DATA607_Project1_RJM

RJM

2019-12-20

```
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
```

```
library(stringr)
```

Regex Example 1 for extra credit in HW3

The code below is the original code with the results from the book:

```
strings <- c("12 Jun 2002", " 8 September 2004 ", "22-July-2009 ", "01 01 2001", "date",
             "02.06.2000",
             "xxx-yyy-zzzz", "$2,600")
dates <- "([0-9]{1,2})[- .]([a-zA-Z]+)[- .]([0-9]{4})"
str_detect(strings, dates)
```

```
## [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

The above results were not capturing the other two dates in the string (example from “Handling and processing strings in R”, p. 80) The following changes were made to capture other styles of dates:

```
dates <- "([0-9]{1,2})[- .]\\s?([[:alnum:]]+)\\s?[- .]([0-9]{4})"
str_detect(strings, dates)
```

```
## [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
```

Regex Example 2 for extra credit in HW3

An example to extract all the digits from a string:

```
x <- "this sentence has 5 words, and 0 animal names"
```

Note that the code only returns the first digit with this str_extract:

```r
y <- str_extract(x, "\\d")
y
```

```
## [1] "5"
```

Note that the code now returns all the digits with the function str_extract_all:

```r
y <- str_extract_all(x, "\\d")
y
```

```
## [[1]]
## [1] "5" "0"
```

Project 1

Notes: This project presented various challenges. The first one was to decide how to read the text files. I had to consult various resources and some previous solutions to make the conclusion that the required data is in a line format so the best format would be to read it as lines (below).

A quick look at the lines below will reveal that the first few lines are irrelevant and the meaningful data starts from line 5.

```r
chess_results <- readLines("/Users/rmirza/Documents/DATA607/tournamentinfo.txt")
```

```
## Warning in readLines("/Users/rmirza/Documents/DATA607/tournamentinfo.txt"):
## incomplete final line found on '/Users/rmirza/Documents/DATA607/
## tournamentinfo.txt'
```

```r
head(chess_results, 10)
```

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

The following line was to count the number of rows in the data which turned out be 196.

```r
n_rows <- length(chess_results)

n_rows
```

```
## [1] 196
```

Upon further consultation from the online sources it seemed reasonable to slice the data based on the information in the rows and start from the rows that have the meaningful information in them. The rows with that had the players' numbers, name, and other information started from row 5 and repeated as every third row after.

```r
row_names <- chess_results[seq(5, n_rows, 3)]

head(row_names, 10)
```

```
##  [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|"
##  [7] "    7 | GARY DEE SWATHELL               |5.0  |W  57|W  46|W  13|W  11|L   1|W   9|L   2|"
##  [8] "    8 | EZEKIEL HOUGHTON                |5.0  |W   3|W  32|L  14|L   9|W  47|W  28|W  19|"
##  [9] "    9 | STEFANO LEE                     |5.0  |W  25|L  18|W  59|W   8|W  26|L   7|W  20|"
## [10] "   10 | ANVIT RAO                       |5.0  |D  16|L  19|W  55|W  31|D   6|W  25|W  18|"
```

An interesting find was to keep checking for the length of the rows to ascertain the data matching.

```r
length(row_names)
```

```
## [1] 64
```

The rows with that had the players' states, and other information started from row 6 and repeated as every third row after.

```r
row_states <- chess_results[seq(6, n_rows, 3)]
head(row_states, 10)
```

```
##  [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    |"
##  [7] "   MI | 11146376 / R: 1649   ->1673     |N:3  |W    |B    |W    |B    |B    |W    |W    |"
##  [8] "   MI | 15142253 / R: 1641P17->1657P24  |N:3  |B    |W    |B    |W    |B    |W    |W    |"
##  [9] "   ON | 14954524 / R: 1411   ->1564     |N:2  |W    |B    |W    |B    |W    |B    |B    |"
## [10] "   MI | 14150362 / R: 1365   ->1544     |N:3  |W    |W    |B    |B    |W    |B    |W    |"
```

```r
length(row_states)
```

```
## [1] 64
```

To extract the players' numbers for matching in calculating average pre rating scores:

```
player_no <- as.integer(str_extract(row_names, "\\d+"))

head(player_no, 10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
length(player_no)
```

```
## [1] 64
```

To extract the players' names:

```
player_name <- as.character(str_trim (str_extract(row_names,
                                                  "(\\w+\\s){2,3}")))
head(player_name, 10)
```

```
##  [1] "GARY HUA"              "DAKSHESH DARURI"     "ADITYA BAJAJ"
##  [4] "PATRICK H SCHILLING" "HANSHI ZUO"           "HANSEN SONG"
##  [7] "GARY DEE SWATHELL"   "EZEKIEL HOUGHTON"    "STEFANO LEE"
## [10] "ANVIT RAO"
```

```
length(player_name)
```

```
## [1] 64
```

To extract the players' points:

```
player_points <- as.numeric(str_extract(row_names,
                                        "\\d+\\.\\d+"))
head(player_points, 10)
```

```
##  [1] 6.0 6.0 6.0 5.5 5.5 5.0 5.0 5.0 5.0 5.0
```

```
length(player_points)
```

```
## [1] 64
```

To extract the player numbers of the opponents:

```
player_opponents <- str_extract_all(str_extract_all(row_names,
                                                    "\\d+\\|"), "\\d+")
```

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

```
head(player_opponents, 5)
```

```
## [[1]]
## [1] "39" "21" "18" "14" "7"  "12" "4" 
## 
## [[2]]
## [1] "63" "58" "4"  "17" "16" "20" "7" 
## 
## [[3]]
## [1] "8"  "61" "25" "21" "11" "13" "12"
## 
## [[4]]
## [1] "23" "28" "2"  "26" "5"  "19" "1" 
## 
## [[5]]
## [1] "45" "37" "12" "13" "4"  "14" "17"
```

```
length(player_opponents)
```

```
## [1] 64
```

To extract the players' states:

```
player_state <- as.character(str_extract(row_states, "\\w+"))
head(player_state, 5)
```

```
## [1] "ON" "MI" "MI" "MI" "MI"
```

```
length(player_state)
```

```
## [1] 64
```

To extract the players' ratings before the tournament:

```
player_prerating <- as.integer(str_extract(str_extract
                                           (row_states,
                                             "[^\\d]\\d{3,4}[^\\d]"),
"\\d+"))
head(player_prerating, 5)
```

```
## [1] 1794 1553 1384 1716 1655
```

```
length(player_prerating)
```

```
## [1] 64
```

The following code was written to calculate the average pre chess ratings of the players as asked in the question. Since there were 64 players, it was important to ensure that there were 64 iterations for the for loop and it is saved as a rounded number.

```
avg_opp_scores <- length(player_no)
for (i in 1:(length(player_no))) {
avg_opp_scores[i] <-
round(mean(player_prerating[as.numeric(unlist
                                       (player_opponents
                                         [player_no[i]]))]))
}
```

A data frame was created to capture the results:

```
df <- data.frame(player_name,player_state,
player_points,player_prerating,avg_opp_scores)
head(df, 5)
```

```
##           player_name player_state player_points player_prerating
## 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_opp_scores
## 1           1605
## 2           1469
## 3           1564
## 4           1574
## 5           1501
```

```
length(df)
```

```
## [1] 5
```

Updated column names to match the requirements of the project:

```
colnames(df) <- c("Player's Name","Player's State",
"Total Number of Points", "Player's Pre-Rating",
"Average Pre Chess Rating of Opponents")
```

```
head(df, 5)
```

```
##         Player's Name Player's State Total Number of Points
## 1            GARY HUA             ON                    6.0
## 2     DAKSHESH DARURI             MI                    6.0
## 3        ADITYA BAJAJ             MI                    6.0
## 4 PATRICK H SCHILLING             MI                    5.5
## 5          HANSHI ZUO             MI                    5.5
##   Player's Pre-Rating Average Pre Chess Rating of Opponents
## 1                1794                                  1605
## 2                1553                                  1469
## 3                1384                                  1564
## 4                1716                                  1574
## 5                1655                                  1501
```

Finally, a csv file was created as per the requirements:

```
write.csv(df, "Chess_results_Project1.csv", row.names=FALSE)
```