Lab 8: Scatter Plots and Correlation

The following worksheet is due by 8pm one day after this lab. You can find the submission dropbox in Brightspace by clicking on Content $->$ Lab Content.

0. Open a new R Markdown file.

   Note: Your worksheet is to be submitted as the output of an R Markdown file (you can knit it to HTML and then convert it to PDF, or you can knit it to PDF if you have LaTeX on your computer, or you can knit it to Word and then convert that to a PDF).

0.1 Download the data set `nba_player_data_2020.csv` and save it to whatever directory you are using for this course.

In the sport of basketball a player earns points for their team by putting the ball into the opposing team's basket. A player earns $1, 2$ or 3 points depending on where the shot is taken from and whether it is during the course of regular play.

When a player is shooting the ball at the basket and an opposing player bumps into him the referee will call a foul and stop the game. The fouled player then gets 2 shots from the "free-throw line" without anyone guarding them. Each of these shots is called a free-throw and earns 1 point each time they score a free-throw.

During regular play a player earns either 2 or 3 points for a made basket. The basketball court has an arc drawn on it called the "3-point line." A shot made from behind the 3-point line is worth 3 points and any shot made from inside the 3-point line is worth 2 points.

Over the past decade teams have focused more and more on 3-point shooting and have tried to figure out how best to determine if a player being drafted from college or another league (where the 3-point line is closer to the basket than in the NBA) could develop into an efficient 3pt shooter. It is also very advantageous to try to determine if players already in the league (who may not take many 3 point shots currently) will have success if they start taking "more threes.".

Today we are going to use data from the $2019-2020$ NBA season to determine which shooting categories are the most positively correlated with a player having a high 3-point percentage (makes / attempts). The statistics we're going to compare are

- Number of made 2-pointers
- Number of attempted 2-pointers
- 2 point shot percentage
- Number of free-throws made
- Number of free-throws taken
- Free-Throw shot percentage.

Take a moment to think about which statistic you think will have the strongest positive linear relationship with 3 point shot percentage?

1. (a) Load the `nba_player_data_2020.csv` dataset into R and save it to `df`.

   (b) Copy and paste each of the following three lines into your code to eliminate rows with zeros or NA's.

   - `df = na.omit(df)`
   - `row_sub = apply(df, 1, function(row) all(row !=0 ))`
   - `df = df[row_sub,]`

   (c) We are only concerned with the categories listed above so use the following code to save only the relevant columns to a new dataframe called `dfc`:

   ```
   dfc = df[, c(13:16, 18:20)]
   ```

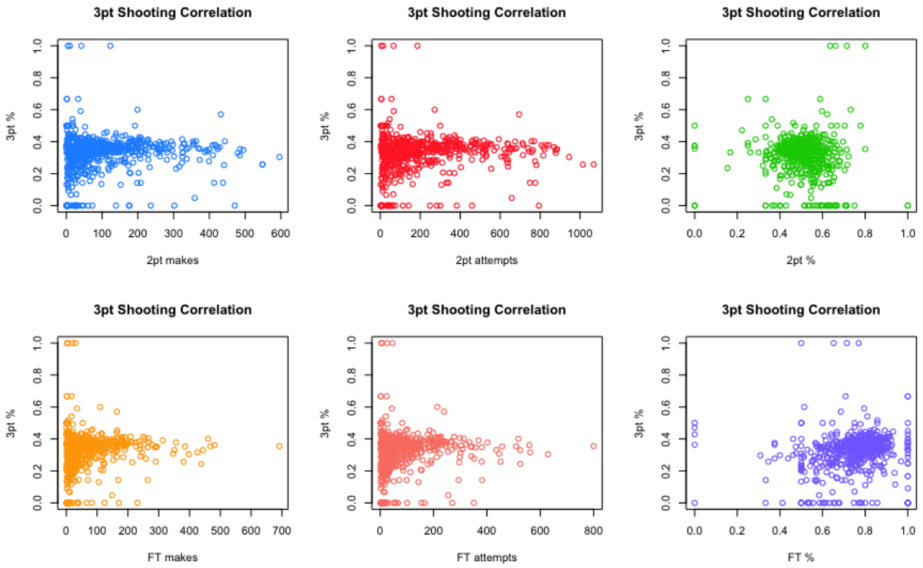
2. (a) The first way we'll compare correlations is by using scatter plots. We want to plot six scatter plots. Each will have 3-point percentage (column 1) as the y-axis and the x-axis will be one of each of the other columns in `dfc`. Recall that the basic function for scatter plots is `plot(x, y)`.

   (b) We will be using a for-loop to populate the plots. Each plot will have the same title: "3pt Shooting Correlation.", and same y-axis label, "3pt %". Each plot should be a different color and each x-axis should be properly labeled with the appropriate category. You may use the following code to set up vectors for different colours and labels if you wish:

   - `colours = c("dodgerblue", "firebrick1", "green3", "orange", "salmon", "slateblue1")` If you don't like these colours, you may check out R Colours to see many other options.
   - `cnames = c("2pt makes", "2pt attempts", "2pt %", "FT makes", "FT attempts", "FT %")`

   (c) Write a for-loop that produces the six necessary scatter plots. Above the for-loop code, use the following to set up a grid so that your plots are presented nicely: `par(mfrow = c(2, 3))`

   (d) Your finished plots should look something like the following:



3. Next, we will use a for-loop to calculate the correlations between 3pt % and each of the other columns.

   (a) Create an empty numeric vector to store the correlations and name it `cor_vec`.

   (b) Write a for-loop to populate `cor_vec` with the corresponding correlations.

   (c) Run the following code to properly name the elements in `cor_vec`:
   `names(cor_vec) = c("3pt %", cnames)`

   (d) Print out `cor_vec`.

4. (a) Now that we've calculated these correlations the hard way, use the following code to see if your answer to question 3 is correct:
   - `easy_way = cor(dfc)[,1]`
   - `names(easy_way) = c("3pt %", cnames)`

   (b) Print out `easy_way`

   (c) Which category is most positively correlated with 3pt %?

Congratulations! You are done. Enjoy the rest of your day.