Part 1

This is a tutorial article that teaching people who is new to R how to do web scraping. First of all, we will need to that what is web scrapping. It is a process of extracting data or information from website, and the scrapped data can be used for some further analyzing. Generally speaking, a good website contains text, pictures, and links. These interactions are organized in a way that stresses the key point and easy to understand. However, there are information that is useless for data analyze or statistics. So, we will need to extract data to meet our needs. Web scrapping uses the code that behind the webpage, which we call it as source code. Source code is usually written in three coding languages, which are Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and PHP (Erolin, 2023).

HTML (HyperText Markup Language) is the language that we will use in this article. Compared to other programming language like R or C++, it is a markup language. “Hypertext” refers to links that connect web pages to one another. By adding links, a webpage will be more interactive. This language is a text-encoding system which states the content and structure of a webpage. HTML reveals features by using tags with <> symbols. The markup includes special elements such as <p> and <title>. <p> represents a paragraph that are usually showed as blocks of text separated by blank lines (MDN Web Docs, 2018). Also, string manipulation with stringr helps in web scrapping.

Before doing actual web scrapping, I did a read in practice. I use the read_html function to read the IMDb website using the web link as the input. To achieve this function, “rvest” package is installed in advanced. This is a step that letting R studio knows that we are dealing with the source code located in a specific link.

After this, I attempt to do some real web scrapping. https://www.premierleague.com/tables?co=1&se=578&ha=-1&team=FIRST. This URL links contains the latest result for Premier League. There are 20 teams in this league, and my goal is to scrap the names for each team. I use three packages here, which are rvest, tidyverse, and stringr. I use read-html function to firstly read in the website, and then use html-nodes function to locate the container that contains the team names. It took me a long time finding the correct container. After browsing the source code again and again, I found out that a div class called "league-table__expandable-container" is the right thing I want. I save the nodes into club_nodes, and using html_text to convert the codes to text message. The following is my code (*Get Element Text — Html_text*, n.d.), (*Select Nodes from an HTML Document — Html_nodes*, n.d.).

```
club_nodes <- html_nodes(webpage, "div.league-table__expandable-container")

club_names <- html_text(club_nodes)
```

Now the names have been saved into "club_names". But now the text contains other information that we don't need. We need to use stringr to extract names. After looking at source code, I found out that all these nodes contain 4 blank lines first, and then names come out. I use the following code to extract data, and save the names in "names"

```
names <- str_extract(club_names, "\\b[A-Z][a-z]+(?=\\n)")
```

However, the result is not what I want. I can only show one word for the team name. But there are several teams have 2 or three names, such as "Manchester United" and "West Ham United".

```
> names
 [1] "Arsenal"      "Liverpool"    "City"         "Villa"        "Hotspur"      "United"       "United"       "Albion"
 [9] "Wanderers"    "United"       "Chelsea"      "Fulham"       "Bournemouth"  "Palace"       "Brentford"    "Everton"
[17] "Town"         "Forest"       "Burnley"      "United"       "Hotspur"      "United"       "Fulham"       "Arsenal"
[25] "Wanderers"    "Liverpool"    "Chelsea"      "Albion"       "Middlesbrough" "Reading"     "Sunderland"   "United"
[33] "Palace"       "City"         "City"         "Southampton"  "Rovers"       "Villa"        "Forest"       "Albion"
[41] "Everton"      "City"         "County"       "United"       "City"         "United"       "United"       "City"
[49] "Liverpool"    "County"       "Wanderers"    "United"       "United"       "Middlesbrough" "Forest"      "Everton"
[57] "City"         "Rovers"       "Sunderland"   "United"       "Chelsea"      "Hotspur"      "Fulham"       "Arsenal"
[65] "Palace"       "City"         "Albion"       "Albion"       "Villa"        "Southampton"  "City"         "Reading"
```

I did some revisions and my final version is this.

names <- str_extract(club_names, "([A-Z][a-z]+(?: [A-Z][a-z]+))")*

The result contains total of 72 teams. The reason is that it contains expandable content such as match history and future match prediction. We only need the first 20, so we use the code to extract the first 20 names.

league_names <- names[1:20]

```
> names
 [1] "Arsenal"                 "Liverpool"               "Manchester City"         "Aston Villa"             "Tottenham Hotspur"
 [6] "Manchester United"       "West Ham United"         "Brighton"                "Wolverhampton Wanderers" "Newcastle United"
[11] "Chelsea"                 "Fulham"                  "Bournemouth"             "Crystal Palace"          "Brentford"
[16] "Everton"                 "Luton Town"              "Nottingham Forest"       "Burnley"                 "Sheffield United"
[21] "Tottenham Hotspur"       "West Ham United"         "Fulham"                  "Arsenal"                 "Wolverhampton Wanderers"
[26] "Liverpool"               "Chelsea"                 "Brighton"                "Middlesbrough"           "Reading"
[31] "Sunderland"              "Manchester United"       "Crystal Palace"          "Stoke City"              "Norwich City"
[36] "Southampton"             "Blackburn Rovers"        "Aston Villa"             "Nottingham Forest"       "West Bromwich Albion"
[41] "Everton"                 "Leicester City"          "Derby County"            "Newcastle United"        "Manchester City"
[46] "Leeds United"            "Manchester United"       "Manchester City"         "Liverpool"               "Derby County"
[51] "Wolverhampton Wanderers" "Leeds United"            "Newcastle United"        "Middlesbrough"           "Nottingham Forest"
[56] "Everton"                 "Stoke City"              "Blackburn Rovers"        "Sunderland"              "West Ham United"
[61] "Chelsea"                 "Tottenham Hotspur"       "Fulham"                  "Arsenal"                 "Crystal Palace"
[66] "Leicester City"          "Brighton"                "West Bromwich Albion"    "Aston Villa"             "Southampton"
[71] "Norwich City"            "Reading"
> names[1:20]
 [1] "Arsenal"                 "Liverpool"               "Manchester City"         "Aston Villa"             "Tottenham Hotspur"
 [6] "Manchester United"       "West Ham United"         "Brighton"                "Wolverhampton Wanderers" "Newcastle United"
[11] "Chelsea"                 "Fulham"                  "Bournemouth"             "Crystal Palace"          "Brentford"
[16] "Everton"                 "Luton Town"              "Nottingham Forest"       "Burnley"                 "Sheffield United"
```

Finally, I put all these data into a data frame, so it provides a clear look. I use data.frame function to do that.

```
> table <- data.frame(league_names)
> table
              league_names
1                  Arsenal
2                Liverpool
3          Manchester City
4              Aston Villa
5        Tottenham Hotspur
6        Manchester United
7          West Ham United
8                 Brighton
9  Wolverhampton Wanderers
10        Newcastle United
11                 Chelsea
12                  Fulham
13             Bournemouth
14          Crystal Palace
15               Brentford
16                 Everton
17              Luton Town
18       Nottingham Forest
19                 Burnley
20        Sheffield United
>
```

References

Erolin, J. (2023). *Best Programming Languages for Web Development | BairesDev*. Www.bairesdev.com. https://www.bairesdev.com/blog/best-programming-languages-web-development/

Get element text — html_text. (n.d.). Rvest.tidyverse.org. Retrieved March 28, 2024, from https://rvest.tidyverse.org/reference/html_text.html

MDN Web Docs. (2018, December 10). *HTML: HyperText Markup Language*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/HTML

Select nodes from an HTML document — html_nodes. (n.d.). Rvest.tidyverse.org. Retrieved March 28, 2024, from https://rvest.tidyverse.org/reference/html_nodes.html