Project 1

Matthew Roland

2023-09-22

Loading the txt File

The first step is to read the txt file into R as a table, and save it

```
chess <- read.table("https://raw.githubusercontent.com/Mattr5541/DATA-607/main/Project%201/Chess.txt",
                    header = T, sep = "|", skip = 1, fill = T, quote = "")

glimpse(chess)
```

```
## Rows: 194
## Columns: 11
## $ Pair        <chr> " Num  ", "-----------------------------------------------~
## $ Player.Name <chr> " USCF ID / Rtg (Pre->Post)       ", "", " GARY HUA       ~
## $ Total       <chr> " Pts ", "", "6.0  ", "N:2  ", "", "6.0  ", "N:2  ", "", "~
## $ Round       <chr> "  1  ", "", "W  39", "W    ", "", "W  63", "B    ", "", "~
## $ Round.1     <chr> "  2  ", "", "W  21", "B    ", "", "W  58", "W    ", "", "~
## $ Round.2     <chr> "  3  ", "", "W  18", "W    ", "", "L   4", "B    ", "", "~
## $ Round.3     <chr> "  4  ", "", "W  14", "B    ", "", "W  17", "W    ", "", "~
## $ Round.4     <chr> "  5  ", "", "W   7", "W    ", "", "W  16", "B    ", "", "~
## $ Round.5     <chr> "  6  ", "", "D  12", "B    ", "", "W  20", "W    ", "", "~
## $ Round.6     <chr> "  7  ", "", "D   4", "W    ", "", "W   7", "B    ", "", "~
## $ X           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

Cleaning the dataframe

Since the resulting dataframe is less than interpretable, the next step will be to clean the dataframe by removing any extraneous lines, characters, and columns. I started by removing all hyphens, cutting out some empty columns, and then by merging columns and rows where appropriate. This was accomplished by making a grouping variable called "merge" that groups every two together; I then created a new dataframe called chess_clean where all instances of "merge" that matched would be grouped into one row, and then, of course, I dropped the merge variable. Finally, I cleaned up any trailing spaces that were present in the observations

```
chess <- data_frame(chess)
```

```
## Warning: ‘data_frame()‘ was deprecated in tibble 1.1.0.
## i Please use ‘tibble()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
chess <- subset(chess, Pair != '-----------------------------------------------------------------------

chess <- chess[-1,]

chess <- chess[-11]


chess$merge <- rep(1:(nrow(chess) / 2), each = 2)
chess_clean <- chess %>% group_by(merge) %>% summarize_all(~paste(., collapse = "")) %>%
  ungroup() %>% select(-merge)

chess_clean <- chess_clean %>% mutate_all(trimws)

chess_clean$Pair <- trimws(chess_clean$Pair)

chess_clean$Player.Name <- trimws(chess_clean$Player.Name)
```

Separating Variables

I then separated the now-cleaned chess dataset where appropriate by using regular expressions and dplyr's separate function. This took quite a bit of trial and error to properly parse out the correct values, primarily due to the many uneven spaces throughout the observations

```r
chess_sep <- chess_clean %>% separate(Pair, c('Pair', 'Player_State'))

chess_sep <- chess_sep %>% separate(Player.Name, c('Player.Name', 'Rating'), sep = " / R: ")

chess_sep$Player.Name <- gsub("[[0-9]+", "", chess_sep$Player.Name)

chess_sep$Player.Name <- trimws(chess_sep$Player.Name)

chess_sep$Rating <- gsub("^[P].+|>.+", "", chess_sep$Rating)
chess_sep$Rating <- gsub("P\\d*|[- ]", "", chess_sep$Rating)

chess_sep$Total <- gsub("N:\\d+", "", chess_sep$Total)
```

Converting to long format

I then converted the dataframe to a long format in order to more easily match the opponents' ratings with each player (essentially, I wanted to convert the rounds into a grouping variable so I could match the opponents with the "Rating" column, and eventually, the "Pair" column). After that, I created a new dataframe consisting of the Pair IDs, and renamed "Pair" to "Opponent" and "Rating" to "Opponent_Rating." I then merged this into the chess dataframe, in order to match each "player with their corresponding opponents' ratings

```r
chess_long <- chess_sep %>% gather("Round", "Opponent", 6:12)

chess_long$Opponent <- gsub("[A-Za-z]", "", chess_long$Opponent)

chess_long$Opponent <- as.numeric(chess_long$Opponent)
```

```
Ratings_sep <- chess_long %>% select(Opponent_Rating = Rating, Opponent = Pair)


chess_long <- chess_long %>% arrange(Opponent)

Ratings_sep <- Ratings_sep %>% arrange(Opponent)

chess_merge <- merge(chess_long, Ratings_sep, by = "Opponent") %>% distinct()
##Just to fix the overall layout of the players
chess_merge <- chess_merge %>% arrange(Pair)
```

Setting to Wide & Calculating Averages

Finally, I set the dataframe back to a wide format, calculated the row averages for every round, in order to determine opponent averages, and performed some last-minute cleaning procedures (removing unnecessary columns/renaming columns)

```
chess_wide <- chess_merge %>% pivot_wider(id_cols = c(Pair, Player.Name, Player_State, Total, Rating),

chess_wide$Pair <- as.numeric(chess_wide$Pair)

Rounds <- chess_wide[,c(6:12)]
Rounds <- Rounds %>% mutate_at(1:7, as.numeric)

Rounds$Opponent_Average <- rowMeans(Rounds, na.rm = T)
Rounds$Opponent_Average <- round(Rounds$Opponent_Average, digits = 0)

chess_wide$Opponent_Average <- Rounds$Opponent_Average

chess_wide <- chess_wide %>% select(-c(1, 6:12))

chess_wide$Total <- as.numeric(chess_wide$Total)
chess_wide <- chess_wide %>% rename("Player_Name" = "Player.Name")

kable(chess_wide)
```

Player_Name	Player_State	Total	Rating	Opponent_Average
GARY HUA	ON	6.0	1794	1605
ANVIT RAO	MI	5.0	1365	1554
CAMERON WILLIAM MC LEMAN	MI	4.5	1712	1468
KENNETH J TACK	MI	4.5	1663	1506
TORRANCE HENRY JR	MI	4.5	1666	1498
BRADLEY SHAW	MI	4.5	1610	1515
ZACHARY JAMES HOUGHTON	MI	4.5	1220	1484
MIKE NIKITIN	MI	4.0	1604	1386
RONALD GRZEGORCZYK	MI	4.0	1629	1499
DAVID SUNDEEN	MI	4.0	1600	1480
DIPANKAR ROY	MI	4.0	1564	1426
DAKSHESH DARURI	MI	6.0	1553	1469
JASON ZHENG	MI	4.0	1595	1411
DINH DANG BUI	ON	4.0	1563	1470

Player_Name	Player_State	Total	Rating	Opponent_Average
EUGENE L MCCLURE	MI	4.0	1555	1300
ALAN BUI	ON	4.0	1363	1214
MICHAEL R ALDRICH	MI	4.0	1229	1357
LOREN SCHWIEBERT	MI	3.5	1745	1363
MAX ZHU	ON	3.5	1579	1507
GAURAV GIDWANI	MI	3.5	1552	1222
SOFIA ADINA STANESCU-BELLU	MI	3.5	1507	1522
CHIEDOZIE OKORIE	MI	3.5	1602	1314
ADITYA BAJAJ	MI	6.0	1384	1564
GEORGE AVERY JONES	ON	3.5	1522	1144
RISHI SHETTY	MI	3.5	1494	1260
JOSHUA PHILIP MATHEWS	ON	3.5	1441	1379
JADE GE	MI	3.5	1449	1277
MICHAEL JEFFERY THOMAS	MI	3.5	1399	1375
JOSHUA DAVID LEE	MI	3.5	1438	1150
SIDDHARTH JHA	MI	3.5	1355	1388
AMIYATOSH PWNANANDAM	MI	3.5	980	1385
BRIAN LIU	MI	3.0	1423	1539
JOEL R HENDON	MI	3.0	1436	1430
PATRICK H SCHILLING	MI	5.5	1716	1574
FOREST ZHANG	MI	3.0	1348	1391
KYLE WILLIAM MURPHY	MI	3.0	1403	1248
JARED GE	MI	3.0	1332	1150
ROBERT GLEN VASEY	MI	3.0	1283	1107
JUSTIN D SCHILLING	MI	3.0	1199	1327
DEREK YAN	MI	3.0	1242	1152
JACOB ALEXANDER LAVALLEY	MI	3.0	377	1358
ERIC WRIGHT	MI	2.5	1362	1392
DANIEL KHAIN	MI	2.5	1382	1356
MICHAEL J MARTIN	MI	2.5	1291	1286
HANSHI ZUO	MI	5.5	1655	1501
SHIVAM JHA	MI	2.5	1056	1296
TEJAS AYYAGARI	MI	2.5	1011	1356
ETHAN GUO	MI	2.5	935	1495
JOSE C YBARRA	MI	2.0	1393	1345
LARRY HODGE	MI	2.0	1270	1206
ALEX KONG	MI	2.0	1186	1406
MARISA RICCI	MI	2.0	1153	1414
MICHAEL LU	MI	2.0	1092	1363
VIRAJ MOHILE	MI	2.0	917	1391
SEAN M MC CORMICK	MI	2.0	853	1319
HANSEN SONG	OH	5.0	1686	1519
JULIA SHEN	MI	1.5	967	1330
JEZZEL FARKAS	ON	1.5	955	1327
ASHWIN BALAJI	MI	1.0	1530	1186
THOMAS JOSEPH HOSMER	MI	1.0	1175	1350
BEN LI	MI	1.0	1163	1263
GARY DEE SWATHELL	MI	5.0	1649	1372
EZEKIEL HOUGHTON	MI	5.0	1641	1468
STEFANO LEE	ON	5.0	1411	1523

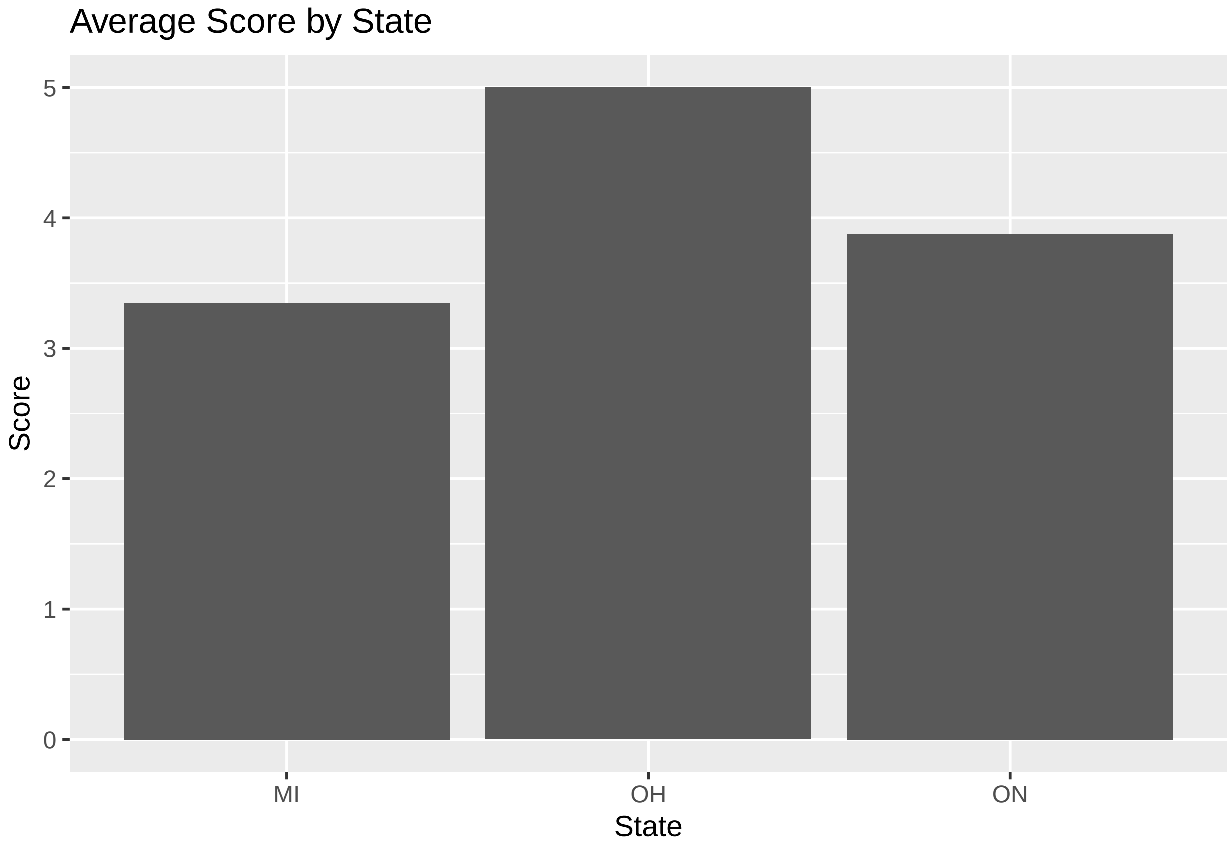
Visualization

Now that everything is set up, it's time to make a little visual demonstration for some of the values

```
chess_avg <- chess_wide %>% select(Player_State, Total)

Avg_by_State <- chess_avg %>%
  group_by(Player_State) %>%
  summarise_at(vars(Total), list(Total = mean))

ggplot(Avg_by_State, aes(x = Player_State, y = Total)) +
  geom_bar(stat = "identity") + labs(title = "Average Score by State", x = "State", y = "Score")
```



The graph above demonstrates that Ohio had the highest average score when compared to ON (which I am assuming is Ontario?) and Michigan. However, that is not entirely meaningful, since Ohio only had one player. The more meaningful comparison would be that Ontario(?) had a higher average player score than Michigan

##Saving as a CSV

The final step would be to save the cleaned and modified dataframe as a CSV file

```
chess_csv <- chess_wide

write.csv(chess_csv, "chess.csv", row.names = F)
```