HW 01 - More Vectors

Stat 133, Fall 2017, Prof. Sanchez

Due date: Sat Sep-23 (before midnight)

The purpose of this assignment is to keep working with vectors of different data types, factors, and some basic plots. Also, we want you to use git and github to version control and host your assignments. Use this assignment to keep developing your manipulation skills of basic data objects in R: use of bracket notation, understanding vectorization, coercion rules, and become more familiar with the associated NBA data set.

General Instructions

- 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 `hw01-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `hw01-gaston-sanchez.Rmd`).
- Save the `Rmd` file in the folder `hw01/` of your (local) repository `stat133-hws-fall17`.
- Please do not use code chunk options such as: `echo = FALSE`, `eval = FALSE`, `results = 'hide'`. All chunks must be visible and evaluated.
- If necessary, modify the contents of the `README.md` file located in `hw01/`.
- 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 grade the work in the `hw01-first-last.md` file of your github repo.
- We WON'T grade any work under html format (because it does not get rendered nicely in github; no exceptions).
- We WON'T grade any files submitted to bCourses (no exceptions).
- If you have questions/problems, don't hesitate to ask us for help in OH or piazza.
- This assignment is not anymore a warm-up assignment; i.e. this HW does count towards your final grade.

Data

The data objects for this assignment are in the file `nba2017-salary-points.RData`, inside the `data/` folder of the `hw01/` directory of your `stat133-hws-fall17` repo. There is also the

data dictionary file `nba2017-salary-points-dictionary.md`.

Importing the data

You don't need to download any data file (since it's already in your local repository). To import the data, you just need to use `load()`. Assuming that your `Rmd` file is already saved inside the `hw01` folder, you can *load* the `.RData` file with the following command (put it inside a code chunk):

```
# load data file
load("data/nba2017-salary-points.RData")
```

Call `ls()` to *list* the objects contained in the `.RData`:

```
# list the available objects
ls()
```

Research Question

In this assignment, you are going to start answering the research question: **"the more points a player scores, the higher his salary?"**

Tackle each of the sections described below

1) A bit of data preprocessing

- The variable `salary` is measured in dollars. In order to have a more convenient measurement scale, create a new salary variable measured in millions of dollars, up to 2 decimal digits.
- The variable `experience` is a character vector. If you take a look at its elements, you will see that there are numbers as well as the character `"R"`. The character `"R"` means that the player is a rookie (i.e. first season playing in the NBA). Replace the values `"R"` by `"0"`, and create a new experience variable as an **integer** vector.
- The variable `position` is a character vector. Create a new position variable as an R factor (see `?factor`). Provide more descriptive labels for the factor's levels (i.e. the categories), for instance:
  - `'center'` instead of `"C"`
  - `'small_fwd'` instead of `"SF"`
  - `'power_fwd'` instead of `"PF"`
  - `'shoot_guard'` instead of `"SG"`

- `'point_guard'` instead of `"PG"`

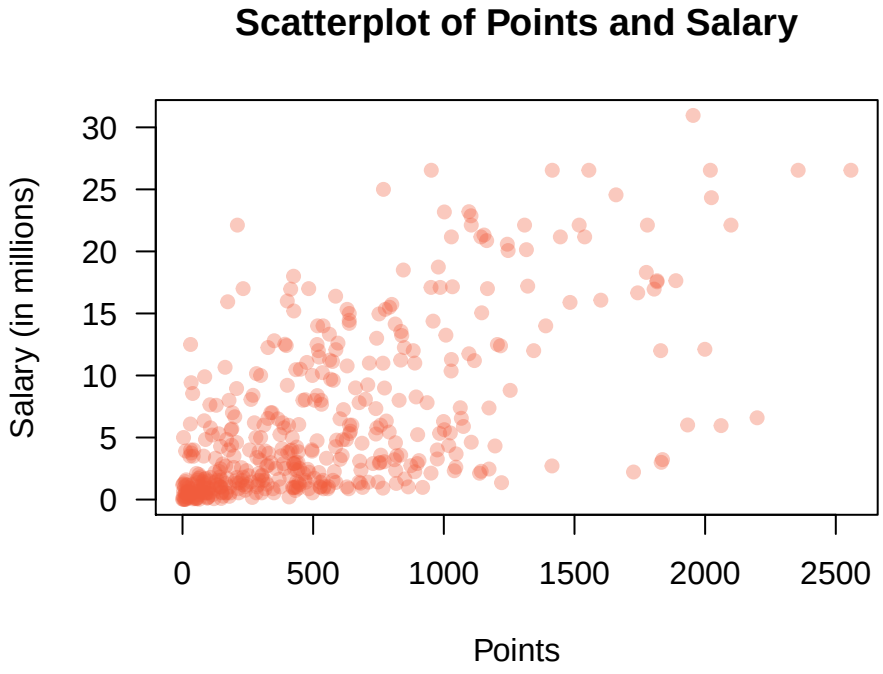
Once you have the factor for positions, compute the frequencies (i.e. number of occurrences) with the function `table()`.

2) Scatterplot of Points and Salary

Begin your analysis of the relationship between Points and Salary with a scatterplot using the function `plot()`. Play with the following graphical parameters of `plot()` (see `?par` for more information about these parameters):

- `pch`
- `col`
- `cex`
- `xlab`
- `ylab`
- `main`

Don't worry too much about the visual appearance of your plot. Later in the course we will spend some time talking about data visualization in a more formal way. Focus instead on providing a concise description of the patterns observed in the scatterplot.



3) Correlation between Points and Salary

Another way to study the relationship between Points and Salary consists of calculating their correlation (i.e. linear correlation coefficient). R has the function `cor()` that computes this

type of correlation. However, we want you to practice writing commands. creating objects, and working with vectors. So instead of using `cor()`—and other related functions—you will have to "manually" compute the correlation as well as other summary statistics. To achieve this task, you have to create R objects (and display their values) for:

- n: number of individuals
- $\bar{x}$: mean of variable X (`points`)
- $\bar{y}$: mean of variable Y (`salary`)
- $var(X)$: variance of X
- $var(Y)$: variance of Y
- $sd(X)$: standard deviation of X
- $sd(Y)$: standard deviation of Y
- $cov(X, Y)$: covariance between X and Y
- $cor(X, Y)$: correlation between X and Y

Note: you are NOT allowed to use functions such as `mean()`, `var()`, `cov()`, `cor()`, or `lm()`. Nor you can use any type of loop (e.g. `for`, `while`, `repeat`). The only auxiliary function that you are allowed to use is `sum()` to implement operations that involve summation $\sum_{i=1}^{n}$. The most important concept here is knowing that most operations with vectors in R are **vectorized**.

The **mean** of a variable X, denoted by $\bar{x}$, is given by:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

where:

- n is the number of individuals (i.e. number of elements in X)
- x_i is the i-th element of X
- $\sum_{i=1}^{n}$ is the summation symbol

Similarly, the **mean** of Y, denoted by $\bar{y}$, is given by:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Use the function `sum()` to implement operations that involve summation $\sum_{i=1}^{n}$ (recall vectorized operations in R).

The (sample) **variance** of a variable is given by:

$$var(X) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

From the variance, you can derive the (sample) **standard deviation** as:

$$sd(X) = \sqrt{var(X)}$$

In turn, the (sample) **covariance** between two variables X and Y is given by:

$$cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

Finally, the **correlation** between X and Y is given by:

$$cor(X, Y) = \frac{cov(X, Y)}{sd(X)sd(Y)}$$

4) Simple Linear Regression

In addition to looking at the correlation between Points and Salary, you can also compute a linear regression equation that could be used, among other things, to predict the Salary of a player in terms of the scored Points.

The linear regression equation between a response variable Y and a predictor variable X is given by:

$$\hat{Y} = b_0 + b_1 X$$

where:

- $\hat{Y}$ are the fitted or predicted values of Y
- b_0 is the estimated intercept of the regression line
- b_1 is the estimated slope of the regression line

The **slope** b_1 can be obtained as:

$$b_1 = cor(X, Y) \times \frac{sd(Y)}{sd(X)}$$

and the **intercept** b_0 can be computed as:

$$b_0 = \bar{y} - b_1 \bar{x}$$

Create objects for:

- b_1 the slope term
- b_0 the intercept term
- $\hat{Y}$ the vector of predicted values ("y-hat")

Provide answers to the following questions:

- Summary statistics (use `summary()`) of $\hat{Y}$.
- What is the regression equation? Use *inline code* to write the equation.
- How do you interpret the slope coefficient b_1?
- How do you interpret the intercept term b_0?
- What is the predicted salary for a player that scores:
  - 0 points?
  - 100 points?
  - 500 points?
  - 1000 points?
  - 2000 points?

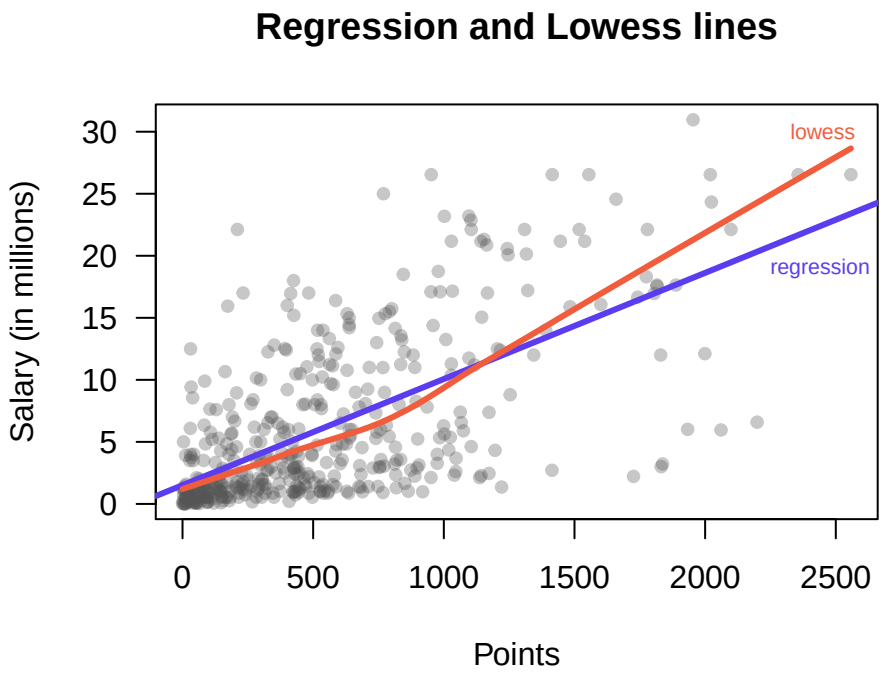
5) Plotting the regression line

Make a second scatterplot of Points and Salary using `plot()`, but this time include the regression line. You can include the regression line by calling the function `abline()`. This function allows you to add lines to a plot by specifying values for the arguments `a` (the intercept) and `b` (the slope). Read more about the details of this function with `?abline`, and find out how to use its arguments:

- `lwd`
- `col`

In addition to including the regression line, you will also have to include a *lowess* smooth line (locally weighted scatterplot smoothing) in the plot. Use the `lines()` and `lowess()` functions to display such a line. For more information on how to do this, see the examples in `?lowess`.

In order to have a better visual display of the scatterplot with the regression and the lowess lines, use the function `text()` to add labels next to each line. Here's a sample plot (you don't need to get the exact same format).



6) Regression residuals and Coefficient of Determination R^2

Having obtained the predicted values $\hat{y}_i$, we can compare them against the observed values y_i. In a regression analysis, the starting point for this comparison is based on the so-called **residuals**. The residuals, denoted by e_i, are obtained by calculating the difference between observed and predicted values:

$$e_i = y_i - \hat{y}_i$$

The better the fit of the regression equation, the smaller the residuals e_i should be.

There are several measures based on the residuals. The most common one is known as the **Residual Sum of Squares** (RSS), given by the following equation:

$$\begin{aligned} RSS &= \sum_{i=1}^{n} e_i^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \end{aligned}$$

One simple way to assess the quality of a fitted regression line involves calculating the coefficient of determination R^2 given by:

$$
\begin{aligned}
R^2 &= 1 - \frac{RSS}{TSS} \\
&= 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\end{aligned}
$$

where:

- RSS is the Residual Sum of Squares
- TSS is the Total Sum of Squares (proportional to the variance of Y)

The R^2 coefficient is the proportion of the variance in Y that is predicted from the X. In other words, R^2 provides a measure of how well observed values y_i are replicated by the fitted valed $\hat{y}_i$, based on the proportion of total variation of Y explained by the regression equation.

Create R objects (and display their values) for:

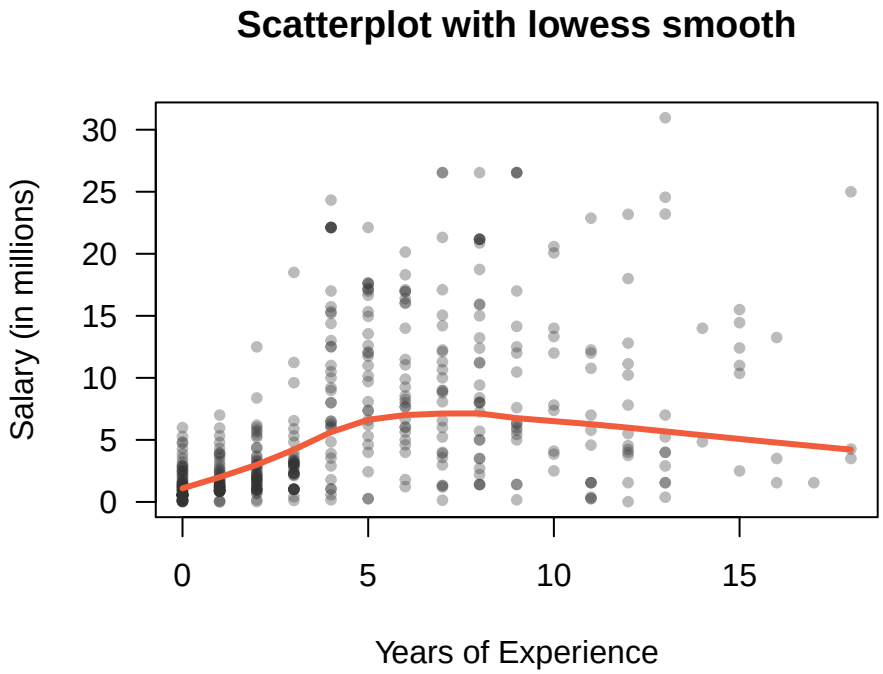
- the vector of residuals (display only its `summary()` statistics)
- the Residual Sum of Squares
- the Total Sum of Squares
- the coefficient of determination R^2

7) Exploring Position and Experience

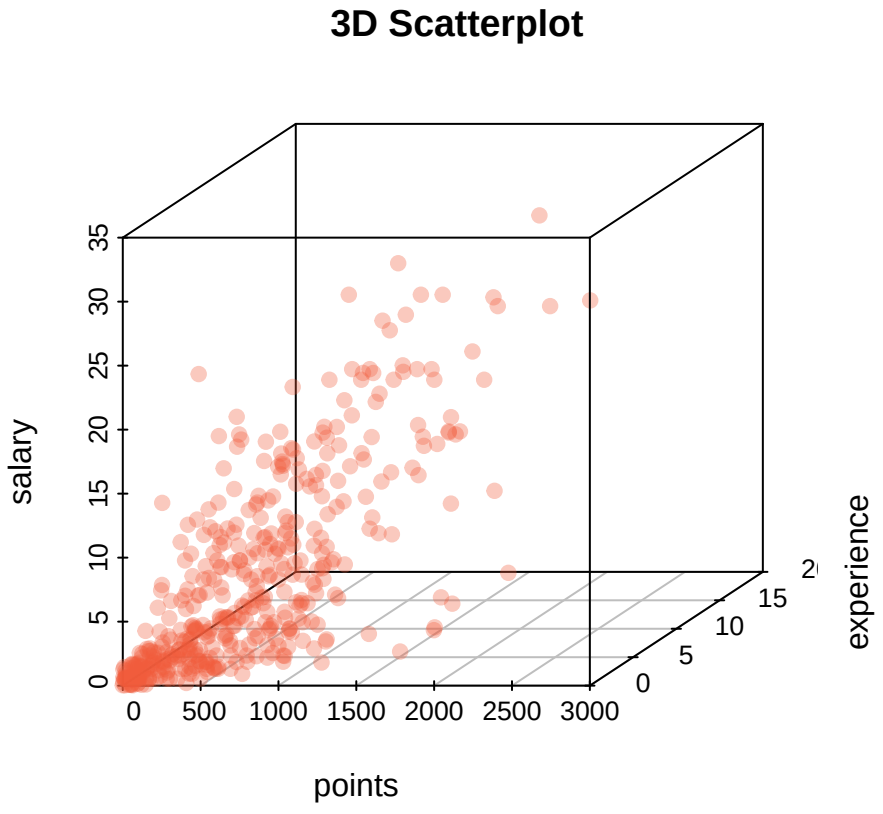
So far you've looked at the relationship between Points and Salary from a narrow point of view: assuming a linear relationship, and without taking into account other variables. However, it is reasonable to assume that the Salary of a player depends on other factors such as his age, or the years of experience in the NBA. Likewise, the number of scored Points comprises free-trows, 2-points, and 3-points. And it is very likely that the number of scored points will be affected by the player's position (among other factors).

In order to expand the scope of the analysis, you will have to consider the relation of Points and Salary according to: 1) years of Experience in the NBA, and 2) Position. To do this, create:

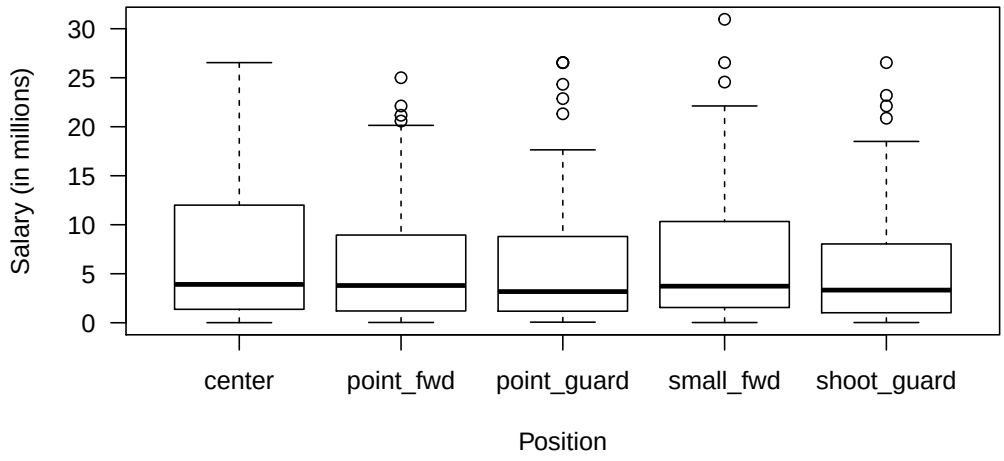
- A scatterplot of Years-of-Experience and Salary, including a *lowess* smooth line (locally weighted scatterplot smoothing).



- Use the function `scatterplot3d()` from the homonym package `"scatterplot3d"` to create a 3D-scatterplot of Points, Experience, and Salary. Use Points for the x-axis, Experience for y-axis, and Salary for the z-axis.



- A conditional boxplot of Salary in terms of Position (use `boxplot()`)



- Provide concise descriptions for the plots of this section.
- From the scatterplots, does Experience seem to be related with Salary?
- From the boxplot, does Position seem to be related with Salary?

You will keep exploring and analyzing the NBA data in subsequent HWs.

8) 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?
- What was easy(-ish) even though we haven't done it in class?
- If this was the first time you were using git, how do you feel about it?
- If this was the first time using GitHub, how do you feel about it?
- Did you need help to complete the assignment? If so, what kind of help? Who helped you?
- 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).