Project 1

Abdelmalek Hajjam

9/20/2019

|  |
| --- |
| 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. |
| 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. |

**Loading the data from the tournament file**

library(stringr)  
  
chess\_dataset <- readLines('https://raw.githubusercontent.com/theoracley/Data607/master/Project1/tournamentinfo.txt')  
  
#chess\_dataset <- readLines("./tournamentinfo.txt")  
head(chess\_dataset)

## [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 |"

tail(chess\_dataset)

## [1] " 63 | THOMAS JOSEPH HOSMER |1.0 |L 2|L 48|D 49|L 43|L 45|H |U |"  
## [2] " MI | 15057092 / R: 1175 ->1125 | |W |B |W |B |B | | |"  
## [3] "-----------------------------------------------------------------------------------------"  
## [4] " 64 | BEN LI |1.0 |L 22|D 30|L 31|D 49|L 46|L 42|L 54|"  
## [5] " MI | 15006561 / R: 1163 ->1112 | |B |W |W |B |W |B |B |"  
## [6] "-----------------------------------------------------------------------------------------"

**Lets start cleaning our data by removing headers**

chess\_dataset\_cleaned <- chess\_dataset[-c(0:4)]  
head(chess\_dataset\_cleaned, 15)

## [1] " 1 | GARY HUA |6.0 |W 39|W 21|W 18|W 14|W 7|D 12|D 4|"  
## [2] " ON | 15445895 / R: 1794 ->1817 |N:2 |W |B |W |B |W |B |W |"  
## [3] "-----------------------------------------------------------------------------------------"  
## [4] " 2 | DAKSHESH DARURI |6.0 |W 63|W 58|L 4|W 17|W 16|W 20|W 7|"  
## [5] " MI | 14598900 / R: 1553 ->1663 |N:2 |B |W |B |W |B |W |B |"  
## [6] "-----------------------------------------------------------------------------------------"  
## [7] " 3 | ADITYA BAJAJ |6.0 |L 8|W 61|W 25|W 21|W 11|W 13|W 12|"  
## [8] " MI | 14959604 / R: 1384 ->1640 |N:2 |W |B |W |B |W |B |W |"  
## [9] "-----------------------------------------------------------------------------------------"  
## [10] " 4 | PATRICK H SCHILLING |5.5 |W 23|D 28|W 2|W 26|D 5|W 19|D 1|"  
## [11] " MI | 12616049 / R: 1716 ->1744 |N:2 |W |B |W |B |W |B |B |"  
## [12] "-----------------------------------------------------------------------------------------"  
## [13] " 5 | HANSHI ZUO |5.5 |W 45|W 37|D 12|D 13|D 4|W 14|W 17|"  
## [14] " MI | 14601533 / R: 1655 ->1690 |N:2 |B |W |B |W |B |W |B |"  
## [15] "-----------------------------------------------------------------------------------------"

**then trim the characters**

chess\_dataset\_cleaned <- chess\_dataset\_cleaned[sapply(chess\_dataset\_cleaned, nchar) > 0]  
head(chess\_dataset\_cleaned)

## [1] " 1 | GARY HUA |6.0 |W 39|W 21|W 18|W 14|W 7|D 12|D 4|"  
## [2] " ON | 15445895 / R: 1794 ->1817 |N:2 |W |B |W |B |W |B |W |"  
## [3] "-----------------------------------------------------------------------------------------"  
## [4] " 2 | DAKSHESH DARURI |6.0 |W 63|W 58|L 4|W 17|W 16|W 20|W 7|"  
## [5] " MI | 14598900 / R: 1553 ->1663 |N:2 |B |W |B |W |B |W |B |"  
## [6] "-----------------------------------------------------------------------------------------"

**then extract the rows (starting from 1) that have names in them into a vector. We use seq() which return those rows numbers. We skip by 3 each time.**

ourSeq\_rows <- c(seq(1, length(chess\_dataset\_cleaned), 3))  
ourSeq\_rows

## [1] 1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49  
## [18] 52 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100  
## [35] 103 106 109 112 115 118 121 124 127 130 133 136 139 142 145 148 151  
## [52] 154 157 160 163 166 169 172 175 178 181 184 187 190

**get the data corresponding to those rows**

ourSeq\_data <- chess\_dataset\_cleaned[ourSeq\_rows]  
head(ourSeq\_data)

## [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|"

**Let’s extract the names using regular expression**

names <- str\_extract(ourSeq\_data, "[[:alpha:]]{2,}([[:blank:]][[:alpha:]]{1,}){1,}")  
head(names)

## [1] "GARY HUA" "DAKSHESH DARURI" "ADITYA BAJAJ"   
## [4] "PATRICK H SCHILLING" "HANSHI ZUO" "HANSEN SONG"

**Let’s extract the rows numbers into a vector starting from row 2 and skipping 3**

ourseq2 <- c(seq(2, length(chess\_dataset\_cleaned), 3))  
ourseq2

## [1] 2 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50  
## [18] 53 56 59 62 65 68 71 74 77 80 83 86 89 92 95 98 101  
## [35] 104 107 110 113 116 119 122 125 128 131 134 137 140 143 146 149 152  
## [52] 155 158 161 164 167 170 173 176 179 182 185 188 191

**get the data corresponding to those rows**

ourSeq2\_data <- chess\_dataset\_cleaned[ourseq2]  
head(ourSeq2\_data)

## [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 |"

**Let’s extract the states**

states <- str\_extract(ourSeq2\_data, "[[:alpha:]]{2}")  
states

## [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"

**Let’s extract the points from ourSeq\_data**

thepoints <- str\_extract(ourSeq\_data, "[[:digit:]]+\\.[[:digit:]]")  
thepoints <- as.numeric(as.character(thepoints))  
thepoints

## [1] 6.0 6.0 6.0 5.5 5.5 5.0 5.0 5.0 5.0 5.0 4.5 4.5 4.5 4.5 4.5 4.0 4.0  
## [18] 4.0 4.0 4.0 4.0 4.0 4.0 4.0 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5  
## [35] 3.5 3.5 3.5 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 2.5 2.5 2.5 2.5 2.5  
## [52] 2.5 2.0 2.0 2.0 2.0 2.0 2.0 2.0 1.5 1.5 1.0 1.0 1.0

**Let’s extract the pre-rating from ourSeq2\_data**

pre\_ratings <- str\_extract(ourSeq2\_data, ".\\: \\s?[[:digit:]]{3,4}")  
pre\_ratings

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

**Let’s extract the digits and convert them to numeric**

pre\_ratings <- as.numeric(str\_extract(pre\_ratings, "\\(?[0-9,.]+\\)?"))  
pre\_ratings

## [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

**Let’s do the same for the opponent**

opponent\_numbers <- str\_extract\_all(ourSeq\_data, "[[:digit:]]{1,2}\\|")  
opponent\_numbers <- str\_extract\_all(opponent\_numbers, "[[:digit:]]{1,2}")  
opponent\_numbers <- lapply(opponent\_numbers, as.numeric)  
head(opponent\_numbers)

## [[1]]  
## [1] 39 21 18 14 7 12 4  
##   
## [[2]]  
## [1] 63 58 4 17 16 20 7  
##   
## [[3]]  
## [1] 8 61 25 21 11 13 12  
##   
## [[4]]  
## [1] 23 28 2 26 5 19 1  
##   
## [[5]]  
## [1] 45 37 12 13 4 14 17  
##   
## [[6]]  
## [1] 34 29 11 35 10 27 21

**What’s the prerating average for our opponent?**

opponent\_prerating\_average <- list()  
  
for (i in 1:length(opponent\_numbers)){  
 opponent\_prerating\_average[i] <- round(mean(pre\_ratings[unlist(opponent\_numbers[i])]),2)  
}  
opponent\_prerating\_average <- lapply(opponent\_prerating\_average, as.numeric)  
opponent\_prerating\_average <- data.frame(unlist(opponent\_prerating\_average))  
  
ourFinalTable <- cbind.data.frame(names, states, thepoints, pre\_ratings, opponent\_prerating\_average)  
colnames(ourFinalTable) <- c("Player\_Name", "Player\_State", "Player\_Points", "Player\_Pre\_Rating", "Opponent\_Pre\_Rating\_AVG")  
ourFinalTable

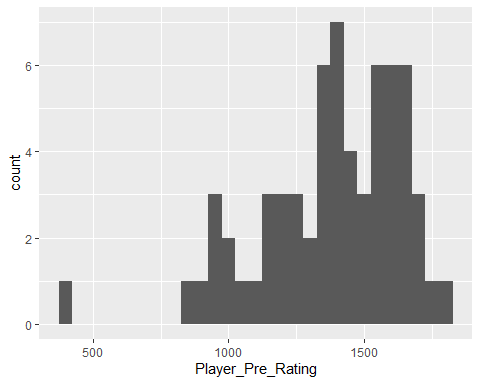
## Player\_Name Player\_State Player\_Points Player\_Pre\_Rating  
## 1 GARY HUA ON 6.0 1794  
## 2 DAKSHESH DARURI MI 6.0 1553  
## 3 ADITYA BAJAJ MI 6.0 1384  
## 4 PATRICK H SCHILLING MI 5.5 1716  
## 5 HANSHI ZUO MI 5.5 1655  
## 6 HANSEN SONG OH 5.0 1686  
## 7 GARY DEE SWATHELL MI 5.0 1649  
## 8 EZEKIEL HOUGHTON MI 5.0 1641  
## 9 STEFANO LEE ON 5.0 1411  
## 10 ANVIT RAO MI 5.0 1365  
## 11 CAMERON WILLIAM MC LEMAN MI 4.5 1712  
## 12 KENNETH J TACK MI 4.5 1663  
## 13 TORRANCE HENRY JR MI 4.5 1666  
## 14 BRADLEY SHAW MI 4.5 1610  
## 15 ZACHARY JAMES HOUGHTON MI 4.5 1220  
## 16 MIKE NIKITIN MI 4.0 1604  
## 17 RONALD GRZEGORCZYK MI 4.0 1629  
## 18 DAVID SUNDEEN MI 4.0 1600  
## 19 DIPANKAR ROY MI 4.0 1564  
## 20 JASON ZHENG MI 4.0 1595  
## 21 DINH DANG BUI ON 4.0 1563  
## 22 EUGENE L MCCLURE MI 4.0 1555  
## 23 ALAN BUI ON 4.0 1363  
## 24 MICHAEL R ALDRICH MI 4.0 1229  
## 25 LOREN SCHWIEBERT MI 3.5 1745  
## 26 MAX ZHU ON 3.5 1579  
## 27 GAURAV GIDWANI MI 3.5 1552  
## 28 SOFIA ADINA STANESCU MI 3.5 1507  
## 29 CHIEDOZIE OKORIE MI 3.5 1602  
## 30 GEORGE AVERY JONES ON 3.5 1522  
## 31 RISHI SHETTY MI 3.5 1494  
## 32 JOSHUA PHILIP MATHEWS ON 3.5 1441  
## 33 JADE GE MI 3.5 1449  
## 34 MICHAEL JEFFERY THOMAS MI 3.5 1399  
## 35 JOSHUA DAVID LEE MI 3.5 1438  
## 36 SIDDHARTH JHA MI 3.5 1355  
## 37 AMIYATOSH PWNANANDAM MI 3.5 980  
## 38 BRIAN LIU MI 3.0 1423  
## 39 JOEL R HENDON MI 3.0 1436  
## 40 FOREST ZHANG MI 3.0 1348  
## 41 KYLE WILLIAM MURPHY MI 3.0 1403  
## 42 JARED GE MI 3.0 1332  
## 43 ROBERT GLEN VASEY MI 3.0 1283  
## 44 JUSTIN D SCHILLING MI 3.0 1199  
## 45 DEREK YAN MI 3.0 1242  
## 46 JACOB ALEXANDER LAVALLEY MI 3.0 377  
## 47 ERIC WRIGHT MI 2.5 1362  
## 48 DANIEL KHAIN MI 2.5 1382  
## 49 MICHAEL J MARTIN MI 2.5 1291  
## 50 SHIVAM JHA MI 2.5 1056  
## 51 TEJAS AYYAGARI MI 2.5 1011  
## 52 ETHAN GUO MI 2.5 935  
## 53 JOSE C YBARRA MI 2.0 1393  
## 54 LARRY HODGE MI 2.0 1270  
## 55 ALEX KONG MI 2.0 1186  
## 56 MARISA RICCI MI 2.0 1153  
## 57 MICHAEL LU MI 2.0 1092  
## 58 VIRAJ MOHILE MI 2.0 917  
## 59 SEAN M MC CORMICK MI 2.0 853  
## 60 JULIA SHEN MI 1.5 967  
## 61 JEZZEL FARKAS ON 1.5 955  
## 62 ASHWIN BALAJI MI 1.0 1530  
## 63 THOMAS JOSEPH HOSMER MI 1.0 1175  
## 64 BEN LI MI 1.0 1163  
## Opponent\_Pre\_Rating\_AVG  
## 1 1605.29  
## 2 1469.29  
## 3 1563.57  
## 4 1573.57  
## 5 1500.86  
## 6 1518.71  
## 7 1372.14  
## 8 1468.43  
## 9 1523.14  
## 10 1554.14  
## 11 1467.57  
## 12 1506.17  
## 13 1497.86  
## 14 1515.00  
## 15 1483.86  
## 16 1385.80  
## 17 1498.57  
## 18 1480.00  
## 19 1426.29  
## 20 1410.86  
## 21 1470.43  
## 22 1300.33  
## 23 1213.86  
## 24 1357.00  
## 25 1363.29  
## 26 1506.86  
## 27 1221.67  
## 28 1522.14  
## 29 1313.50  
## 30 1144.14  
## 31 1259.86  
## 32 1378.71  
## 33 1276.86  
## 34 1375.29  
## 35 1149.71  
## 36 1388.17  
## 37 1384.80  
## 38 1539.17  
## 39 1429.57  
## 40 1390.57  
## 41 1248.50  
## 42 1149.86  
## 43 1106.57  
## 44 1327.00  
## 45 1152.00  
## 46 1357.71  
## 47 1392.00  
## 48 1355.80  
## 49 1285.80  
## 50 1296.00  
## 51 1356.14  
## 52 1494.57  
## 53 1345.33  
## 54 1206.17  
## 55 1406.00  
## 56 1414.40  
## 57 1363.00  
## 58 1391.00  
## 59 1319.00  
## 60 1330.20  
## 61 1327.29  
## 62 1186.00  
## 63 1350.20  
## 64 1263.00

**Write this result table to an output file**

write.csv(ourFinalTable, "./Results.csv")

**Let’s Plot**

library(ggplot2)  
  
ggplot(ourFinalTable, aes(x=Player\_Pre\_Rating)) + geom\_histogram(binwidth = 50)



ggplot(ourFinalTable, aes(x=Opponent\_Pre\_Rating\_AVG)) + geom\_histogram(binwidth = 50)

