Chess rankings data analysis

Henry Owens

assignment

Produce csv file with the following rows: Player's Name, Player's State, Total Number of Points, Player's Pre-Rating, and Average Pre Chess Rating of Opponents

Approach

First I will put the messy table into a single row per player and then extract and pivot some new tables.

```
url <- "https://raw.githubusercontent.com/hankowens/CUNY-MSDS/main/607%20Project%201/tournamentinfo.txt
# make sure you get the raw file!

# Read in lines
# alternative: dflines <- data.frame(readLines(url))
df <- data.frame(read.delim(url, header = FALSE, stringsAsFactors = FALSE))
head(df)
```

```
##                                                                                            V1
## 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    |
```

```
# testing out adding the sep arg, but i think this makes it harder to get everything into one row
df_sep <- data.frame(read.delim(url, header = FALSE, stringsAsFactors = FALSE, sep = "|"))
head(df)
```

```
##                                                                                            V1
## 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    |
```

```
dashrow <- df[1,1]
df <- filter(df, V1 != dashrow)
```

```
head(df)
```

```
##                                                                                           V1
## 1  Pair | Player Name                     |Total|Round|Round|Round|Round|Round|Round|Round|
## 2  Num  | USCF ID / Rtg (Pre->Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  |
## 3      1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|
## 4     ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |
## 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    |
```

```
# this says make data frame with two columns:
#       (1) df column one from row 1 to n-1, and
#       (2) df column one from row 2 to n

df2 <- data.frame(cbind(df[1:(nrow(df)-1),], df[2:nrow(df),]))
```

```
# easy peasy:
# https://stackoverflow.com/questions/39106128/delete-every-evenuneven-row-from-a-dataset

my_rows <- seq(1, nrow(df2), 2)
df3 <- df2[my_rows,]
```

```
df4 <- data.frame(paste(df3$X1, df3$X2))
df4<- df4 %>%
  rename("col1" = "paste.df3.X1..df3.X2.")
```

```
# it worked! this might throw an error because it was expecting (an empty) 21st column

col_names <- c("player_num", "player_name", "total_points", "r1", "r2", "r3","r4", "r5", "r6", "r7", "s
df4<- df4 %>%
  separate(col1, into = col_names, sep = "\\|")
```

```
df4 <- df4[2:nrow(df4),1:13]
```

```
# I dont think this needs to be as.numeric (but just add %>% as.numeric() at the end if so)
df4$player_num <- df4$player_num %>% str_trim
df4$player_name <- df4$player_name %>% str_trim
df4$r1 <- df4$r1 %>% str_extract("\\d+")
df4$r2 <- df4$r2 %>% str_extract("\\d+")
df4$r3 <- df4$r3 %>% str_extract("\\d+")
df4$r4 <- df4$r4 %>% str_extract("\\d+")
df4$r5 <- df4$r5 %>% str_extract("\\d+")
df4$r6 <- df4$r6 %>% str_extract("\\d+")
df4$r7 <- df4$r7 %>% str_extract("\\d+")
```

```
# messy column:
df5 <- df4 %>% separate(uscfetc, into = c("uscf", "ratings"), sep = "\\s/\\s*R:\\s") %>%
  separate(ratings, into = c("preratings", "postratings"), sep = "->")

# deal with the "P" in some of the ratings, assuming that what follows P is irrelevant
# first regex arg is everything up to and excluding "P", OR 3 digits or more
```

```r
# gets me what i wanted
df5$preratings <- str_extract(df5$preratings, pattern = "(.+(?=P))|\\d\\d\\d+")
df5$postratings <- str_extract(df5$postratings, pattern = "(.+(?=P))|\\d\\d\\d+")

df5$preratings <- as.numeric(df5$preratings)
df5$postratings <- as.numeric(df5$postratings)
```

```r
df_player_ratings <- data.frame(df5[1], df5[13])
# looks good
```

```r
# taking player_num and round-wise opponent rating (seven columns of matches)
df_results <- data.frame(df5[1], df5[4:10])
# looks good
```

```r
# first column input is default first column, player_num here
# useful explanation of pivots/tidyr here:
# https://mgimond.github.io/ES218/Week03b.html
df_results_pivot <- pivot_longer(df_results, cols = c(r1, r2, r3, r4, r5, r6, r7), names_to = "round",
                           values_to = "opponent")
# looks good
```

```r
# this is working except for where the ratings num had soething like this: 1403P5
df_results_joined <- left_join(df_results_pivot, df_player_ratings, by = c("opponent" = "player_num"))
```

```r
# create table of player_num and mean opp_rating with summarize groupby
df_opp_rating <- df_results_joined %>%
  group_by(player_num) %>%
  summarize(mean_opp_rating = mean(preratings, na.rm = TRUE))
```

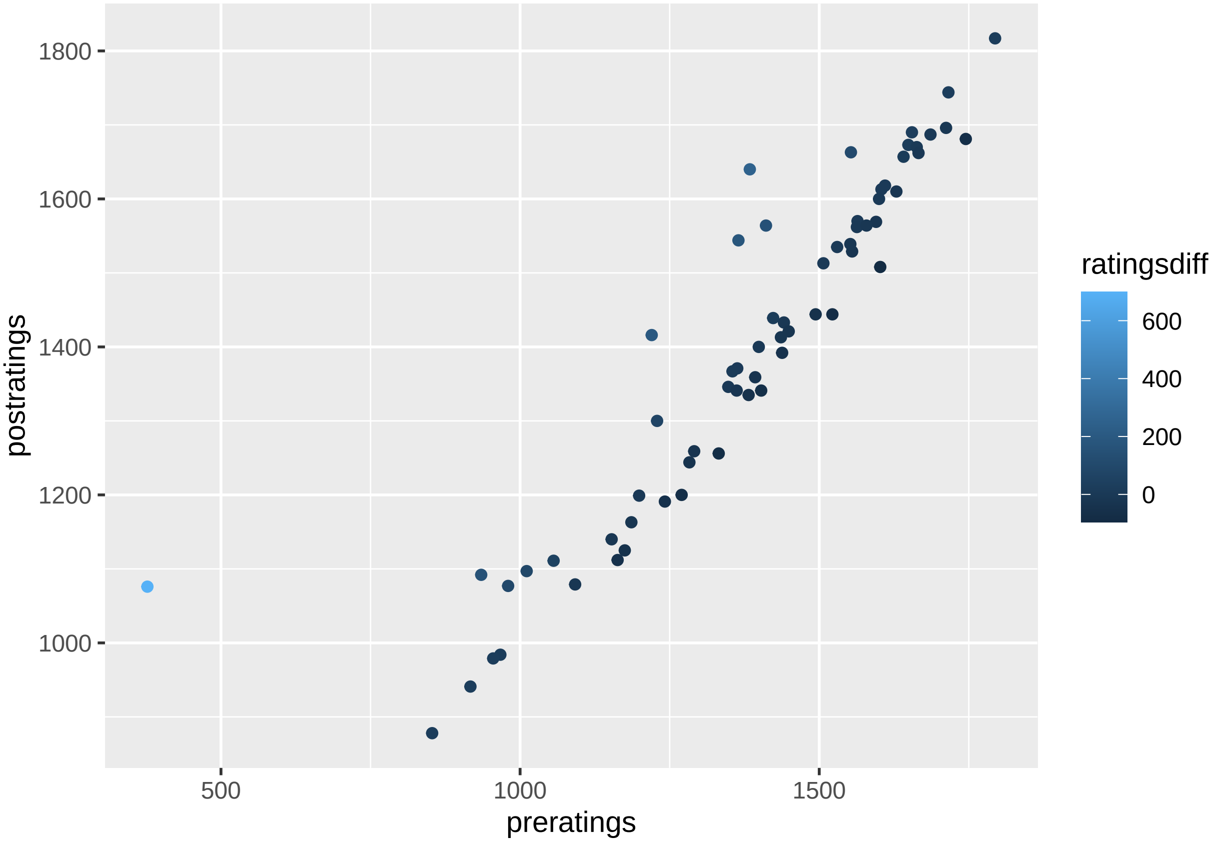
```r
#select the columns from larger table:
df_final <- data.frame(df5[1:3], df5[11], df5[13:14])
# join average opponent rating:
df_final <- left_join(df_final, df_opp_rating)
```

```
## Joining, by = "player_num"
```

```r
# add column for diff btw pre and post ratings
df_final$ratingsdiff <- df_final$postratings - df_final$preratings
```

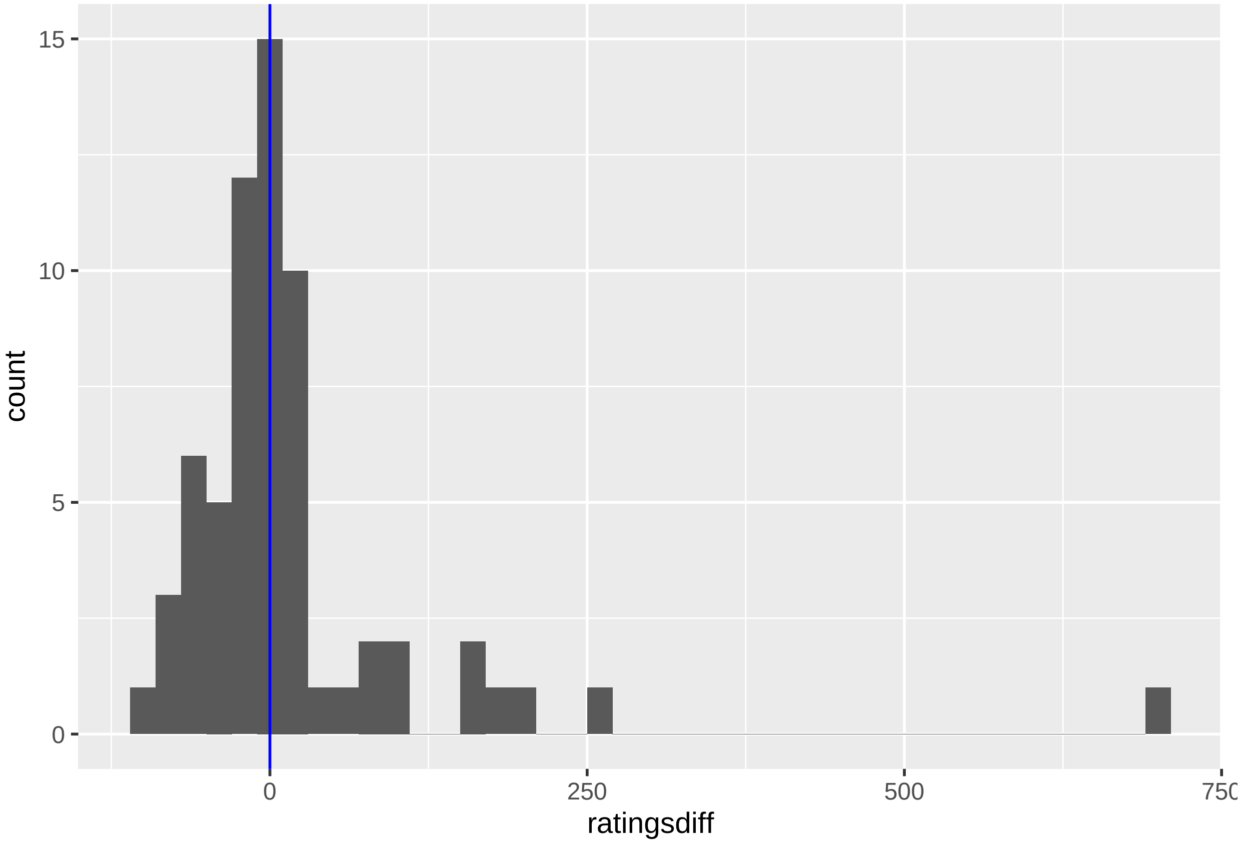
- There is a strong relationship between pre-rating and post-rating

```r
library(ggplot2)
ggplot(df_final, aes(preratings, postratings,colour = ratingsdiff)) +
  geom_point()
```



* The distribution of the difference between post and pre ratings is interesting. The median difference is zero, which I suspect is a function of the ratings algorithm. Most players dropped in the ratings, with a few making large gains over 100 and one about 700.

```
m <- median(df_final$ratingsdiff)
ggplot(df_final, aes(ratingsdiff)) +
  geom_histogram(binwidth = 20) +
  geom_vline(xintercept = m, color = "blue")
```



* Jacob Alexander Lavalley came in with a rating of 377 and left with 1076 after winning 3 points. Of the top six ratings gainers, only Jacob faced opponents with a mean rating below the average, but his opponents were only slightly below the mean (1379 vs. 1358). Of the top six ratings losers, four entered the tournament with above average ratings, and the mean opponent rating for all of them was below average. So I suspect that opponent rating has a big impact on the rating change.

```
head(
  df_final[
    order(
      df_final$ratingsdiff,
      decreasing=TRUE),
    ]
  )
```

```
##    player_num              player_name total_points   state preratings
## 46         46 JACOB ALEXANDER LAVALLEY          3.0      MI        377
## 3           3             ADITYA BAJAJ          6.0      MI       1384
## 15         15   ZACHARY JAMES HOUGHTON          4.5      MI       1220
## 10         10                ANVIT RAO          5.0      MI       1365
## 52         52                ETHAN GUO          2.5      MI        935
## 9           9             STEFANO LEE          5.0      ON       1411
##    postratings mean_opp_rating ratingsdiff
## 46        1076        1357.714         699
## 3         1640        1563.571         256
## 15        1416        1483.857         196
## 10        1544        1554.143         179
## 52        1092        1494.571         157
## 9         1564        1523.143         153
```

```
mean(df_final$preratings)
```

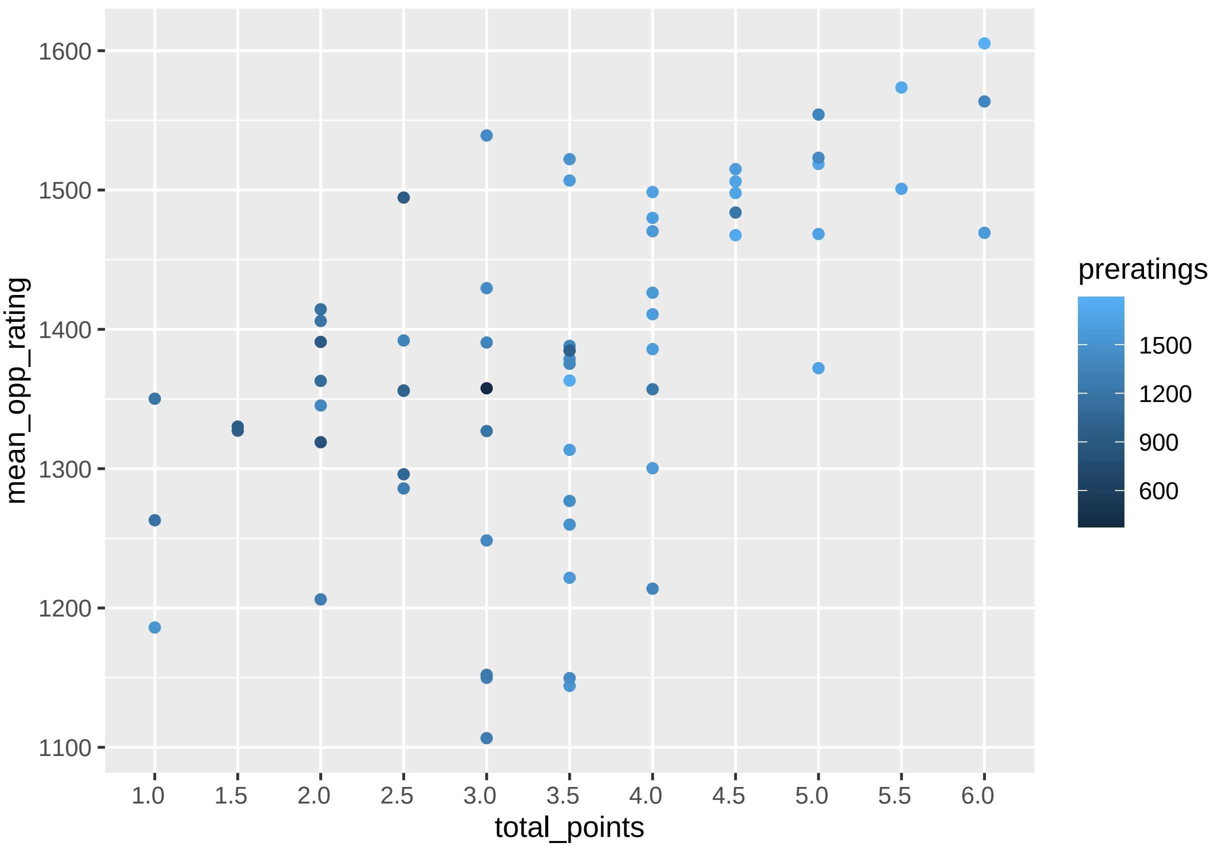
```
## [1] 1378.5
```

```
head(
  df_final[
    order(
      df_final$ratingsdiff,
      decreasing=FALSE),
    ]
  )
```

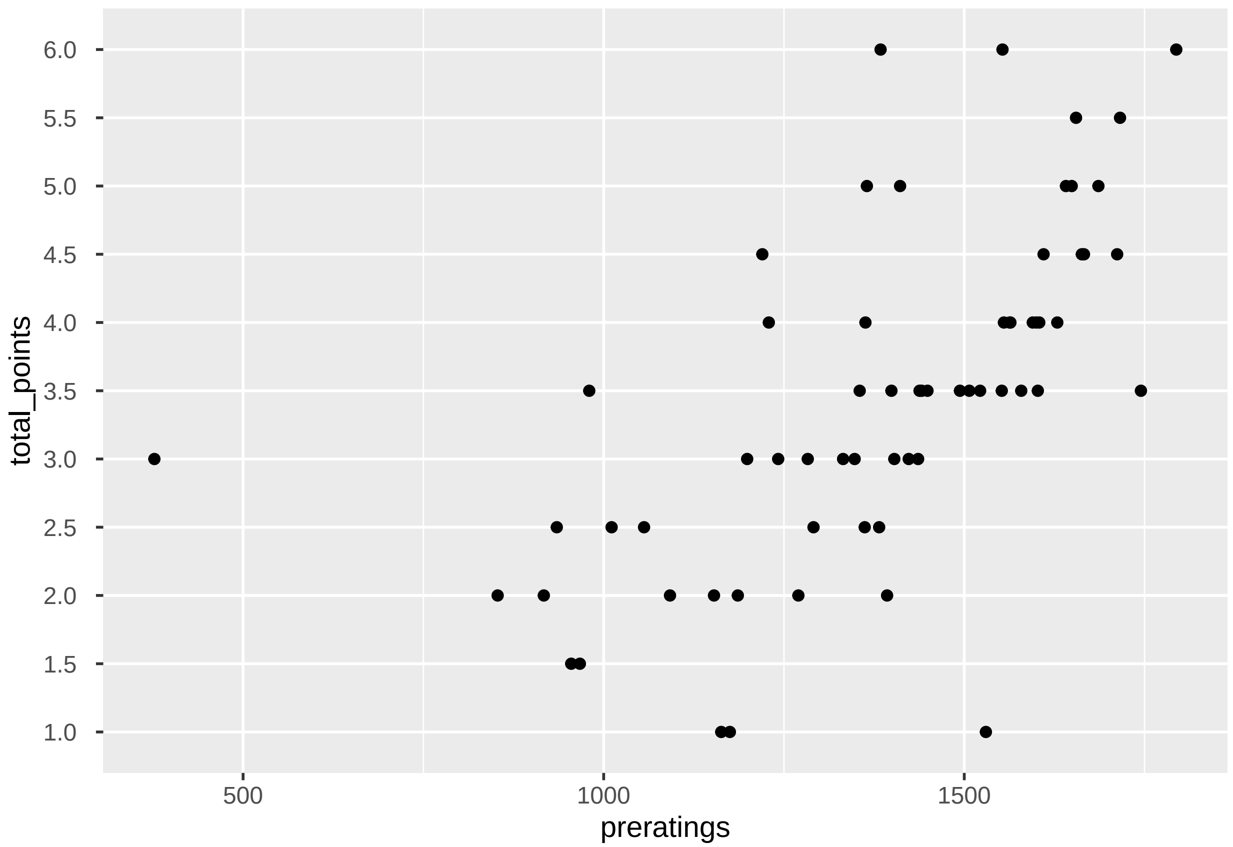
```
##    player_num         player_name total_points   state preratings postratings
## 29         29    CHIEDOZIE OKORIE          3.5      MI       1602        1508
## 30         30   GEORGE AVERY JONES         3.5      ON       1522        1444
## 42         42            JARED GE          3.0      MI       1332        1256
## 54         54         LARRY HODGE          2.0      MI       1270        1200
## 25         25    LOREN SCHWIEBERT          3.5      MI       1745        1681
## 41         41 KYLE WILLIAM MURPHY          3.0      MI       1403        1341
##    mean_opp_rating ratingsdiff
## 29        1313.500         -94
## 30        1144.143         -78
## 42        1149.857         -76
## 54        1206.167         -70
## 25        1363.286         -64
## 41        1248.500         -62
```

- Plotting the total points with mean opponent rating, we can see that players who faced tougher opponents won more points. That is obviously a feature of most any tournament as you face tougher opponents as you win and progress. Eyeballing the chart, I can see a tendency for higher pre-ratings (color) for players who won more points.

```
ggplot(df_final, aes(total_points, mean_opp_rating, color = preratings)) +
  geom_point()
```



```
ggplot(df_final, aes(preratings, total_points)) +
  geom_point()
```



```r
write.csv(df_final, "~/chess_df.csv", row.names = TRUE)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.