DATA 607 Project 1 Revised

Vinicio Haro

March 2, 2018

The raw data can be found here https://raw.githubusercontent.com/vindication09/DATA607_Project1/master/RawChessData

```
ELOsheet <-
read.csv(paste0("https://raw.githubusercontent.com/vindication09/DATA607_Proj
ect1/master/RawChessData"))
head(ELOsheet)

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

goal: we want to create a structured data set that can be uploaded into MySQL that contains the following columns Player's Name, Player's State, Total Number of Points, Player's Pre-Rating, and Average Pre Rating of Opponents.

Before doing anything, I notice that there are headings in the first 3 rows. Lets remove them

```
ELOsheet2<-ELOsheet[-c(1:2),]
head(ELOsheet2, 10)

##  [1] ---------------------------------------------------------------------
--------------------
##  [2]     1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W
14|W   7|D  12|D   4|
##  [3]    ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B
|W    |B    |W    |
##  [4] ---------------------------------------------------------------------
```

```
--------------------
##  [5]     2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W
17|W  16|W  20|W   7|
##  [6]    MI | 14598900 / R: 1553   ->1663     |N:2  |B    |W    |B    |W
|B    |W    |B    |
##  [7] ------------------------------------------------------------------
--------------------
##  [8]     3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W
21|W  11|W  13|W  12|
##  [9]    MI | 14959604 / R: 1384   ->1640     |N:2  |W    |B    |W    |B
|W    |B    |W    |
## [10] ------------------------------------------------------------------
--------------------
## 131 Levels: ------------------------------------------------------------
-------------------------- ...
```

we need to use the stingr package in order to massage the data and extract what we need. The DT library is used to make the data.

```
library(stringr)
library(DT)

## Warning: package 'DT' was built under R version 3.4.3
```

We want to obtain the names of players from the score sheet. I notice that there is a specific structure to each row. With the heading gone, names, scores, and opponents are found in the first row of each subsection. States, ids, and pre rating are all on the second row of each subsection

I can use this to my advantge and create a subset of data that grabs the first row of each subsection and another that grabs the 2nd row of each subsection.

We will call these subsheets ELOsubsheet1 and ELOsubsheet 2. They are defined as follows:

```
#to grab the 1st row of each subsection, I want to skip the first row, grab
the second, skip the third and 4th then repeat
ELOsubsheet1<-ELOsheet2[seq(2, length(ELOsheet2), 3)]
head(ELOsubsheet1)

## [1]     1 | GARY HUA                       |6.0  |W  39|W  21|W  18|W
14|W   7|D  12|D   4|
## [2]     2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W
17|W  16|W  20|W   7|
## [3]     3 | ADITYA BAJAJ                    |6.0  |L   8|W  61|W  25|W
21|W  11|W  13|W  12|
## [4]     4 | PATRICK H SCHILLING             |5.5  |W  23|D  28|W   2|W
26|D   5|W  19|D   1|
## [5]     5 | HANSHI ZUO                      |5.5  |W  45|W  37|D  12|D
13|D   4|W  14|W  17|
## [6]     6 | HANSEN SONG                     |5.0  |W  34|D  29|L  11|W
```

```
35|D  10|W  27|W  21|
## 131 Levels: ----------------------------------------------------------
------------------------- ...

#to grab the 1st row of each subsection, I want to skip the first row, grab
the second, skip the third and 4th then repeat
ELOsubsheet2<-ELOsheet2[seq(3, length(ELOsheet2), 3)]
head(ELOsubsheet2)

## [1]    ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B
|W    |B    |W    |
## [2]    MI | 14598900 / R: 1553   ->1663     |N:2  |B    |W    |B    |W
|B    |W    |B    |
## [3]    MI | 14959604 / R: 1384   ->1640     |N:2  |W    |B    |W    |B
|W    |B    |W    |
## [4]    MI | 12616049 / R: 1716   ->1744     |N:2  |W    |B    |W    |B
|W    |B    |B    |
## [5]    MI | 14601533 / R: 1655   ->1690     |N:2  |B    |W    |B    |W
|B    |W    |B    |
## [6]    OH | 15055204 / R: 1686   ->1687     |N:3  |W    |B    |W    |B
|B    |W    |B    |
## 131 Levels: ----------------------------------------------------------
------------------------- ...
```

1) Names from ELOsubsheet1

```
ELOname <- str_trim(str_extract(ELOsubsheet1, "(\\w+\\s){2,3}"))
df.ELOname <- data.frame(ELOname)
head(df.ELOname)

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

2) States from ELOsubsheet2

```
ELOstate <- str_extract(ELOsubsheet2, "\\w+")
df.ELOstate <- data.frame(ELOstate)
head(df.ELOstate)

##   ELOstate
## 1       ON
## 2       MI
## 3       MI
## 4       MI
## 5       MI
## 6       OH
```

3) Total Points from ELOsubsheet1

```
ELOtotalpoints <- as.numeric(str_extract(ELOsubsheet1, "\\d+\\.\\d+"))
df.ELOtotalpoints <- data.frame(as.numeric(ELOtotalpoints))
head(df.ELOtotalpoints)
```

```
##   as.numeric.ELOtotalpoints.
## 1                        6.0
## 2                        6.0
## 3                        6.0
## 4                        5.5
## 5                        5.5
## 6                        5.0
```

4) Player Pre Rating from ELOsubsheet2

```
ELOprerating <- as.integer(str_extract(str_extract(ELOsubsheet2,
"[^\\d]\\d{3,4}[^\\d]"), "\\d+"))
ELOprerating <- as.numeric(ELOprerating)
df.ELOprerating<-data.frame(as.numeric(ELOprerating))
head(df.ELOprerating)
```

```
##   as.numeric.ELOprerating.
## 1                     1794
## 2                     1553
## 3                     1384
## 4                     1716
## 5                     1655
## 6                     1686
```

Lets check our progress. We can put together a partial csv to make sure the data has been collected correctly so far.

```
partialcsv<-data.frame(df.ELOname, df.ELOstate, df.ELOtotalpoints,
ELOprerating)
head(partialcsv)
```

```
##               ELOname ELOstate as.numeric.ELOtotalpoints. ELOprerating
## 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
```

5) Average opponent Pre-Rating

5a) Opponents by their player number from ELOsubsheet1

```
ELOopponent<-str_extract_all(str_extract_all(ELOsubsheet1, "\\d+\\|"),
"\\d+")
```

5b) Player ID number from ELOsubsheet1

```
ELOplayer<-as.integer(str_extract(ELOsubsheet1, "\\d+"))
df.ELOplayer<-data.frame(as.integer(ELOplayer))
head(df.ELOplayer)
```

```
##   as.integer.ELOplayer.
## 1                     1
## 2                     2
## 3                     3
## 4                     4
## 5                     5
## 6                     6
```

We have collected a list of the player numbers and the opponents. To compute the average pre-rating or each player, we need to write a loop.

The loop then fetches the pre ratings for each opponent per player number and divides by number of rounds played.

I noticed that I ended up with over 100 rows. After the 64th row, every row showed up as NA. A quick fix was to use the Na omit capability.

```
avg_ELOopp_rating <- length(ELOsheet2)
for (i in 1:length(ELOsheet2))
{
  avg_ELOopp_rating[i] <-
round(mean(ELOprerating[as.numeric(unlist(ELOopponent[ELOplayer[i]]))]),
digits = 0)
}
head(avg_ELOopp_rating)
```

```
## [1] 1605 1469 1564 1574 1501 1519
```

```
#Put the avg ratings in a data frame
df.avg_ELOopp_rating<-data.frame(na.omit(avg_ELOopp_rating))
head(df.avg_ELOopp_rating)
```

```
##   na.omit.avg_ELOopp_rating.
## 1                       1605
## 2                       1469
## 3                       1564
## 4                       1574
## 5                       1501
## 6                       1519
```

In order to validate the data, it is encouraged to check the averages by hand for the first few rows and the last few rows and see if they match up to the averages produced by the loop.

Put together in a data frame

```
csv<-data.frame(df.ELOplayer, df.ELOname, df.ELOstate, df.ELOtotalpoints,
df.ELOprerating, df.avg_ELOopp_rating)
head(csv)
```

```
##   as.integer.ELOplayer.             ELOname ELOstate
## 1                     1           GARY HUA       ON
## 2                     2     DAKSHESH DARURI       MI
## 3                     3        ADITYA BAJAJ       MI
## 4                     4 PATRICK H SCHILLING       MI
## 5                     5          HANSHI ZUO       MI
## 6                     6         HANSEN SONG       OH
##   as.numeric.ELOtotalpoints. as.numeric.ELOprerating.
## 1                        6.0                      1794
## 2                        6.0                      1553
## 3                        6.0                      1384
## 4                        5.5                      1716
## 5                        5.5                      1655
## 6                        5.0                      1686
##   na.omit.avg_ELOopp_rating.
## 1                       1605
## 2                       1469
## 3                       1564
## 4                       1574
## 5                       1501
## 6                       1519
```

We can change the names of the columns to make the data more user friendly.

```
#use a better naming convention
colnames(csv)[colnames(csv)=="as.integer.ELOplayer."]<-"PlayerNumber"
colnames(csv)[colnames(csv)=="ELOname"]<-"Name"
colnames(csv)[colnames(csv)=="ELOstate"]<-"State"
colnames(csv)[colnames(csv)=="as.numeric.ELOtotalpoints."]<-"TotalPoints"
colnames(csv)[colnames(csv)=="ELOprerating"]<-"PreRating"
colnames(csv)[colnames(csv)=="na.omit.avg_ELOopp_ratingb."]<-
"AvgOppPreRating"
head(csv)
```

```
##   PlayerNumber               Name State TotalPoints
## 1            1           GARY HUA    ON         6.0
## 2            2     DAKSHESH DARURI    MI         6.0
## 3            3        ADITYA BAJAJ    MI         6.0
## 4            4 PATRICK H SCHILLING    MI         5.5
## 5            5          HANSHI ZUO    MI         5.5
## 6            6         HANSEN SONG    OH         5.0
##   as.numeric.ELOprerating. na.omit.avg_ELOopp_rating.
## 1                     1794                       1605
## 2                     1553                       1469
## 3                     1384                       1564
## 4                     1716                       1574
## 5                     1655                       1501
## 6                     1686                       1519
```

Export the csv to a location of your source. The data is now in a format that can be loaded into mysql or any local database.

Change the destination to one of your own choosing using the following command:
write.table(csv, “C:/Users/traveler/Desktop/”)