DATA607 Project 1

Gabriel Campos

February 25 2021

Assignment Requirements

Procedure

Import Data

Data is imported from the text file path leading to `tournamentinfo.txt` from my github Project 1 folder. Function `read.delim` from the `utils` package is used.

```
# Store Github url to variable
txtfile=
"https://raw.githubusercontent.com/gcampos100/DATA607Spring2021/main/Projects/Project%201/
tournamentinfo.txt"
```

The text file is converted into a data.frame separated by the delimiter `|` and not claiming a `header`, resulting in 11 columns vs. the actual 10 in the file.

```
my_data <- as.data.frame(read.delim(txtfile,header = FALSE,stringsAsFactors = FALSE, sep = "|"))
```

Note[1]

```
##                                                                                           V1
## 1 -----------------------------------------------------------------------------------------
## 2                                                                                     Pair 
## 3                                                                                     Num  
## 4 -----------------------------------------------------------------------------------------
##                                   V2    V3    V4    V5    V6    V7    V8    V9
## 1                                                                               
## 2  Player Name                      Total Round Round Round Round Round Round
## 3  USCF ID / Rtg (Pre->Post)         Pts    1     2     3     4     5     6  
## 4                                                                               
```

Creating .CSV

Step 1: Remove `---` rows and `NULL` values

The data.frame my_data is a 2 dimensional data.frame containing `11 columns` and `195 rows`. Rows composed completely of `---` and the additional all `NULL` column created on import needs to be removed.

[1] Column 10 was ommitted from head() for formatting:

```
#Find rows to delete, multiples of 3
toDelete    <- seq(1, length(my_data$V1), 3)
#remove rows
my_data     <- my_data[-toDelete ,]
# remove column with NULL Values
my_data[11] <-NULL
```

Note[2]

```
##       V1                                V2    V3    V4    V5    V6    V7    V8
## 2  Pair   Player Name                      Total Round Round Round Round Round
## 3  Num    USCF ID / Rtg (Pre->Post)         Pts    1     2     3     4     5
## 5     1   GARY HUA                         6.0   W  39 W  21 W  18 W  14 W   7
```

Step 2: Subset header and body

The header is composed of 2 rows that require merging and separation by column. In order to do so and to avoid affecting the additional data, I subset the first two rows individually and separate the table's body content.

```
# Subset column names and remainder of vectors
# Subset row 1 of my data
my_data_names <- my_data[1,]
# Subset row 2 of my data
my_data_names_2 <- my_data[2,]
# Subset remainder of my data
my_data<-my_data[3:NROW(my_data),]
```

Step 3: Use header subsets to create column names

Using `str_replace` in combination with `gsub()` and some `regex` expressions, I am able to remove the unnecessary information from the rows then combine them. The end result will be the exact or near exact column names needed for this dataset.

```
# Merge both vectors and seperate by ‘,‘
my_data_names <- paste(my_data_names,my_data_names_2,collapse = ",")
# Replace any character issues
my_data_names <- str_replace(my_data_names, "/",",")
my_data_names <- str_replace(my_data_names, "->"," , ")
my_data_names <- str_replace(my_data_names, "\\(","")
my_data_names <- str_replace(my_data_names, "Post\\)","Rtg Post")
# Remove excessive spacing
my_data_names <-str_replace(gsub("\\s+", " ", str_trim(my_data_names)), "B", "b")
# Add comma to attribute ‘Player Name‘
my_data_names <-str_replace(gsub("Player Name", "Player Name,",my_data_names),"B", "b")
```

```
## [1] "Pair Num , Player Name, USCF ID , Rtg Pre , Rtg Post ,Total Pts ,Round 1 ,Round 2 ,Round 3 ,Rou
```

[2] Column 9 & 10 was ommitted from head() for formatting:

Step 4 : Use body subset to create columns

The relevant data is split among 2 rows, ∴ I separated the data set into 2 halves in order to rejoin in combined columns. Similar as with the column names, I used a `sequence` to store the `index` of all odd number `indexes` in the `data frame` and created `subsets` of the relevant data.

```r
# Subset of top using index
top    <- seq(1, length(my_data$V1), 2)
my_data_top         <- my_data[top ,]
# Subset of bottom using index
bottom     <- seq(2, length(my_data$V1), 2)
my_data_bottom          <- my_data[bottom ,]
```

Note[3]

```
## [1] "TOP"
```

```
##                                  V2    V3    V4    V5    V6    V7    V8    V9
## 5  GARY HUA                         6.0   W  39 W  21 W  18 W  14 W   7 D  12
## 8  DAKSHESH DARURI                  6.0   W  63 W  58 L   4 W  17 W  16 W  20
```

```
## [1] "BOTTOM"
```

```
##                                  V2    V3    V4    V5    V6    V7    V8    V9
## 6  15445895 / R: 1794   ->1817      N:2   W     B     W     B     W     B
## 9  14598900 / R: 1553   ->1663      N:2   B     W     B     W     B     W
```

Step 5 : First Data cleanup and consolidation

In order to merge the data cleanly and to allow easy edit of output, I:

- converted each `subset` into a `list`
- Removed data in `my_data_bottom` not needed for the assignment
- Combined the rows into a single column
- Then cleanup any characters or symbols that inhibite creating of a table

```r
# List conversion
my_data_top <- as.list(t(my_data_top))
my_data_bottom<- as.list(t(my_data_bottom))
```

```r
# Unnecessary Data removal
my_data_bottom<-str_replace_all(gsub("MI", " ",my_data_bottom),"B", "b")
my_data_bottom<-str_replace_all(gsub("ON", " ",my_data_bottom),"B", "b")
my_data_bottom<-str_replace_all(gsub("W", " ",my_data_bottom),"B", "b")
my_data_bottom<-str_replace_all(gsub("B", " ",my_data_bottom),"B", "b")
my_data_bottom<-str_replace_all(gsub("b", " ",my_data_bottom),"B", "b")
my_data_bottom<-str_replace_all(gsub("N\\:.", " ",my_data_bottom),"B", "b")
```

[3] Columns 1 & 10 was ommitted from head() for formatting:

```
# clean paste or merger of rows
my_data <- paste(my_data_top,my_data_bottom,collapse = ",")
```

```
# Overall cleanup of data
my_data <-str_replace(gsub("/ R:", ",",my_data),"B", "b")
my_data <-str_replace(gsub("->", ",",my_data),"B", "b")
my_data <-str_replace(gsub("\\s+", " ", str_trim(my_data)), "B", "b")
```

The result of the cleanup is a list of all variables, comma separated, **WITH EXCEPTION** of `Player Name` and `USCF ID`.

```
##  chr "1 , GARY HUA 15445895 , 1794 ,1817 ,6.0 ,W 39 ,W 21 ,W 18 ,W 14 ,W 7 ,D 12 ,D 4 , 2 , DAKSHESH
```

Step 6: Data Frame recreation

Using the list I recreate the data body data frame with column name data

```
my_data<-strsplit(my_data[1],",")
my_data <- data.frame(matrix(unlist(my_data), ncol=12, byrow=TRUE))
```

Note[4]

```
##    X1                         X2     X3    X4   X5    X6    X7    X8    X9
## 1  1          GARY HUA 15445895   1794  1817  6.0  W 39  W 21  W 18  W 14
## 2  2   DAKSHESH DARURI 14598900   1553  1663  6.0  W 63  W 58   L 4  W 17
```

Specifically column `X2` needs to be separated using `str_split_fixed`

```
## [1] " GARY HUA 15445895 "
```

```
## [1] " DAKSHESH DARURI 14598900 "
```

```
quick.split<-str_split_fixed(my_data$X2, "[A-Z]\\s[0-9]", 2)
my_data<-cbind(my_data[1],quick.split,my_data[3:12])
```

This will to create the 13 columns we require with `my_data_names`

```
##   X1        1        2     X3    X4   X5    X6    X7    X8    X9  X10   X11
## 1 1   GARY HU 5445895   1794  1817  6.0  W 39  W 21  W 18  W 14  W 7  D 12
##    X12
## 1 D 4
```

Then the names can be added using the vector

```
colnames(my_data)<-unlist(strsplit(my_data_names, ","))
```

```
##   Pair Num   Player Name  USCF ID   Rtg Pre   Rtg Post  Total Pts  Round 1
## 1        1       GARY HU  5445895      1794       1817        6.0     W 39
##   Round 2  Round 3  Round 4  Round 5  Round 6  Round 7
## 1    W 21     W 18     W 14      W 7     D 12      D 4
```

[4] Column 10-12 was ommitted from head() for formatting: