To Generate a **.csv** file from a chess tournament results

Nnaemeka Okereafor

About the project

In this project, we're given a text file with chess tournament results where the information has some structure. Our job was 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**. Illustration, for the first player, the information would be: Gary Hua, ON, 6.0, 1794, 1605

```
library(readr)
library(stringr)

df1<-read.table(file="https://raw.githubusercontent.com/nnaemeka-git/global-datasets/main/tournamentinfo
```

To remove dashes

```
df2 <- df1$V1
df2 <- str_replace_all(df2, pattern = "\\s+", replacement = " " )
df2 <- str_remove_all(df2, pattern = "-")
```

Names

To match **Players Names**

```
names <- unlist(str_match_all(df1, " [A-Za-z]{1,}\\s*[A-Za-z.,-]{1,}\\s*[A-Za-z.,-]{1,}\\s?"))
names
```

```
##  [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 "            " 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-BELLU "
## [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 "                  " JULIA SHEN "
## [61] " JEZZEL FARKAS "              " ASHWIN BALAJI "
## [63] " THOMAS JOSEPH HOSMER "       " BEN LI "
```

States

To match the names of the **States**

```
state <- unlist(str_match_all(df1, " [A-Za-z]{2}\\s\\|"))
state <-str_remove_all(state,"\\|")
state
```

```
##  [1] " ON " " MI " " MI " " MI " " MI " " OH " " MI " " MI " " ON " " MI "
## [11] " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI "
## [21] " ON " " MI " " ON " " MI " " MI " " ON " " MI " " MI " " MI " " ON "
## [31] " MI " " ON " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI "
## [41] " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI "
## [51] " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI " " MI "
## [61] " ON " " MI " " MI " " MI "
```

Points

To match the **Player Points**

```
pts <- unlist(str_match_all(df1,"\\|\\d+.."))
#To remove pipe
pts <-str_remove_all(pts,"\\|")
pts
```

```
##  [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"
## [13] "4.5" "4.5" "4.5" "4.0" "4.0" "4.0" "4.0" "4.0" "4.0" "4.0" "4.0" "4.0"
## [25] "3.5" "3.5" "3.5" "3.5" "3.5" "3.5" "3.5" "3.5" "3.5" "3.5" "3.5" "3.5"
## [37] "3.5" "3.0" "3.0" "3.0" "3.0" "3.0" "3.0" "3.0" "3.0" "3.0" "2.5" "2.5"
## [49] "2.5" "2.5" "2.5" "2.5" "2.0" "2.0" "2.0" "2.0" "2.0" "2.0" "2.0" "1.5"
## [61] "1.5" "1.0" "1.0" "1.0"
```

Pre Rating

To match the **pre-rating scores**

```
Pre_rate <-unlist(str_match_all(df1,"R.\\s.\\d+"))
Pre_rate <-as.numeric(str_remove_all(Pre_rate,'R:'))

# Convert into a dataframe
Pre_rate<-data.frame(Pre_rate)
colnames(Pre_rate)<-"rating"
index<-data.frame(as.numeric(row.names(Pre_rate)))
colnames(index)<-"Index"
pre_rate_df <- data.frame(index,Pre_rate)

head(pre_rate_df)
```

```
##   Index rating
## 1     1   1794
## 2     2   1553
## 3     3   1384
## 4     4   1716
## 5     5   1655
## 6     6   1686
```

Number of Games Played

To extract the **number of games played and the opponents' numbers**

```
num_game <- unlist(str_extract_all(df2,"\\|[0-9].*"))
num_game <- unlist(str_replace_all(num_game, "\\|[BUXH] ", replacement = "\\|R 0"))
```

```
#To remove the first part containing the total points
oppon <- unlist(str_remove_all(num_game, pattern = "\\|\\d\\.\\d\\s"))

#To remove the alphabets and the pipe
oppone <- unlist(str_remove_all(oppon, pattern = "[:alpha:]\\|"))

oppone <- as.numeric(unlist(str_extract_all(oppone, pattern = "[:digit:]{1,2}")))
head(oppone)
```

```
## [1] 39 21 18 14  7 12
```

To convert vectors into matrix and add columns to the matrix to represent the Opponents in each round of game

```
col_names=c("O1","O2","O3","O4","O5","O6","O7")
op_matr <- as.data.frame(matrix(oppone,byrow=TRUE,ncol=7,))
colnames(op_matr)<-col_names
op_matr
```

```
##    01 02 03 04 05 06 07
## 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  0  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  0 39  2 36  0
## 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  0 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  0
## 28 24  4 22 19 20  8 36
## 29 50  6 38 34 52 48  0
## 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  0 16 28
## 37  0  5 34 27  0 23 61
## 38 11 35 29 12  0 18 15
## 39  1 54 40 16 44 21 24
## 40 20 26 39 59 21 56 22
## 41 59 17 58 20  0  0  0
## 42 12 50 57 60 61 64 56
## 43 21 23 24 63 59 46 55
## 44  0 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  0 52  0 29 35
## 49 26 20 63 64 58  0  0
## 50 29 42 33 46  0 31 30
## 51 27 45 36 57 32 47 33
## 52 30 22 19 48 29 35 34
## 53  0 25  0 44  0 57  0
## 54 14 39 61  0 15 59 64
## 55 62 31 10 30  0 45 43
## 56  0 11 35 45  0 40 42
## 57  7 36 42 51 35 53  0
## 58 31  2 41 23 49  0 45
## 59 41  0  9 40 43 54 44
## 60 33 34 45 42 24  0  0
## 61 32  3 54 47 42 30 37
## 62 55  0  0  0  0  0  0
## 63  2 48 49 43 45  0  0
## 64 22 30 31 49 46 42 54
```

To extract the opponents's pre-rating scores

```
#nexted forloop does it better

for (row in 1:nrow(op_matr)){
  for (col in 1:ncol(op_matr)){
    if (op_matr[row,col] != 0){
      op_matr[row,col] = pre_rate_df$rating[op_matr[row,col]]
    } else {
       op_matr[row,col] = NA
     }
  }
}

op_matr
```

```
##      01   02   03   04   05   06   07
## 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  1092  377 1666 1712 1794 1411 1553
## 8  1384 1441 1610 1411 1362 1507 1564
## 9  1745 1600  853 1641 1579 1649 1595
## 10 1604 1564 1186 1494 1686 1745 1600
## 11 1423 1153 1686 1649 1384 1399 1579
## 12 1332 1449 1655 1423   NA 1794 1384
## 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   NA 1436 1553 1355   NA
## 17 1382 1403 1579 1553 1363 1555 1655
## 18 1362 1411 1794 1441 1564 1423 1365
## 19 1220 1365  935 1507 1600 1716 1641
## 20 1348 1291 1363 1403 1507 1553 1411
## 21 1283 1794 1362 1384 1348 1436 1686
## 22 1163  935 1507 1220   NA 1629 1348
## 23 1716 1283 1595  917 1629  980  377
## 24 1507 1362 1283 1745  967 1199 1436
## 25 1411 1393 1384 1229 1399 1365 1362
## 26 1291 1348 1629 1716 1411 1441 1712
## 27 1011 1666  377  980 1610 1686   NA
## 28 1229 1716 1555 1564 1595 1641 1355
## 29 1056 1686 1423 1399  935 1382   NA
## 30  935 1163 1220 1186 1494  955 1056
## 31  917 1186 1163 1365 1522 1056 1610
## 32  955 1641 1199 1600 1011 1579 1666
## 33  967 1663 1056 1355 1666 1220 1011
## 34 1686  967  980 1602 1745 1712  935
## 35  377 1423 1153 1686 1092  935 1382
## 36 1666 1092 1011 1449   NA 1604 1507
## 37   NA 1655 1399 1552   NA 1363  955
## 38 1712 1438 1602 1663   NA 1600 1220
## 39 1794 1270 1348 1604 1199 1563 1229
## 40 1595 1579 1436  853 1563 1153 1555
## 41  853 1629  917 1595   NA   NA   NA
## 42 1663 1056 1092  967  955 1163 1153
## 43 1563 1363 1229 1175  853  377 1186
## 44   NA 1610 1441 1393 1436 1229  853
## 45 1655 1011  967 1153 1175 1186  917
## 46 1438 1649 1552 1056 1163 1283 1363
## 47 1600 1229 1563  955 1641 1011 1745
## 48 1629 1175   NA  935   NA 1602 1438
## 49 1579 1595 1175 1163  917   NA   NA
## 50 1602 1332 1449  377   NA 1494 1522
## 51 1552 1242 1355 1092 1441 1362 1449
## 52 1522 1555 1564 1382 1602 1438 1399
## 53   NA 1745   NA 1199   NA 1092   NA
## 54 1610 1436  955   NA 1220  853 1163
## 55 1530 1494 1365 1522   NA 1242 1283
## 56   NA 1712 1438 1242   NA 1348 1332
## 57 1649 1355 1332 1011 1438 1393   NA
## 58 1494 1553 1403 1363 1291   NA 1242
## 59 1403   NA 1411 1348 1283 1270 1199
## 60 1449 1399 1242 1332 1229   NA   NA
## 61 1441 1384 1270 1362 1332 1522  980
## 62 1186   NA   NA   NA   NA   NA   NA
## 63 1553 1382 1291 1283 1242   NA   NA
## 64 1555 1522 1494 1291  377 1332 1270
```

To label opponents' pre-rating scores columns, count Number of Opponents, sum the total opponents' pre-rating score and calculate the Average opponents' pre-rating score

```
matr_names=c("O1 Pre-rating","O2 Pre-rating","O3 Pre-rating","O4 Pre-rating","O5 Pre-rating","O6 Pre-ra
colnames(op_matr)<-matr_names
op_matr$Num.of.Opponents <- rowSums(!is.na(op_matr))
op_matr$TotalPreOp_rating <- rowSums(op_matr[,1:7],na.rm=TRUE)
op_matr$AvgPreOp_rating <- round((op_matr$TotalPreOp_rating/op_matr$Num.of.Opponents),0)
head(op_matr,n=20)
```

```
##    O1 Pre-rating O2 Pre-rating O3 Pre-rating O4 Pre-rating O5 Pre-rating
## 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
##    O6 Pre-rating O7 Pre-rating Num.of.Opponents TotalPreOp_rating
## 1           1663          1716                7             11237
## 2           1595          1649                7             10285
## 3           1666          1663                7             10945
## 4           1564          1794                7             11015
## 5           1610          1629                7             10506
## 6           1552          1563                7             10631
## 7           1411          1553                7              9605
## 8           1507          1564                7             10279
## 9           1649          1595                7             10662
## 10          1745          1600                7             10879
## 11          1399          1579                7             10273
## 12          1794          1384                6              9037
## 13          1384          1441                7             10485
## 14          1655          1494                7             10605
## 15          1449          1423                7             10387
## 16          1355            NA                5              6929
## 17          1555          1655                7             10490
## 18          1423          1365                7             10360
## 19          1716          1641                7              9984
## 20          1553          1411                7              9876
##    AvgPreOp_rating
## 1             1605
## 2             1469
## 3             1564
## 4             1574
## 5             1501
## 6             1519
## 7             1372
## 8             1468
## 9             1523
## 10            1554
## 11            1468
## 12            1506
## 13            1498
## 14            1515
## 15            1484
## 16            1386
## 17            1499
## 18            1480
## 19            1426
## 20            1411
```

Combine dataframes to produce a single table

```
library(dplyr)
table <- data.frame("Player Name"=names, state=state, "Number of Points"=pts, pre_rate_df,op_matr)
head(table,n=15)
```

```
##                 Player.Name state Number.of.Points Index rating 01.Pre.rating
## 1                 GARY HUA    ON               6.0     1   1794          1436
## 2          DAKSHESH DARURI    MI               6.0     2   1553          1175
## 3             ADITYA BAJAJ    MI               6.0     3   1384          1641
## 4      PATRICK H SCHILLING    MI               5.5     4   1716          1363
## 5               HANSHI ZUO    MI               5.5     5   1655          1242
## 6              HANSEN SONG    OH               5.0     6   1686          1399
## 7        GARY DEE SWATHELL    MI               5.0     7   1649          1092
## 8         EZEKIEL HOUGHTON    MI               5.0     8   1641          1384
## 9              STEFANO LEE    ON               5.0     9   1411          1745
## 10               ANVIT RAO    MI               5.0    10   1365          1604
## 11      CAMERON WILLIAM MC    MI               4.5    11   1712          1423
## 12          KENNETH J TACK    MI               4.5    12   1663          1332
## 13       TORRANCE HENRY JR    MI               4.5    13   1666          1355
## 14            BRADLEY SHAW    MI               4.5    14   1610          1270
## 15  ZACHARY JAMES HOUGHTON    MI               4.5    15   1220          1564
##    02.Pre.rating 03.Pre.rating 04.Pre.rating 05.Pre.rating 06.Pre.rating
## 1           1563          1600          1610          1649          1663
## 2            917          1716          1629          1604          1595
## 3            955          1745          1563          1712          1666
## 4           1507          1553          1579          1655          1564
## 5            980          1663          1666          1716          1610
## 6           1602          1712          1438          1365          1552
## 7            377          1666          1712          1794          1411
## 8           1441          1610          1411          1362          1507
## 9           1600           853          1641          1579          1649
## 10          1564          1186          1494          1686          1745
## 11          1153          1686          1649          1384          1399
## 12          1449          1655          1423            NA          1794
## 13          1552          1649          1655          1449          1384
## 14          1199          1641          1794          1552          1655
## 15          1604          1522          1555          1270          1449
##    07.Pre.rating Num.of.Opponents TotalPreOp_rating AvgPreOp_rating
## 1           1716                7             11237            1605
## 2           1649                7             10285            1469
## 3           1663                7             10945            1564
## 4           1794                7             11015            1574
## 5           1629                7             10506            1501
## 6           1563                7             10631            1519
## 7           1553                7              9605            1372
## 8           1564                7             10279            1468
## 9           1595                7             10662            1523
## 10          1600                7             10879            1554
## 11          1579                7             10273            1468
## 12          1384                6              9037            1506
## 13          1441                7             10485            1498
## 14          1494                7             10605            1515
## 15          1423                7             10387            1484
```

Create a table with the required columns from the final table

```
chess_rating_table <- table%>% select(Player.Name,state,Number.of.Points,rating,AvgPreOp_rating)
chess_rating_table
```

```
##                     Player.Name state Number.of.Points rating AvgPreOp_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            1372
## 8              EZEKIEL HOUGHTON    MI               5.0   1641            1468
## 9                   STEFANO LEE    ON               5.0   1411            1523
## 10                    ANVIT RAO    MI               5.0   1365            1554
## 11           CAMERON WILLIAM MC    MI               4.5   1712            1468
## 12               KENNETH J TACK    MI               4.5   1663            1506
## 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            1386
## 17           RONALD GRZEGORCZYK    MI               4.0   1629            1499
## 18                DAVID SUNDEEN    MI               4.0   1600            1480
## 19                 DIPANKAR ROY    MI               4.0   1564            1426
## 20                  JASON ZHENG    MI               4.0   1595            1411
## 21                DINH DANG BUI    ON               4.0   1563            1470
## 22             EUGENE L MCCLURE    MI               4.0   1555            1300
## 23                     ALAN BUI    ON               4.0   1363            1214
## 24            MICHAEL R ALDRICH    MI               4.0   1229            1357
## 25             LOREN SCHWIEBERT    MI               3.5   1745            1363
## 26                      MAX ZHU    ON               3.5   1579            1507
## 27               GAURAV GIDWANI    MI               3.5   1552            1222
## 28   SOFIA ADINA STANESCU-BELLU    MI               3.5   1507            1522
## 29             CHIEDOZIE OKORIE    MI               3.5   1602            1314
## 30           GEORGE AVERY JONES    ON               3.5   1522            1144
## 31                 RISHI SHETTY    MI               3.5   1494            1260
## 32        JOSHUA PHILIP MATHEWS    ON               3.5   1441            1379
## 33                      JADE GE    MI               3.5   1449            1277
## 34      MICHAEL JEFFERY THOMAS    MI               3.5   1399            1375
## 35            JOSHUA DAVID LEE    MI               3.5   1438            1150
## 36               SIDDHARTH JHA    MI               3.5   1355            1388
## 37        AMIYATOSH PWNANANDAM    MI               3.5    980            1385
## 38                   BRIAN LIU    MI               3.0   1423            1539
## 39               JOEL R HENDON    MI               3.0   1436            1430
## 40                FOREST ZHANG    MI               3.0   1348            1391
## 41         KYLE WILLIAM MURPHY    MI               3.0   1403            1248
## 42                    JARED GE    MI               3.0   1332            1150
## 43           ROBERT GLEN VASEY    MI               3.0   1283            1107
## 44          JUSTIN D SCHILLING    MI               3.0   1199            1327
## 45                   DEREK YAN    MI               3.0   1242            1152
## 46    JACOB ALEXANDER LAVALLEY    MI               3.0    377            1358
## 47                 ERIC WRIGHT    MI               2.5   1362            1392
## 48                DANIEL KHAIN    MI               2.5   1382            1356
## 49            MICHAEL J MARTIN    MI               2.5   1291            1286
## 50                  SHIVAM JHA    MI               2.5   1056            1296
## 51              TEJAS AYYAGARI    MI               2.5   1011            1356
## 52                   ETHAN GUO    MI               2.5    935            1495
## 53               JOSE C YBARRA    MI               2.0   1393            1345
## 54                 LARRY HODGE    MI               2.0   1270            1206
## 55                   ALEX KONG    MI               2.0   1186            1406
## 56                MARISA RICCI    MI               2.0   1153            1414
## 57                  MICHAEL LU    MI               2.0   1092            1363
## 58                VIRAJ MOHILE    MI               2.0    917            1391
## 59                   SEAN M MC    MI               2.0    853            1319
## 60                  JULIA SHEN    MI               1.5    967            1330
## 61               JEZZEL FARKAS    ON               1.5    955            1327
## 62               ASHWIN BALAJI    MI               1.0   1530            1186
## 63        THOMAS JOSEPH HOSMER    MI               1.0   1175            1350
## 64                      BEN LI    MI               1.0   1163            1263
```

Write the dataframe to a csv file

```
write.csv(chess_rating_table, row.names = FALSE)
```

```
## "Player.Name","state","Number.of.Points","rating","AvgPreOp_rating"
## " GARY HUA "," ON ","6.0",1794,1605
## " DAKSHESH DARURI "," MI ","6.0",1553,1469
## " ADITYA BAJAJ "," MI ","6.0",1384,1564
## " PATRICK H SCHILLING "," MI ","5.5",1716,1574
## " HANSHI ZUO "," MI ","5.5",1655,1501
## " HANSEN SONG "," OH ","5.0",1686,1519
## " GARY DEE SWATHELL "," MI ","5.0",1649,1372
## " EZEKIEL HOUGHTON "," MI ","5.0",1641,1468
## " STEFANO LEE "," ON ","5.0",1411,1523
## " ANVIT RAO "," MI ","5.0",1365,1554
## " CAMERON WILLIAM MC "," MI ","4.5",1712,1468
## " KENNETH J TACK "," MI ","4.5",1663,1506
## " TORRANCE HENRY JR "," MI ","4.5",1666,1498
## " BRADLEY SHAW "," MI ","4.5",1610,1515
## " ZACHARY JAMES HOUGHTON "," MI ","4.5",1220,1484
## " MIKE NIKITIN "," MI ","4.0",1604,1386
## " RONALD GRZEGORCZYK "," MI ","4.0",1629,1499
## " DAVID SUNDEEN "," MI ","4.0",1600,1480
## " DIPANKAR ROY "," MI ","4.0",1564,1426
## " JASON ZHENG "," MI ","4.0",1595,1411
## " DINH DANG BUI "," ON ","4.0",1563,1470
## " EUGENE L MCCLURE "," MI ","4.0",1555,1300
## " ALAN BUI "," ON ","4.0",1363,1214
## " MICHAEL R ALDRICH "," MI ","4.0",1229,1357
## " LOREN SCHWIEBERT "," MI ","3.5",1745,1363
## " MAX ZHU "," ON ","3.5",1579,1507
## " GAURAV GIDWANI "," MI ","3.5",1552,1222
## " SOFIA ADINA STANESCU-BELLU "," MI ","3.5",1507,1522
## " CHIEDOZIE OKORIE "," MI ","3.5",1602,1314
## " GEORGE AVERY JONES "," ON ","3.5",1522,1144
## " RISHI SHETTY "," MI ","3.5",1494,1260
## " JOSHUA PHILIP MATHEWS "," ON ","3.5",1441,1379
## " JADE GE "," MI ","3.5",1449,1277
## " MICHAEL JEFFERY THOMAS "," MI ","3.5",1399,1375
## " JOSHUA DAVID LEE "," MI ","3.5",1438,1150
## " SIDDHARTH JHA "," MI ","3.5",1355,1388
## " AMIYATOSH PWNANANDAM "," MI ","3.5",980,1385
## " BRIAN LIU "," MI ","3.0",1423,1539
## " JOEL R HENDON "," MI ","3.0",1436,1430
## " FOREST ZHANG "," MI ","3.0",1348,1391
## " KYLE WILLIAM MURPHY "," MI ","3.0",1403,1248
## " JARED GE "," MI ","3.0",1332,1150
## " ROBERT GLEN VASEY "," MI ","3.0",1283,1107
## " JUSTIN D SCHILLING "," MI ","3.0",1199,1327
## " DEREK YAN "," MI ","3.0",1242,1152
## " JACOB ALEXANDER LAVALLEY "," MI ","3.0",377,1358
## " ERIC WRIGHT "," MI ","2.5",1362,1392
## " DANIEL KHAIN "," MI ","2.5",1382,1356
## " MICHAEL J MARTIN "," MI ","2.5",1291,1286
## " SHIVAM JHA "," MI ","2.5",1056,1296
## " TEJAS AYYAGARI "," MI ","2.5",1011,1356
## " ETHAN GUO "," MI ","2.5",935,1495
## " JOSE C YBARRA "," MI ","2.0",1393,1345
## " LARRY HODGE "," MI ","2.0",1270,1206
## " ALEX KONG "," MI ","2.0",1186,1406
## " MARISA RICCI "," MI ","2.0",1153,1414
## " MICHAEL LU "," MI ","2.0",1092,1363
## " VIRAJ MOHILE "," MI ","2.0",917,1391
## " SEAN M MC "," MI ","2.0",853,1319
## " JULIA SHEN "," MI ","1.5",967,1330
## " JEZZEL FARKAS "," ON ","1.5",955,1327
## " ASHWIN BALAJI "," MI ","1.0",1530,1186
## " THOMAS JOSEPH HOSMER "," MI ","1.0",1175,1350
## " BEN LI "," MI ","1.0",1163,1263
```

Writing chess_rating_table into a directory

```
#write.csv(chess_rating_table, "choice_file_name.csv", row.names = FALSE)
```