Warm-Up 04 - Exploring NBA Data

Stat 133, Fall 2018, Prof. Sanchez

Due date: Sep-25 (before midnight)

The purpose of this assignment is to keep working with data frames. Use this HW to keep developing your manipulation skills of basic data objects in R: reading data tables, understanding data frames, use of bracket notation, the dollar operator, and become more and more familiar with the associated NBA data set which now includes more variables.

General Instructions

- Write your narrative and code in an `Rmd` (R markdown) file.
- Name this file as `warmup04-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `warmup04-gaston-sanchez.Rmd`).
- Please do not use code chunk options such as: `echo = FALSE`, `eval = FALSE`, `results = 'hide'`. All chunks must be visible and evaluated.
- Submit your Rmd and html files to bCourses.

Data

The data set for this assignment is in the file `nba2018.csv`, inside the `data/` folder of the github repo `stat133-fall-2018`.

Download the data

To get a copy of the data file, use the shell command `curl`. We recomend that you create a directory especifically dedicated to this warmup: e.g. `warmup04/`.

```
# directory for this warmup
mkdir warmup04
cd markup04

# assuming you are inside directory warmup04/
# (run in a single line of text)
curl -O https://raw.githubusercontent.com/ucb-stat133/
stat133-fall-2018/master/data/nba2018.csv
```

Below is the description of variables in `nba2018.csv`:

- `player`: first and last names of player
- `number`: number on jersey
- `team`: 3-letter team abbreviation
- `position`: player's position
- `height`: height in feet-inches
- `weight`: weight in pounds
- `birth_date`: date of birth ("Month day, year")
- `country`: 2-letter country abbreviation
- `experience`: years of experience in NBA (a value of `R` means rookie)
- `college`: attended college in USA
- `salary`: player salary in dollars
- `rank`: Rank of player in his team
- `age`: Age of Player at the start of February 1st of that season.
- `games`: Games Played furing regular season
- `sames_started`: Games Started
- `minutes`: Minutes Played during regular season
- `field_goals`: Field Goals Made
- `field_goals_atts`: Field Goal Attempts
- `field_goals_perc`: Field Goal Percentage
- `points3`: 3-Point Field Goals
- `points3_atts`: 3-Point Field Goal Attempts
- `points3_perc`: 3-Point Field Percentage
- `points2`: 2-Point Field Goals
- `points2_atts`: 2-Point Field Goal Attempts
- `points2_perc`: 2-Point Field Goal Percentage
- `effective_field_goal_perc`: Effective Field Goal Percentage
- `points1`: Free Throws Made
- `points1_atts`: Free Throw Attempts
- `points1_perc`: Free Throw Percentage
- `off_rebounds`: Offensive Rebounds
- `def_rebounds`: Defensive Rebounds
- `assists`: Assists
- `steals`: Steals
- `blocks`: Blocks
- `turnovers`: Turnovers
- `fouls`: Fouls
- `points`: Total points

1) Import the data in R

In previous assignments you've practiced importing data tables using *R base* functions such as: `read.table()` and friends—e.g. `read.csv()`, `read.delim()`. Another major approach is the one provided by the family of functions in the package `"readr"`.

In this assignment, you will have to import the data using "readr". Here's a couple of resources for importing data (if you google about this topic you'll find more links):

- https://www.r-bloggers.com/using-colclasses-to-load-data-more-quickly-in-r/
- https://cran.r-project.org/web/packages/readr/vignettes/readr.html

Include one chunk with the code to import the data with the function read_csv(). And use str() to display its structure.

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.

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.

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.

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().

- center instead of C
- power_fwd' instead of PF
- point_guard' instead of PG
- small_fwd instead of SF
- shoot_guard instead of SG

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, Los Angeles")?
- How many players went to Cal ("University of California, Berkeley")?
- What's the largest weight value?
- Who are the players with the largest weight value?
- What's the overall average weight?
- What is the median salary of all players?
- What is the median salary of the players with 10 years of experience or more?
- What is the median salary of Shooting Guards (SG) and Point Guards (PG)?
- What is the median salary of Power Forwards (PF), 30 years or older, weighing 240 pounds or more?
- Create a data frame `gsw` with the player name, position, height, weight, and age of Golden State Warriors (GSW). Display this data frame.

4) Performance of players

Performance of NBA players can be measured in various ways. Perhaps the most popular performance measure is known as the "Efficiency" statistic, simply referred to as **EFF**

https://en.wikipedia.org/wiki/Efficiency_(basketball)

According to Wikipedia, EFF computes performance as an index that takes into account basic individual statistics: points, rebounds, assists, steals, blocks, turnovers, and shot attempts (per game). It is derived by a simple formula:

```
EFF = (PTS + REB + AST + STL + BLK - Missed FG - Missed FT - TO) / GP
```

- `EFF`: efficiency
- `PTS`: total points
- `REB`: total rebounds
- `AST`: assists
- `STL`: steals
- `BLK`: blocks
- `Missed FG`: missed field goals
- `Missed FT`: missed free throws
- `TO`: turnovers
- `GP`: games played

In case you are curious, you can find more information about the player statistics and related acronyms in the following wikipedia entry:

https://en.wikipedia.org/wiki/Basketball_statistics

You will have to compute the efficiency (`EFF`) for each player. In order to do this, you'll have to add the following variables to the main data frame:

- `missed_field_goals` (missed field goals)
- `missed_free_throws` (missed free throws)
- `rebounds` (total rebounds: offensive and defensive)
- `mins_game` (minutes per game; NOT to be used when calculating `EFF`)

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

Compute `summary()` statistics for `efficiency` and graph its histogram. Add color to the bars in hte histograms, and make sure it includes descriptive axis labels, as well as a title.

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).

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

5) Further Exploration

Answer the following questions, and provide numerical and/or graphical evidence to support your answer:

- The more efficient a player is, the higher his salary?
- As players get older, do they tend to become more efficient?
- Does the rank of a player seem to be associated with his efficiency (i.e. the more importnat the rank, the more efficient)?

6) Comments and Reflections

Reflect on what was hard/easy, problems you solved, helpful tutorials you read, etc.

- How much time did it take to complete this HW?
- What things were hard, even though you saw them in class/lab?
- What was easy(-ish) even though we haven't done it in class/lab?
- Did you need help to complete the assignment? If so, what kind of help?
- What was the most time consuming part?
- Was there anything that you did not understand? or fully grasped?
- Was there anything frustrating in particular?