607_Project1_NoahCollin

Noah Collin

9/16/2021

Project 1

Reading in the file

This look a lot of trial and error to figure out which read function to use, much more than 4 attempts.

```
test4 <- readLines(file("tournamentinfo.txt"))
```

```
## Warning in readLines(file("tournamentinfo.txt")): incomplete final line found on
## 'tournamentinfo.txt'
```

```
tail(test4)
```

```
## [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] "-----------------------------------------------------------------------------------------"
```

Divide Data into header (which was 2 lines) and data:

```
test5 <- test4[-seq(from = 1,to=length(test4), by=3)]
#head(test5, 12)

header <- paste(test5[1],test5[2])
#header
test5 <-  test5[3:length(test5)]
#head(test5)
```

Strings Extract

```
player_names <- c()
Total_Col <- c()
player_states <- c()
pre_ranks <- c()
post_ranks <- c()
whole_lines <- c()

library(stringr)
for (line in 1:length(test5)) {
  if (line %%2 == 1) {
    player_names <- c(player_names,unlist(str_extract_all(test5[line],"\\w+[^USCF|a-z] ?\\w+ \\w+")))

    whole_lines <-  c(whole_lines,(str_extract_all(test5[line],"(\\d+\\.?\\d?)")))

  }
  if (line %%2 ==0) {
    player_states <-  c(player_states,
                        unlist(str_extract_all(test5[line],"[A-Z][A-Z]")))

    pre_ranks <- c(pre_ranks,
                   unlist(str_extract_all(test5[line], "R:\\s+(\\d+)")))

    post_ranks <- c(post_ranks,
                   unlist(str_extract_all(test5[line], "->\\s*(\\d+)")))

  }

}
Total_Col <- as.numeric(Total_Col)
head(player_states)
```

```
## [1] "ON" "MI" "MI" "MI" "MI" "OH"
```

```
# Turn ranks to numerics:
pre_ranks_cleaned <- as.numeric(unlist(str_extract_all(pre_ranks,"\\d+")))
post_ranks_cleaned <- as.numeric(unlist(str_extract_all(post_ranks,"\\d+")))


#To test the first RegEx of player names:
#unlist((str_extract_all(test5[1],"\\w+[^USCF|a-z] ?\\w+ \\w+")))
```

Get means of ranks and fix The total score column

```
Total_Col <- c()
rank_means <- c()
for (p in whole_lines) {
  #print(p[2])
  p <- (as.numeric(p))
  matches <- p[3:length(p)]
  #print(matches)
  #print(mean(pre_ranks_cleaned[matches]))
```

```
  rank_means <- c(rank_means,mean(pre_ranks_cleaned[matches]))
  Total_Col <- c(Total_Col, p[2])
}
Total_Col <-  Total_Col[1:64]
# I don't know why the above line was necessary but it was.  When I printed p[2] above, it was length 6
```

DataFrame

```
df <- data.frame(player_names,player_states,Total_Col,pre_ranks_cleaned,post_ranks_cleaned,rank_means)
df
```

```
##                player_names player_states Total_Col pre_ranks_cleaned
## 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            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           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
##    post_ranks_cleaned rank_means
## 1                1817   1605.286
## 2                1663   1469.286
## 3                1640   1563.571
## 4                1744   1573.571
## 5                1690   1500.857
## 6                1687   1518.714
## 7                1673   1372.143
## 8                1657   1468.429
## 9                1564   1523.143
## 10               1544   1554.143
## 11               1696   1467.571
## 12               1670   1506.167
## 13               1662   1497.857
## 14               1618   1515.000
## 15               1416   1483.857
## 16               1613   1385.800
## 17               1610   1498.571
## 18               1600   1480.000
## 19               1570   1426.286
## 20               1569   1410.857
## 21               1562   1470.429
## 22               1529   1300.333
## 23               1371   1213.857
## 24               1300   1357.000
## 25               1681   1363.286
## 26               1564   1506.857
## 27               1539   1221.667
## 28               1513   1522.143
```

```
## 29              1508   1313.500
## 30              1444   1144.143
## 31              1444   1259.857
## 32              1433   1378.714
## 33              1421   1276.857
## 34              1400   1375.286
## 35              1392   1149.714
## 36              1367   1388.167
## 37              1077   1384.800
## 38              1439   1539.167
## 39              1413   1429.571
## 40              1346   1390.571
## 41              1341   1248.500
## 42              1256   1149.857
## 43              1244   1106.571
## 44              1199   1327.000
## 45              1191   1152.000
## 46              1076   1357.714
## 47              1341   1392.000
## 48              1335   1355.800
## 49              1259   1285.800
## 50              1111   1296.000
## 51              1097   1356.143
## 52              1092   1494.571
## 53              1359   1345.333
## 54              1200   1206.167
## 55              1163   1406.000
## 56              1140   1414.400
## 57              1079   1363.000
## 58               941   1391.000
## 59               878   1319.000
## 60               984   1330.200
## 61               979   1327.286
## 62              1535   1186.000
## 63              1125   1350.200
## 64              1112   1263.000
```

Write CSV

```
write.csv(df,"NoahCollin_607_Project1.csv",quote=FALSE,row.names = F)
```