

# Project 1

Eddie Perez

2022-02-20

## Project #1

In this project, you're given a text file with chess tournament results where the information has some structure. Your job is to create an R Markdown file that generates a .CSV file (that could for example be imported into a SQL database) with the following information for all of the players: Player's Name, Player's State, Total Number of Points, Player's Pre-Rating, and Average Pre Chess Rating of Opponents For the first player, the information would be:

Gary Hua, ON, 6.0, 1794, 1605

1605 was calculated by using the pre-tournament opponents' ratings of 1436, 1563, 1600, 1610, 1649, 1663, 1716, and dividing by the total number of games played.

```
# setting proper libraries
```

```
library(stringr)
library(knitr)
```

## Importing the data.

I used the read csv function to get my data and paste0 to concatenate because of the spaces on the chess file.

```
# Assign my Chess Data file located on github so that I can use it with the Read table Function
data = "https://raw.githubusercontent.com/Eperez54/Dat-607/main/Project%201/ChessData.txt"
chessData <- read.csv(paste0(data), header = F)
```

```
head(chessData)
```

```
##                                                                 V1
## 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 |
```

## Cleaning up

Removing unnecessary data the first four rows doesn't really contain information that we need I decided to truncate

*#omitting the first four lines as they do not have valid data but keeping everything else*

```
chessData <- chessData[-c(1:4),]  
head(chessData)
```

```
## [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] "-----"
```

## Separating Data

I noticed that both rows could be separated and extracted

```
player <- chessData[seq(1, length(chessData), 3)]  
rating <- chessData[seq(2, length(chessData), 3)]  
  
head(player)
```

```
## [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|"
## [6] "      6 | HANSEN SONG                  |5.0 |W 34|D 29|L 11|W 35|D 10|W 27|W 21|"
```

```
head (rating)
```

```
## [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   |"
## [6] "    OH | 15055204 / R: 1686    ->1687 |N:3 |W   |B   |W   |B   |B   |W   |B   |"
```

Right now I will be separating based on information need for the new chessdata.csv file. Here the skills that I learned from last week's homework came into effect and it was very useful in separating and extracting data, based on patterns

```
pairNumber <- as.integer(str_extract(player, "\\d+"))  
player_Name <- str_trim(str_extract(player, "(\\w+\\s){2,3}"))  
points <- as.numeric(str_extract(player, "\\d+\\.\\d+"))  
opponents <- str_extract_all(str_extract_all(player, "\\d+\\|"), "\\d+")
```

```
## Warning in stri_extract_all_regex(string, pattern, simplify = simplify, :  
## argument is not an atomic vector; coercing
```

```
draw <- str_count(player, "\\Q|D  \\E")
lost <- str_count(player, "\\Q|L  \\E")
Won <- str_count(player, "\\Q|W  \\E")

state <- str_extract(rating, "\\w+")
player_Rating <- as.integer(str_extract(str_extract(rating, "[^\\d]\\d{3,4}[^\\d]"), "\\d+"))
```

## Calculating average

```
opp_Rating <- length(player)
for (i in 1:length(player))
  opp_Rating[i] <- round( mean ( player_Rating [as.numeric (unlist( opponents[ pairNumber[i]]))]), digits = 1)
```

## Creating a new dataframe to hold my final chess data ready for export

```
finalChessData <- data.frame(pairNumber, player_Name, state, points, player_Rating, opp_Rating, Won, lost)
head (finalChessData)
```

```
##   pairNumber      player_Name state points player_Rating opp_Rating Won lost
## 1          1          GARY HUA   ON    6.0          1794          1605    5    0
## 2          2    DAKSHESH DARURI   MI    6.0          1553          1469    6    1
## 3          3      ADITYA BAJAJ   MI    6.0          1384          1564    6    1
## 4          4 PATRICK H SCHILLING   MI    5.5          1716          1574    4    0
## 5          5        HANSHI ZUO   MI    5.5          1655          1501    4    0
## 6          6      HANSEN SONG    OH    5.0          1686          1519    4    1
##   draw
## 1     2
## 2     0
## 3     0
## 4     3
## 5     3
## 6     2
```

## Exporting to a csv file

I use the write to csv file to export my chess data to file chessData.csv

```
write.csv(finalChessData, file = "chessData.csv")
```

## Conclusion

This project was a bit tricky because I knew where I wanted to end up but getting there was hard. Thankfully I used some of string manipulation that we learned from last week lab which helped me.