Project 1

2024-09-21

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
# Remove the header lines
data <- data[-(1:3)]

# Replace "->" with ">>"
data <- gsub("->", ">>", data)

# View the cleaned data
head(data)
```

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

```r
library(stringr)
# Format 1: Records starting with a number (player data)
data1 <- data[str_detect(data, "^\\s*\\d")]

# Format 2: Records starting with uppercase letters (e.g., state codes)
data2 <- data[str_detect(data, "^\\s*[A-Z]{2}")]



head(data1)
```

```
## [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|"
```

```r
head(data2)
```

```
## [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     |"
```

```
format2_df <- data.frame(
  player_state  = str_trim(substr(data2, 1,  6), side = "both"),
  uscf_id       = str_extract(substr(data2, 8,  40), "\\d+"),
  pre_rating    = as.numeric(str_extract(substr(data2, 8,  40), "(?<=R: )\\d+")),
  post_rating   = as.numeric(str_extract(substr(data2, 8,  40), "(?<=>>)\\d+")),
  stringsAsFactors = FALSE
)

format1_df <- data.frame(
  player_num = as.numeric(str_trim(substr(data1, 1, 6))),  # Player number
  player_name = str_trim(substr(data1, 8, 40)),             # Player name
  total_pts = as.numeric(str_trim(substr(data1, 42, 46))),  # Total points
  round1 = str_trim(substr(data1, 48, 52)),                 # Round 1 result
  round2 = str_trim(substr(data1, 54, 58)),                 # Round 2 result
  round3 = str_trim(substr(data1, 60, 64)),                 # Round 3 result
  round4 = str_trim(substr(data1, 66, 70)),                 # Round 4 result
  round5 = str_trim(substr(data1, 72, 76)),                 # Round 5 result
  round6 = str_trim(substr(data1, 78, 82)),                 # Round 6 result
  round7 = str_trim(substr(data1, 84, 88)),                 # Round 7 result
  stringsAsFactors = FALSE
)

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
result_df <- cbind(format1_df, format2_df)
```

```
# Initialize the vector for average opponent ratings
AvgOppPreChessRating <- numeric(nrow(result_df))

# Loop through each player's rounds
for (i in seq_len(nrow(result_df))) {
  # Extract round data for the current player
  Rounds <- str_extract_all(c(result_df$round1[i], result_df$round2[i], result_df$round3[i],
                              result_df$round4[i], result_df$round5[i], result_df$round6[i],
                              result_df$round7[i]), "\\d+")

  # Convert extracted rounds to numeric indices
  round_indices <- as.numeric(unlist(Rounds))

  # Check if there are valid rounds played
  if (length(round_indices) > 0) {
    # Calculate the average pre-rating for valid opponents
    total_rating <- sum(result_df$pre_rating[round_indices], na.rm = TRUE)
```

```r
    num_opponents <- sum(!is.na(result_df$pre_rating[round_indices]))

    # Calculate the average and store it
    if (num_opponents > 0) {
      AvgOppPreChessRating[i] <- round(total_rating / num_opponents, 0)
    } else {
      AvgOppPreChessRating[i] <- NA  # No valid opponents
    }
  } else {
    AvgOppPreChessRating[i] <- NA  # No rounds played
  }
}

# Store the average opponent ratings in the result_df dataframe
result_df$AvgOppPreChessRating <- AvgOppPreChessRating


# Final selection
Final_df <- result_df %>%
 select(player_name, player_state, total_pts, pre_rating, AvgOppPreChessRating)





# Write to CSV
write.csv(Final_df, "chess_tournament_resultsre_do.csv", row.names = FALSE)
```

“‘