Workout 1 - Data Wrangling and Visualization

Stat 133, Fall 2018, Prof. Sanchez

Due date: Oct-05 (before midnight)

From the logistical point of view, the purpose of this assignment is twofold. On one hand, you will work with `data.frames/tibbles` and producing plots but now using the packages `"dplyr"` and `"ggplot2"`. On the other hand, we want you to start working with a more complex file structure. Moreover, instead of submitting a combo of `Rmd-html` files to bCourses, you will have to upload all the files to your Github Classroom Repository.

Motivation

In this assignment, you are going to consider different ways to rank the NBA teams. From simple rankings based on a given observed variable, to rankings based on derived indices like efficiency (i.e. `EFF`) or any other composite index.

To make things more interesting, let's pretend that the NBA does not work the way it does. Let's also pretend that the only available data is the player statistics, and nothing else. In other words, we don't know the number of wins (and losses) of each team, or which team won the championship. Moreover, let's assume there is no such championship. All we have is the information about the players, and the goal is to find a ranking for the teams.

If these assumptions and the ranking idea seem awkward, think about the ranking systems of universities, the ranking of companies in a certain industry, or the ranking of countries according to some economic or socio-demographic indicators (see examples below):

- U.S. News National University Rankings
- U.S. News Overall Best Countries Ranking
- Fortune Tech The 30 Best Workplaces in Technology

1) File Structure

After completing this assignment, the file structure of your workout assignment should look like this:

```
workout1/
  README.md
  data/
    nba2018.csv
    nba2018-dictionary.md
    nba2018-teams.csv
  code/
    make-teams-data.R
  output/
    efficiency-summary.txt
    teams-summary.txt
  report/
    workout01-first-last.Rmd
    workout01-first-last.md
    workout01-first-last_files/
      ...  # image files generated by knitr
```

- Create a folder (i.e. subdirectory) `workout1` in your `stat133-hws-fall18` local repository. This is where you will save all the associated files for this assignment.
- Create a `README.md` file.
- Create a folder `data` which will contain the data files.
- Create a folder `code` which will contain an `R` script file.
- Create a folder `output` which will contain some `R` outputs.
- Create a folder `report` which will contain the files for your dynamic document.
- In the yaml header of the `Rmd` file, set the `output` field as `output: github_document` (Do NOT use the default `"output: html_document"`).
- No html files will be taken into account (no exceptions).
- Name this file as `workout01-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `workout01-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.
- 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).**

Download the data

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

To get a copy of the data file, use the shell command `curl`.

```
# assuming you are inside directory workout1/data/
# (run in a single line of text)
curl -O https://raw.githubusercontent.com/ucb-stat133/
stat133-fall-2018/master/data/nba2018.csv
```

2) Create a `README.md` File

Create a `README.md` file and include a description of what the HW is about. If this is your first time creating this type of file, and you are not sure about what to include, then think in your future self. Picture yourself 6 months later (or one year later) and coming back to see what you did for this assignment. What things would you like to see in the `README` file in order to refresh your memory?

Another suggestion is to think of a potential user/reader that looks at your work. What would you like to tell me in case they quickly inspect this assignment and the first thing they look at is the `README` file?

3) 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 `nba2018.csv`

- Create a data dictionary—using markdown syntax—in a separate text file.
- Name this file as `nba2018-dictionary.md`
- Save this file inside the `data` folder of the `workout1` 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/2018.html
- Use bullets (or a table) to list all the variables: names, descriptions,units of measurement, and possible missing values.

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

4) Data Preparation

The first stage of the assignment has to do with the so-called *data preparation* phase. The primary goal of this stage is to create a csv data file `nba2018-teams.csv` that will contain the required variables to be used in the ranking analysis.

All the R code to complete the data preparation stage must be written in an `.R` script file (do NOT confuse with an `Rmd` file). Name the `R` script file as `make-teams-table.R` and save it inside the `code/` folder. Include a header (but NOT a yaml header) in the file containing:

- title: short title
- description: a short description of what the script is about
- input(s): what are the inputs required by the script?
- output(s): what are the outputs created when running the script?

A bit of preprocessing

The data preparation involves preprocessing columns `salary`, `experience`, and `position`.

- `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.
- `salary` is originally measured in dollars. Transform `salary` so that you have salaries in millions: e.g. 1000000 should be converted to 1.
- `position` should be a factor with 5 levels: `'C', 'PF', 'PG', 'SF', 'SG'`. Relabel these factors using more descriptive names (see below):
  - `center` instead of `C`
  - `power_fwd` instead of `PF`
  - `point_guard` instead of `PG`
  - `small_fwd` instead of `SF`
  - `shoot_guard` instead of `SG`

Adding new variables

Use `"dplyr"` function `mutate()` to add the following variables to the imported data frame:

- `missed_fg` = missed field goals
- `missed_ft` = missed free throws
- `rebounds` = offensive rebounds + defensive rebounds
- `efficiency` = efficiency index

Recall that `efficiency` is given by:

```
efficiency = (points + rebounds + assists + steals + blocks
              - missed_fg - missed_ft - turnovers) / games_played
```

Once you've computed the `efficiency` index, use `sink()` to send the R output of `summary()` on `efficiency` to a text file named `efficiency-summary.txt` inside the `output/` folder. Use a relative path when exporting the R output (assume `code/` is your working directory).

Creating `nba2018-teams.csv`

With your updated data frame you will do some data aggregation—or grouped by operations—to create a data frame `teams`, computing total values, for each team, of the following required variables:

- `team`: 3-letter team abbreviation
- `experience`: sum of years of experience (up to 2 decimal digits)
- `salary`: total salary (in millions, up to 2 decimal digits)
- `points3`: total 3-Point Field Goals
- `points2`: total 2-Point Field Goals
- `points1`: total free throws
- `points`: total Points
- `off_rebounds`: total Offensive Rebounds
- `def_rebounds`: total Defensive Rebounds
- `assists`: total Assists
- `steals`: total Steals
- `blocks`: total Blocks
- `turnovers`: total Turnovers
- `fouls`: total fouls
- `efficiency`: total efficiency

Use `sink()` to send the R output of the `teams` summary to a text file named `teams-summary.txt` inside the `data/` folder. Use a relative path when exporting the R output (assume `code/` is your working directory).

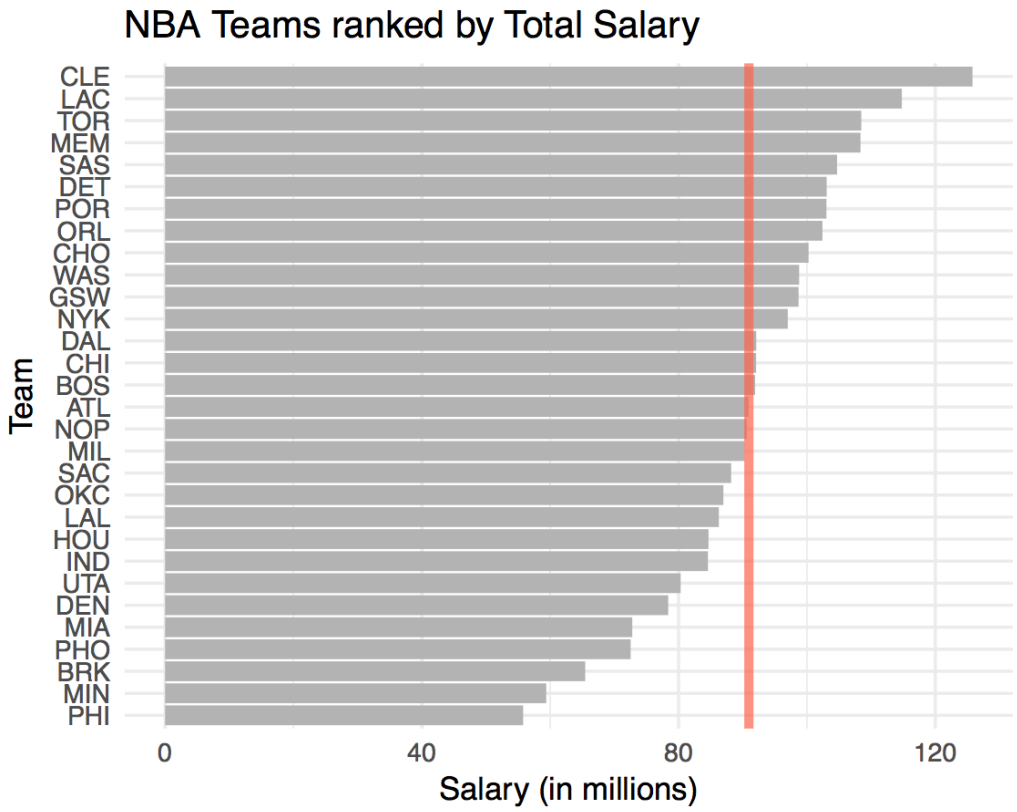
In addition to sinking the above summary, export the `teams` table to a csv file named `nba2018-teams.csv`, inside the `data/` folder. You can use the R base function `write.csv()`, or if you prefer, you can use the `"readr"` function `write_csv()`. Like with all exporting operations, you should specify the file destination using a relative path (assume `code/` is your working directory).

5) Ranking of Teams

The analysis stage of this assignment has to do with looking at various ways to rank the teams. Use an `Rmd` file for this part of your assignment.

Basic Rankings

Start by ranking the teams according to salary, arranged in decreasing order. Use `ggplot()` to create a barchart (horizontally oriented), like the one shown below. By the way, the figure below is based on data from 2016 (the graph you obtain will likely be different). The vertical red line is the average total salary.



You will have to look at the following resources to learn how to obtain such type of ggplot.

- Horizontal barplot in ggplot

https://stackoverflow.com/questions/10941225/horizontal-barplot-in-ggplot2

- axis labels in ggplot2

http://ggplot2.tidyverse.org/reference/labs.html

Create another bar chart of teams ranked by total points. Include a vertical line to indicate the average team points.

Use `efficiency` to obtain a third kind of ranking, and create an associated bar chart of teams ranked by total efficiency. Include a vertical line to indicate the average team efficiency.

Create a fourth bar chart but this time using your own index. In other words, if you had to come up with your own index (e.g. your own efficiency index or something like that), how

would you calculate it? Explain your rationale behind your own index. And then use it to graph the barchart.

Comments and Reflections

In your `Rmd` report include a section to reflect on what was hard/easy, problems you solved, helpful tutorials you read, etc.

- Was this your first time working on a project with such file structure? If yes, how do you feel about it?
- Was this your first time using relative paths? If yes, can you tell why they are important for reproducibility purposes?
- Was this your first time using an `R` script? If yes, what do you think about just writing code (without markdown syntax)?
- 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 anyone help you completing the assignment? If so, who?
- How much time did it take to complete this HW?
- What was the most time consuming part?
- Was there anything interesting?

Don't forget to submit the link of your github classroom repository to bCourses.