete the assignment? If so, what kind of help?**
  - Yes, I have a friend that helps me understand what I'm asked to do. Also, It is always helpful to discuss with someone what is the best way to approach a question to provide a most efficient and clear answer.
- **What was the most time consuming part?**
  - Part one and part 3. Part 1 because I had to research by reading and whatching videos online to learn how to use read_csv with its different arguments and part 3 because it was a bit hard to find information on how to count an specific value of a column inside a dta frame.
- **Was there anything that you did not understand? or fully grasped?**
  - Maybe, I would like to keep practicing more about importing different type of data with different arguments. I know we've been doing a lot since the biggining of the semester, but I would like to continue to get better at it, so ma data can be more manageable since the beggining of my data analysis.
- **Was there anything frustrating in particular?**
  - Part 1 was time consuming, but I was learning a lot by researching the web. However: Part 3 was just to get the right specific code and most things I was finding on my search were things that I

knew already or I didn't need it at all. Part 3 was also time consuming as Part 1, but without the fan stuff of learning. I feel that if I knew someone that knows **R**, this would of be the easiest one.