Project 1 Data 607

Coco Donovan

2023-02-13

Necessary Packages

```
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)
library(tidyr)
```

Loading Raw Txt File

```
chess <- 'https://raw.githubusercontent.com/cocodono/Project-1-Data-607/main/Chess%20Players'

chess <- readLines(chess)

chess <- data.frame(chess)
```

Removing the lines of dashes

```
chess <- data.frame(chess[chess != '-----------------------------------------------------------------
```

Even and Odd Rows

```
row_odd <- seq_len(nrow(chess)) %% 2

odd_chess <- data.frame(chess[row_odd == 1,])[-1,]
even_chess <- data.frame(chess[row_odd == 0,])[-1,]
```

Making a dataframe out of odd_chess

```
odd_cols <- c('Player_Number','Player_Name','Total','Round_1_opponent','Round_2_opponent','Round_3_oppon

separated_odd <- odd_chess %>%
  as.data.frame() %>%
  separate(1, into = odd_cols, sep = "\\|")
```

```
## Warning: Expected 10 pieces. Additional pieces discarded in 64 rows [1, 2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
row_id <- c(1:nrow(separated_odd))
separated_odd$row_id <- row_id
```

Making a dataframe out of even_chess

```
even_cols <- c('State','USCF_ID / Rtg (Pre->Post)','N','Round_1_result','Round_2_result','Round_3_resul

separated_even <- even_chess %>%
  as.data.frame() %>%
  separate(1, into = even_cols, sep = "\\|")
```

```
## Warning: Expected 10 pieces. Additional pieces discarded in 64 rows [1, 2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
row_id <- c(1:nrow(separated_even))
separated_even$row_id <- row_id
```

Merging even and odd

```
full_chess <- merge(separated_odd, separated_even, by = 'row_id') %>%
  subset(select = -c(row_id)) %>%
  separate('USCF_ID / Rtg (Pre->Post)', c('USCF_ID', 'Rtg_Pre_Post'), '\\/') %>%
  separate('Rtg_Pre_Post', c('pre_rating','post_rating'), '->')
```

Making necessary columns numbers

```
full_chess$pre_rating <- strtoi(str_extract(full_chess$pre_rating, '[0-9]+'))
full_chess$post_rating <- strtoi(str_extract(full_chess$post_rating, '[0-9]+'))

for (item in colnames(full_chess)[grepl("opponent", colnames(full_chess))]) {
   full_chess[[item]] <- full_chess$pre_rating[strtoi(str_extract(full_chess[[item]], '[0-9]+'))]
}
```

Average Opponent Rating

```
full_chess$avg_opp_rate <- NA

for (i in 1:nrow(full_chess)) {
  full_chess$avg_opp_rate[i] <- round(rowMeans(full_chess[i,grep('Round_1_opponent',colnames(full_chess
}
```

Writing the csv

```
filtered_chess <- full_chess %>%
  select(Player_Name, State, Total, pre_rating, avg_opp_rate)

write.csv(filtered_chess, 'chess_stats.csv')
```