CUNY DATA607_Wk3_Herold_Chess

Project 1: Wrangling data from a Chess Tournament Cross Table

Zachary Herold, Submitted 9/20/18

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.1
```

```
chess <- NULL
opponents.rating.matrix <- NULL
```

Loading the Chess Tournament Cross Table from the Raw Github URL. Removing the dash lines, with solution by redmode. See: https://stackoverflow.com/questions/21114598/importing-a-text-file-into-r (https://stackoverflow.com/questions/21114598/importing-a-text-file-into-r)

```
url <- "https://raw.githubusercontent.com/ZacharyHerold/chinafundnews/master/tournamentinfo.txt"

lines <- readLines(url)
```

```
## Warning in readLines(url): incomplete final line found on 'https://
## raw.githubusercontent.com/ZacharyHerold/chinafundnews/master/
## tournamentinfo.txt'
```

```
lines <- sapply(lines, gsub, pattern="[-]{2,}|[|]", replacement="")
lines <- c(lines[2], lines[lines!="" & lines!=lines[2]])

head(lines)
```

```
##  Pair | Player Name                     |Total|Round|Round|Round|Round|Round|Round|Round|
##         " Pair  Player Name                     TotalRoundRoundRoundRoundRoundRoundRound "
##  Num  | USCF ID / Rtg (Pre->Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  |
##         " Num   USCF ID / Rtg (Pre->Post)        Pts   1    2    3    4    5    6    7   "
##      1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|
##          "    1  GARY HUA                        6.0  W  39W  21W  18W  14W   7D  12D   4"
##     ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |
##          "   ON  15445895 / R: 1794   ->1817     N:2  W    B    W    B    W    B    W    "
##      2 | DAKSHESH DARURI                 |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|
##          "    2  DAKSHESH DARURI                 6.0  W  63W  58L   4W  17W  16W  20W   7"
##     MI | 14598900 / R: 1553   ->1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |
##          "   MI  14598900 / R: 1553   ->1663     N:2  B    W    B    W    B    W    B    "
```

Extracting out the names with alphabetic characters, allowing for multiple names.

```
name <- unlist(str_extract_all(lines, "[[:alpha:]]{2,}([[:blank:]][[:alpha:]]{1,}){1,}"))

name <- name[-c(1,2)]
name
```

```
##  [1] "GARY HUA"                 "DAKSHESH DARURI"
##  [3] "ADITYA BAJAJ"             "PATRICK H SCHILLING"
##  [5] "HANSHI ZUO"               "HANSEN SONG"
##  [7] "GARY DEE SWATHELL"        "EZEKIEL HOUGHTON"
##  [9] "STEFANO LEE"              "ANVIT RAO"
## [11] "CAMERON WILLIAM MC LEMAN" "KENNETH J TACK"
## [13] "TORRANCE HENRY JR"        "BRADLEY SHAW"
## [15] "ZACHARY JAMES HOUGHTON"   "MIKE NIKITIN"
## [17] "RONALD GRZEGORCZYK"       "DAVID SUNDEEN"
## [19] "DIPANKAR ROY"             "JASON ZHENG"
## [21] "DINH DANG BUI"            "EUGENE L MCCLURE"
## [23] "ALAN BUI"                 "MICHAEL R ALDRICH"
## [25] "LOREN SCHWIEBERT"         "MAX ZHU"
## [27] "GAURAV GIDWANI"           "SOFIA ADINA STANESCU"
## [29] "CHIEDOZIE OKORIE"         "GEORGE AVERY JONES"
## [31] "RISHI SHETTY"             "JOSHUA PHILIP MATHEWS"
## [33] "JADE GE"                  "MICHAEL JEFFERY THOMAS"
## [35] "JOSHUA DAVID LEE"         "SIDDHARTH JHA"
## [37] "AMIYATOSH PWNANANDAM"     "BRIAN LIU"
## [39] "JOEL R HENDON"            "FOREST ZHANG"
## [41] "KYLE WILLIAM MURPHY"      "JARED GE"
## [43] "ROBERT GLEN VASEY"        "JUSTIN D SCHILLING"
## [45] "DEREK YAN"                "JACOB ALEXANDER LAVALLEY"
## [47] "ERIC WRIGHT"              "DANIEL KHAIN"
## [49] "MICHAEL J MARTIN"         "SHIVAM JHA"
## [51] "TEJAS AYYAGARI"           "ETHAN GUO"
## [53] "JOSE C YBARRA"            "LARRY HODGE"
## [55] "ALEX KONG"                "MARISA RICCI"
## [57] "MICHAEL LU"               "VIRAJ MOHILE"
## [59] "SEAN M MC CORMICK"        "JULIA SHEN"
## [61] "JEZZEL FARKAS"            "ASHWIN BALAJI"
## [63] "THOMAS JOSEPH HOSMER"     "BEN LI"
```

Trimming the lines and taking the first two characters, then subsetting out the strings of two alphabetic characters, indicating the state abbreviations. As with name, removing the first two rows.

```
trim_lines <- str_trim(lines, side = "both")
state_collect <-  str_sub(trim_lines, start = 1, end = 2)
state <- unlist(str_extract_all(state_collect, "[[:alpha:]]{2}"))
state <- state[-c(1,2)]
state
```

```
##  [1] "ON" "MI" "MI" "MI" "MI" "OH" "MI" "MI" "ON" "MI" "MI" "MI" "MI" "MI"
## [15] "MI" "MI" "MI" "MI" "MI" "MI" "ON" "MI" "ON" "MI" "MI" "ON" "MI" "MI"
## [29] "MI" "ON" "MI" "ON" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI"
## [43] "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI" "MI"
## [57] "MI" "MI" "MI" "MI" "ON" "MI" "MI" "MI"
```

Checking the length of the state vector.

```
length(state)
```

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

Separating out the digits divided by a decimal point.

```
points <- unlist(str_extract_all(lines, "[[:digit:]]\\.[[:digit:]]"))
```

```
length(points)
```

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

The “R:” characters indicate the player’s ratings.

```
pre_ratings <- unlist(str_extract_all(lines, "R:( ){1,2}[[:digit:]]{3,4}"))
pre_ratings <- unlist(str_extract_all(pre_ratings, "[[:digit:]]{3,4}"))
pre_ratings
```

```
##  [1] "1794" "1553" "1384" "1716" "1655" "1686" "1649" "1641" "1411" "1365"
## [11] "1712" "1663" "1666" "1610" "1220" "1604" "1629" "1600" "1564" "1595"
## [21] "1563" "1555" "1363" "1229" "1745" "1579" "1552" "1507" "1602" "1522"
## [31] "1494" "1441" "1449" "1399" "1438" "1355" "980"  "1423" "1436" "1348"
## [41] "1403" "1332" "1283" "1199" "1242" "377"  "1362" "1382" "1291" "1056"
## [51] "1011" "935"  "1393" "1270" "1186" "1153" "1092" "917"  "853"  "967"
## [61] "955"  "1530" "1175" "1163"
```

```
length(pre_ratings)
```

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

COnstructing the chess dataframe.

```
chess <- data.frame(name, ST = state, points, pre_ratings, stringsAsFactors = FALSE)
head(chess)
```

```
##                  name ST points pre_ratings
## 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
```

Retrieving the contest outcome and opponent., looking for Wins (“W”), Losses (“L”), or Draws (“D”).

```
results <- unlist(str_extract_all(lines, "[WLD][[:blank:]]{2,3}[[:digit:]]{0,2}"))
results
```

```
##   [1] "W  39" "W  21" "W  18" "W  14" "W   7" "D  12" "D   4" "W    "
##   [9] "W    " "W    " "W    " "W  63" "W  58" "L   4" "W  17" "W  16"
##  [17] "W  20" "W   7" "W    " "W    " "W    " "L   8" "W  61" "W  25"
##  [25] "W  21" "W  11" "W  13" "W  12" "W    " "W    " "W    " "W    "
##  [33] "W  23" "D  28" "W   2" "W  26" "D   5" "W  19" "D   1" "W    "
##  [41] "W    " "W    " "W  45" "W  37" "D  12" "D  13" "D   4" "W  14"
##  [49] "W  17" "W    " "W    " "W    " "W  34" "D  29" "L  11" "W  35"
##  [57] "D  10" "W  27" "W  21" "W    " "W    " "W    " "L    " "W  57"
##  [65] "W  46" "W  13" "W  11" "L   1" "W   9" "L   2" "W    " "W    "
##  [73] "W    " "W    " "W   3" "W  32" "L  14" "L   9" "W  47" "W  28"
##  [81] "W  19" "W    " "W    " "W    " "W    " "W  25" "L  18" "W  59"
##  [89] "W   8" "W  26" "L   7" "W  20" "W    " "W    " "W    " "D  16"
##  [97] "L  19" "W  55" "W  31" "D   6" "W  25" "W  18" "W    " "W    "
## [105] "W    " "W    " "D  38" "W  56" "W   6" "L   7" "L   3" "W  34"
## [113] "W  26" "W    " "W    " "W    " "W  42" "W  33" "D   5" "W  38"
## [121] "D   1" "L   3" "W    " "W    " "W    " "W  36" "W  27" "L   7"
## [129] "D   5" "W  33" "L   3" "W  32" "W    " "W    " "W    " "W    "
## [137] "W  54" "W  44" "W   8" "L   1" "D  27" "L   5" "W  31" "W    "
## [145] "W    " "W    " "W    " "D  19" "L  16" "W  30" "L  22" "W  54"
## [153] "W  33" "W  38" "W    " "W    " "W    " "D  10" "W  15" "W  39"
## [161] "L   2" "W  36" "W    " "W    " "W  48" "W  41" "L  26" "L   2"
## [169] "W  23" "W  22" "L   5" "W    " "W    " "W    " "W    " "W  47"
## [177] "W   9" "L   1" "W  32" "L  19" "W  38" "L  10" "W    " "W    "
## [185] "W    " "D  15" "W  10" "W  52" "D  28" "W  18" "L   4" "L   8"
## [193] "W    " "W    " "W    " "W    " "L  40" "W  49" "W  23" "W  41"
## [201] "W  28" "L   2" "L   9" "W    " "W    " "W    " "W    " "W  43"
## [209] "L   1" "W  47" "L   3" "W  40" "W  39" "L   6" "W    " "W    "
## [217] "W    " "W    " "W  64" "D  52" "L  28" "W  15" "L  17" "W  40"
## [225] "W    " "W    " "W    " "L   4" "W  43" "L  20" "W  58" "L  17"
## [233] "W  37" "W  46" "W    " "W    " "W    " "L  28" "L  47" "W  43"
## [241] "L  25" "W  60" "W  44" "W  39" "W    " "W    " "W    " "L   9"
## [249] "W  53" "L   3" "W  24" "D  34" "L  10" "W  47" "W    " "W    "
## [257] "W    " "W  49" "W  40" "W  17" "L   4" "L   9" "D  32" "L  11"
## [265] "W    " "W    " "W    " "W    " "W  51" "L  13" "W  46" "W  37"
## [273] "D  14" "L   6" "W    " "W    " "W    " "W  24" "D   4" "W  22"
## [281] "D  19" "L  20" "L   8" "D  36" "W    " "W    " "W    " "W    "
## [289] "W  50" "D   6" "L  38" "L  34" "W  52" "W  48" "W    " "W    "
## [297] "W    " "L  52" "D  64" "L  15" "W  55" "L  31" "W  61" "W  50"
## [305] "W    " "W    " "W    " "L  58" "D  55" "W  64" "L  10" "W  30"
## [313] "W  50" "L  14" "W    " "W    " "W    " "W  61" "L   8" "W  44"
## [321] "L  18" "W  51" "D  26" "L  13" "W    " "W    " "W    " "W    "
## [329] "W  60" "L  12" "W  50" "D  36" "L  13" "L  15" "W  51" "W    "
## [337] "W    " "W    " "L   6" "W  60" "L  37" "W  29" "D  25" "L  11"
## [345] "W  52" "W    " "W    " "W    " "L  46" "L  38" "W  56" "L   6"
## [353] "W  57" "D  52" "W  48" "W    " "W    " "W    " "W    " "L  13"
## [361] "W  57" "W  51" "D  33" "L  16" "D  28" "W    " "W    " "W    "
## [369] "L   5" "W  34" "L  27" "L  23" "W  61" "W    " "W    " "W    "
## [377] "D  11" "W  35" "W  29" "L  12" "L  18" "L  15" "W    " "W    "
## [385] "W    " "L   1" "W  54" "W  40" "L  16" "W  44" "L  21" "L  24"
## [393] "W    " "W    " "W    " "W    " "W  20" "L  26" "L  39" "W  59"
## [401] "L  21" "W  56" "L  22" "W    " "W    " "W    " "W    " "W  59"
## [409] "L  17" "W  58" "L  20" "W    " "W    " "L  12" "L  50" "L  57"
## [417] "D  60" "D  61" "W  64" "W  56" "W    " "W    " "W    " "L  21"
## [425] "L  23" "L  24" "W  63" "W  59" "L  46" "W  55" "W    " "W    "
## [433] "W    " "W    " "L  14" "L  32" "W  53" "L  39" "L  24" "W  59"
## [441] "W    " "W    " "W    " "L   5" "L  51" "D  60" "L  56" "W  63"
## [449] "D  55" "W  58" "W    " "W    " "W    " "W    " "W  35" "L   7"
## [457] "L  27" "L  50" "W  64" "W  43" "L  23" "W    " "W    " "W    "
## [465] "W    " "L  18" "W  24" "L  21" "W  61" "L   8" "D  51" "L  25"
## [473] "W    " "W    " "W    " "W    " "L  17" "W  63" "D  52" "L  29"
## [481] "L  35" "W    " "W    " "L  26" "L  20" "D  63" "D  64" "W  58"
## [489] "W    " "W    " "W    " "L  29" "W  42" "L  33" "W  46" "L  31"
## [497] "L  30" "W    " "W    " "W    " "L  27" "W  45" "L  36" "W  57"
## [505] "L  32" "D  47" "L  33" "W    " "W    " "W    " "W    " "W  30"
## [513] "D  22" "L  19" "D  48" "L  29" "D  35" "L  34" "W    " "W    "
## [521] "W    " "L  25" "L  44" "W  57" "W    " "W    " "L  14" "L  39"
## [529] "L  61" "L  15" "L  59" "W  64" "W    " "W    " "W    " "L  62"
## [537] "D  31" "L  10" "L  30" "D  45" "L  43" "W    " "W    " "W    "
## [545] "L  11" "L  35" "W  45" "L  40" "L  42" "W    " "W    " "W    "
## [553] "L   7" "L  36" "W  42" "L  51" "L  35" "L  53" "W    " "W    "
## [561] "W    " "W  31" "L   2" "L  41" "L  23" "L  49" "L  45" "W    "
## [569] "W    " "W    " "L  41" "L   9" "L  40" "L  43" "W  54" "L  44"
## [577] "W    " "W    " "W    " "L  33" "L  34" "D  45" "D  42" "L  24"
## [585] "W    " "W    " "L  32" "L   3" "W  54" "L  47" "D  42" "L  30"
## [593] "L  37" "W    " "W    " "W    " "W  55" "L   2" "L  48" "D  49"
## [601] "L  43" "L  45" "W    " "W    " "L  22" "D  30" "L  31" "D  49"
## [609] "L  46" "L  42" "L  54" "W    " "W    " "W    "
```

With 64 players and 7 rounds, there should be 448 total scores. Checking the data, should remove the “W” followed by blanks with no opponent number.

```
length(results)
```

```
## [1] 614
```

```
64 * 7
```

```
## [1] 448
```

```
results.raw <- subset(results, results != "W    ")
length(results.raw)
```

```
## [1] 409
```

Just one short. Finding the abberation. The one “B” in line 75.

```
lines[75:77]
```

```
##    37 | AMIYATOSH PWNANANDAM            |3.5  |B    |L   5|W  34|L  27|H    |L  23|W  61|
##         "   37  AMIYATOSH PWNANANDAM            3.5  B    L   5W  34L  27H    L  23W  61"
##    MI | 15489571 / R:  980P12->1077P17  |     |     |B    |W    |W    |     |B    |W    |
##         "   MI  15489571 / R:  980P12->1077P17           B    W    W         B    W    "
##    38 | BRIAN LIU                       |3.0  |D  11|W  35|W  29|L  12|H    |L  18|L  15|
##         "   38  BRIAN LIU                       3.0  D  11W  35W  29L  12H    L  18L  15"
```

```
results <- unlist(str_extract_all(lines, "[WLDUH][[:blank:]]{2,3}[[:digit:]]{0,2}"))
results.raw <- subset(results, results != "W   ")
results.raw
```

```
##   [1] "W  39" "W  21" "W  18" "W  14" "W   7" "D  12" "D   4" "W  63"
##   [9] "W  58" "L   4" "W  17" "W  16" "W  20" "W   7" "L   8" "W  61"
##  [17] "W  25" "W  21" "W  11" "W  13" "W  12" "W  23" "D  28" "W   2"
##  [25] "W  26" "D   5" "W  19" "D   1" "W  45" "W  37" "D  12" "D  13"
##  [33] "D   4" "W  14" "W  17" "W  34" "D  29" "L  11" "W  35" "D  10"
##  [41] "W  27" "W  21" "H  15" "L    " "W  57" "W  46" "W  13" "W  11"
##  [49] "L   1" "W   9" "L   2" "W   3" "W  32" "L  14" "L   9" "W  47"
##  [57] "W  28" "W  19" "W  25" "L  18" "W  59" "W   8" "W  26" "L   7"
##  [65] "W  20" "D  16" "L  19" "W  55" "W  31" "D   6" "W  25" "W  18"
##  [73] "D  38" "W  56" "W   6" "L   7" "L   3" "W  34" "W  26" "W  42"
##  [81] "W  33" "D   5" "W  38" "H    " "D   1" "L   3" "W  36" "W  27"
##  [89] "L   7" "D   5" "W  33" "L   3" "W  32" "W  54" "W  44" "W   8"
##  [97] "L   1" "D  27" "L   5" "W  31" "D  19" "L  16" "W  30" "L  22"
## [105] "W  54" "W  33" "W  38" "D  10" "W  15" "H    " "W  39" "L   2"
## [113] "W  36" "U    " "W  48" "W  41" "L  26" "L   2" "W  23" "W  22"
## [121] "L   5" "W  47" "W   9" "L   1" "W  32" "L  19" "W  38" "L  10"
## [129] "D  15" "W  10" "W  52" "D  28" "W  18" "L   4" "L   8" "L  40"
## [137] "W  49" "W  23" "W  41" "W  28" "L   2" "L   9" "W  43" "L   1"
## [145] "W  47" "L   3" "W  40" "W  39" "L   6" "W  64" "D  52" "L  28"
## [153] "W  15" "H    " "L  17" "W  40" "L   4" "W  43" "L  20" "W  58"
## [161] "L  17" "W  37" "W  46" "H    " "L  28" "L  47" "W  43" "L  25"
## [169] "W  60" "W  44" "W  39" "L   9" "W  53" "L   3" "W  24" "D  34"
## [177] "L  10" "W  47" "U    " "W  49" "W  40" "W  17" "L   4" "L   9"
## [185] "D  32" "L  11" "W  51" "L  13" "W  46" "W  37" "D  14" "L   6"
## [193] "U    " "U    " "W  24" "D   4" "W  22" "D  19" "L  20" "L   8"
## [201] "D  36" "W  50" "D   6" "L  38" "L  34" "W  52" "W  48" "U    "
## [209] "L  52" "D  64" "L  15" "W  55" "L  31" "W  61" "W  50" "L  58"
## [217] "D  55" "W  64" "L  10" "W  30" "W  50" "L  14" "W  61" "L   8"
## [225] "W  44" "L  18" "W  51" "D  26" "L  13" "W  60" "L  12" "W  50"
## [233] "D  36" "L  13" "L  15" "W  51" "L   6" "W  60" "L  37" "W  29"
## [241] "D  25" "L  11" "W  52" "L  46" "L  38" "W  56" "L   6" "W  57"
## [249] "D  52" "W  48" "L  13" "W  57" "W  51" "D  33" "H    " "L  16"
## [257] "D  28" "L   5" "W  34" "L  27" "H    " "L  23" "W  61" "U    "
## [265] "D  11" "W  35" "W  29" "L  12" "H    " "L  18" "L  15" "L   1"
## [273] "W  54" "W  40" "L  16" "W  44" "L  21" "L  24" "W  20" "L  26"
## [281] "L  39" "W  59" "L  21" "W  56" "L  22" "W  59" "L  17" "W  58"
## [289] "L  20" "U    " "U    " "L  12" "L  50" "L  57" "D  60" "D  61"
## [297] "W  64" "W  56" "L  21" "L  23" "L  24" "W  63" "W  59" "L  46"
## [305] "W  55" "L  14" "L  32" "W  53" "L  39" "L  24" "W  59" "L   5"
## [313] "L  51" "D  60" "L  56" "W  63" "D  55" "W  58" "W  35" "L   7"
## [321] "L  27" "L  50" "W  64" "W  43" "L  23" "L  18" "W  24" "L  21"
## [329] "W  61" "L   8" "D  51" "L  25" "L  17" "W  63" "H    " "D  52"
## [337] "H    " "L  29" "L  35" "L  26" "L  20" "D  63" "D  64" "W  58"
## [345] "H    " "U    " "L  29" "W  42" "L  33" "W  46" "H    " "L  31"
## [353] "L  30" "L  27" "W  45" "L  36" "W  57" "L  32" "D  47" "L  33"
## [361] "W  30" "D  22" "L  19" "D  48" "L  29" "D  35" "L  34" "H    "
## [369] "L  25" "H    " "L  44" "U    " "W  57" "U    " "L  14" "L  39"
## [377] "L  61" "L  15" "L  59" "W  64" "L  62" "D  31" "L  10" "L  30"
## [385] "D  45" "L  43" "H    " "L  11" "L  35" "W  45" "H    " "L  40"
## [393] "L  42" "U    " "L   7" "L  36" "W  42" "L  51" "L  35" "L  53"
## [401] "W  31" "L   2" "L  41" "L  23" "L  49" "L  45" "L  41" "L   9"
## [409] "L  40" "L  43" "W  54" "L  44" "L  33" "L  34" "D  45" "D  42"
## [417] "L  24" "H    " "U    " "L  32" "L   3" "W  54" "L  47" "D  42"
## [425] "L  30" "L  37" "W  55" "U    " "U    " "U    " "U    " "U    "
## [433] "U    " "L   2" "L  48" "D  49" "L  43" "L  45" "H    " "U    "
## [441] "L  22" "D  30" "L  31" "D  49" "L  46" "L  42" "L  54"
```

```
length(results.raw)
```

```
## [1] 447
```

Manually discovering that the extra “B” is in position 408, then inserting it. Would be nice to automate this process somehow.

```
results.raw <- c(results.raw[1:407],"B", results.raw[408:length(results.raw)])
length(results.raw)
```

```
## [1] 448
```

From the results vector, removing the result so that only the opponent number remains.

```
opponents <- NULL

opponents <- ifelse(str_detect(results.raw, "[[:digit:]]{1,2}"), unlist(str_extract_all(results.ra
w, "[[:digit:]]{1,2}")), NA)
opponents
```

```
##   [1] "39" "21" "18" "14" "7"  "12" "4"  "63" "58" "4"  "17" "16" "20" "7"
##  [15] "8"  "61" "25" "21" "11" "13" "12" "23" "28" "2"  "26" "5"  "19" "1"
##  [29] "45" "37" "12" "13" "4"  "14" "17" "34" "29" "11" "35" "10" "27" "21"
##  [43] "15" NA   "46" "13" "11" "1"  "9"  "2"  "3"  "32" "14" "9"  "47" "28"
##  [57] "19" "25" "18" "59" "8"  "26" "7"  "20" "16" "19" "55" "31" "6"  "25"
##  [71] "18" "38" "56" "6"  "7"  "3"  "34" "26" "42" "33" "5"  "38" "1"  NA
##  [85] "36" "27" "7"  "5"  "33" "3"  "32" "54" "44" "8"  "1"  "27" "5"  "31"
##  [99] "19" "16" "30" "22" "54" "33" "38" "10" "15" "39" "2"  NA   "48" "41"
## [113] "26" NA   "23" "22" "5"  "47" "9"  "1"  "32" "19" "38" "10" "15" "10"
## [127] "52" "28" "18" "4"  "8"  "40" "49" "23" "41" "28" "2"  "9"  "43" "1"
## [141] "47" "3"  "40" "39" "6"  "64" "52" "28" "15" "17" "40" "4"  "43" NA
## [155] "58" "17" "37" "46" "28" "47" "43" "25" "60" NA   "39" "9"  "53" "3"
## [169] "24" "34" "10" "47" "49" "40" "17" "4"  "9"  "32" NA   "51" "13" "46"
## [183] "37" "14" "6"  "24" "4"  "22" "19" "20" "8"  "36" NA   NA   "38" "34"
## [197] "52" "48" "52" "64" "15" "55" "31" "61" "50" "58" "55" NA   "10" "30"
## [211] "50" "14" "61" "8"  "44" "18" "51" "26" "13" "60" "12" "50" "36" "13"
## [225] "15" "51" "6"  "60" "37" "29" "25" "11" "52" "46" "38" "56" "6"  "57"
## [239] "52" "48" "13" "57" "51" "33" "16" "28" "5"  "34" "27" "23" "61" "11"
## [253] "35" "29" NA   "18" "15" "1"  "54" "40" NA   "44" "21" NA   "20" "26"
## [267] "39" "59" NA   "56" "22" "59" "17" "58" "20" "12" "50" "57" "60" "61"
## [281] "64" "56" "21" "23" "24" "63" "59" "46" "55" NA   NA   "53" "39" "24"
## [295] "59" "5"  "51" "60" "56" "63" "55" "58" "35" "7"  "27" "50" "64" "43"
## [309] "23" "18" "24" "21" "61" "8"  "51" "25" "17" "63" "52" "29" "35" "26"
## [323] "20" "63" "64" "58" "29" "42" "33" "46" "31" "30" "27" "45" NA   "57"
## [337] NA   "47" "33" "30" "22" "19" "48" "29" NA   NA   "25" "44" "57" "14"
## [351] NA   "61" "15" "59" "64" "62" "31" "10" "30" "45" "43" "11" "35" "45"
## [365] "40" "42" "7"  NA   "42" NA   "35" NA   "31" NA   "41" "23" "49" "45"
## [379] "41" "9"  "40" "43" "54" "44" "33" "34" NA   "42" "24" "32" NA   "54"
## [393] "47" NA   "30" "37" "55" "2"  "48" "49" "43" "45" "22" "30" "31" "49"
## [407] "46" NA   "54" "39" "21" "18" "14" "7"  "12" "4"  "63" "58" NA   NA
## [421] "16" "20" "7"  "8"  "61" "25" "21" "11" NA   NA   NA   NA   NA   NA
## [435] "5"  "19" "1"  "45" "37" NA   NA   "4"  "14" "17" "34" "29" "11" "35"
```

Checking the class of the variable opponents. Finding it is a character, converting that to a numeric.

```
class(opponents)
```

```
## [1] "character"
```

```
opponents <- as.numeric(opponents)
class(opponents)
```

```
## [1] "numeric"
```

Creating an opponents matrix, with 64 rows for the players and 7 columns for the rounds.

```
opponents.mat <- matrix(opponents, nrow = 64, ncol = 7, byrow = TRUE)
opponents.mat
```

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

Replacing the player ID with their ratings, again ensuring it is a numeric value. The matrix is dissolved.

```
opponents.rating <- as.numeric(pre_ratings[opponents])
opponents.rating
```

```
##   [1] 1436 1563 1600 1610 1649 1663 1716 1175  917 1716 1629 1604 1595 1649
##  [15] 1641  955 1745 1563 1712 1666 1663 1363 1507 1553 1579 1655 1564 1794
##  [29] 1242  980 1663 1666 1716 1610 1629 1399 1602 1712 1438 1365 1552 1563
##  [43] 1220   NA  377 1666 1712 1794 1411 1553 1384 1441 1610 1411 1362 1507
##  [57] 1564 1745 1600  853 1641 1579 1649 1595 1604 1564 1186 1494 1686 1745
##  [71] 1600 1423 1153 1686 1649 1384 1399 1579 1332 1449 1655 1423 1794   NA
##  [85] 1355 1552 1649 1655 1449 1384 1441 1270 1199 1641 1794 1552 1655 1494
##  [99] 1564 1604 1522 1555 1270 1449 1423 1365 1220 1436 1553   NA 1382 1403
## [113] 1579   NA 1363 1555 1655 1362 1411 1794 1441 1564 1423 1365 1220 1365
## [127]  935 1507 1600 1716 1641 1348 1291 1363 1403 1507 1553 1411 1283 1794
## [141] 1362 1384 1348 1436 1686 1163  935 1507 1220 1629 1348 1716 1283   NA
## [155]  917 1629  980  377 1507 1362 1283 1745  967   NA 1436 1411 1393 1384
## [169] 1229 1399 1365 1362 1291 1348 1629 1716 1411 1441   NA 1011 1666  377
## [183]  980 1610 1686 1229 1716 1555 1564 1595 1641 1355   NA   NA 1423 1399
## [197]  935 1382  935 1163 1220 1186 1494  955 1056  917 1186   NA 1365 1522
## [211] 1056 1610  955 1641 1199 1600 1011 1579 1666  967 1663 1056 1355 1666
## [225] 1220 1011 1686  967  980 1602 1745 1712  935  377 1423 1153 1686 1092
## [239]  935 1382 1666 1092 1011 1449 1604 1507 1655 1399 1552 1363  955 1712
## [253] 1438 1602   NA 1600 1220 1794 1270 1348   NA 1199 1563   NA 1595 1579
## [267] 1436  853   NA 1153 1555  853 1629  917 1595 1663 1056 1092  967  955
## [281] 1163 1153 1563 1363 1229 1175  853  377 1186   NA   NA 1393 1436 1229
## [295]  853 1655 1011  967 1153 1175 1186  917 1438 1649 1552 1056 1163 1283
## [309] 1363 1600 1229 1563  955 1641 1011 1745 1629 1175  935 1602 1438 1579
## [323] 1595 1175 1163  917 1602 1332 1449  377 1494 1522 1552 1242   NA 1092
## [337]   NA 1362 1449 1522 1555 1564 1382 1602   NA   NA 1745 1199 1092 1610
## [351]   NA  955 1220  853 1163 1530 1494 1365 1522 1242 1283 1712 1438 1242
## [365] 1348 1332 1649   NA 1332   NA 1438   NA 1494   NA 1403 1363 1291 1242
## [379] 1403 1411 1348 1283 1270 1199 1449 1399   NA 1332 1229 1441   NA 1270
## [393] 1362   NA 1522  980 1186 1553 1382 1291 1283 1242 1555 1522 1494 1291
## [407]  377   NA 1270 1436 1563 1600 1610 1649 1663 1716 1175  917   NA   NA
## [421] 1604 1595 1649 1641  955 1745 1563 1712   NA   NA   NA   NA   NA   NA
## [435] 1655 1564 1794 1242  980   NA   NA 1716 1610 1629 1399 1602 1712 1438
```

Restructuring the opponent ratings into a matrix.

```
opponents.rating.matrix <- matrix(opponents.rating, nrow = 64, ncol = 7, byrow = TRUE)
opponents.rating.matrix
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7]
##  [1,] 1436 1563 1600 1610 1649 1663 1716
##  [2,] 1175  917 1716 1629 1604 1595 1649
##  [3,] 1641  955 1745 1563 1712 1666 1663
##  [4,] 1363 1507 1553 1579 1655 1564 1794
##  [5,] 1242  980 1663 1666 1716 1610 1629
##  [6,] 1399 1602 1712 1438 1365 1552 1563
##  [7,] 1220   NA  377 1666 1712 1794 1411
##  [8,] 1553 1384 1441 1610 1411 1362 1507
##  [9,] 1564 1745 1600  853 1641 1579 1649
## [10,] 1595 1604 1564 1186 1494 1686 1745
## [11,] 1600 1423 1153 1686 1649 1384 1399
## [12,] 1579 1332 1449 1655 1423 1794   NA
## [13,] 1355 1552 1649 1655 1449 1384 1441
## [14,] 1270 1199 1641 1794 1552 1655 1494
## [15,] 1564 1604 1522 1555 1270 1449 1423
## [16,] 1365 1220 1436 1553   NA 1382 1403
## [17,] 1579   NA 1363 1555 1655 1362 1411
## [18,] 1794 1441 1564 1423 1365 1220 1365
## [19,]  935 1507 1600 1716 1641 1348 1291
## [20,] 1363 1403 1507 1553 1411 1283 1794
## [21,] 1362 1384 1348 1436 1686 1163  935
## [22,] 1507 1220 1629 1348 1716 1283   NA
## [23,]  917 1629  980  377 1507 1362 1283
## [24,] 1745  967   NA 1436 1411 1393 1384
## [25,] 1229 1399 1365 1362 1291 1348 1629
## [26,] 1716 1411 1441   NA 1011 1666  377
## [27,]  980 1610 1686 1229 1716 1555 1564
## [28,] 1595 1641 1355   NA   NA 1423 1399
## [29,]  935 1382  935 1163 1220 1186 1494
## [30,]  955 1056  917 1186   NA 1365 1522
## [31,] 1056 1610  955 1641 1199 1600 1011
## [32,] 1579 1666  967 1663 1056 1355 1666
## [33,] 1220 1011 1686  967  980 1602 1745
## [34,] 1712  935  377 1423 1153 1686 1092
## [35,]  935 1382 1666 1092 1011 1449 1604
## [36,] 1507 1655 1399 1552 1363  955 1712
## [37,] 1438 1602   NA 1600 1220 1794 1270
## [38,] 1348   NA 1199 1563   NA 1595 1579
## [39,] 1436  853   NA 1153 1555  853 1629
## [40,]  917 1595 1663 1056 1092  967  955
## [41,] 1163 1153 1563 1363 1229 1175  853
## [42,]  377 1186   NA   NA 1393 1436 1229
## [43,]  853 1655 1011  967 1153 1175 1186
## [44,]  917 1438 1649 1552 1056 1163 1283
## [45,] 1363 1600 1229 1563  955 1641 1011
## [46,] 1745 1629 1175  935 1602 1438 1579
## [47,] 1595 1175 1163  917 1602 1332 1449
## [48,]  377 1494 1522 1552 1242   NA 1092
## [49,]   NA 1362 1449 1522 1555 1564 1382
## [50,] 1602   NA   NA 1745 1199 1092 1610
## [51,]   NA  955 1220  853 1163 1530 1494
## [52,] 1365 1522 1242 1283 1712 1438 1242
## [53,] 1348 1332 1649   NA 1332   NA 1438
## [54,]   NA 1494   NA 1403 1363 1291 1242
## [55,] 1403 1411 1348 1283 1270 1199 1449
## [56,] 1399   NA 1332 1229 1441   NA 1270
## [57,] 1362   NA 1522  980 1186 1553 1382
## [58,] 1291 1283 1242 1555 1522 1494 1291
## [59,]  377   NA 1270 1436 1563 1600 1610
## [60,] 1649 1663 1716 1175  917   NA   NA
## [61,] 1604 1595 1649 1641  955 1745 1563
## [62,] 1712   NA   NA   NA   NA   NA   NA
## [63,] 1655 1564 1794 1242  980   NA   NA
## [64,] 1716 1610 1629 1399 1602 1712 1438
```

Taking the mean of the matrix to find the average rating of opponents, after removing the NA values.

```
chess$opponents.rating <- round(rowMeans(opponents.rating.matrix, na.rm = T, dims = 1))
chess
```

```
##                        name ST points pre_ratings opponents.rating
## 1                  GARY HUA ON    6.0        1794             1605
## 2           DAKSHESH DARURI MI    6.0        1553             1469
## 3              ADITYA BAJAJ MI    6.0        1384             1564
## 4       PATRICK H SCHILLING MI    5.5        1716             1574
## 5                HANSHI ZUO MI    5.5        1655             1501
## 6               HANSEN SONG OH    5.0        1686             1519
## 7         GARY DEE SWATHELL MI    5.0        1649             1363
## 8          EZEKIEL HOUGHTON MI    5.0        1641             1467
## 9               STEFANO LEE ON    5.0        1411             1519
## 10                ANVIT RAO MI    5.0        1365             1553
## 11 CAMERON WILLIAM MC LEMAN MI    4.5        1712             1471
## 12           KENNETH J TACK MI    4.5        1663             1539
## 13        TORRANCE HENRY JR MI    4.5        1666             1498
## 14             BRADLEY SHAW MI    4.5        1610             1515
## 15   ZACHARY JAMES HOUGHTON MI    4.5        1220             1484
## 16             MIKE NIKITIN MI    4.0        1604             1393
## 17       RONALD GRZEGORCZYK MI    4.0        1629             1488
## 18            DAVID SUNDEEN MI    4.0        1600             1453
## 19             DIPANKAR ROY MI    4.0        1564             1434
## 20              JASON ZHENG MI    4.0        1595             1473
## 21            DINH DANG BUI ON    4.0        1563             1331
## 22         EUGENE L MCCLURE MI    4.0        1555             1450
## 23                 ALAN BUI ON    4.0        1363             1151
## 24        MICHAEL R ALDRICH MI    4.0        1229             1389
## 25         LOREN SCHWIEBERT MI    3.5        1745             1375
## 26                  MAX ZHU ON    3.5        1579             1270
## 27           GAURAV GIDWANI MI    3.5        1552             1477
## 28     SOFIA ADINA STANESCU MI    3.5        1507             1483
## 29         CHIEDOZIE OKORIE MI    3.5        1602             1188
## 30       GEORGE AVERY JONES ON    3.5        1522             1167
## 31             RISHI SHETTY MI    3.5        1494             1296
## 32    JOSHUA PHILIP MATHEWS ON    3.5        1441             1422
## 33                  JADE GE MI    3.5        1449             1316
## 34   MICHAEL JEFFERY THOMAS MI    3.5        1399             1197
## 35         JOSHUA DAVID LEE MI    3.5        1438             1306
## 36            SIDDHARTH JHA MI    3.5        1355             1449
## 37     AMIYATOSH PWNANANDAM MI    3.5         980             1487
## 38                BRIAN LIU MI    3.0        1423             1457
## 39            JOEL R HENDON MI    3.0        1436             1246
## 40             FOREST ZHANG MI    3.0        1348             1178
## 41      KYLE WILLIAM MURPHY MI    3.0        1403             1214
## 42                 JARED GE MI    3.0        1332             1124
## 43        ROBERT GLEN VASEY MI    3.0        1283             1143
## 44       JUSTIN D SCHILLING MI    3.0        1199             1294
## 45                DEREK YAN MI    3.0        1242             1337
## 46 JACOB ALEXANDER LAVALLEY MI    3.0         377             1443
## 47              ERIC WRIGHT MI    2.5        1362             1319
## 48             DANIEL KHAIN MI    2.5        1382             1213
## 49         MICHAEL J MARTIN MI    2.5        1291             1472
## 50               SHIVAM JHA MI    2.5        1056             1450
## 51           TEJAS AYYAGARI MI    2.5        1011             1202
## 52                ETHAN GUO MI    2.5         935             1401
## 53            JOSE C YBARRA MI    2.0        1393             1420
## 54              LARRY HODGE MI    2.0        1270             1359
## 55                ALEX KONG MI    2.0        1186             1338
## 56             MARISA RICCI MI    2.0        1153             1334
## 57               MICHAEL LU MI    2.0        1092             1331
## 58             VIRAJ MOHILE MI    2.0         917             1383
## 59        SEAN M MC CORMICK MI    2.0         853             1309
## 60               JULIA SHEN MI    1.5         967             1424
## 61            JEZZEL FARKAS ON    1.5         955             1536
## 62            ASHWIN BALAJI MI    1.0        1530             1712
## 63     THOMAS JOSEPH HOSMER MI    1.0        1175             1447
## 64                   BEN LI MI    1.0        1163             1587
```

Finally, capturing the data into a csv file.

```
write.csv(chess, file = "chess.csv")
```