Project 1

Adam Douglas

9/16/2018

Text File

First we read in the raw chess text file. By using the `read_table` method from the `readr` package gets us a single long character field for each row:

```
chess <- read_table("chess.txt",col_names="txt")
```

```
head(chess)
```

```
## # A tibble: 6 x 1
##   txt
##   <chr>
## 1 -----------------------------------------------------------------------…
## 2 Pair | Player Name                     |Total|Round|Round|Round|Round|R…
## 3 Num  | USCF ID / Rtg (Pre->Post)       | Pts |  1  |  2  |  3  |  4  | …
## 4 -----------------------------------------------------------------------…
## 5 1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   …
## 6 ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B    |W   …
```

Data Prep

Before we can extract the actual data we are looking to record, we need to do a few steps to prepare the raw text data:

```
# Filter to the rows with data only by looking for the pipe character
chess <- chess %>% filter(str_detect(chess$txt,"\\|"))

# Eliminate the first two header rows
chess <- chess[3:length(chess$txt),]

# Split the lines by pipe "|" characters
temp <- str_split(chess$txt,"\\|")
```

Now that we have a cleaner dataset, we now have to parse through and get the specific data we're after. First we parse the odd rows:

```
# We parse the odd rows to get: number, name, points, and opponent numbers

# Set-up the initial vectors, by initializing with dummy data
number <- -999
names(number) <- "number"

name <- "DUMMY DATA"
names(name) <- "name"

point <- 0.0
names(point) <- "points"

opponent <- data.frame(rbind(c(0,0,0,0,0,0,0)))
names(opponent) <- c("opp1","opp2","opp3","opp4","opp5","opp6","opp7")

# Iterate through each odd line
for(i in seq(1,length(temp),2)){
  x <- temp[[i]][[1]] %>% str_trim() %>% parse_number() # number
  n <- temp[[i]][[2]] %>% str_trim() # name
  p <- temp[[i]][[3]] %>% str_trim() %>% parse_number() # points
  s <- data.frame(rbind(str_extract(temp[[i]][4:10],"[[:digit:]]+") %>% parse_number(
))) # opponents
  names(s) <- c("opp1","opp2","opp3","opp4","opp5","opp6","opp7")

  number <- rbind(number, x)
  name <- rbind(name,n)
  point <- rbind(point,p)
  opponent <- rbind(opponent,s)
}
```

Next, the even rows:

```
# We parse the even rows to get: state, and rating

# Set-up the initial vectors, by initializing with dummy data
state <- "XX"
names(state) <- "state"

rating <- 0
names(rating) <- "rating"

# Iterate through each even line
for(i in seq(2,length(temp),2)){
  t <- temp[[i]][[1]] %>% str_trim() # state
  r <- temp[[i]][[2]] %>% str_extract("(?<=R:\\s{0,4})[[:digit:]]+") %>% parse_number() # rating

  state <- rbind(state,t)
  rating <- rbind(rating,r)
}
```

Now we put it all together:

```
# Combine all the elements
data <- data.frame(number,name,state,rating,point,opponent,stringsAsFactors = FALSE)
row.names(data) <- data$number
data <- filter(data, number != -999) # Remove our dummy row
```

Finally, we do our opponent rating lookups:

```
# A helper function to get a player's rating
getRating <- function(n){
  if(is.na(n)) return(NA)
  data %>% filter(number == n) %>% select(rating)
}

# Get the ratings into a new data frame
opp <- data.frame(opp1 = unlist(lapply(data$opp1,getRating), use.names = FALSE),
                  opp2 = unlist(lapply(data$opp2,getRating), use.names = FALSE),
                  opp3 = unlist(lapply(data$opp3,getRating), use.names = FALSE),
                  opp4 = unlist(lapply(data$opp4,getRating), use.names = FALSE),
                  opp5 = unlist(lapply(data$opp5,getRating), use.names = FALSE),
                  opp6 = unlist(lapply(data$opp6,getRating), use.names = FALSE),
                  opp7 = unlist(lapply(data$opp7,getRating), use.names = FALSE)
                  )
# Fine the mean opponent rating
data$oppMean <- round(apply(opp,1,mean, na.rm=TRUE),0)
```

Export

We have all of the data elements we need and can now export to a CSV file:

```
# Select and export our fields
data %>% select(name,state,points,rating,oppMean) %>% write_csv("data.csv", col_names
=FALSE)
```