Elements of Data Processing - Project 1: Task 6

Fu sheng (Jeremy) Huang (1046432)

Crawling In Task1

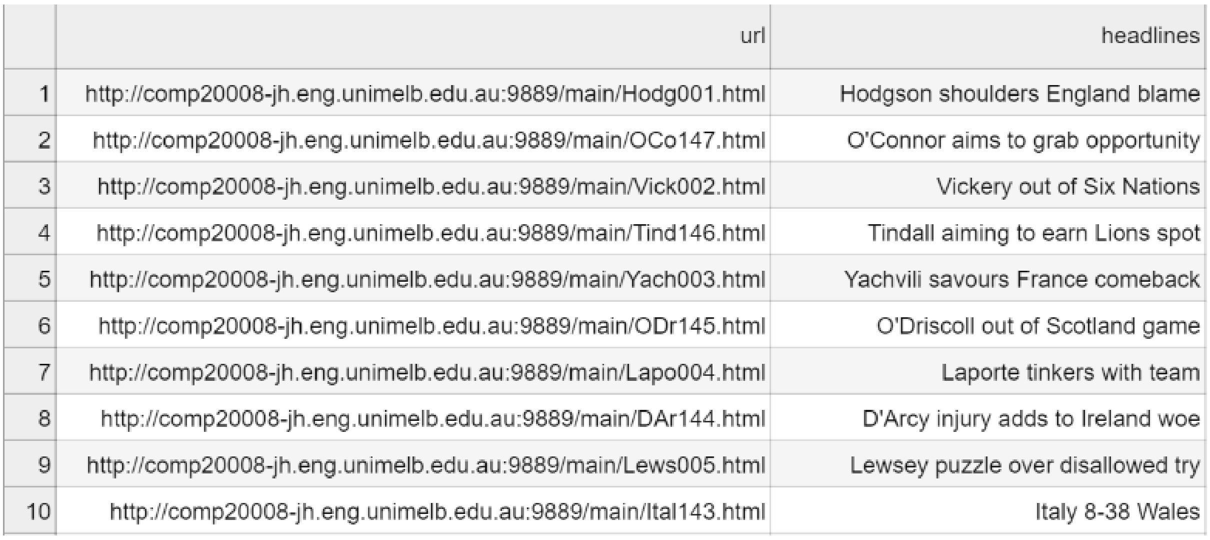
The crawling method requires a ***base link*** in which the website will be crawled from. In this project, this is the given website: http://comp20008-jh.eng.unimelb.edu.au:9889/main/.

To crawl the data for each website, it needs to be access to the code in which the website is written in. In this case, the website was written in HTML, and hence a seed item, `index.html` was added to the link, which then a parser, from the Python library, BeautifulSoup, enables the 'parsing', or retrieving of the website data directly from the HTML code.

Once the HTML parser is set up, a dictoinary, `visited`, was set up so that there would be ***no article parsed more than once***. Then, the base link was parsed and checked all sections of the code with the `<a>` tag. The output is a direct copy of the HTML code, so the link cannot be accessed in this state, which is why each link was converted into an appropriate format, particularly by using the `urljoin` command. These newly converted links were placed in a list, `to_visit`.

Following this, the process of accessing each article on the website begins. The website is accessed in the same way the base link was accessed and parsed (as shown above) to collect links for other articles. Each website is checked to see whether it has already been accessed, and whether it is already in the list of websites to be checked, `to_visit`. As the link currently being visited gets removed from the list, the process is repeated until the `to_visit` list becomes empty. The links and headlines for every website accessed was recorded, which are also formed in **DataFrame** from pandas library to be converted to the csv file, 'task1.csv'.

In 'task1.csv', is a csv with 148 lines; 1 line for the column headings, and 147 lines for the 147 websites found with its url link and headline. As the task required the crawling of every article, the base page its link and headline was excluded. Figure 1 shown below is a small segment of the csv file.



	url	headlines
1	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Hodg001.html	Hodgson shoulders England blame
2	http://comp20008-jh.eng.unimelb.edu.au:9889/main/OCo147.html	O'Connor aims to grab opportunity
3	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Vick002.html	Vickery out of Six Nations
4	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Tind146.html	Tindall aiming to earn Lions spot
5	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Yach003.html	Yachvili savours France comeback
6	http://comp20008-jh.eng.unimelb.edu.au:9889/main/ODr145.html	O'Driscoll out of Scotland game
7	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Lapo004.html	Laporte tinkers with team
8	http://comp20008-jh.eng.unimelb.edu.au:9889/main/DAr144.html	D'Arcy injury adds to Ireland woe
9	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Lews005.html	Lewsey puzzle over disallowed try
10	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Ital143.html	Italy 8-38 Wales

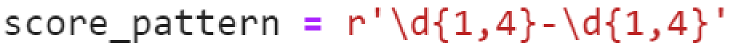
Figure 1: part of task1.csv file

Elements of Data Processing - Project 1: Task 6

Fu sheng (Jeremy) Huang (1046432)

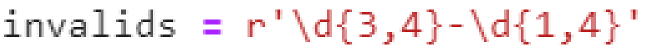
Scraping In Task 2

The scraping method begins by making a list of the data that in tag 'teams' mentioned in the json file, `team_names` created by collecting names and there is a list of the links to access which was obtained by accessing the `visited.keys()` dictionary made for scraping. For every link in the `visited.keys()`, the headline and the article contents were combined and parsed into the variable, `article`, which is then compiled so it can be searched using **`score_pattern`**. This was done to match a pattern made using **`score_pattern`** to find all kind of score in this form "xxxx-xxxx" and extract the largest match score identified in the article. The RE pattern code is the following:



```
score_pattern = r'\d{1,4}-\d{1,4}'
```

The above RE code checks series of numbers matching the possible scores. However, ***it will not check date*** that in this pattern. Thus, I use **`invalid`** pattern to get the invalid data stored in **`errors`**.

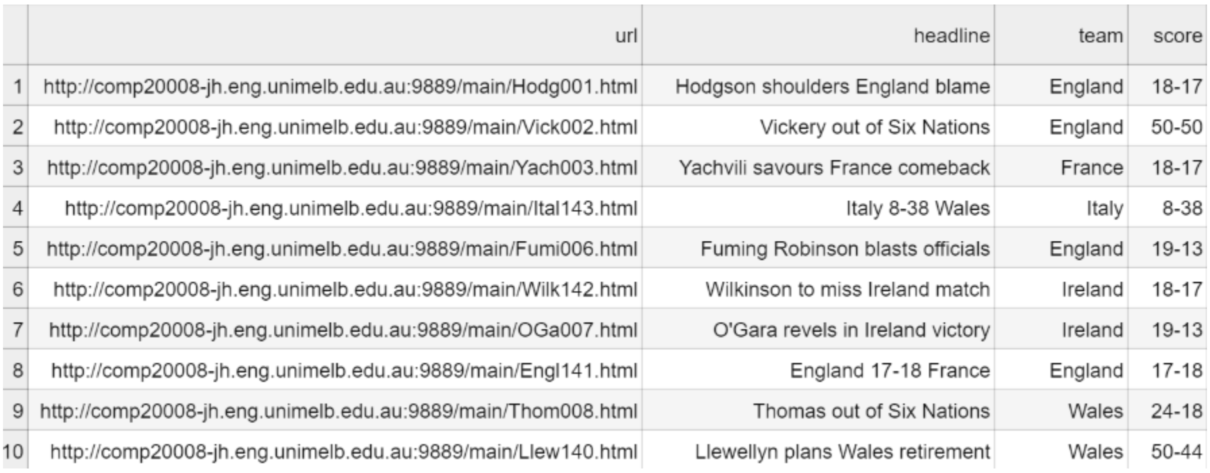


```
invalids = r'\d{3,4}-\d{1,4}'
```

Then use **`Valid`** to obtain a valid list without invalid data from **`searching`**. Next, using **`sum_or_diff`** to collect the number of sums from each valid match score stored in **`scores`** dictionary. I will use **`largest`** to get the max value from **`scores`**. In my findings, I noticed that there were some invalid data like 50-50(showed in), which are ignored in my project because in real world, we may work with billions of data or even more in our job. In this situation, ***we cannot check every single page or every single sentence just to find specific invalid information***. Having said that, we can still check if we are asked to.

To find the first name written in the article, the index and name (in that order) of every team mentioned in the article was recorded in a dictionary. And we used the built function to find the name with lowest index, which was stored in the list **`teams`**. Lastly, we made them in the order of url, headline, team, score, which was then used to make the csv file, 'task2.csv'.

In 'task2.csv', is a csv file with 65 lines; 1 line for the column headings, 64 links and headlines with names and valid largest sum scores. Figure 2 shown below is a small segment of the csv file.



	url	headline	team	score
1	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Hodg001.html	Hodgson shoulders England blame	England	18-17
2	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Vick002.html	Vickery out of Six Nations	England	50-50
3	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Yach003.html	Yachvili savours France comeback	France	18-17
4	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Ital143.html	Italy 8-38 Wales	Italy	8-38
5	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Fumi006.html	Fuming Robinson blasts officials	England	19-13
6	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Wilk142.html	Wilkinson to miss Ireland match	Ireland	18-17
7	http://comp20008-jh.eng.unimelb.edu.au:9889/main/OGa007.html	O'Gara revels in Ireland victory	Ireland	19-13
8	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Engl141.html	England 17-18 France	England	17-18
9	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Thom008.html	Thomas out of Six Nations	Wales	24-18
10	http://comp20008-jh.eng.unimelb.edu.au:9889/main/Llew140.html	Llewellyn plans Wales retirement	Wales	50-44

Figure 2: part of task2.csv file

Elements of Data Processing - Project 1: Task 6

Fu sheng (Jeremy) Huang (1046432)

Plot Analysis for task 4 & 5

As we already retrieved the average game difference in task 3, we will use its DataFrame in the following plotting process. In task 4, we were told to plot the top five teams that were most frequently written in articles with the numbers the articles. By using **groupby** , **sort_values** and **count** function by seeking team name and url, respectively. And slicing them to take the top 5. Then we got "**England**", "**Ireland**", "**Wales**", "**France**" and "**Italy**", from the most to the least. By setting the names of the players in the x-axis and the frequency in the y-axis, the following bar graph was produced named "task4.png".

In Figure 3, we can see the number of articles relating to "England" showing that there were more articles written about "**England**" than anyone else has the absolute dominance with 20 articles while "**Ireland**" is the second with over 5 articles less than the first in the 64 articles obtained from task 2. But there were 2 teams mentioned less than 10 times, which are "**France**" and "**Italy**". As the variable, teams, is a categorical variable, mapped against a numerical variable, frequency, a bar chart was used, showing the 5 most frequently mentioned players out of the 64 articles from task 2.

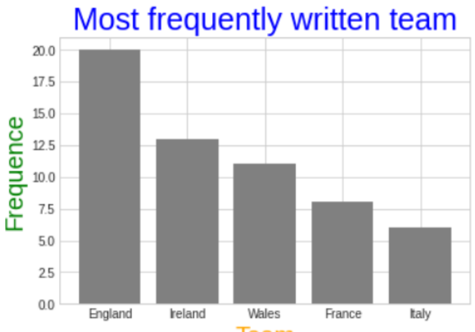


Figure 3: task4.png

As part of task 3, a csv file 'task3.csv' was created, having a list of 7 teams with their average game difference, which was made through looking at every match score corresponding to the teams from 'task2.csv'. A new dictionary was made to collect the average game difference of each team with the number of articles mentioned so that the plot would show the correct correlation between the two variables. This data was then have double-bar in single chart by setting the values for the average game difference on the right bar and the number of articles at the left bar, then these double-bars were labelled team name accordingly , the following chart was produced. A double-bar plot was chosen as the correlation between two numeric variables, average game difference and number of articles mentioned, aiming to be better observed.

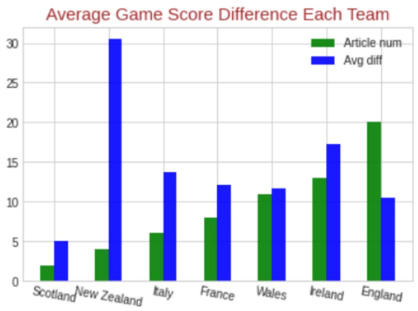


Figure 4: task5.png

This image displayed the overall trend that the average game difference score is not proportional to the mentioned times. Take "**England**", for example, it had the most mentioned times with the second least value for average difference score. In my perspective, the reason behind probably "**England**" is the most popular team, however, they played quite terrible comparing to the expectation while the other teams were probably better than their anticipations. But there was also another explanation for this.

Elements of Data Processing - Project 1: Task 6

Fu sheng (Jeremy) Huang (1046432)

In task 3, it specifically determines the absolute value of the game difference. This means that the value of the game difference score merely indicates the difference and not the wins or losses of that game. For the plot from task 5, we can see fairly obvious trend that the number of mentioned times has a positive relationship with the score, however, we have no idea which team has the better winning percentage or their playing statement in the game. And the relationships between them has not much sense to readers. Basically, ***taking the absolute value of the game difference was not appropriate*** since it led to an untidy data set. The untidiness in the plot could have come from the fact that it may also be an outlier; for example, a team which only wins or loses games with a large margin. ***Thus, we are supposed to take outcome of each game into consideration and either the positive or negative value should be handled carefully.***

Appropriateness Of First Team Having First Match Score

There are many cases where the first name mentioned in the article is related to the first match score in the article. However, this does not apply to all articles. For example, the first name mentioned may not even be a part of the match of discussion, or the match score may be from a past game as a reference for the article. Although the first name mentioned in the article does somewhat relate to the first match score, ***it is still hard to applies to every article***, and therefore using this data for a presentation can become inaccurate. Checking that the name in the title matches with the first name mentioned in the article will probably be a good verification to make sure the first name mentioned is related to the article, rather than just a reference. Overall, I will say that it is not appropriate to associate the first name mentioned with the first match score in the article. Since we may handle billions of data or over in real document, it will then lead us to have low accuracy and pretty high risk for our analysis.

Methods For Determining Win/Loss Of The First Named Team

Suggestion 1:

Firstly, I would like to collect both right side and left sides of match score, which are the team names, then calculate the difference to know if is either positive or negative to each side of team according to if the larger score is at the side of the team. Next, the first mentioned team name will have the total difference score with proper value.

Suggestion 2:

If we assume this article is main discussing about the first named team, we can collect the sides from each valid match score of the named team and the absolute difference value. Then, checking which side of each match score has larger score and if the side is according to the first named team, positive for yes and negative for no. Finally, we will get proper value.

As I mentioned in plotting question, the inappropriateness of just taking absolute value for the difference from match score, we can combine these two methods above to determine whether a team has won or lost the game.

Elements of Data Processing - Project 1: Task 6

Fu sheng (Jeremy) Huang (1046432)

Suggestion Of Knowing Team Performance

By determining who has the most mentioned times in each article and assuming the team is mainly discussed in the article, then checking the positive word in each sentence such as “good”, “excellence”, “wonderful”, and the negative word as well, such as “disappointed”, ”mistakes”. Then, we collect the numbers of each kind, using the former minus the latter gets a score. Making a list that have all the name with relative score, then group them by their names and calculate the overall score for each team. We will get nearly perfect performance report for each team.