HW 02 - Basics of Data Frames

Stat 133, Fall 2017, Prof. Sanchez

Due date: Mon Oct-02 (before midnight)

The purpose of this assignment is to start 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

- Create a folder (i.e. subdirectory) `hw02` in your `stat133-hws-fall17` local repository. This is where you will save all the associated files for this assignment.
- Inside the folder `hw02`, create a `README.md` file with similar contents to the `README.md` file of the first assignment in `hw01`.
- Inside the folder `hw02` create a `data` folder which will contain the csv data file, and the data dictionary.
- Write your narrative and code in an `Rmd` (R markdown) file.
- In the yaml header, set the `output` field as `output: github_document` (Do NOT use the default `"output: html_document"`).
- Name this file as `hw02-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `hw02-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.
- Use Git to *add* and *commit* the changes as you progress with your HW. Track changes in the `Rmd` and `md` files, as well as the generated folder and files containing the plot images.
- And don't forget to *push* your commits to your github repository; you should push the `Rmd` and `md` files, as well as the generated folder and files containing the plot images.
- Submit the link of your repository to bCourses. Do NOT submit any files (we will actually turn off the uploading files option).
- We will review, and you will self grade, the work in the knitted `hw02-first-last.md` file of your github repo.
- No html files will be taken into account (no exceptions).
- If you have questions/problems, don't hesitate to ask us for help in OH or piazza.

About the Research Question

In the previous assignment you started to tackle the initial research question: **"the more points a player scores, the higher his salary?"**

Admittedly, this question is kind of open ended. From a narrow point of view, and based on the correlation between points and salary, we could answer the research question like so: *"Yes, in general, the more points scored by a player, the higher his salary"*. We can even take one more step and use the results of the simple linear regression to state that *"on average, for every extra point that a player scores, his salary will tend to increase by 8556.68 dollars"*.

However, the salary of a player will hardly depend on only the number of scored points. Other factors like years of experience, offensive performance, defensive performance, and physical attributes, for instance, may contribute as well to his salary.

In this assignment, you are going to expand the scope of the analysis by taking into account the overall performance of a player. More precisely, the idea is to look at common individual statistics (e.g. points, rebounds, assists, steals, etc.) to try to capture ther *performance* of a player, and see how this may be related with his salary.

Data

The data set for this assignment is in the file `nba2017-player-statistics.csv`, inside the `data/` folder of the course github repo `stat133-fall-2017`.

Download the data

You will need to download a copy of the data file to your local repository. One way to do this from R is with the `download.file()` function. By the way, you should NOT include a code chunk with the previous command in your Rmd file.

```
# download csv file to your working directory
# (do NOT include this code in your Rmd)
github <- "https://github.com/ucb-stat133/stat133-fall-2017/raw/master/"
file <- "data/nba2017-player-statistics.csv"
csv <- paste0(github, file)
download.file(url = csv, destfile = 'nba2017-player-statistics.csv')
```

1) Create a data dictionary

As we saw in lecture, in addition to having a text file for the data table, there should also be a file with the **data dictionary** describing various details about the contents of the data file. For instance, things like:

- what is the data about?
- how many rows?
- how many columns?
- what are the column labels?
- if the column names are abbreviations, what is the full description of each column?
- what are the units of measurement (e.g. inches, pounds, km/h, etc)?
- how missing values are codified?

You need to create a data dictionary file for `nba2017-player-statistics.csv`

- Use a text file to create a data dictionary file.
- Name this file as `nba2017-player-statistics-dictionary.md`
- Save this file inside the `data` folder of the `hw02` subdirectory.
- Use markdown syntax to write the content of the dictionary.
- Include a short title.
- Provide a description of what the data is about.
- Include the main source: www.basketball-reference.com
- Also include a sample link for the data source of a given team (e.g. GS Warriors)
- https://www.basketball-reference.com/teams/GSW/2017.html

Below is the description of variables according to the glossary of basketball-reference

- `Player`: first and last names of player
- `Team`: 3-letter team abbreviation
- `Position`: player's position
- `Experience`: years of experience in NBA (a value of `R` means rookie)
- `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.
- `GP`: Games Played furing regular season
- `GS`: Games Started
- `MIN`: Minutes Played during regular season
- `FGM`: Field Goals Made
- `FGA`: Field Goal Attempts
- `Points3`: 3-Point Field Goals
- `Points3_atts`: 3-Point Field Goal Attempts
- `Points2`: 2-Point Field Goals
- `Points2_atts`: 2-Point Field Goal Attempts
- `FTM`: Free Throws Made
- `FTA`: Free Throw Attempts
- `ORB`: Offensive Rebounds
- `DRB`: Defensive Rebounds
- `TRB`: Total Rebounds
- `AST`: Assists
- `STL`: Steals
- `BLK`: Blocks
- `TO`: Turnovers

2) Import the data in R

We want you to get some practice importing data tables in R. As we saw in lecture, there are multiple ways for reading in tables in R. One major approach consists of using *R base* functions: `read.table()` and friends—e.g. `read.csv()`, `read.delim()`. Another major approach, recently introduced, is the one provided by the family of functions in the package `"readr"`.

You will have to import the data using both approaches. This means that you are going to import the data twice, but this is just for practicing purposes, and so that you can compare both importing styles (`"base"` vs `"readr"`).

What you should include in your Rmd is two code chunks with the commands to import the data, one chunk with `read.csv()`, and the other with `read_csv()`. Create one data frame for the output of `read.csv()`, and use `str()` to display its structure. Likewise, create another data frame for the output of `read_csv()` and use `str()` to check its structure.

In both cases you have to explicitly specify the data-type for each column as follows:

- the columns `Player`, `Team`, and `Experience` 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 column `Salary` has to be declared as type `double` (or real).
- the rest of the 19 columns have to be declared as type `integer`.
- recall that `read.csv()` uses the argument `colClasses` to specify data types.
- recall that `read_csv()` uses the argument `col_types` to specify data types.

In case you need, 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

---

3) Right after importing the data

Once you have the data in R, do a bit of preprocessing on the column `Experience`. This column should be of type character because of the presence of the `R` values that indicate rookie players.

Replace all th eoccurrences of `"R"` with `1`, and then convert the entire column into integers.

4) Performance of players

As we mention above, in this assignment you will take into account other variables rather than just the points scored by each player. More precisely, you are going to consider basic individual

statistics commonly used in the NBA in order to get a proxy of a player's performance.

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)

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

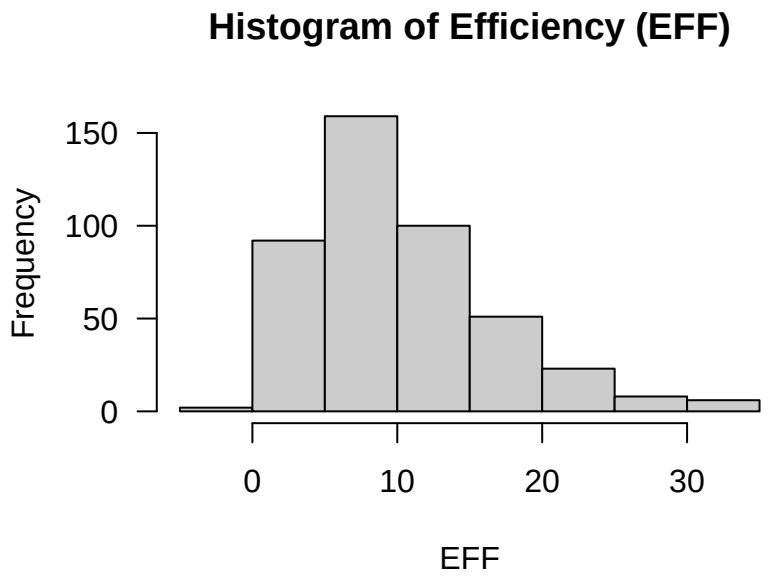
To compute `EFF`, you will have to add the following variables to your data frame:

- `Missed_FG` (missed field goals)
- `Missed_FT` (missed free throws)
- `PTS` (total points)
- `REB` (total rebounds: offensive and defensive)
- `MPG` (minutes per game; NOT to be used when calculating `EFF`)

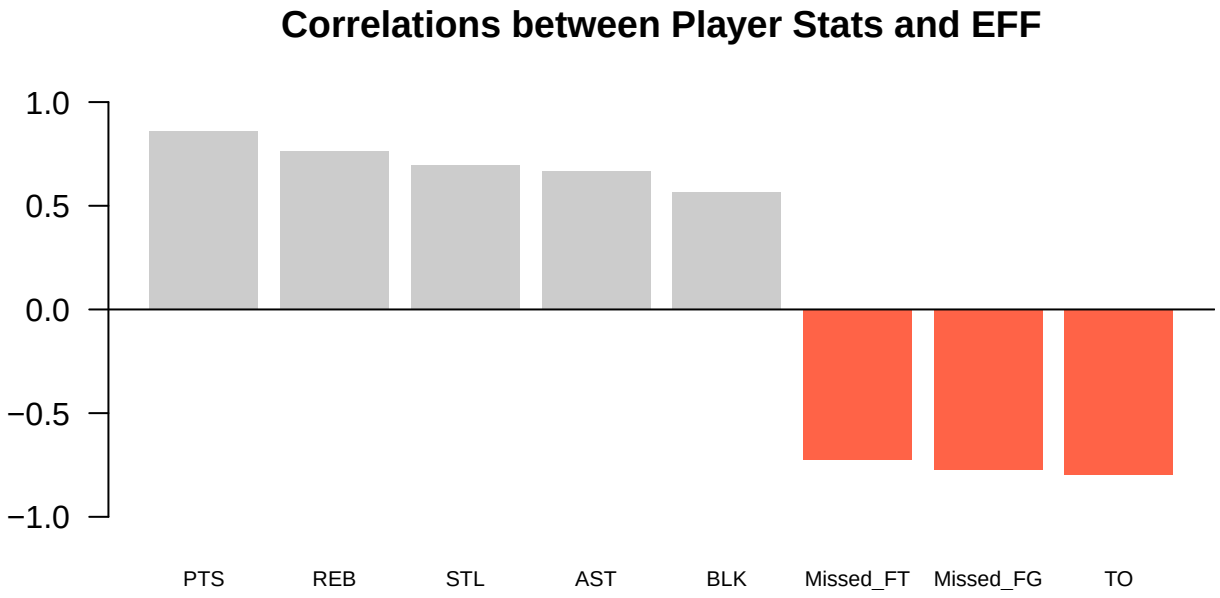
Once you have all the necessary statistics, add a variable `EFF` to the data frame using the formula provided above.

Compute summary statistics for `EFF` and confirm that you have the following results, as well as a similar histogram:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.600   5.452   9.090  10.140  13.250  33.840
```



- Display the player name, team, salary, and `EFF` value of the top-10 players by `EFF` in decreasing order (display this information in a data frame).
- Provide the names of the players that have a negative `EFF`.
- Use the function `cor()` to compute the correlation coefficients between `EFF` and all the variables used in the EFF formula.
- Notice that `Missed_FG`, `Missed_FT`, and `TO` contribute negatively to `EFF`, so make sure to take into account this negative association when calculating the correlation coefficients.
- Display the computed correlations in descending order, either in a vector or a data frame. And create a barchart with the correlations (bars in decreasing order) like the one below.



5) Efficiency and Salary

Once you've calculated the Efficiency statistic, produce a scatterplot between Efficiency (x-axis) and Salary (y-axis), including a *lowess* smooth line (locally weighted scatterplot smoothing). Also, compute the linear correlation coefficient between them. What can you say about the relationship between these two variables?

One aspect that has an important contribution in a player's salary has to do with the years of experience playing in the NBA. In fact, salaries of rookie players follow a special scale, as explained by Tom Ziller in this post

Why NBA players get paid so much more than NFL stars

Because rookie players (and other low-experience players) seem to form a different universe of players, let's see what's happening with those individuals with a more "solid" or more established trajectory in the NBA.

- Taking into account the column `MPG` (minutes per game) select those players that have an `MPG` value of 20 or more minutes per game.
- Create a data frame `players2` with these players.
- Use this data frame to create a scatterplot between Efficiency and Salary, including a *lowess* smooth line.
- Compute the linear correlation coefficient between these variables.
- What can you say about the relationship between these two variables for the set of "more established players"?

In future assignments, we'll consider other research questions, as well as other ways to explore more variables, and perform a deeper multivariate analysis.

6) Comments and Reflections

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

- 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?
- How much time did it take to complete this HW?
- What was the most time consuming part?
- Was there anything that you did not understand? or fully grasped?
- Was there anything frustrating in particular?
- Was there anything exciting? Something that you feel proud of? (Don't be shy, we won't tell anyone).