Project1

Alice Ding

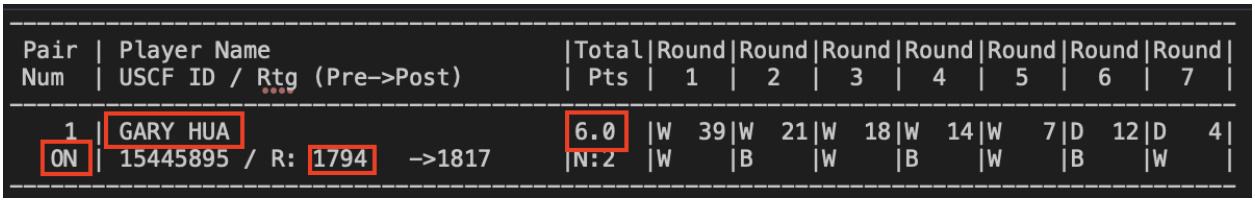
2023-02-13

Overview

The point of this project is to take a text file formatted in a certain way, mold the data into a .csv, and do some calculations in order to get certain aggregations of data for each row. The necessary fields in this .csv are:

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

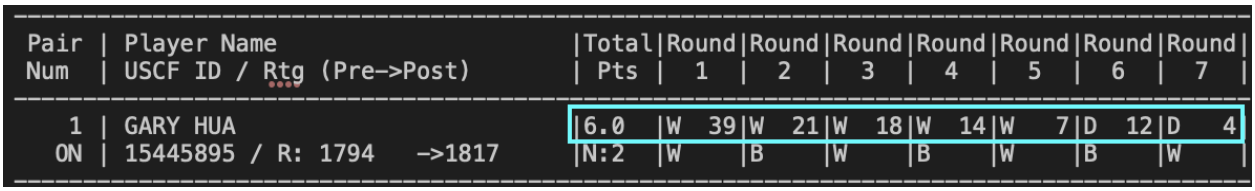
The first four are found easily in the text file:



Pair Num	Player Name / USCF ID / Rtg (Pre->Post)	Total Pts	Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Round 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

Figure 1: First row of data with fields for the .csv indicated with a red box.

The last requires calculating an average pre-rating based on the opponents faced which are outlined below:



Pair Num	Player Name / USCF ID / Rtg (Pre->Post)	Total Pts	Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Round 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

Figure 2: Cyan box denotes the player IDs that Gary Hua faced (ID 39 in Round 1, ID 21 in Round 2, etc.).

For Gary Hua, his output in the final .csv should read like this: Gary Hua, ON, 6.0, 1794, 1605

Data Clean-Up

Based on the data, we see that | is used as a separation indicator in the file.

```
# Get the data from tournamentinfo.txt
chess_df <- read.delim(
  file=
    "https://raw.githubusercontent.com/addsding/data607/main/project1/tournamentinfo.txt"
  , header = FALSE, sep="|"
  )
head(chess_df, 10)
```

```
##                                                                                                 V1
## 1  -----------------------------------------------------------------------------------------------
## 2                                                                                            Pair
## 3                                                                                            Num
## 4  -----------------------------------------------------------------------------------------------
## 5                                                                                             1
## 6                                                                                            ON
## 7  -----------------------------------------------------------------------------------------------
## 8                                                                                             2
## 9                                                                                            MI
## 10 -----------------------------------------------------------------------------------------------
##                                  V2    V3    V4    V5    V6    V7    V8    V9
## 1
## 2   Player Name                    Total Round Round Round Round Round Round
## 3   USCF ID / Rtg (Pre->Post)      Pts    1     2     3     4     5     6
## 4
## 5   GARY HUA                       6.0   W  39 W  21 W  18 W  14 W   7 D  12
## 6   15445895 / R: 1794   ->1817    N:2   W     B     W     B     W     B
## 7
## 8   DAKSHESH DARURI                6.0   W  63 W  58 L   4 W  17 W  16 W  20
## 9   14598900 / R: 1553   ->1663    N:2   B     W     B     W     B     W
## 10
##      V10 V11
## 1         NA
## 2  Round  NA
## 3    7    NA
## 4         NA
## 5  D   4  NA
## 6  W      NA
## 7         NA
## 8  W   7  NA
## 9  B      NA
## 10        NA
```

We can see here that rows 1-4 are not necessary as they were just the headers, and then every other row after that with a bunch of dashes (-) are just filler. We can remove these.

```
chess_df <- filter(chess_df, !grepl(pattern = "[-]+", V1)) |>
  filter(row_number() > 2)
head(chess_df)
```

```
##       V1                                V2    V3    V4    V5    V6    V7    V8
## 1     1   GARY HUA                        6.0   W  39 W  21 W  18 W  14 W   7
## 2    ON   15445895 / R: 1794   ->1817     N:2   W     B     W     B     W
```

```
## 3     2   DAKSHESH DARURI                  6.0   W  63 W  58 L   4 W  17 W  16
## 4    MI   14598900 / R: 1553   ->1663      N:2   B     W     B     W     B
## 5     3   ADITYA BAJAJ                     6.0   L   8 W  61 W  25 W  21 W  11
## 6    MI   14959604 / R: 1384   ->1640      N:2   W     B     W     B     W
##      V9   V10 V11
## 1 D  12 D   4  NA
## 2 B     W      NA
## 3 W  20 W   7  NA
## 4 W     B      NA
## 5 W  13 W  12  NA
## 6 B     W      NA
```

Now, the data holds all relevant information, just broken up into different rows. For example, for Gary, we can see that rows one and two hold information just for him. Row one has his player number, player name, total points, and all the players he faced. Row two has his pre-rating.

We can split this one data frame into two now based on every other row, then merge them after.

```
chess_df_2 = chess_df[seq(1, nrow(chess_df), 2), ]
chess_df_3 = chess_df[seq(0, nrow(chess_df), 2), ]
```

Now we have two data frames to clean up – let's start with `chess_df_2`

Part 1: Player Name, Total Points, Round Information

```
head(chess_df_2)
```

```
##        V1                              V2    V3    V4    V5    V6    V7    V8
## 1       1   GARY HUA                     6.0   W  39 W  21 W  18 W  14 W   7
## 3       2   DAKSHESH DARURI              6.0   W  63 W  58 L   4 W  17 W  16
## 5       3   ADITYA BAJAJ                 6.0   L   8 W  61 W  25 W  21 W  11
## 7       4   PATRICK H SCHILLING          5.5   W  23 D  28 W   2 W  26 D   5
## 9       5   HANSHI ZUO                   5.5   W  45 W  37 D  12 D  13 D   4
## 11      6   HANSEN SONG                  5.0   W  34 D  29 L  11 W  35 D  10
##       V9   V10 V11
## 1  D  12 D   4  NA
## 3  W  20 W   7  NA
## 5  W  13 W  12  NA
## 7  W  19 D   1  NA
## 9  W  14 W  17  NA
## 11 W  27 W  21  NA
```

We can start with renaming fields.

```
colnames(chess_df_2)[colnames(chess_df_2) == "V1"] = "player_id"
colnames(chess_df_2)[colnames(chess_df_2) == "V2"] = "player_name"
colnames(chess_df_2)[colnames(chess_df_2) == "V3"] = "total_points"
colnames(chess_df_2)[colnames(chess_df_2) == "V4"] = "round_1"
colnames(chess_df_2)[colnames(chess_df_2) == "V5"] = "round_2"
colnames(chess_df_2)[colnames(chess_df_2) == "V6"] = "round_3"
colnames(chess_df_2)[colnames(chess_df_2) == "V7"] = "round_4"
```

```
colnames(chess_df_2)[colnames(chess_df_2) == "V8"] = "round_5"
colnames(chess_df_2)[colnames(chess_df_2) == "V9"] = "round_6"
colnames(chess_df_2)[colnames(chess_df_2) == "V10"] = "round_7"
head(chess_df_2)
```

```
##    player_id                      player_name total_points round_1 round_2
## 1          1   GARY HUA                              6.0     W  39   W  21
## 3          2   DAKSHESH DARURI                       6.0     W  63   W  58
## 5          3   ADITYA BAJAJ                          6.0     L   8   W  61
## 7          4   PATRICK H SCHILLING                   5.5     W  23   D  28
## 9          5   HANSHI ZUO                            5.5     W  45   W  37
## 11         6   HANSEN SONG                           5.0     W  34   D  29
##    round_3 round_4 round_5 round_6 round_7 V11
## 1    W  18   W  14   W   7   D  12   D   4  NA
## 3    L   4   W  17   W  16   W  20   W   7  NA
## 5    W  25   W  21   W  11   W  13   W  12  NA
## 7    W   2   W  26   D   5   W  19   D   1  NA
## 9    D  12   D  13   D   4   W  14   W  17  NA
## 11   L  11   W  35   D  10   W  27   W  21  NA
```

Next, the round numbers should just have the `player_id` and not a W/L/D indication – time to fix that. There may be nulls for those that aren't W/L/D, such as those that are H/B.

```
chess_df_2$round_1 <- gsub("[WDL]\\s+", '', chess_df_2$round_1)
chess_df_2$round_2 <- gsub("[WDL]\\s+", '', chess_df_2$round_2)
chess_df_2$round_3 <- gsub("[WDL]\\s+", '', chess_df_2$round_3)
chess_df_2$round_4 <- gsub("[WDL]\\s+", '', chess_df_2$round_4)
chess_df_2$round_5 <- gsub("[WDL]\\s+", '', chess_df_2$round_5)
chess_df_2$round_6 <- gsub("[WDL]\\s+", '', chess_df_2$round_6)
chess_df_2$round_7 <- gsub("[WDL]\\s+", '', chess_df_2$round_7)

chess_df_2$round_1 <- gsub("[HB]\\s+", NA, chess_df_2$round_1)
chess_df_2$round_2 <- gsub("[HB]\\s+", NA, chess_df_2$round_2)
chess_df_2$round_3 <- gsub("[HB]\\s+", NA, chess_df_2$round_3)
chess_df_2$round_4 <- gsub("[HB]\\s+", NA, chess_df_2$round_4)
chess_df_2$round_5 <- gsub("[HB]\\s+", NA, chess_df_2$round_5)
chess_df_2$round_6 <- gsub("[HB]\\s+", NA, chess_df_2$round_6)
chess_df_2$round_7 <- gsub("[HB]\\s+", NA, chess_df_2$round_7)

head(chess_df_2)
```

```
##    player_id                      player_name total_points round_1 round_2
## 1          1   GARY HUA                              6.0      39       21
## 3          2   DAKSHESH DARURI                       6.0      63       58
## 5          3   ADITYA BAJAJ                          6.0       8       61
## 7          4   PATRICK H SCHILLING                   5.5      23       28
## 9          5   HANSHI ZUO                            5.5      45       37
## 11         6   HANSEN SONG                           5.0      34       29
##    round_3 round_4 round_5 round_6 round_7 V11
## 1       18      14       7      12       4  NA
## 3        4      17      16      20       7  NA
## 5       25      21      11      13      12  NA
```

```
## 7        2      26       5      19       1  NA
## 9       12      13       4      14      17  NA
## 11      11      35      10      27      21  NA
```

Next, time to reformat some of these fields to be numbers if they're numbers.

```
chess_df_2$player_id <- as.numeric(as.character(chess_df_2$player_id))
chess_df_2$total_points <- as.numeric(as.character(chess_df_2$total_points))
chess_df_2$round_1 <- as.numeric(as.character(chess_df_2$round_1))
chess_df_2$round_2 <- as.numeric(as.character(chess_df_2$round_2))
```

```
## Warning: NAs introduced by coercion
```

```
chess_df_2$round_3 <- as.numeric(as.character(chess_df_2$round_3))
```

```
## Warning: NAs introduced by coercion
```

```
chess_df_2$round_4 <- as.numeric(as.character(chess_df_2$round_4))
```

```
## Warning: NAs introduced by coercion
```

```
chess_df_2$round_5 <- as.numeric(as.character(chess_df_2$round_5))
```

```
## Warning: NAs introduced by coercion
```

```
chess_df_2$round_6 <- as.numeric(as.character(chess_df_2$round_6))
```

```
## Warning: NAs introduced by coercion
```

```
chess_df_2$round_7 <- as.numeric(as.character(chess_df_2$round_7))
```

```
## Warning: NAs introduced by coercion
```

```
glimpse(chess_df_2)
```

```
## Rows: 64
## Columns: 11
## $ player_id    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
## $ player_name  <chr> " GARY HUA                        ", " DAKSHESH DARURI   ~
## $ total_points <dbl> 6.0, 6.0, 6.0, 5.5, 5.5, 5.0, 5.0, 5.0, 5.0, 5.0, 4.5, 4.~
## $ round_1      <dbl> 39, 63, 8, 23, 45, 34, 57, 3, 25, 16, 38, 42, 36, 54, 19,~
## $ round_2      <dbl> 21, 58, 61, 28, 37, 29, 46, 32, 18, 19, 56, 33, 27, 44, 1~
## $ round_3      <dbl> 18, 4, 25, 2, 12, 11, 13, 14, 59, 55, 6, 5, 7, 8, 30, NA,~
## $ round_4      <dbl> 14, 17, 21, 26, 13, 35, 11, 9, 8, 31, 7, 38, 5, 1, 22, 39~
## $ round_5      <dbl> 7, 16, 11, 5, 4, 10, 1, 47, 26, 6, 3, NA, 33, 27, 54, 2, ~
## $ round_6      <dbl> 12, 20, 13, 19, 14, 27, 9, 28, 7, 25, 34, 1, 3, 5, 33, 36~
## $ round_7      <dbl> 4, 7, 12, 1, 17, 21, 2, 19, 20, 18, 26, 3, 32, 31, 38, NA~
## $ V11          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

We can see that `player_id`, `total_points`, and all the rounds are now numbers as they should be.

Now, let's try to clean up `chess_df_3`.

Part 2: State, Pre-Rating

```
head(chess_df_3)
```

```
##        V1                                 V2    V3    V4    V5    V6    V7    V8
## 2     ON   15445895 / R: 1794   ->1817      N:2   W     B     W     B     W
## 4     MI   14598900 / R: 1553   ->1663      N:2   B     W     B     W     B
## 6     MI   14959604 / R: 1384   ->1640      N:2   W     B     W     B     W
## 8     MI   12616049 / R: 1716   ->1744      N:2   W     B     W     B     W
## 10    MI   14601533 / R: 1655   ->1690      N:2   B     W     B     W     B
## 12    OH   15055204 / R: 1686   ->1687      N:3   W     B     W     B     B
##       V9   V10 V11
## 2  B     W      NA
## 4  W     B      NA
## 6  B     W      NA
## 8  B     B      NA
## 10 W     B      NA
## 12 W     B      NA
```

We can start with renaming the columns and also extracting pre-rating.

```
# rename the state column
colnames(chess_df_3)[colnames(chess_df_3) == "V1"] = "state"

# find "R: [number] in the string with the pre-rating
chess_df_3$match <- sub(pattern = ".*(R:\\s+([0-9]+)).*",
                replacement = "\\1", x = chess_df_3$V2)
# \\s+ because it seems like there are some cases where
# the data is formatted inconsistently
# (more than one space before the number)

# remove the "R: " part
chess_df_3$pre_rating <- gsub("R:\\s+", '', chess_df_3$match)

# make it numeric
chess_df_3$pre_rating <- as.numeric(as.character(chess_df_3$pre_rating))

head(select(chess_df_3, state, pre_rating))
```

```
##     state pre_rating
## 2     ON        1794
## 4     MI        1553
## 6     MI        1384
## 8     MI        1716
## 10    MI        1655
## 12    OH        1686
```

Next, I want to join the two data frames together. There's no key to join them on, but I know that the `player_id` column in `chess_df_2` (also known as pair num in the original data field) is an incrementing value starting at 1, so I'll be just numbering each row in `chess_df_3` and then joining both data frames on that new field since I know that the first row in `chess_df_2` matches the first row in `chess_df_3` and so on.

```
# add new player_id field to the second data frame
chess_df_3 <- chess_df_3 |>
  mutate(player_id = row_number())

# join them together
chess_df_all <- merge(chess_df_2, chess_df_3, by = "player_id")

# take only the relevant columns
chess_df_all<- select(
  chess_df_all
  , player_id
  , player_name
  , state
  , total_points
  , pre_rating
  , round_1
  , round_2
  , round_3
  , round_4
  , round_5
  , round_6
  , round_7
  )

head(chess_df_all)
```

```
##   player_id                      player_name  state total_points pre_rating
## 1         1  GARY HUA                          ON           6.0       1794
## 2         2  DAKSHESH DARURI                   MI           6.0       1553
## 3         3  ADITYA BAJAJ                      MI           6.0       1384
## 4         4  PATRICK H SCHILLING               MI           5.5       1716
## 5         5  HANSHI ZUO                        MI           5.5       1655
## 6         6  HANSEN SONG                       OH           5.0       1686
##   round_1 round_2 round_3 round_4 round_5 round_6 round_7
## 1      39      21      18      14       7      12       4
## 2      63      58       4      17      16      20       7
## 3       8      61      25      21      11      13      12
## 4      23      28       2      26       5      19       1
## 5      45      37      12      13       4      14      17
## 6      34      29      11      35      10      27      21
```

We have 4/5 of the desired fields now! It's time to calculate average pre-rating chess of each person's opponents now.

Part 3: Average Opponent Pre-Rating

To find the average opponent pre-rating, we can do this by pivoting the table by person with all the round numbers, appending the opponent's pre-rating, and then averaging all them.

```
# create the pivot
round_pivot <- chess_df_all |>
  pivot_longer(
```

```
    cols = starts_with("round_"),
    names_to = "round",
    values_to = "opponent_player_id",
    values_drop_na = TRUE
  )

# append the opponent's pre-rating
round_pivot <- left_join(
  round_pivot
  , select(chess_df_all, player_id, pre_rating)
  , by=c('opponent_player_id' = 'player_id')
  )

# average the opponent's pre-rating by the player
average_opponent_pre_rating <- round_pivot |>
  group_by(player_id, player_name) |>
  summarise(average_opponent_pre_rating = mean(pre_rating.y), .groups = 'drop') |>
  arrange(average_opponent_pre_rating)

head(average_opponent_pre_rating)
```

```
## # A tibble: 6 x 3
##   player_id player_name                          average_opponent_pre_rating
##       <dbl> <chr>                                                    <dbl>
## 1        43 " ROBERT GLEN VASEY                "                      1107.
## 2        30 " GEORGE AVERY JONES               "                      1144.
## 3        35 " JOSHUA DAVID LEE                 "                      1150.
## 4        42 " JARED GE                         "                      1150.
## 5        45 " DEREK YAN                        "                      1152
## 6        62 " ASHWIN BALAJI                    "                      1186
```

Now that we have the average opponent pre-rating, it's time to append that to the final data frame.

```
# join the data frame with average_opponent_pre_rating to our original one
chess_df_all <- merge(
  chess_df_all
  , select(average_opponent_pre_rating, player_id, average_opponent_pre_rating)
  , by = "player_id"
  )

# select only relevant columns for the final output
chess_df_final <- select(
  chess_df_all
  , player_name
  , state
  , total_points
  , pre_rating
  , average_opponent_pre_rating
  )

# write it all to a .csv
write.csv(chess_df_final, "tournament.csv", row.names=TRUE)
```

```
tournament_final <- read.csv("tournament.csv")
head(tournament_final)
```

```
##   X                      player_name  state total_points pre_rating
## 1 1  GARY HUA                           ON           6.0       1794
## 2 2  DAKSHESH DARURI                    MI           6.0       1553
## 3 3  ADITYA BAJAJ                       MI           6.0       1384
## 4 4  PATRICK H SCHILLING                MI           5.5       1716
## 5 5  HANSHI ZUO                         MI           5.5       1655
## 6 6  HANSEN SONG                        OH           5.0       1686
##   average_opponent_pre_rating
## 1                    1605.286
## 2                    1469.286
## 3                    1563.571
## 4                    1573.571
## 5                    1500.857
## 6                    1518.714
```

The .csv has been created and then pulled from – done!

Note: I see that `average_opponent_pre_rating` is an integer in the example given, however I kept it a double as I personally think it's helpful to see some decimals. To round it though, it would be as simple as doing `tournament_final$average_opponent_pre_rating <- round(tournament_final$average_opponent_pre_rating)`.

Conclusion

This text file was definitely a little tricky to parse, but at the very least it was consistently formatted so it was able to end up in a data frame without too much work. With this file now, one could potentially set up a simple model to predict how many points a player is going to get with their pre-rating and the average pre-ratings of their opponents. It could also just generally be used to predict whether the player would win a match or not, as well as predict perhaps what their after-tournament rating is. These cleaning scripts could also be used again with a file formatted the same way if next year's results come out in the same fashion too.