HW 03 - Ranking NBA Teams

Stat 133, Fall 2017, Prof. Sanchez

Due date: Sun Oct-15 (before midnight)

From the logistical point of view, the purpose of this assignment is twofold. On one hand, we want you to keep working with data frames and producing plots but now using the packages `"dplyr"` and `"ggplot2"`. On the other hand, we want you to start working with a little bit more complex file structure.

From the analytical point of view, we will focus on ranking tasks. This will give us an excuse to introduce Principal Components Analysis (PCA), from a narrow yet useful perspective. One of the deliverables is to calculate a composite index to rank NBA teams.

General Instructions

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

```
hw03/
  README.md
  data/
    nba2017-roster.csv
    nba2017-roster-dictionary.md
    nba2017-stats.csv
    nba2017-stats-dictionary.md
    nba2017-teams.csv
  code/
    make-teams-data.R
  output/
    efficiency-summary.txt
    teams-summary.txt
  images/

  report/
    hw03-first-last.Rmd
    hw03-first-last.md
    hw03-first-last_files/
      ...  # image files generated by knitr
```

- Create a folder (i.e. subdirectory) `hw03` in your `stat133-hws-fall17` local repository. This is where you will save all the associated files for this assignment.
- Create a `README.md` file with similar contents to the `README.md` file of the first assignment in `hw01`.

- 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 `images` which will contain some secondary plot images.
- Create a folder `report` which will contain the files for your dynamic document (e.g. `Rmd` and derived files).
- In the yaml header of the `Rmd` file, set the `output` field as `output: github_document` (Do NOT use the default `"output: html_document"`).
- Name this file as `hw03-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `hw03-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 not only in the knitted `hw03-first-last.md` file, but also the entire structure of the project.
- 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 in Piazza.

About the Research Question

In this assignment you will focus on a hypothetical question: **If you had to come up with a ranking system for the teams, how would you rank them?**

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

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`), to rankings based on a composite index using Principal Component Analysis (PCA).

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 `nba2017-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?

Raw data and dictionaries

The *raw* data for this assignment consists of two data files (available in the course github repository):

- `nba2017-roster.csv`
- `nba2017-stats.csv`

Include these files in the `data/` folder of your `hw03`, and create data dictionary files for them: `nba2017-roster-dictionary.md` and `nba2017-roster-dictionary.md`

Adding new variables

In your R script, write code to read these data tables in R. You can use `read.csv()` or `read_csv()`, but make sure you specify a **relative path**. After importing the tables, use `"dplyr"` function `mutate()` to add the following variables to the data frame associated with `nba2017-stats.csv`:

- `missed_fg` = missed field goals
- `missed_ft` = missed free throws
- `points` = total points
- `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.

Merging Tables

The next step is to merge the *roster* and *stats* data frames (i.e. join them) to form a larger table, from which you will derive an aggregated table with team statistics. The merging can be performed either with R base `merge()` or with the `join()` function from `"dplyr"`.

Creating `nba2017-teams.csv`

With your merged data table 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
- `free_throws`: 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

The `summary()` of your `teams` data frame should look like this:

```
     team             experience        salary          points3     
 Length:30          Min.   : 34.00   Min.   : 55.78   Min.   : 513.0  
 Class :character   1st Qu.: 56.00   1st Qu.: 84.59   1st Qu.: 617.0  
 Mode  :character   Median : 63.00   Median : 91.41   Median : 704.0  
                    Mean   : 68.73   Mean   : 90.95   Mean   : 730.7  
                    3rd Qu.: 73.25   3rd Qu.:101.87   3rd Qu.: 805.8  
                    Max.   :128.00   Max.   :125.79   Max.   :1141.0  
    points2      free_throws       points       off_rebounds  
 Min.   :1769   Min.   : 998   Min.   :6348   Min.   :524.0  
 1st Qu.:2115   1st Qu.:1238   1st Qu.:7561   1st Qu.:699.2  
 Median :2252   Median :1384   Median :8164   Median :762.5  
 Mean   :2242   Mean   :1359   Mean   :8035   Mean   :768.7  
 3rd Qu.:2413   3rd Qu.:1492   3rd Qu.:8452   3rd Qu.:865.8  
 Max.   :2638   Max.   :1605   Max.   :9473   Max.   :961.0  
  def_rebounds      assists          steals           blocks     
 Min.   :1878   Min.   :1291   Min.   :475.0   Min.   :234.0  
 1st Qu.:2435   1st Qu.:1546   1st Qu.:544.8   1st Qu.:311.0  
 Median :2536   Median :1738   Median :590.0   Median :351.5  
 Mean   :2524   Mean   :1732   Mean   :583.3   Mean   :360.3  
 3rd Qu.:2644   3rd Qu.:1858   3rd Qu.:612.0   3rd Qu.:389.5  
 Max.   :2854   Max.   :2475   Max.   :779.0   Max.   :551.0  
   turnovers           fouls         efficiency   
 Min.   : 703.0   Min.   :1164   Min.   :125.1  
 1st Qu.: 973.5   1st Qu.:1355   1st Qu.:143.8  
 Median :1021.5   Median :1519   Median :146.7  
 Mean   :1013.5   Mean   :1496   Mean   :149.0  
 3rd Qu.:1087.2   3rd Qu.:1599   3rd Qu.:152.9  
 Max.   :1184.0   Max.   :1886   Max.   :177.9  
```

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.

In addition to sinking the above summary, export the `teams` table to a csv file named `nba2017-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.

Some graphics

The last data preparation tasks to be completed within the `.R` script, consist of making some exploratory plots, and saving the produced graphics as image files—in `.pdf` format—inside the `images/` folder. Again, use a relative path when exporting the images:

- use `stars()` to get a *star plot* of the teams. Save the plot in the file `teams_star_plot.pdf` (insise the `images/` folder).

  ```
  stars(teams[ ,-1], labels = teams$team)
  ```

- use `ggplot()` to get a scatterplot of `experience` and `salary`, in which the names of the teams are included. Save the plot in the file `experience_salary.pdf` (insise the `images/` folder).

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 project.

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. The vertical red line is the average team salary.

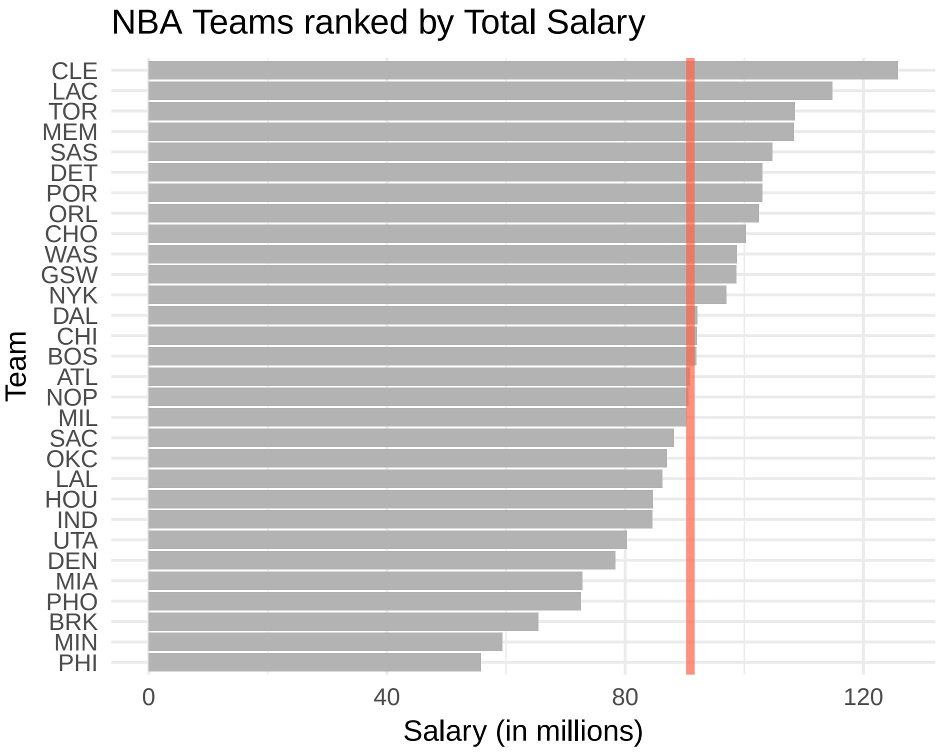
You will have to look at the following resources to learn how to obtain such type of ggplot.

- Horizontal barplot in ggplot

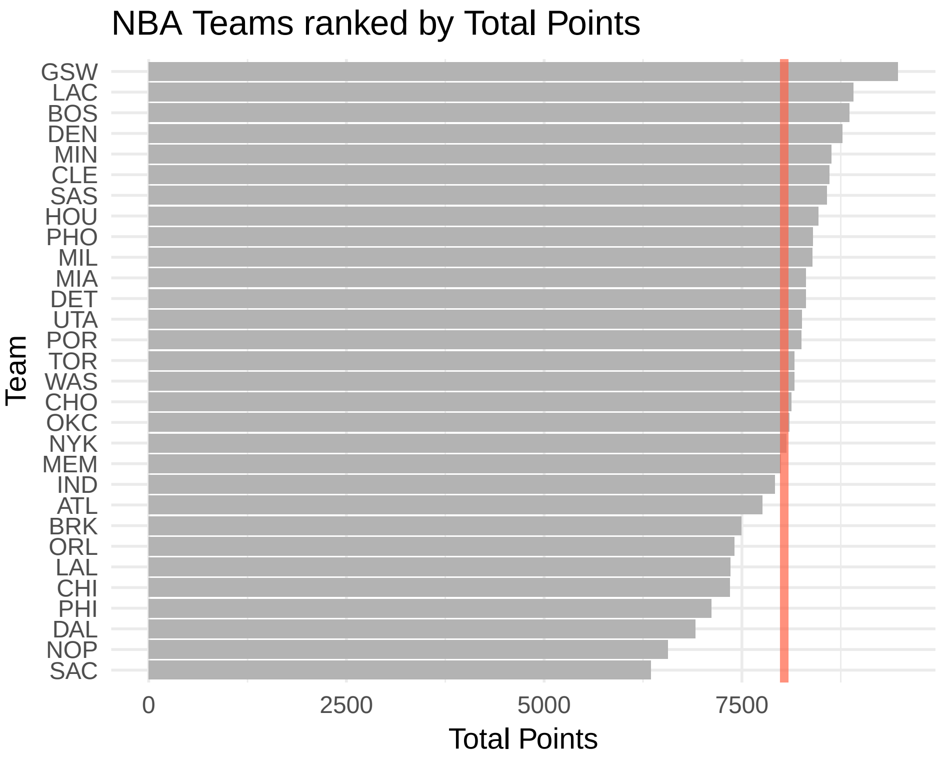
https://stackoverflow.com/questions/10941225/horizontal-barplot-in-ggplot2

- axis labels in ggplot2

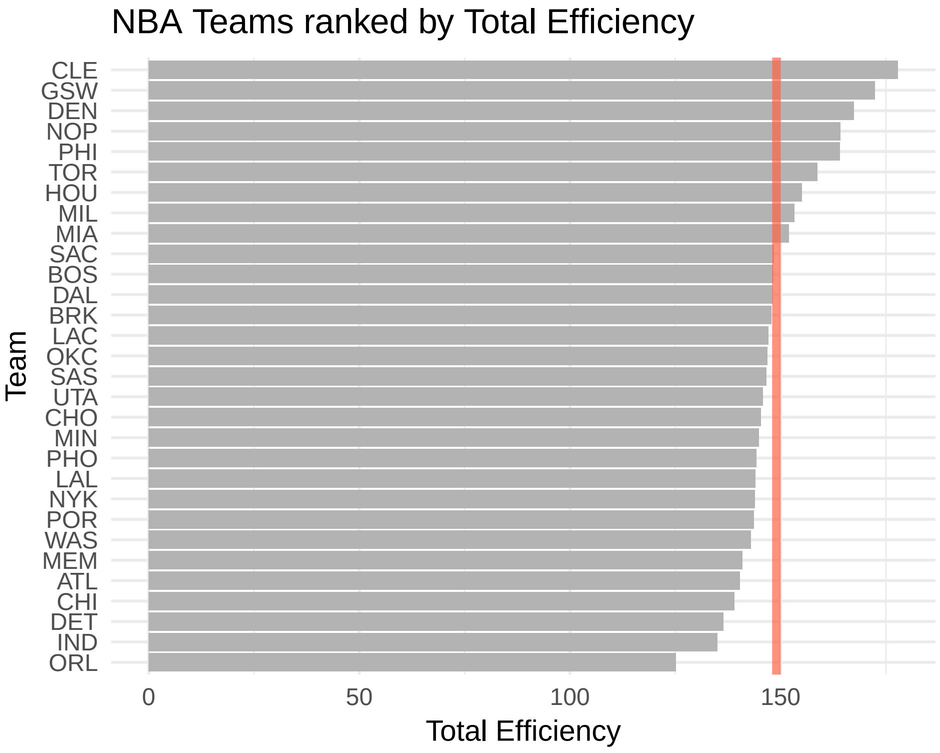
http://ggplot2.tidyverse.org/reference/labs.html



Create another bar chart of teams ranked by total points. The vertical red line is 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. The vertical red line is the average team efficiency.



Provide concise descriptions of the obtained rankings so far.

Principal Components Analysis (PCA)

Perform a principal components analysis (PCA) on the following variables, to use the first principal component (PC1) as another index to rank the teams:

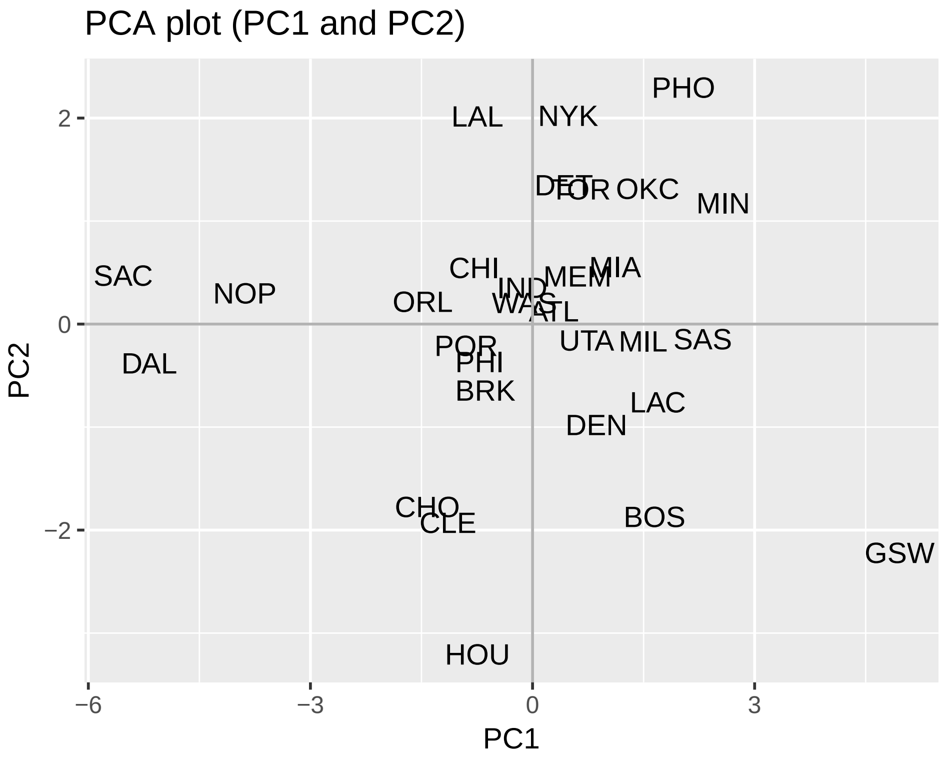
- `points3`
- `points2`
- `free_throws`
- `off_rebounds`
- `def_rebounds`
- `assists`
- `steals`
- `blocks`
- `turnovers`
- `fouls`

Use `prcomp()`—NOT to confuse with `princomp()`—to perform a PCA, specifying the argument `scale. = TRUE` (i.e. PCA on standardized data).

Createa a data frame with the eigenvalues:

```
     eigenvalue     prop   cumprop
1        4.6959   0.4696    0.4696
2        1.7020   0.1702    0.6398
3        0.9795   0.0980    0.7377
4        0.7717   0.0772    0.8149
5        0.5341   0.0534    0.8683
6        0.4780   0.0478    0.9161
7        0.3822   0.0382    0.9543
8        0.2603   0.0260    0.9804
9        0.1336   0.0134    0.9937
10       0.0627   0.0063    1.0000
```

Use the first two PCs to get a scatterplot of the teams



To interpret the PCs you can look at the associated weights (i.e. columns of `$rotation`), or you can compute the correlations between the variables and the PCs.

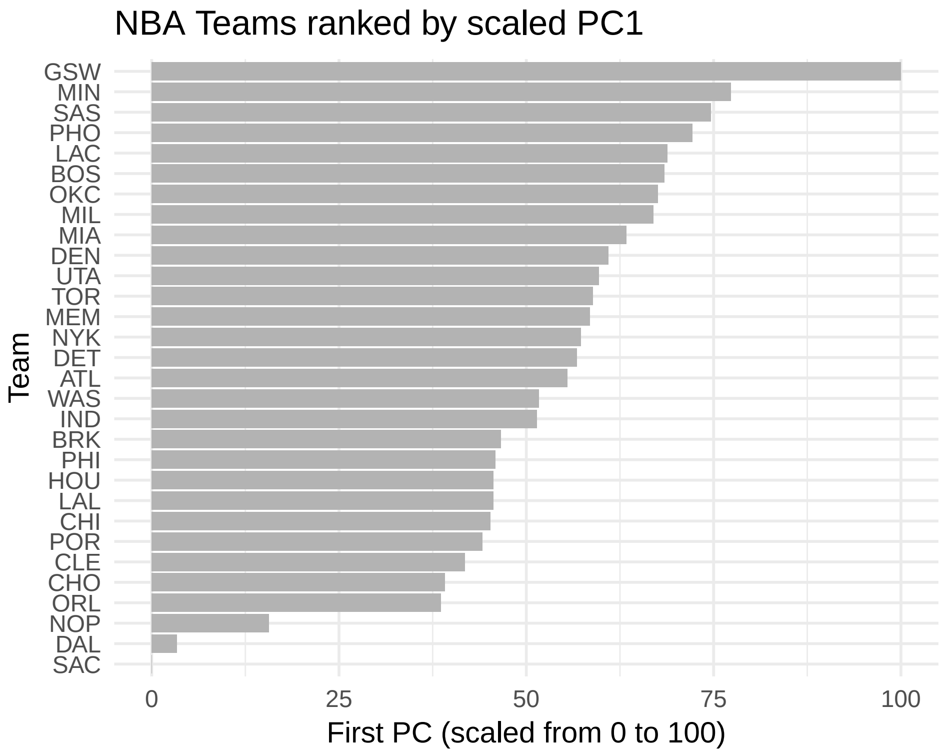
Index based on PC1

In order to build an index based on the first PC, you are going to transform PC1. To get a more meaningful scale, you can rescale the first PC with a new scale ranging from 0 to 100.

Let z_1 be the first principal component. The transformed score s_1, ranging on a scale from 0 to 100, can be obtained as:

$$s_1 = 100 \times \frac{z_1 - min(z_1)}{max(z_1) - min(z_1)}$$

Once you have obtained the rescaled PC1, you can produce a barchart like the previous ones:



Provide a brief description of the PC1 index to rank the teams.

Comments and Reflections

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?
- 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?