Project #1

Anthony Conrardy

2024-02-10

Chess Tournament File

We will pull in the Chess Tournament Results table from the text file to begin manipulating the structure to create the proposed CSV file to be imported into a SQL database. Once loaded we can view the structure of the results and plan a way to extract the data for our CSV file. According to the Project 1 Assignment directions, we will need the following data elements: Player's Name, Player's State, Total Number of Points, Player's Pre-Rating and Average Pre Chess Rating of Opponents.

```
##  [1] "-----------------------------------------------------------------------------------------"
##  [2] " Pair | Player Name                     |Total|Round|Round|Round|Round|Round|Round|Round| "
##  [3] " Num  | USCF ID / Rtg (Pre->Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  | "
##  [4] "-----------------------------------------------------------------------------------------"
##  [5] "    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
##  [6] "   ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [7] "-----------------------------------------------------------------------------------------"
##  [8] "    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
##  [9] "   MI | 14598900 / R: 1553   ->1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
## [10] "-----------------------------------------------------------------------------------------"
```

Tournament Results Table Review

This results table appears to be very complex in structure. The headers take up two lines, and the data is not easily discernible from the header information.

Looking at the first entry (Gary Hua), in the first column we have the players number above what appears to be the state abbreviation. In the second column the players name, USCF ID number and pre- and post ratings are listed. The third column contains the total points and another abbreviation and number with unexplained significance. The fourth through tenth columns contain information on the round, whether the person won, lost, or draw, and against which opponents.

There are some issues that we need to immediately address. The dashed lines need to be removed, which are created with hyphens.

```
##  [1] ""
##  [2] " Pair | Player Name                     |Total|Round|Round|Round|Round|Round|Round|Round| "
##  [3] " Num  | USCF ID / Rtg (Pre>Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  | "
##  [4] ""
##  [5] "    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
##  [6] "   ON | 15445895 / R: 1794   >1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [7] ""
##  [8] "    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
##  [9] "   MI | 14598900 / R: 1553   >1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
## [10] ""
```

However, after running this code we see that it changes the symbol that they were using to show an increase in ratings from -> to just >. The -> also looks exactly the same as the assignment operator, so changing that first before removing the hyphens is probably a better idea.

```
##  [1] ""
##  [2] " Pair | Player Name                     |Total|Round|Round|Round|Round|Round|Round|Round| "
##  [3] " Num  | USCF ID / Rtg (Pre=>Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  | "
##  [4] ""
##  [5] "    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
##  [6] "   ON | 15445895 / R: 1794   =>1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [7] ""
##  [8] "    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
##  [9] "   MI | 14598900 / R: 1553   =>1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
## [10] ""
```

We can now remove the first four (4) rows, since they contain the old headers and empty lines.

```
##  [1] "    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
##  [2] "   ON | 15445895 / R: 1794   =>1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [3] ""
##  [4] "    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
##  [5] "   MI | 14598900 / R: 1553   =>1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
##  [6] ""
##  [7] "    3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|"
##  [8] "   MI | 14959604 / R: 1384   =>1640     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [9] ""
## [10] "    4 | PATRICK H SCHILLING             |5.5  |W  23|D  28|W   2|W  26|D   5|W  19|D   1|"
```

The table is now structured in a way that we can remove every third line. and compress to two lines per chess player entry. This is done by applying a logical vector tho the data lines through the length of data.

```
## [1] "    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
## [2] "   ON | 15445895 / R: 1794   =>1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
## [3] "    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
## [4] "   MI | 14598900 / R: 1553   =>1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
## [5] "    3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|"
```

Now we can subset the data similar to the way we removed the empty row. We can subset using applying a logical vector to the data for the first and second row through the length of data.

```
## [1] "    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
## [2] "    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
## [3] "    3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|"
## [4] "    4 | PATRICK H SCHILLING             |5.5  |W  23|D  28|W   2|W  26|D   5|W  19|D   1|"
## [5] "    5 | HANSHI ZUO                      |5.5  |W  45|W  37|D  12|D  13|D   4|W  14|W  17|"
```

```
## [1] "   ON | 15445895 / R: 1794   =>1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
## [2] "   MI | 14598900 / R: 1553   =>1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
## [3] "   MI | 14959604 / R: 1384   =>1640     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
## [4] "   MI | 12616049 / R: 1716   =>1744     |N:2  |W    |B    |W    |B    |W    |B    |B    |"
## [5] "   MI | 14601533 / R: 1655   =>1690     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
```

Now we can start the process of extracting the data into columns. We can see most data elements are separated by "|", so we can use that to parse the columns. Let's start with the lines that have the have the participants name. We can parse out player number, player name, total points and the results for each match.

```
##   player_number                       player_name total_points round1 round2
## 1             1   GARY HUA                                 6.0    W  39  W  21
## 2             2   DAKSHESH DARURI                          6.0    W  63  W  58
## 3             3   ADITYA BAJAJ                             6.0    L   8  W  61
## 4             4   PATRICK H SCHILLING                      5.5    W  23  D  28
## 5             5   HANSHI ZUO                               5.5    W  45  W  37
##   round3 round4 round5 round6 round7
## 1  W  18  W  14  W   7  D  12  D   4
## 2  L   4  W  17  W  16  W  20  W   7
## 3  W  25  W  21  W  11  W  13  W  12
## 4  W   2  W  26  D   5  W  19  D   1
## 5  D  12  D  13  D   4  W  14  W  17
```

The second line is a little more tricky. While we have the player state, the pre- and post- ratings are mixed in a string with the USCF id. There are a number of columns we simply won't need for assignment. We split the strings based upon "|" as a delimiter, drop the unnecessary columns, and then split the USCF_id and player ratings using ":" and "=>", respectively. We can then bind the columns into a single frame.

```
##   player_state pre_rating post_rating
## 1           ON      1794        1817
## 2           MI      1553        1663
## 3           MI      1384        1640
## 4           MI      1716        1744
## 5           MI      1655        1690
```

Once we have parsed each of the columns to obtain our required data from both the first and second rows, we can now combine the two data frames into a single frame with the required data to continue with calculating the average pre-chess ratings of the opponents for each player.

```
##   player_number                       player_name total_points round1 round2
## 1             1   GARY HUA                                 6.0    W  39  W  21
## 2             2   DAKSHESH DARURI                          6.0    W  63  W  58
## 3             3   ADITYA BAJAJ                             6.0    L   8  W  61
## 4             4   PATRICK H SCHILLING                      5.5    W  23  D  28
## 5             5   HANSHI ZUO                               5.5    W  45  W  37
##   round3 round4 round5 round6 round7 player_state pre_rating post_rating
## 1  W  18  W  14  W   7  D  12  D   4           ON      1794        1817
## 2  L   4  W  17  W  16  W  20  W   7           MI      1553        1663
## 3  W  25  W  21  W  11  W  13  W  12           MI      1384        1640
## 4  W   2  W  26  D   5  W  19  D   1           MI      1716        1744
## 5  D  12  D  13  D   4  W  14  W  17           MI      1655        1690
```

No we can pull out opponent numbers from each of the matches, and then rename the column to indicate that it is the opponent number now represented.

```
##   player_number                       player_name total_points opp_rnd1
## 1             1   GARY HUA                                 6.0       39
## 2            2   DAKSHESH DARURI                          6.0         63
## 3            3   ADITYA BAJAJ                             6.0          8
## 4            4   PATRICK H SCHILLING                      5.5         23
## 5            5   HANSHI ZUO                               5.5         45
##   opp_rnd2 opp_rnd3 opp_rnd4 opp_rnd5 opp_rnd6 opp_rnd7 player_state pre_rating
## 1       21       18       14        7       12        4           ON     1794
## 2       58        4       17       16       20        7           MI     1553
## 3       61       25       21       11       13       12           MI     1384
## 4       28        2       26        5       19        1           MI     1716
## 5       37       12       13        4       14       17           MI     1655
##   post_rating
## 1   1817
## 2   1663
## 3   1640
## 4   1744
## 5   1690
```

Cleaning Up the Frame

Here we do a little housecleaning on the data frame. We will eliminate any letter characters or numbers from the character strings in the pre- and post-rating columns, as well as any white space before any of the strings. This will leave the remaining characters that are the rating themselves We will also replace missing data with "NA" in the strings that are missing.

```
##   player_number                     player_name total_points opp_rnd1
## 1             1   GARY HUA                                 6.0         39
## 2             2   DAKSHESH DARURI                          6.0         63
## 3             3   ADITYA BAJAJ                             6.0          8
## 4             4   PATRICK H SCHILLING                      5.5         23
## 5             5   HANSHI ZUO                               5.5         45
##   opp_rnd2 opp_rnd3 opp_rnd4 opp_rnd5 opp_rnd6 opp_rnd7 player_state pre_rating
## 1       21       18       14        7       12        4           ON       1794
## 2       58        4       17       16       20        7           MI       1553
## 3       61       25       21       11       13       12           MI       1384
## 4       28        2       26        5       19        1           MI       1716
## 5       37       12       13        4       14       17           MI       1655
##   post_rating
## 1        1817
## 2        1663
## 3        1640
## 4        1744
## 5        1690
```

Assigning Pre_Chess Rating to the Opponents

In this section we will take the opponent numbers for each round and replace it with their assigned pre-chess rating. This was done by extracting out the individual player numbers and ratings, and then using that as a reference frame for the numbers assigned to the opponents in each round. A new column is created for each round now containing the associated pre-chess rating to the opponent number in the round column (i.e. opp_rndtable1). While I am sure there is a much quicker way of coding this (for loop or function), I was not able to learn that competently enough for this assignment.

```
##   player_number                       player_name total_points opp_rnd1
## 1             1  GARY HUA                                 6.0       39
## 2             2  DAKSHESH DARURI                          6.0       63
## 3             3  ADITYA BAJAJ                             6.0        8
## 4             4  PATRICK H SCHILLING                      5.5       23
## 5             5  HANSHI ZUO                               5.5       45
##   opp_rnd2 opp_rnd3 opp_rnd4 opp_rnd5 opp_rnd6 opp_rnd7 player_state pre_rating
## 1       21       18       14        7       12        4           ON       1794
## 2       58        4       17       16       20        7           MI       1553
## 3       61       25       21       11       13       12           MI       1384
## 4       28        2       26        5       19        1           MI       1716
## 5       37       12       13        4       14       17           MI       1655
##   post_rating rnd1_rating rnd2_rating rnd3_rating rnd4_rating rnd5_rating
## 1        1817        1436        1563        1600        1610        1649
## 2        1663        1175         917        1716        1629        1604
## 3        1640        1641         955        1745        1563        1712
## 4        1744        1363        1507        1553        1579        1655
## 5        1690        1242         980        1663        1666        1716
##   rnd6_rating rnd7_rating
## 1        1663        1716
## 2        1595        1649
## 3        1666        1663
## 4        1564        1794
## 5        1610        1629
```

Calculating Average Pre-Chess Rating of Opponents

In this section we simply average the pre-chess ratings of all faced by a specific player and place that in a column at the end. We will take each of the columns, that are currently characters, and treat them as numeric for the calculation. We are also doing additional housekeeping by cleaning up the data frame by extracting out the requested columns in the order requested. The final column will contain the Average Pre-Chess Rating of Opponents faced by a participant, rounded to the nearest whole number.

```
##                         player_name player_state total_points pre_rating
## 1  GARY HUA                                   ON          6.0       1794
## 2  DAKSHESH DARURI                            MI          6.0       1553
## 3  ADITYA BAJAJ                               MI          6.0       1384
## 4  PATRICK H SCHILLING                        MI          5.5       1716
## 5  HANSHI ZUO                                 MI          5.5       1655
##   avg_pre_chess_rating_opp
## 1                     1605
## 2                     1469
## 3                     1564
## 4                     1574
## 5                     1501
```

Writing File to CSV for Export

In this section we prepare the file for export as CSV for possible import into a SQL database.

```
write.csv(tdr_average_final, "C:/Users/para2/Documents/GitHub/DATA607/Project_1/Project_1_Conrardy.csv"
print('CSV file written successfully to location')
```

```
## [1] "CSV file written successfully to location"
```