MY607-Week04-Project1-Chess

Michael Y.

September 22, 2019

Contents

Week 4 - Project 1

Parse a provided table of results from a Chess tournament, and perform the requested transformations

Setup

Load libraries

Load the raw datafile (pull from github)

```
#Avoid using setwd for knitting - instead set directory above in knitr$opts_set
#setwd("C:/Users/Michael/Dropbox/priv/CUNY/MSDS/201909-Spring/DATA607_Tati_Andy_Sabrina/20190922_Week04")
### inputfile <- "tournamentinfo.txt"

### Obtain the file from github, rather than local drive
inputfile <- "https://raw.githubusercontent.com/myampol/MY607/master/tournamentinfo.txt"
rawchess <- read_lines(inputfile)
head(rawchess,18)
```

```
##  [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] "-----------------------------------------------------------------------------------------"
## [11] "    3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|"
## [12] "   MI | 14959604 / R: 1384   ->1640     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
## [13] "-----------------------------------------------------------------------------------------"
## [14] "    4 | PATRICK H SCHILLING             |5.5  |W  23|D  28|W   2|W  26|D   5|W  19|D   1|"
## [15] "   MI | 12616049 / R: 1716   ->1744     |N:2  |W    |B    |W    |B    |W    |B    |B    |"
## [16] "-----------------------------------------------------------------------------------------"
## [17] "    5 | HANSHI ZUO                      |5.5  |W  45|W  37|D  12|D  13|D   4|W  14|W  17|"
## [18] "   MI | 14601533 / R: 1655   ->1690     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
```

Analyse the raw datafile:

The datafile contains 196 lines.
One of every three lines is a separator composed of hyphens, which we will ignore, while the subsequent pair of lines contains information about each participant and the opponents he/she faced in the tournament.
The information about the 64 players is preceded by a pair of lines with header titles.
Here we will transform the above into an array of 65 lines (one for the header titles, followed by one for each of 64 players).
### Transform the input data from 196 lines into 65 lines:

```r
TempOutputArray <- NULL
TempOutputLine <- ""
TempLineNum <- 0
for (row in 1:length(rawchess)) {
  if (row %% 3 == 1) {
    ### this row is just hyphens, so we will ignore it
    invisible(NULL)   #no-op
  }
  if (row %% 3 == 2) {
    ### This is the first of two lines for the player
    TempLineNum <- TempLineNum + 1
    TempOutputLine <- rawchess[row]
  }
  if (row %% 3 == 0) {
    ### This is the second of two lines for the player
    TempOutputLine <- str_trim(paste(TempOutputLine, rawchess[row]))
    TempOutputArray[TempLineNum] <-TempOutputLine
  }
}
head(TempOutputArray) %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

<table>
<tr><th>x</th></tr>
<tr><td>Pair | Player Name |Total|Round|Round|Round|Round|Round|Round|Round| Num | USCF ID / Rtg (Pre->Post) | Pts | 1 | 2 | 3 | 4 | 5 | 6 | 7 |</td></tr>
<tr><td>1 | GARY HUA |6.0 |W 39|W 21|W 18|W 14|W 7|D 12|D 4| ON | 15445895 / R: 1794 ->1817 |N:2 |W |B |W |B |W |B |W |</td></tr>
<tr><td>2 | DAKSHESH DARURI |6.0 |W 63|W 58|L 4|W 17|W 16|W 20|W 7| MI | 14598900 / R: 1553 ->1663 |N:2 |B |W |B |W |B |W |B |</td></tr>
<tr><td>3 | ADITYA BAJAJ |6.0 |L 8|W 61|W 25|W 21|W 11|W 13|W 12| MI | 14959604 / R: 1384 ->1640 |N:2 |W |B |W |B |W |B |W |</td></tr>
<tr><td>4 | PATRICK H SCHILLING |5.5 |W 23|D 28|W 2|W 26|D 5|W 19|D 1| MI | 12616049 / R: 1716 ->1744 |N:2 |W |B |W |B |W |B |B |</td></tr>
<tr><td>5 | HANSHI ZUO |5.5 |W 45|W 37|D 12|D 13|D 4|W 14|W 17| MI | 14601533 / R: 1655 ->1690 |N:2 |B |W |B |W |B |W |B |</td></tr>
</table>

Split at the pipe separators into elements, and trim the extra spaces preceding and following each element:

```
# This is actually a list, but the column names are in the first element
TrimmedOutputArray <- lapply(strsplit(x=TempOutputArray, split = "\\|"),str_trim)

# extract the initial column names from the first element of the list
ColumnNames <- as.array(TrimmedOutputArray[[1]])

# create a "list" which has just the remaining 64 entries (sans column names)
TrimmedOutputList <- TrimmedOutputArray[-c(1)]

# make a data frame from the list (initially no column names)
TrimmedOutputDF <- as.data.frame(TrimmedOutputList,  cut.names = TRUE, stringsAsFactors = FALSE)

# this didn't quite give what we wanted, because each player is in a column
# (i.e., there are 20 rows of features, and 64 columns of players)
dim(TrimmedOutputDF)
```

```
## [1] 20 64
```

```
# show just the first two players (otherwise, too wide...)
head(TrimmedOutputDF[,1:2],14)  %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...	c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17...
1	2
GARY HUA	DAKSHESH DARURI
6.0	6.0
W 39	W 63
W 21	W 58
W 18	L 4
W 14	W 17
W 7	W 16
D 12	W 20
D 4	W 7
ON	MI
15445895 / R: 1794 ->1817	14598900 / R: 1553 ->1663
N:2	N:2
W	B

```r
# so we need to transpose the data frame:
TrimmedOutputDF <- t(TrimmedOutputDF)
head(TrimmedOutputDF,3)
```

```
##                                                                       [,1] [,2]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "1"  "GARY HUA"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "2"  "DAKSHESH DARURI"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "3"  "ADITYA BAJAJ"
##                                                                       [,3]  [,4]    [,5]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "6.0" "W  39" "W  21"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "6.0" "W  63" "W  58"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "6.0" "L   8" "W  61"
##                                                                       [,6]    [,7]    [,8]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "W  18" "W  14" "W   7"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "L   4" "W  17" "W  16"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "W  25" "W  21" "W  11"
##                                                                       [,9]    [,10]   [,11]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "D  12" "D   4" "ON"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "W  20" "W   7" "MI"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "W  13" "W  12" "MI"
##                                                                       [,12]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "15445895 / R: 1794   ->1817"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "14598900 / R: 1553   ->1663"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "14959604 / R: 1384   ->1640"
##                                                                       [,13] [,14] [,15] [,16]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "N:2" "W"   "B"   "W"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "N:2" "B"   "W"   "B"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "N:2" "W"   "B"   "W"
##                                                                       [,17] [,18] [,19] [,20]
## c..1....GARY.HUA....6.0....W..39....W..21....W..18....W..14...        "B"   "W"   "B"   "W"
## c..2....DAKSHESH.DARURI....6.0....W..63....W..58....L...4....W..17... "W"   "B"   "W"   "B"
## c..3....ADITYA.BAJAJ....6.0....L...8....W..61....W..25....W..21...    "B"   "W"   "B"   "W"
```

```r
# however, the rownames and colnames are not as expected
# so, delete the row names
rownames(TrimmedOutputDF) <- NULL
# and, set the column names to the items extracted above
colnames(TrimmedOutputDF)<- ColumnNames
```

```r
# Display the info on the first 3 players (display transposed, for clarity)
t(head(TrimmedOutputDF,3))  %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

Pair	1	2	3
Player Name	GARY HUA	DAKSHESH DARURI	ADITYA BAJAJ
Total	6.0	6.0	6.0
Round	W 39	W 63	L 8
Round	W 21	W 58	W 61
Round	W 18	L 4	W 25
Round	W 14	W 17	W 21
Round	W 7	W 16	W 11
Round	D 12	W 20	W 13
Round	D 4	W 7	W 12
Num	ON	MI	MI
USCF ID / Rtg (Pre->Post)	15445895 / R: 1794 ->1817	14598900 / R: 1553 ->1663	14959604 / R: 1384 ->1640
Pts	N:2	N:2	N:2
1	W	B	W
2	B	W	B
3	W	B	W
4	B	W	B
5	W	B	W
6	B	W	B
7	W	B	W

Clean up column headers

The column titles are not unique, and do not always make sense:

```
BadColNames <- TrimmedOutputArray[[1]]
BadColNames
```

```
##  [1] "Pair"                      "Player Name"               "Total"
##  [4] "Round"                     "Round"                     "Round"
##  [7] "Round"                     "Round"                     "Round"
## [10] "Round"                     "Num"                       "USCF ID / Rtg (Pre->Post)"
## [13] "Pts"                       "1"                         "2"
## [16] "3"                         "4"                         "5"
## [19] "6"                         "7"
```

so we will replace them with better titles:

```
BetterColNames <- c("ID",
                    "PlayerName",
                    "TotalPts",
                    sprintf("Round%d",1:7),
                    "State",
                    "USCFID_RatePre_RatePost",
                    "Pts",
                    sprintf("Color%d",1:7))
BetterColNames
```

```
##  [1] "ID"                      "PlayerName"              "TotalPts"
##  [4] "Round1"                  "Round2"                  "Round3"
##  [7] "Round4"                  "Round5"                  "Round6"
## [10] "Round7"                  "State"                   "USCFID_RatePre_RatePost"
## [13] "Pts"                     "Color1"                  "Color2"
## [16] "Color3"                  "Color4"                  "Color5"
## [19] "Color6"                  "Color7"
```

```
TrimmedOutputArray[[1]] = BetterColNames
```

Now convert the above array into a matrix, where the first line from the array becomes the column names, and the remaining 64 lines become the data:

```r
TotalRows <- length(TrimmedOutputArray)
TotalElements <- length(unlist(TrimmedOutputArray))
ColumnCount <- table(unlist(lapply(TrimmedOutputArray,length)))
TrimmedOutputMatrix  <-  matrix(data = unlist(TrimmedOutputArray[2:TotalRows]),
                               nrow = length(TrimmedOutputArray)-1,
                               ncol=length(TrimmedOutputArray[[1]]),
                               byrow = T)
colnames(TrimmedOutputMatrix) <- BetterColNames
head(TrimmedOutputMatrix)
```

```
##      ID  PlayerName            TotalPts Round1  Round2  Round3  Round4  Round5  Round6 
## [1,] "1" "GARY HUA"            "6.0"    "W  39" "W  21" "W  18" "W  14" "W   7" "D  12"
## [2,] "2" "DAKSHESH DARURI"     "6.0"    "W  63" "W  58" "L   4" "W  17" "W  16" "W  20"
## [3,] "3" "ADITYA BAJAJ"        "6.0"    "L   8" "W  61" "W  25" "W  21" "W  11" "W  13"
## [4,] "4" "PATRICK H SCHILLING" "5.5"    "W  23" "D  28" "W   2" "W  26" "D   5" "W  19"
## [5,] "5" "HANSHI ZUO"          "5.5"    "W  45" "W  37" "D  12" "D  13" "D   4" "W  14"
## [6,] "6" "HANSEN SONG"         "5.0"    "W  34" "D  29" "L  11" "W  35" "D  10" "W  27"
##      Round7  State USCFID_RatePre_RatePost       Pts   Color1 Color2 Color3 Color4 Color5
## [1,] "D   4" "ON"  "15445895 / R: 1794   ->1817" "N:2" "W"    "B"    "W"    "B"    "W"   
## [2,] "W   7" "MI"  "14598900 / R: 1553   ->1663" "N:2" "B"    "W"    "B"    "W"    "B"   
## [3,] "W  12" "MI"  "14959604 / R: 1384   ->1640" "N:2" "W"    "B"    "W"    "B"    "W"   
## [4,] "D   1" "MI"  "12616049 / R: 1716   ->1744" "N:2" "W"    "B"    "W"    "B"    "W"   
## [5,] "W  17" "MI"  "14601533 / R: 1655   ->1690" "N:2" "B"    "W"    "B"    "W"    "B"   
## [6,] "W  21" "OH"  "15055204 / R: 1686   ->1687" "N:3" "W"    "B"    "W"    "B"    "B"   
##      Color6 Color7
## [1,] "B"    "W"   
## [2,] "W"    "B"   
## [3,] "B"    "W"   
## [4,] "B"    "B"   
## [5,] "W"    "B"   
## [6,] "W"    "B"
```

There were 65 total rows in the array, corresponding to 1300 total elements.

This is consistent with having 20 columns on each row.

On the rating, ignore whatever number comes after a P, i.e., if a rating is nnnnPxx , use only the first part.

```
df = data.frame(TrimmedOutputMatrix, stringsAsFactors = F)
USCFID <- as.integer(str_extract(df$USCFID_RatePre_RatePost, "^\\d{8}"))
df$USCFID <- USCFID
tempPrerating1 <- str_replace(string = df$USCFID_RatePre_RatePost, "\\d{8} / R: *","")
tempPrerating2 <- str_replace(tempPrerating1, "->.*$","")
tempPrerating3 <- str_replace(tempPrerating2,"P.*$","")
df$RatePre <- as.integer(tempPrerating3)
cat('Pre-tournament ratings for each player:\n')
```

```
## Pre-tournament ratings for each player:
```

```
df$RatePre
```

```
##  [1] 1794 1553 1384 1716 1655 1686 1649 1641 1411 1365 1712 1663 1666 1610 1220 1604 1629 1600
## [19] 1564 1595 1563 1555 1363 1229 1745 1579 1552 1507 1602 1522 1494 1441 1449 1399 1438 1355
## [37]  980 1423 1436 1348 1403 1332 1283 1199 1242  377 1362 1382 1291 1056 1011  935 1393 1270
## [55] 1186 1153 1092  917  853  967  955 1530 1175 1163
```

```
# we're not actually using the Post-tournament ratings, but they might be useful to have...
tempPostrating1=str_trim(str_replace(string = df$USCFID_RatePre_RatePost, "^.*->",""))
tempPostrating2=str_replace(tempPostrating1, "P.*$","")
df$RatePost <- as.integer(tempPostrating2)
cat('Post-tournament ratings for each player:\n')
```

```
## Post-tournament ratings for each player:
```

```
df$RatePost
```

```
##  [1] 1817 1663 1640 1744 1690 1687 1673 1657 1564 1544 1696 1670 1662 1618 1416 1613 1610 1600
## [19] 1570 1569 1562 1529 1371 1300 1681 1564 1539 1513 1508 1444 1444 1433 1421 1400 1392 1367
## [37] 1077 1439 1413 1346 1341 1256 1244 1199 1191 1076 1341 1335 1259 1111 1097 1092 1359 1200
## [55] 1163 1140 1079  941  878  984  979 1535 1125 1112
```

Compute wins/losses/draws and games played for each player

Separate out the win/loss/draw/etc. for each player and tally up, in order to determine now many games each played:

```
### WLD is a 64x7 grid of the individual letters corresponding to each game played
### (or, not played) per player/round
WLD = mapply(function(x) str_extract(string = x,
                                     pattern = "^."),
             subset(x = df,
                    select = Round1:Round7))
cat('Win-Loss-Draw for the various players (first 6 displayed) : \n')
```

```
## Win-Loss-Draw for the various players (first 6 displayed) :
```

```
head(WLD)
```

```
##      Round1 Round2 Round3 Round4 Round5 Round6 Round7
## [1,] "W"    "W"    "W"    "W"    "W"    "D"    "D"
## [2,] "W"    "W"    "L"    "W"    "W"    "W"    "W"
## [3,] "L"    "W"    "W"    "W"    "W"    "W"    "W"
## [4,] "W"    "D"    "W"    "W"    "D"    "W"    "D"
## [5,] "W"    "W"    "D"    "D"    "D"    "W"    "W"
## [6,] "W"    "D"    "L"    "W"    "D"    "W"    "W"
```

```
### collapseWLD collapses each row into a single string
collapseWLD = apply(X=WLD,MARGIN = 1, function(rw) str_c(rw,collapse=''))
cat('The above, collapsed into a string for each player (first 6 displayed) : \n')
```

```
## The above, collapsed into a string for each player (first 6 displayed) :
```

```
t(t(head(collapseWLD)))
```

```
##      [,1]
## [1,] "WWWWWDD"
## [2,] "WWLWWWW"
## [3,] "LWWWWWW"
## [4,] "WDWWDWD"
## [5,] "WWDDDWW"
## [6,] "WDLWDWW"
```

Determine how many games were W, L, or D (ignoring other items):

```r
### compute a table for each row, where the table counts the occurance of each value (W, L, D, or other indicators)
tempResults1 <- apply(X=WLD, MARGIN = 1, FUN=table)
cat('list of tables tallying results for each competitor (only the first 6 displayed here:) \n')
```

```
## list of tables tallying results for each competitor (only the first 6 displayed here:)
```

```r
head(tempResults1)
```

```
## [[1]]
##
## D W
## 2 5
##
## [[2]]
##
## L W
## 1 6
##
## [[3]]
##
## L W
## 1 6
##
## [[4]]
##
## D W
## 3 4
##
## [[5]]
##
## D W
## 3 4
##
## [[6]]
##
## D L W
## 2 1 4
```

Tally the number which are “W”, “L”, or “D” – (other values such as “U” or “B” indicate that no game was played):

```
numgames <- lapply(X = tempResults1,
                   FUN = function(rw) ifelse(is.na(rw["W"]),0,rw["W"])
                                     +ifelse(is.na(rw["L"]),0,rw["L"])
                                     +ifelse(is.na(rw["D"]),0,rw["D"]))
df$numgames <- as.integer(numgames)
cat('Number of "W", "L", "D" games played by each player: \n')
```

```
## Number of "W", "L", "D" games played by each player:
```

```
df$numgames
```

```
##  [1] 7 7 7 7 7 7 7 7 7 7 7 6 7 7 7 5 7 7 7 7 7 6 7 7 7 7 6 7 6 7 7 7 7 7 7 6 5 6 7 7 4 7 7 6 7
## [46] 7 7 5 5 6 7 7 3 6 6 5 6 6 6 5 7 1 5 7
```

Extract the list of opponents from the Round1:Round7 columns, so we can look up their ratings:

(Note that there are “NA” values in cases where no game was played)

```
opponents=mapply(function(x) as.integer(str_trim(str_replace(string = x, pattern = "^.",""))),
                 subset(x = df, select = Round1:Round7))
colnames(opponents) <- sprintf("Opp%d",1:7)
cbind(1:dim(opponents)[1],df$PlayerName,opponents) %>%
  head() %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

		Opp1	Opp2	Opp3	Opp4	Opp5	Opp6	Opp7
1	GARY HUA	39	21	18	14	7	12	4
2	DAKSHESH DARURI	63	58	4	17	16	20	7
3	ADITYA BAJAJ	8	61	25	21	11	13	12
4	PATRICK H SCHILLING	23	28	2	26	5	19	1
5	HANSHI ZUO	45	37	12	13	4	14	17
6	HANSEN SONG	34	29	11	35	10	27	21

Determine the average ratings for each player's opponents, and append all relevant info to the dataframe:

Append the 7 opponent columns to the dataframe:

```
### CAUTION -- this would create duplicate columns
### if this chunk (alone) were re-evaluated manually,
### but this is not a problem when knitting the entire result,
### as it can only be executed once
df <- cbind(df,opponents)
### First 3 players:
head(df,3)
```

```
##   ID      PlayerName TotalPts Round1 Round2 Round3 Round4 Round5 Round6 Round7 State
## 1  1         GARY HUA      6.0  W  39  W  21  W  18  W  14  W   7  D  12  D   4    ON
## 2  2 DAKSHESH DARURI      6.0  W  63  W  58  L   4  W  17  W  16  W  20  W   7    MI
## 3  3    ADITYA BAJAJ      6.0  L   8  W  61  W  25  W  21  W  11  W  13  W  12    MI
##       USCFID_RatePre_RatePost Pts Color1 Color2 Color3 Color4 Color5 Color6 Color7   USCFID
## 1 15445895 / R: 1794   ->1817 N:2      W      B      W      B      W      B      W 15445895
## 2 14598900 / R: 1553   ->1663 N:2      B      W      B      W      B      W      B 14598900
## 3 14959604 / R: 1384   ->1640 N:2      W      B      W      B      W      B      W 14959604
##   RatePre RatePost numgames Opp1 Opp2 Opp3 Opp4 Opp5 Opp6 Opp7
## 1    1794     1817        7   39   21   18   14    7   12    4
## 2    1553     1663        7   63   58    4   17   16   20    7
## 3    1384     1640        7    8   61   25   21   11   13   12
```

```r
### Transpose, for clarity:
t(head(df,3)) %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

	1	2	3
ID	1	2	3
PlayerName	GARY HUA	DAKSHESH DARURI	ADITYA BAJAJ
TotalPts	6.0	6.0	6.0
Round1	W 39	W 63	L 8
Round2	W 21	W 58	W 61
Round3	W 18	L 4	W 25
Round4	W 14	W 17	W 21
Round5	W 7	W 16	W 11
Round6	D 12	W 20	W 13
Round7	D 4	W 7	W 12
State	ON	MI	MI
USCFID__RatePre__RatePost	15445895 / R: 1794 ->1817	14598900 / R: 1553 ->1663	14959604 / R: 1384 ->1640
Pts	N:2	N:2	N:2
Color1	W	B	W
Color2	B	W	B
Color3	W	B	W
Color4	B	W	B
Color5	W	B	W
Color6	B	W	B
Color7	W	B	W
USCFID	15445895	14598900	14959604
RatePre	1794	1553	1384
RatePost	1817	1663	1640
numgames	7	7	7
Opp1	39	63	8
Opp2	21	58	61
Opp3	18	4	25
Opp4	14	17	21
Opp5	7	16	11
Opp6	12	20	13
Opp7	4	7	12

Look up the ratings for each opponent:

```
oppPreRatings=apply(opponents,
                    MARGIN = c(1,2),
                    FUN=function(x)df[x,]$RatePre)
### set the the column names
colnames(oppPreRatings) <- sprintf("OppPreRate%d",1:7)
cat('Opponent ratings, prior to the tournament: \n')
```

```
## Opponent ratings, prior to the tournament:
```

```
head(cbind(1:dim(opponents)[1],df$PlayerName,oppPreRatings),22)  %>%
  kable() %>%  kable_styling(c("striped", "bordered"))
```

		OppPreRate1	OppPreRate2	OppPreRate3	OppPreRate4	OppPreRate5	OppPreRate6	OppPreRate7
1	GARY HUA	1436	1563	1600	1610	1649	1663	1716
2	DAKSHESH DARURI	1175	917	1716	1629	1604	1595	1649
3	ADITYA BAJAJ	1641	955	1745	1563	1712	1666	1663
4	PATRICK H SCHILLING	1363	1507	1553	1579	1655	1564	1794
5	HANSHI ZUO	1242	980	1663	1666	1716	1610	1629
6	HANSEN SONG	1399	1602	1712	1438	1365	1552	1563
7	GARY DEE SWATHELL	1092	377	1666	1712	1794	1411	1553
8	EZEKIEL HOUGHTON	1384	1441	1610	1411	1362	1507	1564
9	STEFANO LEE	1745	1600	853	1641	1579	1649	1595
10	ANVIT RAO	1604	1564	1186	1494	1686	1745	1600
11	CAMERON WILLIAM MC LEMAN	1423	1153	1686	1649	1384	1399	1579
12	KENNETH J TACK	1332	1449	1655	1423	NA	1794	1384
13	TORRANCE HENRY JR	1355	1552	1649	1655	1449	1384	1441
14	BRADLEY SHAW	1270	1199	1641	1794	1552	1655	1494
15	ZACHARY JAMES HOUGHTON	1564	1604	1522	1555	1270	1449	1423
16	MIKE NIKITIN	1365	1220	NA	1436	1553	1355	NA
17	RONALD GRZEGORCZYK	1382	1403	1579	1553	1363	1555	1655
18	DAVID SUNDEEN	1362	1411	1794	1441	1564	1423	1365
19	DIPANKAR ROY	1220	1365	935	1507	1600	1716	1641
20	JASON ZHENG	1348	1291	1363	1403	1507	1553	1411
21	DINH DANG BUI	1283	1794	1362	1384	1348	1436	1686
22	EUGENE L MCCLURE	1163	935	1507	1220	NA	1629	1348

Append the 7 opponent rating columns to the dataframe:

```r
### CAUTION -- this would duplicate columns if this chunk is re-evaluated manually,
### but this is not a problem when knitting the entire result
df <- cbind(df,oppPreRatings)
### display first 3 players, transposed, for clarity
t(head(df,n = 3))   %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

	1	2	3
ID	1	2	3
PlayerName	GARY HUA	DAKSHESH DARURI	ADITYA BAJAJ
TotalPts	6.0	6.0	6.0
Round1	W 39	W 63	L 8
Round2	W 21	W 58	W 61
Round3	W 18	L 4	W 25
Round4	W 14	W 17	W 21
Round5	W 7	W 16	W 11
Round6	D 12	W 20	W 13
Round7	D 4	W 7	W 12
State	ON	MI	MI
USCFID_RatePre_RatePost	15445895 / R: 1794 ->1817	14598900 / R: 1553 ->1663	14959604 / R: 1384 ->1640
Pts	N:2	N:2	N:2
Color1	W	B	W
Color2	B	W	B
Color3	W	B	W
Color4	B	W	B
Color5	W	B	W
Color6	B	W	B
Color7	W	B	W
USCFID	15445895	14598900	14959604
RatePre	1794	1553	1384
RatePost	1817	1663	1640
numgames	7	7	7
Opp1	39	63	8
Opp2	21	58	61
Opp3	18	4	25
Opp4	14	17	21
Opp5	7	16	11
Opp6	12	20	13
Opp7	4	7	12
OppPreRate1	1436	1175	1641
OppPreRate2	1563	917	955
OppPreRate3	1600	1716	1745
OppPreRate4	1610	1629	1563
OppPreRate5	1649	1604	1712
OppPreRate6	1663	1595	1666
OppPreRate7	1716	1649	1663

Compute the average opponent rating, for each participant:

```
# use na.rm=T to omit the NA values
# divide by the number of opponents played
# (in the case where no game was played, opponent=NA and numgames < 7)
AvgOppRatePre <- rowSums(oppPreRatings,na.rm=T)/df$numgames
cat('Average opponent rating, prior to the tournament: \n')
```

```
## Average opponent rating, prior to the tournament:
```

```
AvgOppRatePre
```

```
##  [1] 1605.29 1469.29 1563.57 1573.57 1500.86 1518.71 1372.14 1468.43 1523.14 1554.14 1467.57
## [12] 1506.17 1497.86 1515.00 1483.86 1385.80 1498.57 1480.00 1426.29 1410.86 1470.43 1300.33
## [23] 1213.86 1357.00 1363.29 1506.86 1221.67 1522.14 1313.50 1144.14 1259.86 1378.71 1276.86
## [34] 1375.29 1149.71 1388.17 1384.80 1539.17 1429.57 1390.57 1248.50 1149.86 1106.57 1327.00
## [45] 1152.00 1357.71 1392.00 1355.80 1285.80 1296.00 1356.14 1494.57 1345.33 1206.17 1406.00
## [56] 1414.40 1363.00 1391.00 1319.00 1330.20 1327.29 1186.00 1350.20 1263.00
```

```
### Append the result to the dataframe
df$AvgOppRatePre <- AvgOppRatePre
```

doublecheck to be sure that rowSums/numgames gives the same result as rowMeans:

```
rowMeans(oppPreRatings,na.rm = T)
```

```
##  [1] 1605.29 1469.29 1563.57 1573.57 1500.86 1518.71 1372.14 1468.43 1523.14 1554.14 1467.57
## [12] 1506.17 1497.86 1515.00 1483.86 1385.80 1498.57 1480.00 1426.29 1410.86 1470.43 1300.33
## [23] 1213.86 1357.00 1363.29 1506.86 1221.67 1522.14 1313.50 1144.14 1259.86 1378.71 1276.86
## [34] 1375.29 1149.71 1388.17 1384.80 1539.17 1429.57 1390.57 1248.50 1149.86 1106.57 1327.00
## [45] 1152.00 1357.71 1392.00 1355.80 1285.80 1296.00 1356.14 1494.57 1345.33 1206.17 1406.00
## [56] 1414.40 1363.00 1391.00 1319.00 1330.20 1327.29 1186.00 1350.20 1263.00
```

```
### any differences?
rowSums(oppPreRatings,na.rm=T)/df$numgames - rowMeans(oppPreRatings,na.rm = T)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [46] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Clean up the capitalization of player names:

The input data has the names entirely capitalized:

```
cat('Initial name format supplied in input data: \n')
```

```
## Initial name format supplied in input data:
```

```
t(t(head(df$PlayerName)))
```

```
##      [,1]
## [1,] "GARY HUA"
## [2,] "DAKSHESH DARURI"
## [3,] "ADITYA BAJAJ"
## [4,] "PATRICK H SCHILLING"
## [5,] "HANSHI ZUO"
## [6,] "HANSEN SONG"
```

but the sample output shows just the first letter on each name component capitalized:

```
##library(R.utils) ## loaded at top

# lowercase each name, and then separate each into an array of strings,
# so "capitalize"" will operate on each component
CapitalList <- lapply(strsplit(x = tolower(df$PlayerName),
                               split = " "),
                      capitalize)
cat('Properly Capitalized name components (in list format): \n')
```

```
## Properly Capitalized name components (in list format):
```

```
head(CapitalList)
```

```
## [[1]]
## [1] "Gary" "Hua"
```

```
##
## [[2]]
## [1] "Dakshesh" "Daruri"
##
## [[3]]
## [1] "Aditya" "Bajaj"
##
## [[4]]
## [1] "Patrick"   "H"         "Schilling"
##
## [[5]]
## [1] "Hanshi" "Zuo"
##
## [[6]]
## [1] "Hansen" "Song"
```

paste the resulting names back together again

```
CapitalNameArray=unlist(lapply( X = CapitalList,
                                FUN= function(name) paste(name, collapse=" ")))
cat('Each name pasted back together again: \n')
```

```
## Each name pasted back together again:
```

```
t(t(head(CapitalNameArray)))
```

```
##      [,1]
## [1,] "Gary Hua"
## [2,] "Dakshesh Daruri"
## [3,] "Aditya Bajaj"
## [4,] "Patrick H Schilling"
## [5,] "Hanshi Zuo"
## [6,] "Hansen Song"
```

```
# append Capitalized Name as a new column in the data frame
df$CapitalizedName = CapitalNameArray
```

Select output columns for final result into smaller dataframe:

```
OutputColumns = c("CapitalizedName","State", "TotalPts", "RatePre", "AvgOppRatePre")
FinalOutputDF <- subset(x = df, select = OutputColumns)

##### improve the column headers, to match the assignment
OutputHeaders <- c("Player's Name",
                   "Player's State",
                   "Total Number of Points",
                   "Player's Pre-Rating",
                   "Average Pre Chess Rating of Opponents")
colnames(x = FinalOutputDF) <- OutputHeaders
head(FinalOutputDF,22) %>% kable() %>%  kable_styling(c("striped", "bordered"))
```

Player's Name	Player's State	Total Number of Points	Player's Pre-Rating	Average Pre Chess Rating of Opponents
Gary Hua	ON	6.0	1794	1605.29
Dakshesh Daruri	MI	6.0	1553	1469.29
Aditya Bajaj	MI	6.0	1384	1563.57
Patrick H Schilling	MI	5.5	1716	1573.57
Hanshi Zuo	MI	5.5	1655	1500.86
Hansen Song	OH	5.0	1686	1518.71
Gary Dee Swathell	MI	5.0	1649	1372.14
Ezekiel Houghton	MI	5.0	1641	1468.43
Stefano Lee	ON	5.0	1411	1523.14
Anvit Rao	MI	5.0	1365	1554.14
Cameron William Mc Leman	MI	4.5	1712	1467.57
Kenneth J Tack	MI	4.5	1663	1506.17
Torrance Henry Jr	MI	4.5	1666	1497.86
Bradley Shaw	MI	4.5	1610	1515.00
Zachary James Houghton	MI	4.5	1220	1483.86
Mike Nikitin	MI	4.0	1604	1385.80
Ronald Grzegorczyk	MI	4.0	1629	1498.57
David Sundeen	MI	4.0	1600	1480.00
Dipankar Roy	MI	4.0	1564	1426.29
Jason Zheng	MI	4.0	1595	1410.86
Dinh Dang Bui	ON	4.0	1563	1470.43
Eugene L Mcclure	MI	4.0	1555	1300.33

Write the results into a .csv file:

```
head(FinalOutputDF)
```

```
##         Player's Name Player's State Total Number of Points Player's Pre-Rating
## 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
##   Average Pre Chess Rating of Opponents
## 1                               1605.29
## 2                               1469.29
## 3                               1563.57
## 4                               1573.57
## 5                               1500.86
## 6                               1518.71
```

```
options(encoding = 'UTF-8')
results_filename <- "MY-DATA607-Week04-ChessData.csv"
write.csv(FinalOutputDF, results_filename, row.names=FALSE)
```

Re-read the results (stored in github, rather than local drive) and display them:

```r
# read the results from local disk
# re_read_output <- read_lines(results_filename)

# Read the results from file (previously) posted to github
github_filename <-
  "https://raw.githubusercontent.com/myampol/MY607/master/MY-DATA607-Week04-ChessData.csv"
re_read_output <- read_lines(github_filename)
cat (re_read_output, sep = '\n')
```

```
## "Player's Name","Player's State","Total Number of Points","Player's Pre-Rating","Average Pre Chess Rating of Opponents"
## "Gary Hua","ON","6.0",1794,1605.28571428571
## "Dakshesh Daruri","MI","6.0",1553,1469.28571428571
## "Aditya Bajaj","MI","6.0",1384,1563.57142857143
## "Patrick H Schilling","MI","5.5",1716,1573.57142857143
## "Hanshi Zuo","MI","5.5",1655,1500.85714285714
## "Hansen Song","OH","5.0",1686,1518.71428571429
## "Gary Dee Swathell","MI","5.0",1649,1372.14285714286
## "Ezekiel Houghton","MI","5.0",1641,1468.42857142857
## "Stefano Lee","ON","5.0",1411,1523.14285714286
## "Anvit Rao","MI","5.0",1365,1554.14285714286
## "Cameron William Mc Leman","MI","4.5",1712,1467.57142857143
## "Kenneth J Tack","MI","4.5",1663,1506.16666666667
## "Torrance Henry Jr","MI","4.5",1666,1497.85714285714
## "Bradley Shaw","MI","4.5",1610,1515
## "Zachary James Houghton","MI","4.5",1220,1483.85714285714
## "Mike Nikitin","MI","4.0",1604,1385.8
## "Ronald Grzegorczyk","MI","4.0",1629,1498.57142857143
## "David Sundeen","MI","4.0",1600,1480
## "Dipankar Roy","MI","4.0",1564,1426.28571428571
## "Jason Zheng","MI","4.0",1595,1410.85714285714
## "Dinh Dang Bui","ON","4.0",1563,1470.42857142857
## "Eugene L Mcclure","MI","4.0",1555,1300.33333333333
## "Alan Bui","ON","4.0",1363,1213.85714285714
## "Michael R Aldrich","MI","4.0",1229,1357
## "Loren Schwiebert","MI","3.5",1745,1363.28571428571
## "Max Zhu","ON","3.5",1579,1506.85714285714
## "Gaurav Gidwani","MI","3.5",1552,1221.66666666667
## "Sofia Adina Stanescu-bellu","MI","3.5",1507,1522.14285714286
## "Chiedozie Okorie","MI","3.5",1602,1313.5
## "George Avery Jones","ON","3.5",1522,1144.14285714286
## "Rishi Shetty","MI","3.5",1494,1259.85714285714
## "Joshua Philip Mathews","ON","3.5",1441,1378.71428571429
## "Jade Ge","MI","3.5",1449,1276.85714285714
## "Michael Jeffery Thomas","MI","3.5",1399,1375.28571428571
## "Joshua David Lee","MI","3.5",1438,1149.71428571429
## "Siddharth Jha","MI","3.5",1355,1388.16666666667
## "Amiyatosh Pwnanandam","MI","3.5",980,1384.8
## "Brian Liu","MI","3.0",1423,1539.16666666667
## "Joel R Hendon","MI","3.0",1436,1429.57142857143
## "Forest Zhang","MI","3.0",1348,1390.57142857143
## "Kyle William Murphy","MI","3.0",1403,1248.5
## "Jared Ge","MI","3.0",1332,1149.85714285714
## "Robert Glen Vasey","MI","3.0",1283,1106.57142857143
## "Justin D Schilling","MI","3.0",1199,1327
## "Derek Yan","MI","3.0",1242,1152
## "Jacob Alexander Lavalley","MI","3.0",377,1357.71428571429
## "Eric Wright","MI","2.5",1362,1392
## "Daniel Khain","MI","2.5",1382,1355.8
## "Michael J Martin","MI","2.5",1291,1285.8
## "Shivam Jha","MI","2.5",1056,1296
## "Tejas Ayyagari","MI","2.5",1011,1356.14285714286
## "Ethan Guo","MI","2.5",935,1494.57142857143
## "Jose C Ybarra","MI","2.0",1393,1345.33333333333
## "Larry Hodge","MI","2.0",1270,1206.16666666667
## "Alex Kong","MI","2.0",1186,1406
## "Marisa Ricci","MI","2.0",1153,1414.4
## "Michael Lu","MI","2.0",1092,1363
## "Viraj Mohile","MI","2.0",917,1391
## "Sean M Mc Cormick","MI","2.0",853,1319
## "Julia Shen","MI","1.5",967,1330.2
## "Jezzel Farkas","ON","1.5",955,1327.28571428571
## "Ashwin Balaji","MI","1.0",1530,1186
## "Thomas Joseph Hosmer","MI","1.0",1175,1350.2
## "Ben Li","MI","1.0",1163,1263
```