By looking at the source code of every page, the link of the website can be easily identified after the “a” tag since we use the “page.text” method and then parse in html, therefore, by using the “findAll” method in “BeautifulSoup” library, we can correctly specify all the links. By using the “while loop”, we can put all the links into a list and use a “for loop” to mark the item as visited, so that we can avoid to put the same link into the list.

Then, the headline part, for each web pages, I parse it in html and the headline can be easily found after the “h1” tag. Then I use a “for loop” with a “findAll” method to find all the headlines. Then I put the web URL and headline together into the csv file with the header: ‘URL’ and ‘headline’.

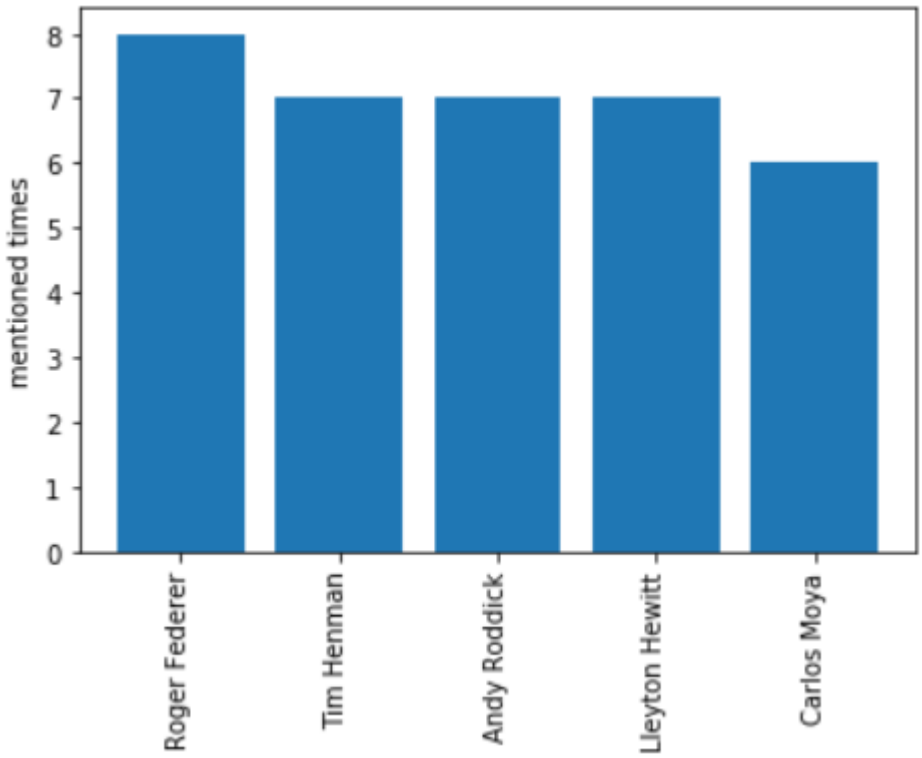
The csv file includes two columns headings ‘URL’ and ‘headline’. Everything is separated by comma and all the headlines go to the right URL.

For Task 2(a), since we need to find the first player’s name in each web page, firstly I import json so that I can get a full name list of players’ names and I created a list called ‘name_list’ to install all the name that I found in the json file. Then I used the ‘findAll('p’)’ method to go through all the web pages and put the articles into a list with their own URL. Then I separated the elements in the list by two at a time. Since most of the players do not have a middle name, so I just used a for loop and an if-else statement to identify if the name appears in the article is in the ‘name_list’. Therefore, if the name appears in the article is in the ‘name_list’, I then store the web URL and all the names that appear under the URL into a new list called ‘player’, and again, I separated the elements in the list by two at a time. Therefore, the name right after the URL will be the first player that occurs in that web page.

For Task 2(b), I used the regular expression to find each score that is mentioned in each article. Then I put the score and its corresponding URL together so that it is easy for me to check if the output is correct or not. I noticed that some of the scores are included in the bracket, so I manually replace the remove the brackets by letting the score equals the whole thing.
The regular expression that I used to search the pattern in the article is “ r'(\(?[\d]+-[\d]+\)? *){2,5}' ”, which means that we can still find the score that is included in the bracket, then, since a full tennis game is either BO3 or BO5, therefore, we need to find at least two scores in an article that describe the whole game and find no more than five scores.

There are only 44 lines including URL, headline, player and score in the task2.csv file, which are 56 lines short than task1.csv file. And the Task2.csv file looks more make sense than Task1.csv because the Task2.csv file is more detailed than Task1.csv file.

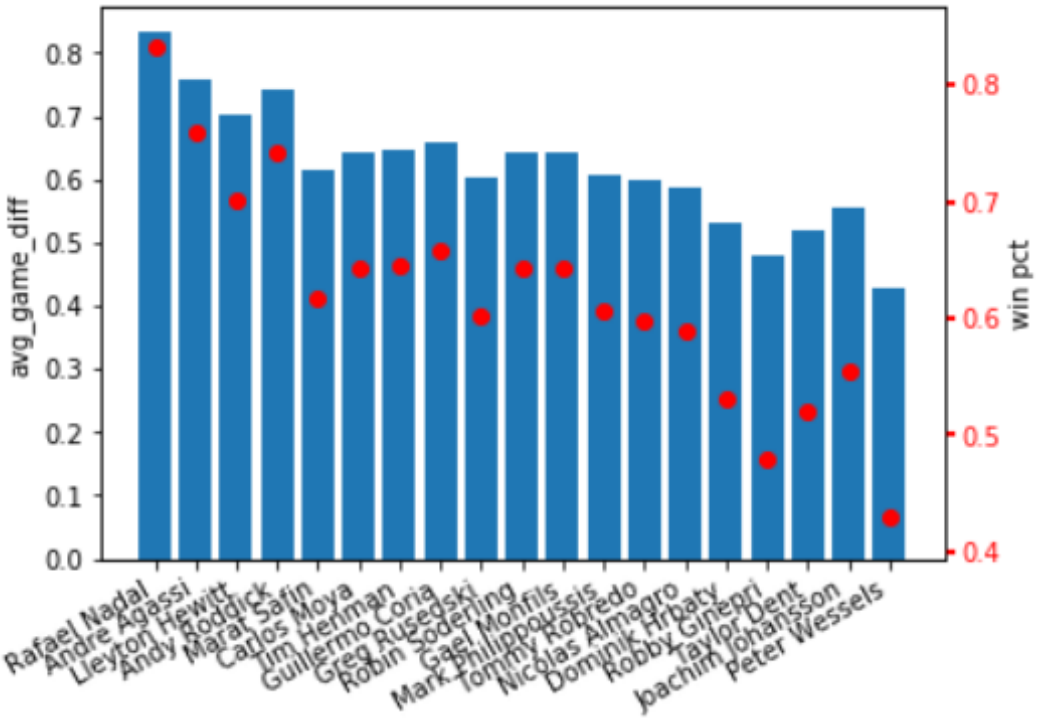
By looking at the plot in Task 4, it is easy to see that the five players that articles are most frequently written about are ‘Roger Federer’, ‘Tim Henman’, ‘Andy Roddick’, ‘Lleyton Hewitt’ and ‘Carlos Moya’. The times these players are mentioned are 8, 7, 7, 7 and 6, which does not have a really big difference compared to each other.



By looking at the plot in Task 5, the blue bar represents the average game difference of each player, and the red dot represents the win percentage of each player. It is clearly to see that none of the red dots are above the blue bars, all the red dots are sitting on the blue bar.

It is easy to identify that 'Nafael Nadal', 'Andre Agassi', 'Lleyton Hewitt' and 'Andy Roddick' have a better win percentage than other players. I personally think that these four players' blue bar and red dot have a strong association. By looking at 'Dominik Hrbaty's and Peter Wessels' association between the blue bar and the red dot, the red dots for both players are low, especially 'Peter Wessels' red dot.

Personally, I think that the reason of causing the results that I talked above could depend on player's performance in every game and their opponent as well. If a player can not player stably, then it will drag down the win percentage and increase the game difference. If the player and player's opponent does not at the same level, then it will affect the result of the win percentage and the win percentage for both the player and player's opponent.



I personally think that the association of the first named player in the article with the first match score is not appropriate. Since we do not know the player's opponent, it is hard for people to judge the player in a specific game because people do not know if the player and player's opponent are at the same level or not.

- At least one suggested method for how you could figure out from the contents of the article whether the first named player won or lost the match being reported on.

  Each article should have a total score, we can try to develop a regular expression to find the total score.

- A discussion of what other information could be extracted from the articles to better understand player performance and a brief suggestion for how this could be done.

  There is one way to extract the article to have more information is that we can try to scrap all the names of the competitions in the corresponding articles and get them a weight, each game's weight can be decided based on the international repercussions, therefore, we can use a csv file to store all the games and the corresponding weight. Then we can simply scrape all the games that are mentioned in one article and calculated the total weight of the winner in that article and make the score positive, then the opponent will have a negative score. Since the winner of one article might be the defeated one of another articles, therefore, we can add these scores up together with their win percentage and do more analysis on each player.