MSBA 615: R for Analytics

Final Project: Custom R Function for EPL Data

**Context and Purpose:** The English Premier League (EPL) is a distinguished soccer league in Great Britain, consisting of 20 teams. The season begins each year in August and concludes in May with each team playing each other team exactly twice (home and away). Each team plays 38 matches in a season while the total number of matches among all teams in a season is 380. A team receives 3 points for a win and if the game is tied, both teams receive 1 point; no points are awarded for a loss. At any point in the season, the team with the most points is the first-place team.

Your task is to develop an analytical understanding of the EPL's game results by creating a comprehensive function in R that retrieves and processes data about EPL match results from the web. Throughout the coding assignment you will implement a custom function that involves various R programming concepts you have learned throughout the course, including data cleaning, data transformation, data aggregation, control structures, and loops. This assignment will also require you to practice breaking a large coding problem down into manageable steps and developing a strategy for completing the various components required to get the specified output.

**Assignment Overview**

1. **Objective:** Create an R function called **EPL\_Standings** that takes two parameters - *date* and *season* - and returns the league standings for the specific date and season that a user specifies. Your function must be capable of handling data variations and inconsistencies.
2. **Data Source:** Access the EPL game results from [this link](https://www.football-data.co.uk/englandm.php) under the heading Premier League. A full walkthrough of how to access is provided in the week 3 lecture content.
3. **General Function Requirements:**
   1. **Function Name:** Your function must be named **EPL\_Standings**.
   2. **Parameters:**

Your function must accept two parameters named exactly as shown and the parameters should expect the input formats shown below.

* + - 1. **date:** mm/dd/yyyy, such as “05/12/2023” given as a string input that specifies the exact date the function should use as a cutoff for the data it summarizes. (It takes the date given as a cutoff and considers everything from the give date downwards).
      2. **season:** yyyy/yy, such as “2022/23” given as a string input that specifies the season from which the function should extract data to summarize. For example, “2022/23” would refer to the 2022-2023 season. You are only required to accept the most recent 3 seasons as inputs.

1. **Example Function Usage:**

If I were to load your script and enter the following command in my R console:

EPL\_Standings(“04/25/2022”, “2021/22”)

…your function should read the correct data file for the 2021/22 season from the web, filter the data so that only data up to the conclusion of all matches played on April 25, 2022 is used, and then perform all required data transformations (specified below) and return a data frame of results.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Overview of the Data**

* The csv data files available via the weblink provided above contain a lot of information that is not needed for this assignment. The only critical information that you need to complete the assignment is:
  + Date (in day/month/year format)
  + Home Team
  + Away Team
  + FTHG – the number of goals scored by the home team
  + FTAG – the number of goals scored by the away team
  + FTR – the result of the match (*H* indicates home team won, *A* indicates the away team won, *D* indicates the match ended in a draw or tie)
* The data for previous seasons is static (unchanging) but the data for the current season changes whenever a game is played. Your function should read the data directly from the web pages and not from a csv file stored on your hard drive. The *read.csv*() command will do this easily where the url is used instead of the filename.
* Notice that the year in the date column sometimes appears as a 2-digit value but sometimes it appears as 4-digit value. Your function will need to address this issue. – and standardize it, I believe to 2-digit values.
* Check that the data column names are consistent from season to season; if they are not, you need to address this issue in your function.
* You are only required to build your function to handle three seasons as inputs: your function should be able to handle 1) the current season and, 2 & 3) the two most recently completed seasons.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Specifications**

Your function must generate the following calculated or grouped fields listed below. Each of the specified fields below are required to be calculated correctly and returned from your function in a single data frame to receive full credit for the coding portion of the assignment.

Along with the team name (named *TeamName*), the data frame returned by the function will contain the following columns where the column name is specified in parentheses:

* record as *wins-loses-ties* *(Record*)
* home record (*HomeRec*)
* away record (*AwayRec*)
* matches played (*MatchesPlayed*)
* points (*Points*),
* points per match (*PPM*),
* point percentage = points / 3 \* the number of games played, (*PtPct*)
* goals scored (*GS*),
* goals scored per match (*GSM*)
* goals allowed (*GA*)
* goals allowed per match (*GAM*)

After calculating the specified columns from the input data, your function should ***display the standings in descending order according to the number of points per match earned up to and including the date parameter value***.

* *Note on tiebreakers when sorting the table standings:* When two teams have the same number of points per match, the teams should appear in descending order according to the number of points per match, then wins, then goals scored per match and finally ascending order according to goals allowed per match.

After calculating the fields and sorting the results according to the above criteria, your function should **return a single data frame**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Additional Challenges (100% Optional) \***

If you are feeling more comfortable with the above requirements and the content in the course in general and wish to push yourself to extend your understanding, it is recommended you attempt to add the following components to your output for an additional challenge:

* A column containing the team’s record over the last 10 games played (*Last10*)
* A column containing the team’s current match result streak where *W6* would mean the team has won their last six games but they have not won their last 7 games (*Streak*)
  + *Streak Example: L2* indicates a team has lost their last two games but they have not lost their last 3 games, while *T1* denotes the team has tied their most recent game but did not tie their last two games.

\*These components are NOT REQUIRED and will not be graded as extra credit. I have offered them as extra credit in the past but have generally found that students who attempt these are more advanced and typically do not benefit from extra credit.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Analytical Components**

* **Understanding the Dataset:** In your script, using code comments, write a brief summary explaining the dataset's structure and important variables.
* **Methodology:** In your script, using code comments, describe your approach to handle variations in date formats, column names, and other potential data inconsistencies.
* **Reflection:** In your script, using code comments, discuss any challenges you faced, the decisions you made, and how your function contributes to a deeper understanding of the EPL standings.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Rubric for Assessment**

* **Primary Criterion:** Correctness and efficiency of your code.
* **Secondary Criterion:** Clarity, consistency, and in-depth understanding through well-documented code comments.
* **Tertiary Criterion:** Engagement with the analytical component, demonstrating critical thinking and problem-solving skills.

**Submission Requirements**

* You should submit a single R script (.R file) containing all of your code for your function, all code comments, and all responses to the analytical / reflection components listed in the assignment prompt.
* **Your script file should be named “EPL\_Assignment\_<your name>.R”** but please also list your name in a code comment at the top of the file.
* Please submit your script via Blackboard. Emailed responses may not be accepted.
* **Be sure to submit your script by the assigned deadline**.

**Evaluation**

Although this assignment may have only one possible correct output for a given input to the function, there are many correct ways to arrive at the correct response. Please add code comments describing your reasoning for the code you wrote where possible so that partial credit may be awarded in cases where the result is not completely correct.

**Additional Notes:**

* **Exclusivity:** This assignment must be programmed entirely in R. Usage of external languages or copy-pasting solutions from other sources including online help forums or homework answering services will lead to disqualification.
* **Integrity:** Your submission should reflect ***your own understanding and coding skills***. While you may certainly seek out help via classmates, Stack Overflow, ChatGPT, or other coding forums and tools, utilizing AI tools or homework services to solve the assignment or sharing complete solutions directly with others in the course is strictly prohibited and will result in disqualification.

**Tips:**

* **Start Early:** I would estimate that a typical student spends around 5-10 hours on this assignment. Begin as soon as possible to avoid last-minute struggles. After years of offering this assignment I have found that students who wait until the final week of the course to work on this typically end up with a lot more stress and are generally submit less successful functions.
* **Seek Help if Needed:** Approach the instructor or teaching assistant for guidance. Please feel free to share any general questions via the course discussion board. There are no bad/wrong questions and if you are confused about something it is likely that classmates are as well.