Project 1 - DATA607

Rachel Greenlee

9/14/2020

Step 1 - Import the text file

I access the .txt file on my github repo. I skip the first 4 lines as it's dashes and headers. In order to to remove the full lines that comprise of dashes every 3 rows, I can write the pattern True, True, False for it to take the first and second rows, skip the third, and repeat.

Now we read that into a table, with no header as we stripped it in line 1, and set the delimiter to a vertical bar/pipe. We tell it to fill in missing values in case the rows have unequal length for any player.

```
raw.tourney <- read_lines("https://raw.githubusercontent.com/rachel-greenlee/data607_proj1/master/tourna
                  skip = 4)[c(TRUE, TRUE, FALSE)]
raw.tourney <- read.table(textConnection(raw.tourney), header = FALSE, sep="|", fill = TRUE)
glimpse(raw.tourney)
```

```
## Rows: 128
## Columns: 11
## $ V1  <chr> "    1 ", "   ON ", "    2 ", "   MI ", "    3 ", "   MI ", "  ...
## $ V2  <chr> " GARY HUA                        ", " 15445895 / R: 1794   ->1...
## $ V3  <chr> "6.0  ", "N:2  ", "6.0  ", "N:2  ", "6.0  ", "N:2  ", "5.5  ", ...
## $ V4  <chr> "W  39", "W    ", "W  63", "B    ", "L   8", "W    ", "W  23", ...
## $ V5  <chr> "W  21", "B    ", "W  58", "W    ", "W  61", "B    ", "D  28", ...
## $ V6  <chr> "W  18", "W    ", "L   4", "B    ", "W  25", "W    ", "W   2", ...
## $ V7  <chr> "W  14", "B    ", "W  17", "W    ", "W  21", "B    ", "W  26", ...
## $ V8  <chr> "W   7", "W    ", "W  16", "B    ", "W  11", "W    ", "D   5", ...
## $ V9  <chr> "D  12", "B    ", "W  20", "W    ", "W  13", "B    ", "W  19", ...
## $ V10 <chr> "D   4", "W    ", "W   7", "B    ", "W  12", "W    ", "D   1", ...
## $ V11 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
```

Step 2 - Create data frame & clean it

I create two data frames, one called first_rows and the other second_rows since this data has one chess players information across two rows. Then I add an ID, merge across on that ID, drop some unnecssary rows, and we have a full dataframe called d.tourney.

```
#create 2 data frames
first_rows <- data.frame(first_rows <- raw.tourney %>%
 filter(row_number() %% 2 == 1))
second_rows <- data.frame(second_rows <- raw.tourney[c(rep(FALSE),TRUE),])

#create IDs for each data frame so we can match on it
first_rows$ID <- seq.int(nrow(first_rows))
second_rows$ID <- seq.int(nrow(first_rows))

#merge these two datasets together so all a player's data is in one row
d.tourney <- merge(first_rows, second_rows, by="ID")
#drop some columns we don't need
d.tourney <- subset(d.tourney, select = -c(V1.x, V11.x, V11.y))
```

Our dataframe columns need some cleaning so I create new columns and select the data I need.

```
#create a pre-rating column and extract characters 15-18 to grab the relevant
#part of the string
d.tourney$PreRating <- str_sub(d.tourney$V2.y, 15, 19)

#create a column for each of the possible 7 opponents, and start taking the
#value from the existing column at character 2 as to avoid the letter
#representing the outcome of the match, we just need the opponent's ID
#there are some blank entries at this point as not all players had 7 opponents
d.tourney$R1opp <- str_sub(d.tourney$V4.x, 2, )
d.tourney$R2opp <- str_sub(d.tourney$V5.x, 2, )
d.tourney$R3opp <- str_sub(d.tourney$V6.x, 2, )
d.tourney$R4opp <- str_sub(d.tourney$V7.x, 2, )
d.tourney$R5opp <- str_sub(d.tourney$V8.x, 2, )
d.tourney$R6opp <- str_sub(d.tourney$V9.x, 2, )
d.tourney$R7opp <- str_sub(d.tourney$V10.x, 2, )

#drop all those columns now that we have what we need from them
d.tourney <- subset(d.tourney, select = -c(V4.x, V5.x, V6.x, V7.x, V8.x, V9.x,
                                           V10.x, V2.y, V3.y, V4.y, V5.y, V6.y,
                                           V7.y, V8.y, V9.y, V10.y))
glimpse(d.tourney)
```

```
## Rows: 64
## Columns: 12
## $ ID        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...
## $ V2.x      <chr> " GARY HUA                        ", " DAKSHESH DARURI   ...
## $ V3.x      <chr> "6.0  ", "6.0  ", "6.0  ", "5.5  ", "5.5  ", "5.0  ", "5....
## $ V1.y      <chr> "   ON ", "   MI ", "   MI ", "   MI ", "   MI ", "   OH ...
## $ PreRating <chr> " 1794", " 1553", " 1384", " 1716", " 1655", " 1686", " 1...
## $ R1opp     <chr> "  39", "  63", "   8", "  23", "  45", "  34", "  57", "...
## $ R2opp     <chr> "  21", "  58", "  61", "  28", "  37", "  29", "  46", "...
## $ R3opp     <chr> "  18", "   4", "  25", "   2", "  12", "  11", "  13", "...
## $ R4opp     <chr> "  14", "  17", "  21", "  26", "  13", "  35", "  11", "...
## $ R5opp     <chr> "   7", "  16", "  11", "   5", "   4", "  10", "   1", "...
## $ R6opp     <chr> "  12", "  20", "  13", "  19", "  14", "  27", "   9", "...
## $ R7opp     <chr> "   4", "   7", "  12", "   1", "  17", "  21", "   2", "...
```

Step 3 - Create calculated variables

Next I calculate the number of rounds each player had, out of a possible 7.

```
#set the 7 variables of opponent IDs to be numeric and this also creates NAs in
#the blank spaces
cols = c(6:12);
d.tourney[,cols] = apply(d.tourney[,cols], 2, function(x) as.numeric(as.character(x)))

#sum up the number of NAs across each row, subtract for 7 possible games to get
#the number of rounds that player plaid - store in "roundsplayed" variable
d.tourney$roundsplayed <- 7 - (apply(is.na(d.tourney), 1, sum))
```

Now we have to look-up the PreRating for all of a player's opponents, based on their ID so we can calculate the average PreRating of their opponents. I did this via a look-up table.

```
#create a 2-variable look-up table with just the player ID and their PreRating
ratinglookup <- subset(d.tourney, select = c(ID, PreRating))

#we can overwrite the player ID that's in each RXopp variable with their
#PreRating, which we can access by using the lookup table created above
d.tourney$R1opp =
  ratinglookup[match(d.tourney$R1opp,
  ratinglookup$ID), "PreRating"]

d.tourney$R2opp =
  ratinglookup[match(d.tourney$R2opp,
  ratinglookup$ID), "PreRating"]

d.tourney$R3opp =
  ratinglookup[match(d.tourney$R3opp,
  ratinglookup$ID), "PreRating"]

d.tourney$R4opp =
  ratinglookup[match(d.tourney$R4opp,
  ratinglookup$ID), "PreRating"]

d.tourney$R5opp =
  ratinglookup[match(d.tourney$R5opp,
  ratinglookup$ID), "PreRating"]

d.tourney$R6opp =
  ratinglookup[match(d.tourney$R6opp,
  ratinglookup$ID), "PreRating"]

d.tourney$R7opp =
  ratinglookup[match(d.tourney$R7opp,
  ratinglookup$ID), "PreRating"]

#make a numeric class so we can do calculations
d.tourney <- d.tourney %>%
    mutate_at(vars(matches('R(.)opp')), list(as.numeric))
```

Finally, we create the variable that holds the average PreRating of the player's opponents.

```
#sum the appropriate rows or the opponents' PreRatings then divide by the rounds
#played
d.tourney$Avg_PreRating_of_Opponents = as.numeric(((rowSums(d.tourney[,c(6:12)],
                                        na.rm = TRUE))/d.tourney$roundsplayed))

#round to nearest whole number
d.tourney$Avg_PreRating_of_Opponents <- round(d.tourney$Avg_PreRating_of_Opponents)

glimpse(d.tourney)
```

```
## Rows: 64
## Columns: 14
## $ ID                         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1...
## $ V2.x                       <chr> " GARY HUA                        ", " D...
## $ V3.x                       <chr> "6.0  ", "6.0  ", "6.0  ", "5.5  ", "5.5...
## $ V1.y                       <chr> "   ON ", "   MI ", "   MI ", "   MI ", ...
## $ PreRating                  <chr> " 1794", " 1553", " 1384", " 1716", " 16...
## $ R1opp                      <dbl> 1436, 1175, 1641, 1363, 1242, 1399, 1092...
## $ R2opp                      <dbl> 1563, 917, 955, 1507, 980, 1602, 377, 14...
## $ R3opp                      <dbl> 1600, 1716, 1745, 1553, 1663, 1712, 1666...
## $ R4opp                      <dbl> 1610, 1629, 1563, 1579, 1666, 1438, 1712...
## $ R5opp                      <dbl> 1649, 1604, 1712, 1655, 1716, 1365, 1794...
## $ R6opp                      <dbl> 1663, 1595, 1666, 1564, 1610, 1552, 1411...
## $ R7opp                      <dbl> 1716, 1649, 1663, 1794, 1629, 1563, 1553...
## $ roundsplayed               <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6, 7, 7...
## $ Avg_PreRating_of_Opponents <dbl> 1605, 1469, 1564, 1574, 1501, 1519, 1372...
```

Step 4 - Final cleaning & export

Do some cleaning of the final dataframe.

```
#remove unnecessary columns
d.tourney <- subset(d.tourney, select = -c(ID, R1opp, R2opp, R3opp, R4opp,
                                           R5opp, R6opp, R7opp, roundsplayed))

#reorder to match assignment
d.tourney <- d.tourney[, c(1, 3, 2, 4, 5)]

#rename columns
d.tourney <- rename(d.tourney, c("V2.x"="Name", "V1.y"="State",
                                 "V3.x"="TotalNumPoints"))

#capitlize names correctly
d.tourney$Name <- str_to_title(d.tourney$Name)

#trim whitespace around state abbreviations
d.tourney$State <- trimws(d.tourney$State, which = c("both"))

head(d.tourney)
```

```
##                                 Name State TotalNumPoints 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
## 6  Hansen Song                         OH            5.0      1686
##   Avg_PreRating_of_Opponents
## 1                       1605
## 2                       1469
## 3                       1564
## 4                       1574
## 5                       1501
## 6                       1519
```

Last, we write the dataframe to a CSV for export to the desktop.

```
write.csv(d.tourney,"C:\\Users\\rgreenlee\\Desktop\\\\chess_tournament_results.csv",
          row.names = FALSE)
```