MY-DATA607-Week04-Chess

Michael Y.

September 23, 2018

Week 4 - Project 1

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

Load libraries

```
library(readr)
library(stringr)
library(R.utils)
```

```
## Loading required package: R.oo
```

```
## Loading required package: R.methodsS3
```

```
## R.methodsS3 v1.7.1 (2016-02-15) successfully loaded. See ?R.methodsS3 for help.
```

```
## R.oo v1.22.0 (2018-04-21) successfully loaded. See ?R.oo for help.
```

```
##
## Attaching package: 'R.oo'
```

```
## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods
```

```
## The following objects are masked from 'package:base':
##
##     attach, detach, gc, load, save
```

```
## R.utils v2.7.0 successfully loaded. See ?R.utils for help.
```

```
##
## Attaching package: 'R.utils'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, inherits, isOpen, parse, warnings
```

Load the raw datafile

```
setwd("C:/Users/Michael/Dropbox/priv/CUNY/MSDS/201809-Fall/DATA607_Andy_Sabrina/Week04-Project 1")
inputfile <- "tournamentinfo.txt"
rawchess <- read_lines(inputfile)
head(rawchess)
```

```
## [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    |"
```

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):

```
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)
```

```
## [1] "Pair | Player Name                    |Total|Round|Round|Round|Round|Round|Round|Round|   Num
## [2] "1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|    ON | 15
## [3] "2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|    MI | 14
## [4] "3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|    MI | 14
## [5] "4 | PATRICK H SCHILLING             |5.5  |W  23|D  28|W   2|W  26|D   5|W  19|D   1|    MI | 11
## [6] "5 | HANSHI ZUO                      |5.5  |W  45|W  37|D  12|D  13|D   4|W  14|W  17|    MI | 14
```

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

```
TrimmedOutputArray <- lapply(strsplit(x=TempOutputArray, split = "\\|"),str_trim)
head(TrimmedOutputArray)
```

```
## [[1]]
##  [1] "Pair"                      "Player Name"
##  [3] "Total"                     "Round"
```

```
##  [5] "Round"                    "Round"
##  [7] "Round"                    "Round"
##  [9] "Round"                    "Round"
## [11] "Num"                      "USCF ID / Rtg (Pre->Post)"
## [13] "Pts"                      "1"
## [15] "2"                        "3"
## [17] "4"                        "5"
## [19] "6"                        "7"
##
## [[2]]
##  [1] "1"                          "GARY HUA"
##  [3] "6.0"                        "W  39"
##  [5] "W  21"                      "W  18"
##  [7] "W  14"                      "W   7"
##  [9] "D  12"                      "D   4"
## [11] "ON"                         "15445895 / R: 1794   ->1817"
## [13] "N:2"                        "W"
## [15] "B"                          "W"
## [17] "B"                          "W"
## [19] "B"                          "W"
##
## [[3]]
##  [1] "2"                          "DAKSHESH DARURI"
##  [3] "6.0"                        "W  63"
##  [5] "W  58"                      "L   4"
##  [7] "W  17"                      "W  16"
##  [9] "W  20"                      "W   7"
## [11] "MI"                         "14598900 / R: 1553   ->1663"
## [13] "N:2"                        "B"
## [15] "W"                          "B"
## [17] "W"                          "B"
## [19] "W"                          "B"
##
## [[4]]
##  [1] "3"                          "ADITYA BAJAJ"
##  [3] "6.0"                        "L   8"
##  [5] "W  61"                      "W  25"
##  [7] "W  21"                      "W  11"
##  [9] "W  13"                      "W  12"
## [11] "MI"                         "14959604 / R: 1384   ->1640"
## [13] "N:2"                        "W"
## [15] "B"                          "W"
## [17] "B"                          "W"
## [19] "B"                          "W"
##
## [[5]]
##  [1] "4"                          "PATRICK H SCHILLING"
##  [3] "5.5"                        "W  23"
##  [5] "D  28"                      "W   2"
##  [7] "W  26"                      "D   5"
##  [9] "W  19"                      "D   1"
## [11] "MI"                         "12616049 / R: 1716   ->1744"
## [13] "N:2"                        "W"
## [15] "B"                          "W"
```

```
## [17] "B"                          "W"
## [19] "B"                          "B"
##
## [[6]]
##  [1] "5"                          "HANSHI ZUO"
##  [3] "5.5"                        "W  45"
##  [5] "W  37"                      "D  12"
##  [7] "D  13"                      "D   4"
##  [9] "W  14"                      "W  17"
## [11] "MI"                         "14601533 / R: 1655   ->1690"
## [13] "N:2"                        "B"
## [15] "W"                          "B"
## [17] "W"                          "B"
## [19] "W"                          "B"
```

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

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

```
##  [1] "Pair"                    "Player Name"
##  [3] "Total"                   "Round"
##  [5] "Round"                   "Round"
##  [7] "Round"                   "Round"
##  [9] "Round"                   "Round"
## [11] "Num"                     "USCF ID / Rtg (Pre->Post)"
## [13] "Pts"                     "1"
## [15] "2"                       "3"
## [17] "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"
##  [3] "TotalPts"                "Round1"
##  [5] "Round2"                  "Round3"
##  [7] "Round4"                  "Round5"
##  [9] "Round6"                  "Round7"
## [11] "State"                   "USCFID_RatePre_RatePost"
## [13] "Pts"                     "Color1"
## [15] "Color2"                  "Color3"
## [17] "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:

```
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) <- TrimmedOutputArray[[1]]
head(TrimmedOutputMatrix)
```

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

```
tail(TrimmedOutputMatrix)
```

```
##       ID   PlayerName             TotalPts Round1  Round2  Round3  Round4
## [59,] "59" "SEAN M MC CORMICK"    "2.0"    "L  41" "B"     "L   9" "L  40"
## [60,] "60" "JULIA SHEN"           "1.5"    "L  33" "L  34" "D  45" "D  42"
## [61,] "61" "JEZZEL FARKAS"        "1.5"    "L  32" "L   3" "W  54" "L  47"
## [62,] "62" "ASHWIN BALAJI"        "1.0"    "W  55" "U"     "U"     "U"
## [63,] "63" "THOMAS JOSEPH HOSMER" "1.0"    "L   2" "L  48" "D  49" "L  43"
## [64,] "64" "BEN LI"               "1.0"    "L  22" "D  30" "L  31" "D  49"
##       Round5  Round6  Round7  State USCFID_RatePre_RatePost          Pts
## [59,] "L  43" "W  54" "L  44" "MI"  "12841036 / R:  853   -> 878"    ""
## [60,] "L  24" "H"     "U"     "MI"  "14579262 / R:  967   -> 984"    ""
## [61,] "D  42" "L  30" "L  37" "ON"  "15771592 / R:  955P11-> 979P18" ""
## [62,] "U"     "U"     "U"     "MI"  "15219542 / R: 1530   ->1535"    ""
## [63,] "L  45" "H"     "U"     "MI"  "15057092 / R: 1175   ->1125"    ""
## [64,] "L  46" "L  42" "L  54" "MI"  "15006561 / R: 1163   ->1112"    ""
##       Color1 Color2 Color3 Color4 Color5 Color6 Color7
```

```
## [59,] "W"    ""     "B"    "B"    "W"    "W"    "B"
## [60,] "W"    "B"    "B"    "W"    "B"    ""     ""
## [61,] "B"    "W"    "B"    "W"    "B"    "W"    "B"
## [62,] "B"    ""     ""     ""     ""     ""     ""
## [63,] "W"    "B"    "W"    "B"    "B"    ""     ""
## [64,] "B"    "W"    "W"    "B"    "W"    "B"    "B"
```

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

This is consistent with having 20 columns on each row.

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
## [15] 1220 1604 1629 1600 1564 1595 1563 1555 1363 1229 1745 1579 1552 1507
## [29] 1602 1522 1494 1441 1449 1399 1438 1355  980 1423 1436 1348 1403 1332
## [43] 1283 1199 1242  377 1362 1382 1291 1056 1011  935 1393 1270 1186 1153
## [57] 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
## [15] 1416 1613 1610 1600 1570 1569 1562 1529 1371 1300 1681 1564 1539 1513
## [29] 1508 1444 1444 1433 1421 1400 1392 1367 1077 1439 1413 1346 1341 1256
## [43] 1244 1199 1191 1076 1341 1335 1259 1111 1097 1092 1359 1200 1163 1140
## [57] 1079  941  878  984  979 1535 1125 1112
```

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 correspondig to each game played (or, not played) per p
WLD = mapply(function(x) str_extract(string = x, pattern = "^."),subset(x = df, select = Round1:Round7))
cat('Win-Loss-Draw for the various players: \n')
```

```
## Win-Loss-Draw for the various players:
tail(WLD)
```

```
##       Round1 Round2 Round3 Round4 Round5 Round6 Round7
## [59,] "L"    "B"    "L"    "L"    "L"    "W"    "L"
## [60,] "L"    "L"    "D"    "D"    "L"    "H"    "U"
## [61,] "L"    "L"    "W"    "L"    "D"    "L"    "L"
## [62,] "W"    "U"    "U"    "U"    "U"    "U"    "U"
## [63,] "L"    "L"    "D"    "L"    "L"    "H"    "U"
## [64,] "L"    "D"    "L"    "D"    "L"    "L"    "L"
```

```
### 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: \n')
```

```
## The above, collapsed into a string for each player:
tail(collapseWLD)
```

```
## [1] "LBLLLWL" "LLDDLHU" "LLWLDLL" "WUUUUUU" "LLDLLHU" "LDLDLLL"
```

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

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

```
## table tallying results for each competitor (only the final 6 displayed here:)
tail(tempResults1)
```

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

```
## [[6]]
##
## D L
## 2 5
```

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
## [36] 6 5 6 7 7 4 7 7 6 7 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)
opponents
```

```
##       Opp1 Opp2 Opp3 Opp4 Opp5 Opp6 Opp7
##  [1,]   39   21   18   14    7   12    4
##  [2,]   63   58    4   17   16   20    7
##  [3,]    8   61   25   21   11   13   12
##  [4,]   23   28    2   26    5   19    1
##  [5,]   45   37   12   13    4   14   17
##  [6,]   34   29   11   35   10   27   21
##  [7,]   57   46   13   11    1    9    2
##  [8,]    3   32   14    9   47   28   19
##  [9,]   25   18   59    8   26    7   20
## [10,]   16   19   55   31    6   25   18
## [11,]   38   56    6    7    3   34   26
## [12,]   42   33    5   38   NA    1    3
## [13,]   36   27    7    5   33    3   32
## [14,]   54   44    8    1   27    5   31
## [15,]   19   16   30   22   54   33   38
## [16,]   10   15   NA   39    2   36   NA
## [17,]   48   41   26    2   23   22    5
## [18,]   47    9    1   32   19   38   10
## [19,]   15   10   52   28   18    4    8
## [20,]   40   49   23   41   28    2    9
## [21,]   43    1   47    3   40   39    6
## [22,]   64   52   28   15   NA   17   40
```

```
## [23,]    4   43   20   58   17   37   46
## [24,]   28   47   43   25   60   44   39
## [25,]    9   53    3   24   34   10   47
## [26,]   49   40   17    4    9   32   11
## [27,]   51   13   46   37   14    6   NA
## [28,]   24    4   22   19   20    8   36
## [29,]   50    6   38   34   52   48   NA
## [30,]   52   64   15   55   31   61   50
## [31,]   58   55   64   10   30   50   14
## [32,]   61    8   44   18   51   26   13
## [33,]   60   12   50   36   13   15   51
## [34,]    6   60   37   29   25   11   52
## [35,]   46   38   56    6   57   52   48
## [36,]   13   57   51   33   NA   16   28
## [37,]   NA    5   34   27   NA   23   61
## [38,]   11   35   29   12   NA   18   15
## [39,]    1   54   40   16   44   21   24
## [40,]   20   26   39   59   21   56   22
## [41,]   59   17   58   20   NA   NA   NA
## [42,]   12   50   57   60   61   64   56
## [43,]   21   23   24   63   59   46   55
## [44,]   NA   14   32   53   39   24   59
## [45,]    5   51   60   56   63   55   58
## [46,]   35    7   27   50   64   43   23
## [47,]   18   24   21   61    8   51   25
## [48,]   17   63   NA   52   NA   29   35
## [49,]   26   20   63   64   58   NA   NA
## [50,]   29   42   33   46   NA   31   30
## [51,]   27   45   36   57   32   47   33
## [52,]   30   22   19   48   29   35   34
## [53,]   NA   25   NA   44   NA   57   NA
## [54,]   14   39   61   NA   15   59   64
## [55,]   62   31   10   30   NA   45   43
## [56,]   NA   11   35   45   NA   40   42
## [57,]    7   36   42   51   35   53   NA
## [58,]   31    2   41   23   49   NA   45
## [59,]   41   NA    9   40   43   54   44
## [60,]   33   34   45   42   24   NA   NA
## [61,]   32    3   54   47   42   30   37
## [62,]   55   NA   NA   NA   NA   NA   NA
## [63,]    2   48   49   43   45   NA   NA
## [64,]   22   30   31   49   46   42   54
```

Append the 7 opponent columns to the dataframe:

```
### 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,opponents)
head(df)
```

```
##   ID          PlayerName TotalPts Round1 Round2 Round3 Round4 Round5
## 1  1            GARY HUA      6.0  W  39  W  21  W  18  W  14  W   7
## 2  2     DAKSHESH DARURI      6.0  W  63  W  58  L   4  W  17  W  16
## 3  3        ADITYA BAJAJ      6.0  L   8  W  61  W  25  W  21  W  11
```

```
## 4  4 PATRICK H SCHILLING      5.5  W  23  D  28  W   2  W  26  D   5
## 5  5          HANSHI ZUO      5.5  W  45  W  37  D  12  D  13  D   4
## 6  6         HANSEN SONG      5.0  W  34  D  29  L  11  W  35  D  10
##   Round6 Round7 State     USCFID_RatePre_RatePost Pts Color1 Color2 Color3
## 1  D  12  D   4    ON 15445895 / R: 1794   ->1817 N:2      W      B      W
## 2  W  20  W   7    MI 14598900 / R: 1553   ->1663 N:2      B      W      B
## 3  W  13  W  12    MI 14959604 / R: 1384   ->1640 N:2      W      B      W
## 4  W  19  D   1    MI 12616049 / R: 1716   ->1744 N:2      W      B      W
## 5  W  14  W  17    MI 14601533 / R: 1655   ->1690 N:2      B      W      B
## 6  W  27  W  21    OH 15055204 / R: 1686   ->1687 N:3      W      B      W
##   Color4 Color5 Color6 Color7   USCFID RatePre RatePost numgames Opp1 Opp2
## 1      B      W      B      W 15445895    1794     1817        7   39   21
## 2      W      B      W      B 14598900    1553     1663        7   63   58
## 3      B      W      B      W 14959604    1384     1640        7    8   61
## 4      B      W      B      B 12616049    1716     1744        7   23   28
## 5      W      B      W      B 14601533    1655     1690        7   45   37
## 6      B      B      W      B 15055204    1686     1687        7   34   29
##   Opp3 Opp4 Opp5 Opp6 Opp7
## 1   18   14    7   12    4
## 2    4   17   16   20    7
## 3   25   21   11   13   12
## 4    2   26    5   19    1
## 5   12   13    4   14   17
## 6   11   35   10   27   21
```

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:
```

Append the 7 opponent rating columns to the dataframe:

```
### CAUTION -- this would duplicate columns if this chunk is re-evaluated manually,
### but this is not a problem when knitting the entire result

oppPreRatings
```

```
##       OppPreRate1 OppPreRate2 OppPreRate3 OppPreRate4 OppPreRate5
##  [1,]        1436        1563        1600        1610        1649
##  [2,]        1175         917        1716        1629        1604
##  [3,]        1641         955        1745        1563        1712
##  [4,]        1363        1507        1553        1579        1655
##  [5,]        1242         980        1663        1666        1716
##  [6,]        1399        1602        1712        1438        1365
##  [7,]        1092         377        1666        1712        1794
##  [8,]        1384        1441        1610        1411        1362
##  [9,]        1745        1600         853        1641        1579
## [10,]        1604        1564        1186        1494        1686
```

```
## [11,]       1423       1153       1686       1649       1384
## [12,]       1332       1449       1655       1423         NA
## [13,]       1355       1552       1649       1655       1449
## [14,]       1270       1199       1641       1794       1552
## [15,]       1564       1604       1522       1555       1270
## [16,]       1365       1220         NA       1436       1553
## [17,]       1382       1403       1579       1553       1363
## [18,]       1362       1411       1794       1441       1564
## [19,]       1220       1365        935       1507       1600
## [20,]       1348       1291       1363       1403       1507
## [21,]       1283       1794       1362       1384       1348
## [22,]       1163        935       1507       1220         NA
## [23,]       1716       1283       1595        917       1629
## [24,]       1507       1362       1283       1745        967
## [25,]       1411       1393       1384       1229       1399
## [26,]       1291       1348       1629       1716       1411
## [27,]       1011       1666        377        980       1610
## [28,]       1229       1716       1555       1564       1595
## [29,]       1056       1686       1423       1399        935
## [30,]        935       1163       1220       1186       1494
## [31,]        917       1186       1163       1365       1522
## [32,]        955       1641       1199       1600       1011
## [33,]        967       1663       1056       1355       1666
## [34,]       1686        967        980       1602       1745
## [35,]        377       1423       1153       1686       1092
## [36,]       1666       1092       1011       1449         NA
## [37,]         NA       1655       1399       1552         NA
## [38,]       1712       1438       1602       1663         NA
## [39,]       1794       1270       1348       1604       1199
## [40,]       1595       1579       1436        853       1563
## [41,]        853       1629        917       1595         NA
## [42,]       1663       1056       1092        967        955
## [43,]       1563       1363       1229       1175        853
## [44,]         NA       1610       1441       1393       1436
## [45,]       1655       1011        967       1153       1175
## [46,]       1438       1649       1552       1056       1163
## [47,]       1600       1229       1563        955       1641
## [48,]       1629       1175         NA        935         NA
## [49,]       1579       1595       1175       1163        917
## [50,]       1602       1332       1449        377         NA
## [51,]       1552       1242       1355       1092       1441
## [52,]       1522       1555       1564       1382       1602
## [53,]         NA       1745         NA       1199         NA
## [54,]       1610       1436        955         NA       1220
## [55,]       1530       1494       1365       1522         NA
## [56,]         NA       1712       1438       1242         NA
## [57,]       1649       1355       1332       1011       1438
## [58,]       1494       1553       1403       1363       1291
## [59,]       1403         NA       1411       1348       1283
## [60,]       1449       1399       1242       1332       1229
## [61,]       1441       1384       1270       1362       1332
## [62,]       1186         NA         NA         NA         NA
## [63,]       1553       1382       1291       1283       1242
## [64,]       1555       1522       1494       1291        377
```

```
##       OppPreRate6 OppPreRate7
##  [1,]        1663        1716
##  [2,]        1595        1649
##  [3,]        1666        1663
##  [4,]        1564        1794
##  [5,]        1610        1629
##  [6,]        1552        1563
##  [7,]        1411        1553
##  [8,]        1507        1564
##  [9,]        1649        1595
## [10,]        1745        1600
## [11,]        1399        1579
## [12,]        1794        1384
## [13,]        1384        1441
## [14,]        1655        1494
## [15,]        1449        1423
## [16,]        1355          NA
## [17,]        1555        1655
## [18,]        1423        1365
## [19,]        1716        1641
## [20,]        1553        1411
## [21,]        1436        1686
## [22,]        1629        1348
## [23,]         980         377
## [24,]        1199        1436
## [25,]        1365        1362
## [26,]        1441        1712
## [27,]        1686          NA
## [28,]        1641        1355
## [29,]        1382          NA
## [30,]         955        1056
## [31,]        1056        1610
## [32,]        1579        1666
## [33,]        1220        1011
## [34,]        1712         935
## [35,]         935        1382
## [36,]        1604        1507
## [37,]        1363         955
## [38,]        1600        1220
## [39,]        1563        1229
## [40,]        1153        1555
## [41,]          NA          NA
## [42,]        1163        1153
## [43,]         377        1186
## [44,]        1229         853
## [45,]        1186         917
## [46,]        1283        1363
## [47,]        1011        1745
## [48,]        1602        1438
## [49,]          NA          NA
## [50,]        1494        1522
## [51,]        1362        1449
## [52,]        1438        1399
## [53,]        1092          NA
```

```
## [54,]         853        1163
## [55,]        1242        1283
## [56,]        1348        1332
## [57,]        1393          NA
## [58,]          NA        1242
## [59,]        1270        1199
## [60,]          NA          NA
## [61,]        1522         980
## [62,]          NA          NA
## [63,]          NA          NA
## [64,]        1332        1270
```

```
df <- cbind(df,oppPreRatings)
```

Compute the average opponent rating, for each participant:

```
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] 1605.286 1469.286 1563.571 1573.571 1500.857 1518.714 1372.143
##  [8] 1468.429 1523.143 1554.143 1467.571 1506.167 1497.857 1515.000
## [15] 1483.857 1385.800 1498.571 1480.000 1426.286 1410.857 1470.429
## [22] 1300.333 1213.857 1357.000 1363.286 1506.857 1221.667 1522.143
## [29] 1313.500 1144.143 1259.857 1378.714 1276.857 1375.286 1149.714
## [36] 1388.167 1384.800 1539.167 1429.571 1390.571 1248.500 1149.857
## [43] 1106.571 1327.000 1152.000 1357.714 1392.000 1355.800 1285.800
## [50] 1296.000 1356.143 1494.571 1345.333 1206.167 1406.000 1414.400
## [57] 1363.000 1391.000 1319.000 1330.200 1327.286 1186.000 1350.200
## [64] 1263.000
```

```
### 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] 1605.286 1469.286 1563.571 1573.571 1500.857 1518.714 1372.143
##  [8] 1468.429 1523.143 1554.143 1467.571 1506.167 1497.857 1515.000
## [15] 1483.857 1385.800 1498.571 1480.000 1426.286 1410.857 1470.429
## [22] 1300.333 1213.857 1357.000 1363.286 1506.857 1221.667 1522.143
## [29] 1313.500 1144.143 1259.857 1378.714 1276.857 1375.286 1149.714
## [36] 1388.167 1384.800 1539.167 1429.571 1390.571 1248.500 1149.857
## [43] 1106.571 1327.000 1152.000 1357.714 1392.000 1355.800 1285.800
## [50] 1296.000 1356.143 1494.571 1345.333 1206.167 1406.000 1414.400
## [57] 1363.000 1391.000 1319.000 1330.200 1327.286 1186.000 1350.200
## [64] 1263.000
```

```
### 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
```

```
## [36] 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
```

The input data has the names entirely capitalized:

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

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

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

```
## [1] "GARY HUA"            "DAKSHESH DARURI"     "ADITYA BAJAJ"
## [4] "PATRICK H SCHILLING" "HANSHI ZUO"          "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 o
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, in array format: \n')
```

```
## Each name pasted back together again, in array format:
```

```
head(CapitalNameArray)
```

```
## [1] "Gary Hua"            "Dakshesh Daruri"     "Aditya Bajaj"
```

```
## [4] "Patrick H Schilling" "Hanshi Zuo"          "Hansen Song"
```

```
# add Capitalized Name to the data frame
df$CapitalizedName = CapitalNameArray
```

Select output columns for final result into smaller dataframe:

```
OutputColumns = c("CapitalizedName","State", "TotalPts", "RatePre", "AvgOppRatePre")
OutputDF <- 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 = OutputDF) <- OutputHeaders
head(OutputDF)
```

```
##         Player's Name Player's State Total Number of Points
## 1            Gary Hua             ON                    6.0
## 2     Dakshesh Daruri             MI                    6.0
## 3        Aditya Bajaj             MI                    6.0
## 4 Patrick H Schilling             MI                    5.5
## 5          Hanshi Zuo             MI                    5.5
## 6         Hansen Song             OH                    5.0
##   Player's Pre-Rating Average Pre Chess Rating of Opponents
## 1                1794                                1605.286
## 2                1553                                1469.286
## 3                1384                                1563.571
## 4                1716                                1573.571
## 5                1655                                1500.857
## 6                1686                                1518.714
```

write the results into a .csv file:

```
head(OutputDF)
```

```
##         Player's Name Player's State Total Number of Points
## 1            Gary Hua             ON                    6.0
## 2     Dakshesh Daruri             MI                    6.0
## 3        Aditya Bajaj             MI                    6.0
## 4 Patrick H Schilling             MI                    5.5
## 5          Hanshi Zuo             MI                    5.5
## 6         Hansen Song             OH                    5.0
##   Player's Pre-Rating Average Pre Chess Rating of Opponents
## 1                1794                                1605.286
## 2                1553                                1469.286
## 3                1384                                1563.571
## 4                1716                                1573.571
## 5                1655                                1500.857
## 6                1686                                1518.714
```

```
options(encoding = 'UTF-8')
filename <- "MY-DATA607-Week04-ChessData.csv"
```

```
write.csv(OutputDF, filename, row.names=FALSE)
```

re-read the results and display them:

```
re_read_output <- read_lines(filename)
cat (re_read_output, sep = '\n')
```

```
## "Player's Name","Player's State","Total Number of Points","Player's Pre-Rating","Average Pre Chess Ra
## "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
```