Week4_Project1

Vishal Arora

February 24, 2019

Introduction

The problem statement is to have the raw data loaded from txt file into R dataframe. The ask from the problem is to give an output in CSV file which contains Player's Name, Player's State, Total Number of Points, Player's Pre-Rating, and Average Pre Chess Rating of Opponents.

Loading necessary libraries:-

'tidyr','dplyr','stringr','sqldf'

Solution to the problem

First load the txt using the read delim method with sep='|' and you get the data loaded in dataframe . Use select function from tidyr package to select the df and limit the columns you want to retain for your dataframe. Using Filter function of dplyr package you filter out the rows which are unwanted, and then construct 2 different data frames from our original data frame.One containing the player(s) playing details mainly comprising of (Player name/Points/Rounds he played) and an additional fields names player id as primary key. Second dataframe contains the remaining information for that player e.g. (State,USCF code, pre-rating, post ratings and what was the color of the pieces s(he) played with) and again player_id as primaey key.

```
my_data <- read.delim("tournamentinfo.txt" , header = FALSE ,  sep = "|" )
glimpse(my_data)
```

```
## Observations: 196
## Variables: 11
## $ V1  <fct> -----------------------------------------------------------...
## $ V2  <fct> ,  Player Name                     ,  USCF ID / Rtg (Pre->...
## $ V3  <fct> , Total,  Pts , , 6.0  , N:2  , , 6.0  , N:2  , , 6.0  , N...
## $ V4  <fct> , Round,   1  , , W  39, W    , , W  63, B    , , L   8, W...
## $ V5  <fct> , Round,   2  , , W  21, B    , , W  58, W    , , W  61, B...
## $ V6  <fct> , Round,   3  , , W  18, W    , , L   4, B    , , W  25, W...
## $ V7  <fct> , Round,   4  , , W  14, B    , , W  17, W    , , W  21, B...
## $ V8  <fct> , Round,   5  , , W   7, W    , , W  16, B    , , W  11, W...
## $ V9  <fct> , Round,   6  , , D  12, B    , , W  20, W    , , W  13, B...
## $ V10 <fct> , Round,   7  , , D   4, W    , , W   7, B    , , W  12, W...
## $ V11 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
```

```
df <-  select(my_data , 1:10)

df <- filter(df,!grepl(pattern = "[-]+" , V1)) %>%
      dplyr::filter(row_number() > 2)

df_player_det <- df %>%
        dplyr::filter(grepl(pattern = "[[:digit:]]", V1)) %>%
        dplyr::mutate(player_id = row_number())

df_player_rating <-  df %>%
```

```
        dplyr::filter(grepl(pattern = "[[:alpha:]]", V1))  %>%
        dplyr::mutate(player_id = row_number())
```

Next step is to seperate the joint values and spread the dataframe where ever possible e.g. seperating out USCF and Ratings fields and then again seperating out Pre-rating and Post-ratings for every player. Then we rearranged the Data frames by using select and having player_id as first column in both the new dataframes.Using the merge function of dplyr to merge both the dataframes inyo one dataframe. Next step we do is using select, mutate and str_extract functions we clean the value of pre-rating for each row and also give our dataframe columns a meaningfull name.

```
rating <- separate( df_player_rating , V2 , c("USCF" , "Rtg") , sep = "R: ")
rating <- separate(rating , Rtg , c("PreRating" , "Postrating") , sep = "->" )

df_player_det <- select(df_player_det, player_id, everything())
df_player_rating <- select(rating ,  player_id , V1 , PreRating , Postrating)
playerdet <- merge(df_player_det , df_player_rating , by="player_id")
playerdet <- select(playerdet , 1:13 , -2 ) %>%
             mutate(PreRating = str_extract(PreRating,"[:digit:]+") )
colnames(playerdet) <- c("Player_ID", "Player Name", "Total Points", "Match1", "Match2", "Match3", "Mat

str(playerdet)
```

```
## 'data.frame':    64 obs. of  12 variables:
##  $ Player_ID   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Player Name : Factor w/ 131 levels "","  10131499 / R: 1610   ->1618     ",..: 89 77 66 116 93 92 8
##  $ Total Points: Factor w/ 18 levels "","     "," Pts ",..: 14 14 14 13 13 12 12 12 12 12 ...
##  $ Match1      : Factor w/ 67 levels "","     ","  1  ",..: 51 66 18 43 54 48 62 41 45 8 ...
##  $ Match2      : Factor w/ 69 levels "","     ","  2  ",..: 47 66 68 8 53 9 60 50 27 28 ...
##  $ Match3      : Factor w/ 68 levels "","     ","  3  ",..: 47 14 50 42 6 18 45 19 67 64 ...
##  $ Match4      : Factor w/ 69 levels "","     ","  4  ",..: 47 49 50 52 6 56 46 23 45 54 ...
##  $ Match5      : Factor w/ 61 levels "","     ","  5  ",..: 40 42 41 6 5 8 16 51 45 7 ...
##  $ Match6      : Factor w/ 66 levels "","     ","  6  ",..: 6 47 44 46 45 50 43 51 21 49 ...
##  $ Match7      : Factor w/ 61 levels "","     ","  7  ",..: 6 37 38 5 39 43 9 41 42 40 ...
##  $ State       : Factor w/ 70 levels "-----------------------------------------------------------------
##  $ PreRating   : chr  "1794" "1553" "1384" "1716" ...
```

In the next steps using the gather function we expand the dataframe in number of rows by having multiple rows for one player id i.e. by removing the round columns. Then arrange the rows by Player_ID and then using seperate function we again bifurcate a single column into multple column(s), in current scenario we bifurcate each round column into 2 columns and in last using transform we coonvert the data type of few columns.

Lastly we use sqldf library to load the dataframe as table which gives us the facility of running sql queries against our sqldf dataframe to have 2 new dataframes constructed from our datafranme , one with al the details and one with only player_id and average opponent rating having group by on player_id gives us one row per player _id with average of all the opponent pre-ratings. Using the merge function we merge the original dataframe before sqldf and the dataframe we got from using sqldf which contains the avereage of opponent ratings for each player. Using Select we construct our final dataframeand using write csv function we write our dataframe to comma seperated csv.

```
data(playerdet1)
```

```
## Warning in data(playerdet1): data set 'playerdet1' not found
```

```
temp <- sqldf("select a.Player_ID , (select PreRating from playerdet1 b where b.Player_ID = a.OppID ) Op
glimpse(temp)
```

```
## Observations: 448
## Variables: 2
## $ Player_ID  <int> 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, ...
## $ Opp_rating <int> 1436, 1563, 1600, 1610, 1649, 1663, 1716, 1175, 917...
```

```
temp <- sqldf("select Player_ID , round(avg(Opp_rating)) 'Opponent Prerating' from temp group by Player_
head(temp)
```

```
##   Player_ID Opponent Prerating
## 1         1               1605
## 2         2               1469
## 3         3               1564
## 4         4               1574
## 5         5               1501
## 6         6               1519
```

```
combinedDataSet <- merge(playerdet,temp, by="Player_ID")
finalDataSet <- select(combinedDataSet, 2, 3,11,12,13, -1 )
glimpse(finalDataSet)
```

```
## Observations: 64
## Variables: 5
## $ `Player Name`        <fct>  GARY HUA                          ,  DAKSH...
## $ `Total Points`       <fct> 6.0  , 6.0  , 6.0  , 5.5  , 5.5  , 5.0  ,...
## $ State                <fct>    ON ,    MI ,    MI ,    MI ,    MI ,  ...
## $ PreRating            <chr> "1794", "1553", "1384", "1716", "1655", "...
## $ `Opponent Prerating` <dbl> 1605, 1469, 1564, 1574, 1501, 1519, 1372,...
```

```
write.csv(finalDataSet,"Project1.csv",row.names=FALSE)
```

Summary

This exercise helped us to use various packages of R e.g. dplyr,tidyr,stringr and sqldf through which we manipulated the dataframes thus achieving the target of cleaning/purging data for visualisation.