Project1

Paul Perez

9/22/2019

Project 1: Chess Tournament Data

In this project, we're given a text file with chess tournament results where the information has some structure. Our job is to create an R Markdown file that generates a .CSV file (that could for example be imported into a SQL database). That file should contain the Player's Name, Player's State, Total Number of Points, Player's Pre-Rating, and Average Pre Chess Ratings of Opponents.

Data

We've receievd a .txt file that is tab delimited ("\t") and need to clean it up. The information provided on the sheet is the players rank, name location, rating (both before & after the tournament), the number of points they've scored, and the results of all their matches.

```
data <- readLines("tournamentinfo.txt")
head(data)
```

```
## [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    |"
```

Cleaning the Data

We can see that our data actually starts on the 5th row, so we'll skip the first 4 rows. To do this, we can create a data frame out of the data from the file, and then slice out those first 4 rows. Note: I'm using the length of the original data import to guage the number of rows available. I can also check using tail() but this will dynamically adjust should there be another tournament file available.

```
df <- data.frame(data)
df <- df[5:length(data),]
head(df)
```

```
## [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] -----------------------------------------------------------------------------------------
## 131 Levels: -----------------------------------------------------------------------------------------
```

After several attemps of ingesting the data and converting the tabs to commas, the only successful way to access the data was through the sequence function. Since the rows are not consistent with data, we'll have to create two variables for different sets of sequences. Looking at the head of the dataframe above, we see that we can extract the players rating, name, score, and results from row 1, and the players location, and ranks pre/post tournament.

```
players <- df[seq(1, length(df), 3)]
ratings <- df[seq(2, length(df), 3)]
```

Next we'll have to extract data from each of these sequences to prep for our final data frame.

```
library(stringr)
id <- as.integer(str_extract(players, "[[:digit:]]+"))
names <- str_trim(str_extract(players, "([[:alpha:]]+\\s){2,3}"))
location <- str_extract(ratings, "[[:alpha:]]+")
scores <- str_extract(players, "\\d+\\.\\d+")
rating <- as.integer(str_extract(str_extract(ratings, "[^\\d]\\d{3,4}[^\\d]"), "\\d+"))
opponents <- str_extract_all(str_extract_all(players,"[[:digit:]]+\\|"), "[[:digit:]]+")
```

```
## Warning in stri_extract_all_regex(string, pattern, simplify = simplify, :
## argument is not an atomic vector; coercing
```

Get Average of Opponents Ratings

We're going to have to identify the opponents of each player, and then get a mean rating for each to help determine the strength of their opponents.

```
opponent_ratings <- length(players)
for (i in 1:length(players)) {
  opponent_ratings[i] <- round(mean(rating[as.numeric(unlist(opponents[as.integer(id)[i]]))]), digits =
}
```

Create Data Frame

Utilizing the knitr library, we can create a table to list the results. Additionally, we'll add titles to each of the columns.

```
library(knitr)
ndf <- data.frame(names, location, scores, rating, opponent_ratings)
colnames(ndf) <- c("Player's Name", "Player's State", "Total Number of Points", "Player's Pre-Rating",
kable(ndf, caption = "Chess Tournament Results", format = "html")
```

Chess Tournament Results

Player's Name

Player's State

Total Number of Points

Player's Pre-Rating

Average Pre Chess Rating of Opponents

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

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 Data Frame to .csv

```r
write.csv(ndf, file = "tournament_results.csv")
```