# The Chess Data File Parser

Tony Fraser / 26 Sept 2023

### Assignment:

Load a data file, manipulate it to a specification, write out a CSV.

### Apporach

First, take each of these records, combine to a single line, and turn the collection of single lines into a dataframe. Next, create a second dataframe of opponent scores and averages and left it join it back to the dataframe. Finally, filter down and manipulate the final dataframe to match the specification, and write out to CSV.

```
-----------------------------------------------------------------------------------------
 Pair | Player Name                     |Total|Round|Round|Round|Round|Round|Round|Round|
 Num  | USCF ID / Rtg (Pre->Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  |
-----------------------------------------------------------------------------------------
    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|
   ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |
-----------------------------------------------------------------------------------------
    2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|
   MI | 14598900 / R: 1553   ->1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |
-----------------------------------------------------------------------------------------
    3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|
   MI | 14959604 / R: 1384   ->1640     |N:2  |W    |B    |W    |B    |W    |B    |W    |
-----------------------------------------------------------------------------------------
    4 | PATRICK H SCHILLING             |5.5  |W  23|D  28|W   2|W  26|D   5|W  19|D   1|
   MI | 12616049 / R: 1716   ->1744     |N:2  |W    |B    |W    |B    |W    |B    |B    |
-----------------------------------------------------------------------------------------
```

*… many records omitted…*

## Source Code

```r
library(stringr)
library(tidyr)
library(magrittr)
library(dplyr)
library(purrr)

## -----------------------------------------------------------------------------------------------------
##  Helper Functions
## -----------------------------------------------------------------------------------------------------
file_to_df <- function(file_name) {
  #load the file, it's in same directory as this R file.

  lines <- readLines(file_name)
  # first, there's a bunch of lines in these files that are all hyphens.
  # let's get rid of those, and at the same time we'll chop off the two
  # top lines, which are static column headers.

  filtered_lines <- c()
  line_counter <- 1
  for (line in lines[seq(4, length(lines))]) {
    if (!str_detect(line, "----")) {
      filtered_lines[line_counter] <- gsub(" *\\| *", "|", str_trim(str_squish(line), side = "both"))
      line_counter <- line_counter + 1
    }
  }

  # Sweet.
  # now we have to deal with this crazy formatting, basically one record
  # spans across two rows one on top of the other. Logically, let's call
  # them "top_of_record" and "bottom_of_record", then we'll move them side
  # by side and collapse them onto a single line.

  odd_numbers <- seq(1, length(filtered_lines), 2)
  even_numbers <- seq(2, length(filtered_lines), 2)

  top_of_record <- filtered_lines[even_numbers]
  bottom_of_record <- filtered_lines[odd_numbers]

  # now let's put them side by side and merge into one line
  single_row_records <- paste(bottom_of_record, top_of_record)
  # records look like this now, perfect.
  # [1] "1|GARY HUA|6.0|W 39|W 21|W 18|W 14|W 7|D 12|D 4| ON|15445895 / R: 1794 ->1817|N:2|W|B|W|B|W|B|W
  # [1] "2|DAKSHESH DARURI|6.0|W 63|W 58|L 4|W 17|W 16|W 20|W 7| MI|14598900 / R: 1553 ->1663|N:2|B|W|B|W
  # [1] "3|ADITYA BAJAJ|6.0|L 8|W 61|W 25|W 21|W 11|W 13|W 12| MI|14959604 / R: 1384 ->1640|N:2|W|B|W|B|W
  # [1] "4|PATRICK H SCHILLING|5.5|W 23|D 28|W 2|W 26|D 5|W 19|D 1| MI|12616049 / R: 1716 ->1744|N:2|W|B

  # we need to get it into a dataframe now so can work with it.
  split_data <- lapply(single_row_records, function(x) strsplit(x, split = "\\|")[[1]])
  df <- as.data.frame(do.call(rbind, split_data), stringsAsFactors = FALSE)
  cols <- c("pair", "playersName", "totalPoints", "r1", "r2", "r3", "r4", "r5",
            "r6", "r7", "playersState", "B", "C", "D", "E", "F", "G", "H", "I", "J")
  colnames(df) <-  cols

  df_reformatted <- subset(df, select = -c(C, D, E, F, G, H, I, J, pair)) %>%
    separate(B,   #<- parse this column "15619130 / R: 1220P13 ->1416P20"
             into = c("ucsfId", "preRating", "postRating"), sep = " / R: | -> |->") %>%
    mutate(preRating = gsub("^(\\d+).*", "\\1", preRating)) %>%   # chop P13 off of 1220P13
    mutate(postRating = gsub("^(\\d+).*", "\\1", postRating)) %>%
    mutate(r1 = gsub(".*?(\\d+).*", "\\1", r1)) %>% # <- R1/Gary says "W 39", remove W
    mutate(r2 = gsub(".*?(\\d+).*", "\\1", r2)) %>% # <- R2/Gary says "W 21", remove W ...
    mutate(r3 = gsub(".*?(\\d+).*", "\\1", r3)) %>%
    mutate(r4 = gsub(".*?(\\d+).*", "\\1", r4)) %>%
    mutate(r5 = gsub(".*?(\\d+).*", "\\1", r5)) %>%
    mutate(r6 = gsub(".*?(\\d+).*", "\\1", r6)) %>%
    mutate(r7 = gsub(".*?(\\d+).*", "\\1", r7)) %>%
    select(-ucsfId)

  # bww, this is the regex, pattern<-"^(\\w+)\\s+/\\s+R:\\s+(\\w+)(->|\\s->)(\\w+)"
  # for that separate line.

  # The dataframe now looks like this.
  # (row),playersName,totalPoints,r1,r2,r3,r4,r5,r6,r7,playersState,preRating,postRating
  # 1,GARY HUA,6.0,39,21,18,14,7,12,4, ON,1794,1817
  # 21,DINH DANG BUI,4.0,43,1,47,3,40,39,6, ON,1563,1562
  # 39,JOEL R HENDON,3.0,1,54,40,16,44,21,24, MI,1436,1413

  ## perfectly workable, send it back.
  return(df_reformatted)

}
```

```
generate_rating_sequence <- function(player_row, df) {
  # this is the method that takes in Gary Hua, and looks up all his r1, r1.. scores.
  # we'll append these three columns to the right of the main dataframe
  # after it's all figured out.
  r_values <- c("r1", "r2", "r3", "r4", "r5", "r6", "r7")
  ratings <- sapply(player_row[r_values], function(r) as.numeric(df[r, "preRating"]))
  players_name <- player_row[["playersName"]]  # Extract player name
  avg_ratings <- round(mean(ratings, na.rm = TRUE))
  result <- data.frame(playersName = players_name,
                       opponentRatings = I(list(ratings)),
                       avgRatings = avg_ratings)
  colnames(result) <- c("playersName", "opponentRatings", "avgRatings")
  return(result)
}

## ----------------------------------------------------------------------------------------------------
## Main program
## ----------------------------------------------------------------------------------------------------

# load the raw data into the base dataframe
file_name <- "act.txt"
df <- file_to_df(file_name)

## create the right two columns of opponentRatings and avgRatings
three_columns_df <- data.frame(playersName = character(0),
                               opponentRatings = I(list()),
                               avgRatings = numeric(0))
result_list <- lapply(1:nrow(df), function(i) generate_rating_sequence(df[i, ], df))
three_columns_df <- do.call(rbind, result_list)

## left join them.
final_join <- left_join(df, three_columns_df, by = "playersName")

# > glimpse(final_join)
# Rows: 64
# Columns: 14
# $ playersName     <chr> "GARY HUA", "DAKSHESH DARURI", "ADITYA BAJAJ", "PATRIC…
# $ totalPoints     <chr> "6.0", "6.0", "6.0", "5.5", "5.5", "5.0", "5.0", "5.0"…
# $ r1              <chr> "39", "63", "8", "23", "45", "34", "57", "3", "25", "1…
# $ r2              <chr> "21", "58", "61", "28", "37", "29", "46", "32", "18", …
# $ r3              <chr> "18", "4", "25", "2", "12", "11", "13", "14", "59", "5…
# $ r4              <chr> "14", "17", "21", "26", "13", "35", "11", "9", "8", "3…
# $ r5              <chr> "7", "16", "11", "5", "4", "10", "1", "47", "26", "6",…
# $ r6              <chr> "12", "20", "13", "19", "14", "27", "9", "28", "7", "2…
# $ r7              <chr> "4", "7", "12", "1", "17", "21", "2", "19", "20", "18"…
# $ playersState    <chr> " ON", " MI", " MI", " MI", " MI", " OH", " MI", " MI"…
# $ preRating       <chr> "1794", "1553", "1384", "1716", "1655", "1686", "1649"…
# $ postRating      <chr> "1817", "1663", "1640", "1744", "1690", "1687", "1673"…
# $ opponentRatings <I<list>> 1436, 15...., 1175, 91...., 1641, 95...., 1363, 15…
# $ avgRatings      <dbl> 1605, 1469, 1564, 1574, 1501, 1519, 1372, 1468, 1523, …
# perfect. Now we delete extra columns, reorder, rename to spec, and write to file.

final_format <- final_join %>%
  select(-c("r1", "r2", "r3", "r4", "r5", "r6", "r7", "opponentRatings", "postRating")) %>%
  select("playersName", "playersState", "totalPoints", "preRating", "avgRatings") %>%
  rename(
    "Player’s Name" = playersName,
    "Player’s State" = playersState,
    "Total Number of Points" = totalPoints,
    "Player’s Pre-Rating" = preRating,
    "Average Pre Chess Rating of Opponents" = avgRatings
  )

file_name_csv <- str_replace(file_name, "txt", "csv")
write.csv(final_format, file = file_name_csv,
        append = FALSE, quote = FALSE, row.names = FALSE )

## ----------------------------------------------------------------------------------------------------
```

### First six records of output

Player’s Name,Player’s State,Total Number of Points,Player’s Pre-Rating,Average Pre Chess Rating of Opponents

GARY HUA, ON,6.0,1794,1605

DAKSHESH DARURI, MI,6.0,1553,1469

ADITYA BAJAJ, MI,6.0,1384,1564

PATRICK H SCHILLING, MI,5.5,1716,1574

HANSHI ZUO, MI,5.5,1655,1501

HANSEN SONG, OH,5.0,1686,1519