**Task 6 Communication on tasks**

* **Task 1 crawling**

Crawling is a method to get data from web, which is a highly linked graph with links from one page to another. In order to crawl all pages, I start from a seed URL and extract the link of another page by parsing and indexing current page, until all pages in the web has been operated. These urls all consist of a base url and their unique url of links. A prioritised list and a visited list are created. The former list has urls that are to be visited, and its url will be removed to visited list if it has been visited. It serves to avoid crawling a page twice, thus the loop could stop.

After the loop of crawling is finished, the program outputs two lists about our wanted information. They are toke as parameters, being constructed in a Data Frame and further be converted into a csv file called “task1.csv”. The output file contains two columns called ‘headlines’ and ‘urls’, and has 100 rows of data in total.

* **Task2 scraping**

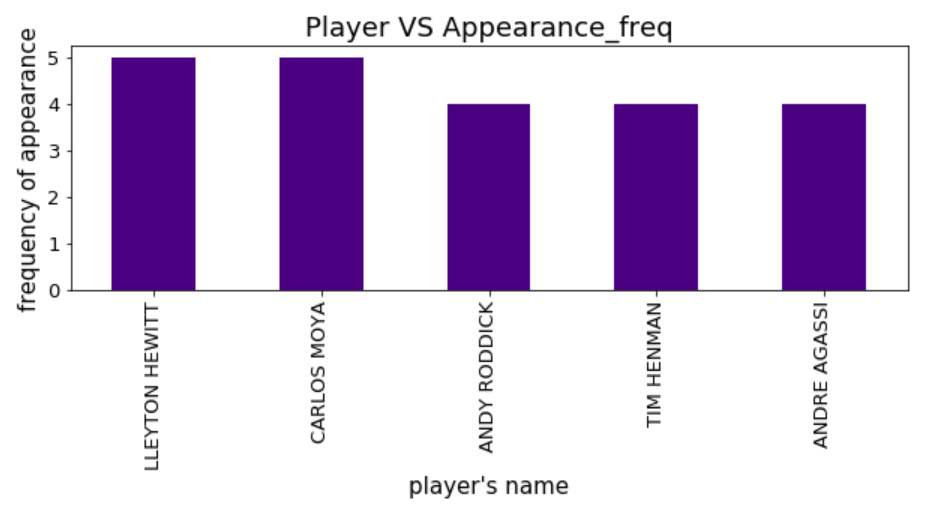
A description of how you scraped data from each page, including any regular expressions used for Task 2 and a brief summary of the output.

To scrap data from each page, I first iterate through each url in visited url list and parse and index HTML text on its tag and attribute name: ('div', id = 'articleDetail') by BeautifulSoup. The text of each article is accessed by accessing ‘text’ attribute of resulted bs4 object.

Firstly, the match score is found by implementing re search function based on score pattern. The pattern is composed of at least 2 and at most 10 repetitions (including tiebreaks) of “#-#” combination. If there is a complete and valid match score, then I use re\_search function again and search for matched names in each article according to names in name\_list (pattern) and record its index position in the text for comparison. The name with smallest index for each article is added into lists, as well as url, headline, and the score. These four are the columns of output “task2.csv”, which has 41 rows of records.

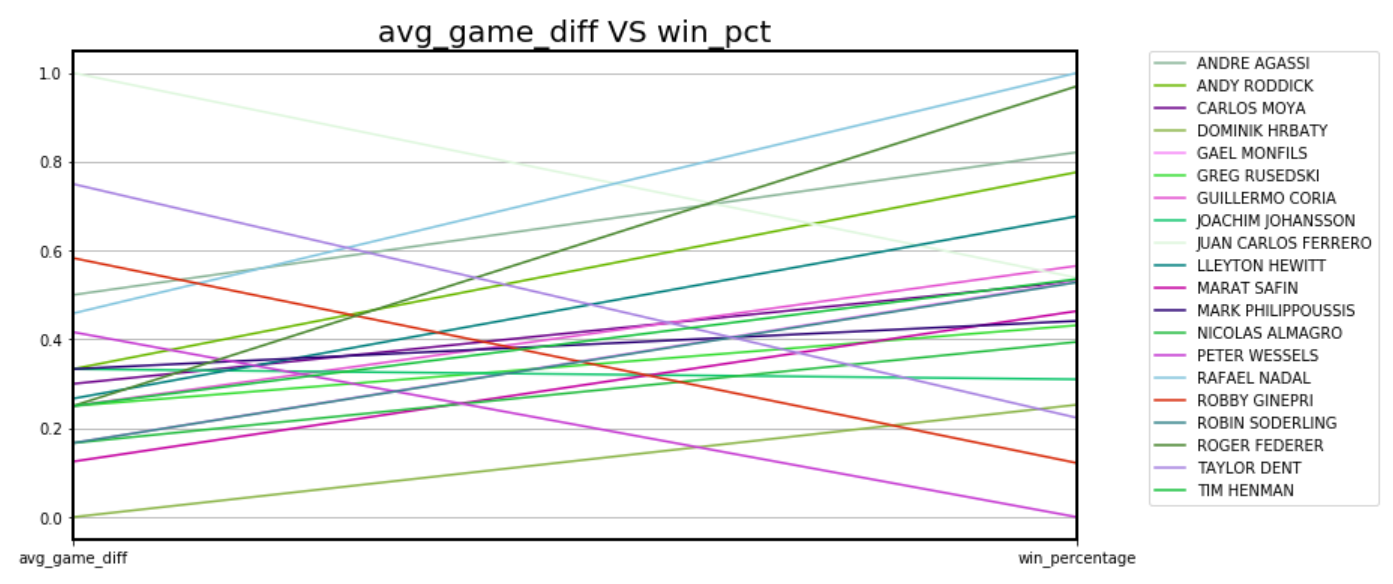
* **Task 4&5. Plot analysis**

The bar chart below represents five players that articles are most frequently written about and their frequency of appearances. I choose the bar chart because two variables are respectively numerical and categorical. The input data(df4) with these two columns is implemented by firstly separating all articles into groups of the same player, count the frequency and sort for the top 5. As the label shows, the vertical bars compare the frequency for each of the five players (horizontal variable), and are placed in a descending order. Lleyton and Carlos are mostly mentioned, for five times. The rest three players are mentioned for four times. Among 40 articles with 20 named players in total, the frequency of top 5 is much higher than the average (40/20 = 2) and shows there are also names appears only once or twice.



For task5, I choose to plot data in parallel coordinates, which allows to compare the average game difference and win percentage of each players on a set of numeric variables. Two vertical axes initially have different units and scales. The domain of average game difference is in [1.0, 13.0], far smaller than the range of win percentage in [42.9, 83.2]. Therefore, I normalised the data of two columns and represent these two features on the same, standard scale of 0-1, so they are approximately in the same domain. Values are then plotted as series of lines connected across each axis, where each line represent a single player.

In terms of the plot, a positive relationship is presented in general: greater game difference, greater win percentage. There are few crossing lines, where all three inverse lines have higher-then-average difference of 0.6, 0.8 and 1.0 on scale (most of players have differences between 0.2 and 0.4, which are all in the lower half, yet all of them were distributed evenly in terms of win percentage.). Among 20 lines in total, 6 lines (one horizontal line, three inverse lines) of exception results in a correlation coefficient of 0.8 and indicates a strong positive linear relationship. Yet it does not represent an absolute relationship, as Juan Carlos who has the greatest average game difference (1.0) only rates 0.5 in terms of win percentage and Rafel who rates top in win percentage has an average game difference of 0.5.



* **appropriateness of the method for deriving ﬁrst named player**

It’s inappropriate to assume that first matched score goes with first named player, since there could be first named player mentioned without writing about his match scores, and later changes the topic and writes match scores about other two players in the same article. This assumption may successfully match sometimes, when the score exactly follows the first name, yet still has a low accuracy and strongly depends on scenarios.

* **Method for determining win/loss of the ﬁrst named player**

Assume the left score stands for the first named player, instead of taking the absolute value of game difference, keep the original difference to see whether the difference is positive or negative. If the game difference is positive, the first named player wins and vice versa.

* **Information extraction and processing method for understanding player performance**

Related players (past opponents) could also be extracted from the article since they are always mentioned on past games they played with the main character in that article, Instead of adding both matched names and their index positions in a dictionary and just take the first one, we could take the rest into a list and save as related players. The number of past opponents could refer to the length of time for tennis match career and as a more experienced player. It could also relate to win percentage for further analysing on each players’ performance in a comprehensive way. Some excellent players may share a low win percentage since they join much more games than others, having greater probability of losing a game.