Project 1

CUNY MSDA - DATA607

Duubar Villalobos Jimenez mydvtech@gmail.com

February 26, 2017

Instructions

In this project, you're given a text file with chess tournament results where the information has some structure. Your job is to create an R Markdown file that generates a .CSV file (that could for example be imported into a SQL database) with the following information for all of the players: Player's Name, Player's State, Total Number of Points, Player's Pre-Rating, and Average Pre Chess Rating of Opponents For the first player, the information would be: Gary Hua, ON, 6.0, 1794, 1605 1605 was calculated by using the pre-tournament opponents' ratings of 1436, 1563, 1600, 1610, 1649, 1663, 1716, and dividing by the total number of games played.

If you have questions about the meaning of the data or the results, please post them on the discussion forum. Data science, like chess, is a game of back and forth...

The chess rating system (invented by a Minnesota statistician named Arpad Elo) has been used in many other contexts, including assessing relative strength of employment candidates by human resource departments.

You may substitute another text file (or set of text files, or data scraped from web pages) of similar or greater complexity, and create your own assignment and solution. You may work in a small team. All of your code should be in an R markdown file (and published to rpubs.com); with your data accessible for the person running the script.

Read data from URL file by using "read.delim" function.

I experienced problems using "read.table" function hense I used the "read.delim" function.

```
url <- 'https://raw.githubusercontent.com/dvillalobos/MSDA/master/607/Projects/Project1/tournamentinfo.t
my.data <- read.delim(url, header=FALSE, stringsAsFactors =FALSE )
head(my.data)
```

```
##                                                                                                 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    |
```

Clean line dividers

```
# Cleaning extra lines and eliminating Empty lines in between
head(my.data)
```

```
##                                                                                                 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    |
```

```
split_data <- data.frame(str_replace_all(my.data$V1,"-----------------------------------------------------
head(split_data)
```

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

```
# Deleting empty lines
split_data <- data.frame(split_data[!apply(split_data == "", 1, all),])
```

Combining two consecutive rows into one column

```
# Need to define an empty new_dataframe
new_table <- data.frame(c())
# Combining two consecutive rows into one column
for (i in 1:dim(split_data)[1]){
  if (i %% 2 == 1) {
    Part1 <- rbind(new_table$Part1, as.character(split_data[i,1]))
    Part2 <- as.character(split_data[i+1,1])
    Combined <- data.frame(paste0(Part1, Part2))
    names(Combined) <- "Combined"
    new_table <- rbind(new_table, Combined)
  }
}
head(new_table)
```

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

Spliting one more column for "|" separator

```
# Creating headers
Headers <- c("Pair","Player Name","Total","Round 1","Round 2","Round 3", "Round 4","Round 5","Round 6",
# Separating into columns
newer_table <- separate(data = new_table, col = Combined, into = Headers, sep = "\\|")
# Need to clear row containing all columns names
newer_table <- newer_table[-c(1), ]

head(newer_table)
```

```
##     Pair                      Player Name Total Round 1 Round 2 Round 3
## 2     1   GARY HUA                          6.0     W  39   W  21   W  18
## 3     2   DAKSHESH DARURI                   6.0     W  63   W  58   L   4
## 4     3   ADITYA BAJAJ                      6.0     L   8   W  61   W  25
## 5     4   PATRICK H SCHILLING               5.5     W  23   D  28   W   2
## 6     5   HANSHI ZUO                        5.5     W  45   W  37   D  12
## 7     6   HANSEN SONG                       5.0     W  34   D  29   L  11
##   Round 4 Round 5 Round 6 Round 7  State         USCF ID / Rtg (Pre->Post)
## 2   W  14   W   7   D  12   D   4    ON   15445895 / R: 1794   ->1817
## 3   W  17   W  16   W  20   W   7    MI   14598900 / R: 1553   ->1663
## 4   W  21   W  11   W  13   W  12    MI   14959604 / R: 1384   ->1640
## 5   W  26   D   5   W  19   D   1    MI   12616049 / R: 1716   ->1744
## 6   D  13   D   4   W  14   W  17    MI   14601533 / R: 1655   ->1690
## 7   W  35   D  10   W  27   W  21    OH   15055204 / R: 1686   ->1687
##     Pts     1     2     3     4     5     6     7 Ave Pre Rating
## 2 N:2   W     B     W     B     W     B     W
## 3 N:2   B     W     B     W     B     W     B
## 4 N:2   W     B     W     B     W     B     W
## 5 N:2   W     B     W     B     W     B     B
## 6 N:2   B     W     B     W     B     W     B
## 7 N:3   W     B     W     B     B     W     B
```

Spliting extra columns that were not splitted

```
# Extracting numerical values from "USCF ID / Rtg (Pre->Post)" unsplitted column
temp <- str_extract_all(newer_table$`USCF ID / Rtg (Pre->Post)`,"\\b\\d{1,}")
temp <- data.frame(as.character(temp))
# Separating the data frame from one couln to three different columns
temp <- separate(data = temp, col = as.character.temp., into = c("col1","col2","col3"), sep = ",")
kable(head(temp))
```

col1	col2	col3
c(“15445895”	“1794”	“1817”)
c(“14598900”	“1553”	“1663”)
c(“14959604”	“1384”	“1640”)
c(“12616049”	“1716”	“1744”)
c(“14601533”	“1655”	“1690”)
c(“15055204”	“1686”	“1687”)

```
# Temporary column vectors
col1 <- str_extract_all(temp$col1,"[[:digit:]]{1,}")
col2 <- str_extract_all(temp$col2,"[[:digit:]]{1,}")
col3 <- str_extract_all(temp$col3,"[[:digit:]]{1,}")

newer_table$`USCF ID` <- as.character(col1)
newer_table$`Pre Rating` <- as.character(col2)
newer_table$`Post Rating` <- as.character(col3)
head(newer_table)
```

```
##     Pair                      Player Name Total Round 1 Round 2 Round 3
## 2     1   GARY HUA                          6.0     W  39   W  21   W  18
## 3     2   DAKSHESH DARURI                   6.0     W  63   W  58   L   4
## 4     3   ADITYA BAJAJ                      6.0     L   8   W  61   W  25
```

```
## 5     4   PATRICK H SCHILLING                 5.5     W  23   D  28   W   2
## 6     5   HANSHI ZUO                          5.5     W  45   W  37   D  12
## 7     6   HANSEN SONG                         5.0     W  34   D  29   L  11
##   Round 4 Round 5 Round 6 Round 7  State         USCF ID / Rtg (Pre->Post)
## 2   W  14   W   7   D  12   D   4    ON   15445895 / R: 1794   ->1817
## 3   W  17   W  16   W  20   W   7    MI   14598900 / R: 1553   ->1663
## 4   W  21   W  11   W  13   W  12    MI   14959604 / R: 1384   ->1640
## 5   W  26   D   5   W  19   D   1    MI   12616049 / R: 1716   ->1744
## 6   D  13   D   4   W  14   W  17    MI   14601533 / R: 1655   ->1690
## 7   W  35   D  10   W  27   W  21    OH   15055204 / R: 1686   ->1687
##     Pts     1     2     3     4     5     6     7 Ave Pre Rating  USCF ID
## 2 N:2   W     B     W     B     W     B     W                     15445895
## 3 N:2   B     W     B     W     B     W     B                     14598900
## 4 N:2   W     B     W     B     W     B     W                     14959604
## 5 N:2   W     B     W     B     W     B     B                     12616049
## 6 N:2   B     W     B     W     B     W     B                     14601533
## 7 N:3   W     B     W     B     B     W     B                     15055204
##   Pre Rating Post Rating
## 2       1794        1817
## 3       1553        1663
## 4       1384        1640
## 5       1716        1744
## 6       1655        1690
## 7       1686        1687
```

Separating needed columns to include in .csv file

```
csv.table <- subset(newer_table, select = c(1,2,11,3,22,23,24,21))
kable(head(csv.table))
```

	Pair	Player Name	State	Total	USCF ID	Pre Rating	Post Rating	Ave Pre Rating
2	1	GARY HUA	ON	6.0	15445895	1794	1817	
3	2	DAKSHESH DARURI	MI	6.0	14598900	1553	1663	
4	3	ADITYA BAJAJ	MI	6.0	14959604	1384	1640	
5	4	PATRICK H SCHILLING	MI	5.5	12616049	1716	1744	
6	5	HANSHI ZUO	MI	5.5	14601533	1655	1690	
7	6	HANSEN SONG	OH	5.0	15055204	1686	1687	

Calculating Ave Pre Rating

```
# Creating the opponent values from the unsplitted data frame
opponent1 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 1`,"[[:digit:]]{1,}")))
opponent2 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 2`,"[[:digit:]]{1,}")))
opponent3 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 3`,"[[:digit:]]{1,}")))
opponent4 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 4`,"[[:digit:]]{1,}")))
opponent5 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 5`,"[[:digit:]]{1,}")))
opponent6 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 6`,"[[:digit:]]{1,}")))
opponent7 <- data.frame(as.numeric(str_extract_all(newer_table$`Round 7`,"[[:digit:]]{1,}")))

# Creeating Opponents data frame.
```

```r
opponents <- cbind(opponent1, opponent2, opponent3, opponent4, opponent5, opponent6, opponent7)
names(opponents) <- c("Opp 1","Opp 2","Opp 3","Opp 4","Opp 5","Opp 6","Opp 7")

# Finding number of games played
for(i in 1:dim(opponents)[1]){
    opponents$playedGames[i] <- 7- (as.numeric(sum((is.na(opponents[i,])))))
}

# Reporting  table to view the opponents table
kable(opponents)
```

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

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

```
# Eliminating NA Cases in order to continue with our calculations, NA replaced by 0.
opponents[is.na(opponents)] <- as.numeric(0)
csv.table[is.na(csv.table$Pair)] <- as.numeric(-1)

# Need to assign zero values in order to add accordingly and to avoid errors.
csv.table$`Ave Pre Rating` <- as.numeric(0)

# Procedure to calculate Average Pre-Rating for each player
for (k in 1:7){
  for (j in 1:dim(csv.table)[1]){
    for (i in 1:dim(csv.table)[1]){
      if (as.numeric(opponents[i,k]) == as.numeric(csv.table$Pair[j])){
        csv.table$`Ave Pre Rating`[j] <- as.numeric(csv.table$`Ave Pre Rating`[j]) + as.numeric(csv.tabl
      }
    }
  }
}

# Final Procedure to find each player's average based on the number of played games
csv.table$`Ave Pre Rating`  <- round(as.numeric(csv.table$`Ave Pre Rating`) / opponents$playedGames,0)
```

Finalized table ready to export

```
kable(csv.table, row.names = FALSE)
```

Pair	Player Name	State	Total	USCF ID	Pre Rating	Post Rating	Ave Pre Rating
1	GARY HUA	ON	6.0	15445895	1794	1817	1605
2	DAKSHESH DARURI	MI	6.0	14598900	1553	1663	1469
3	ADITYA BAJAJ	MI	6.0	14959604	1384	1640	1564
4	PATRICK H SCHILLING	MI	5.5	12616049	1716	1744	1574
5	HANSHI ZUO	MI	5.5	14601533	1655	1690	1501
6	HANSEN SONG	OH	5.0	15055204	1686	1687	1519
7	GARY DEE SWATHELL	MI	5.0	11146376	1649	1673	1372
8	EZEKIEL HOUGHTON	MI	5.0	15142253	1641	1657	1468
9	STEFANO LEE	ON	5.0	14954524	1411	1564	1523
10	ANVIT RAO	MI	5.0	14150362	1365	1544	1554
11	CAMERON WILLIAM MC LEMAN	MI	4.5	12581589	1712	1696	1468
12	KENNETH J TACK	MI	4.5	12681257	1663	1670	1506
13	TORRANCE HENRY JR	MI	4.5	15082995	1666	1662	1498
14	BRADLEY SHAW	MI	4.5	10131499	1610	1618	1515
15	ZACHARY JAMES HOUGHTON	MI	4.5	15619130	1220	1416	1484
16	MIKE NIKITIN	MI	4.0	10295068	1604	1613	1386
17	RONALD GRZEGORCZYK	MI	4.0	10297702	1629	1610	1499
18	DAVID SUNDEEN	MI	4.0	11342094	1600	1600	1480
19	DIPANKAR ROY	MI	4.0	14862333	1564	1570	1426
20	JASON ZHENG	MI	4.0	14529060	1595	1569	1411
21	DINH DANG BUI	ON	4.0	15495066	1563	1562	1470
22	EUGENE L MCCLURE	MI	4.0	12405534	1555	1529	1300
23	ALAN BUI	ON	4.0	15030142	1363	1371	1214
24	MICHAEL R ALDRICH	MI	4.0	13469010	1229	1300	1357
25	LOREN SCHWIEBERT	MI	3.5	12486656	1745	1681	1363
26	MAX ZHU	ON	3.5	15131520	1579	1564	1507
27	GAURAV GIDWANI	MI	3.5	14476567	1552	1539	1222
28	SOFIA ADINA STANESCU-BELLU	MI	3.5	14882954	1507	1513	1522
29	CHIEDOZIE OKORIE	MI	3.5	15323285	1602	1508	1314
30	GEORGE AVERY JONES	ON	3.5	12577178	1522	1444	1144
31	RISHI SHETTY	MI	3.5	15131618	1494	1444	1260
32	JOSHUA PHILIP MATHEWS	ON	3.5	14073750	1441	1433	1379
33	JADE GE	MI	3.5	14691842	1449	1421	1277
34	MICHAEL JEFFERY THOMAS	MI	3.5	15051807	1399	1400	1375
35	JOSHUA DAVID LEE	MI	3.5	14601397	1438	1392	1150
36	SIDDHARTH JHA	MI	3.5	14773163	1355	1367	1388
37	AMIYATOSH PWNANANDAM	MI	3.5	15489571	980	1077	1385
38	BRIAN LIU	MI	3.0	15108523	1423	1439	1539
39	JOEL R HENDON	MI	3.0	12923035	1436	1413	1430
40	FOREST ZHANG	MI	3.0	14892710	1348	1346	1391
41	KYLE WILLIAM MURPHY	MI	3.0	15761443	1403	1341	1248
42	JARED GE	MI	3.0	14462326	1332	1256	1150
43	ROBERT GLEN VASEY	MI	3.0	14101068	1283	1244	1107
44	JUSTIN D SCHILLING	MI	3.0	15323504	1199	1199	1327
45	DEREK YAN	MI	3.0	15372807	1242	1191	1152
46	JACOB ALEXANDER LAVALLEY	MI	3.0	15490981	377	1076	1358
47	ERIC WRIGHT	MI	2.5	12533115	1362	1341	1392
48	DANIEL KHAIN	MI	2.5	14369165	1382	1335	1356
49	MICHAEL J MARTIN	MI	2.5	12531685	1291	1259	1286
50	SHIVAM JHA	MI	2.5	14773178	1056	1111	1296
51	TEJAS AYYAGARI	MI	2.5	15205474	1011	1097	1356
52	ETHAN GUO	MI	2.5	14918803	935	1092	1495

Pair	Player Name	State	Total	USCF ID	Pre Rating	Post Rating	Ave Pre Rating
53	JOSE C YBARRA	MI	2.0	12578849	1393	1359	1345
54	LARRY HODGE	MI	2.0	12836773	1270	1200	1206
55	ALEX KONG	MI	2.0	15412571	1186	1163	1406
56	MARISA RICCI	MI	2.0	14679887	1153	1140	1414
57	MICHAEL LU	MI	2.0	15113330	1092	1079	1363
58	VIRAJ MOHILE	MI	2.0	14700365	917	941	1391
59	SEAN M MC CORMICK	MI	2.0	12841036	853	878	1319
60	JULIA SHEN	MI	1.5	14579262	967	984	1330
61	JEZZEL FARKAS	ON	1.5	15771592	955	979	1327
62	ASHWIN BALAJI	MI	1.0	15219542	1530	1535	1186
63	THOMAS JOSEPH HOSMER	MI	1.0	15057092	1175	1125	1350
64	BEN LI	MI	1.0	15006561	1163	1112	1263

Export csc file

```
#write.csv(csv.table, file = "Villalobos-tournamentInfo.csv")
write.table(csv.table, file = "Villalobos-tournamentInfo.csv",row.names=FALSE, na="",col.names=TRUE, sep
```