AP Assignment

Poonam Nair [sid: 15136653]

2022-06-05

By including this statement, we the authors of this work, verify that: \* We hold a copy of this assignment that we can produce if the original is lost or damaged. \* We hereby certify that no part of this assignment/product has been copied from any other student’s work or from any other source except where due acknowledgement is made in the assignment. No part of this assignment/product has been wrien/produced for us by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned. \* We are aware that this work may be reproduced and submied to plagiarism detection soware programs for the purpose of detecting possible plagiarism (which may retain a copy on its database for future plagiarism checking). \* We hereby certify that we have read and understand what the School of Computing, Engineering and Mathematics denes as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

# QUESTION 1

## Write the code to compute the total scored by each team for each match. Note that each goal is worth 10 points and the team that caught the snitch is awarded an extra 50 points. Run your code on the data and present the results in a table showing the season and round number, the match number (a new variable that identies each match of the season), the home and away team names and their scores.

## DATA PREPROCESSING

df <- read.csv("competitionResults.csv")  
  
head(df)

## Season round homeTeam awayTeam  
## 1 2012 1 United States Chimeras Malaysia Tornadoes  
## 2 2012 1 Durhamstrang Denmark Horntails  
## 3 2012 1 England Nightmares France Hippogriffs Firsts  
## 4 2012 1 Scotland Squirrels New Zealand Daemons  
## 5 2012 1 Wales Longshots Australia Centurions  
## 6 2012 1 Ireland Griffins Japan Falcons  
## snitch homeGoals awayGoals  
## 1 United States Chimeras 8 5  
## 2 Durhamstrang 11 12  
## 3 England Nightmares 14 10  
## 4 Scotland Squirrels 11 6  
## 5 Wales Longshots 5 6  
## 6 Ireland Griffins 1 0

## THE ALGORITHM

# [RESULT] Total scored by each team for each match. Each goal is worth 10 points, team that caught the snitch awarded extra 50 points.   
# \* the season and round number  
# \* the match number (a new variable that identifies each match of the season)  
# \* the home and away team names and their scores  
seasonTable <- data.frame(matrix(ncol=7, nrow=0))  
headings <- c("season", "round", "matchNo", "homeTeam", "homeTeamPoints", "awayTeam", "awayTeamPoints")  
colnames(seasonTable) <- headings  
print(seasonTable)

## [1] season round matchNo homeTeam homeTeamPoints  
## [6] awayTeam awayTeamPoints  
## <0 rows> (or 0-length row.names)

for (match\_number in 1:nrow(df)) {  
 match <- df[match\_number,]  
 totalPointsHomeTeam <- match[["homeGoals"]] \* 10 + ifelse(match[["homeTeam"]] == match[["snitch"]], 50, 0)  
 totalPointsAwayTeam <- match[["awayGoals"]] \* 10 + ifelse(match[["awayTeam"]] == match[["snitch"]], 50, 0)  
  
 seasonTable[match\_number, 1] <- match[["Season"]]  
 seasonTable[match\_number, 2] <- match[["round"]]  
 seasonTable[match\_number, 3] <- match\_number  
 seasonTable[match\_number, 4] <- match[["homeTeam"]]  
 seasonTable[match\_number, 5] <- totalPointsHomeTeam  
 seasonTable[match\_number, 6] <- match[["awayTeam"]]  
 seasonTable[match\_number, 7] <- totalPointsAwayTeam  
}  
  
head(seasonTable)

## season round matchNo homeTeam homeTeamPoints  
## 1 2012 1 1 United States Chimeras 130  
## 2 2012 1 2 Durhamstrang 160  
## 3 2012 1 3 England Nightmares 190  
## 4 2012 1 4 Scotland Squirrels 160  
## 5 2012 1 5 Wales Longshots 100  
## 6 2012 1 6 Ireland Griffins 60  
## awayTeam awayTeamPoints  
## 1 Malaysia Tornadoes 50  
## 2 Denmark Horntails 120  
## 3 France Hippogriffs Firsts 100  
## 4 New Zealand Daemons 60  
## 5 Australia Centurions 60  
## 6 Japan Falcons 0

# QUESTION 2

## The winner of each match has highest score. Each team scores 3 points for a win, 1 point for a draw. Team with most points at the end of season is the season winner.

teams\_associatedPoints <- data.frame(matrix(ncol=2, nrow=0))  
headings <- c("teamName", "totalPoints")  
colnames(teams\_associatedPoints) <- headings  
  
# I built a function and passed a set (dataframe) and n (first n in set) variable  
buildResultsLadder <- function(set, n\_matches) {  
 # aggregate points for each unique team in set  
 for(i in 1:n\_matches) {  
 match <- set[i,]  
   
 # award points: win, draw  
 if(match[["homeTeamPoints"]] > match[["awayTeamPoints"]]) {  
 homeTeamFinalPoints <- match[["homeTeamPoints"]] + 3  
 awayTeamFinalPoints <- match[["awayTeamPoints"]]  
 } else if(match[["homeTeamPoints"]] == match[["awayTeamPoints"]]) {  
 homeTeamFinalPoints <- match[["homeTeamPoints"]] + 1  
 awayTeamFinalPoints <- match[["awayTeamPoints"]] + 1  
 } else if(match[["homeTeamPoints"]] < match[["awayTeamPoints"]]){  
 homeTeamFinalPoints <- match[["homeTeamPoints"]]  
 awayTeamFinalPoints <- match[["awayTeamPoints"]] + 3  
 } else {  
 # do nothing  
 }  
   
 # aggregate team points  
 homeTeamIdx <- which(teams\_associatedPoints$teamName == match[["homeTeam"]])  
 awayTeamIdx <- which(teams\_associatedPoints$teamName == match[["awayTeam"]])  
   
 if(length(homeTeamIdx) != 0) {  
 teams\_associatedPoints[homeTeamIdx, 1] <- match[["homeTeam"]]  
 teams\_associatedPoints[homeTeamIdx, 2] <- homeTeamFinalPoints  
 } else {  
 # get next row index  
 nextRowIdx <- ifelse(length(teams\_associatedPoints$teamName) == 0, 1, length(teams\_associatedPoints$teamName) + 1)  
 teams\_associatedPoints[nextRowIdx, 1] <- match[["homeTeam"]]  
 teams\_associatedPoints[nextRowIdx, 2] <- homeTeamFinalPoints  
 }  
   
 if(length(awayTeamIdx) != 0) {  
 teams\_associatedPoints[awayTeamIdx, 1] <- match[["awayTeam"]]  
 teams\_associatedPoints[awayTeamIdx, 2] <- awayTeamFinalPoints  
 } else {  
 nextRowIdx <- ifelse(length(teams\_associatedPoints$teamName) == 0, 1, length(teams\_associatedPoints$teamName) + 1)  
 teams\_associatedPoints[nextRowIdx, 1] <- match[["awayTeam"]]  
 teams\_associatedPoints[nextRowIdx, 2] <- awayTeamFinalPoints  
 }  
   
 }  
 # Order teams by points descending  
 teams\_associatedPoints\_ordered <- order(teams\_associatedPoints$totalPoints, decreasing = TRUE)   
 # returns index of order, used to ORDER tables  
 resultsLadder <- teams\_associatedPoints[teams\_associatedPoints\_ordered,]  
 print(resultsLadder)  
}  
  
# [RESULT] Results Ladder: Each team’s points as per the first 5 rounds (ordered by points)  
first5MatchesSet <- seasonTable[1:5,]  
first\_n\_matches <- length(first5MatchesSet$season)  
buildResultsLadder(first5MatchesSet, first\_n\_matches)

## teamName totalPoints  
## 5 England Nightmares 193  
## 3 Durhamstrang 163  
## 7 Scotland Squirrels 163  
## 1 United States Chimeras 133  
## 4 Denmark Horntails 120  
## 9 Wales Longshots 103  
## 6 France Hippogriffs Firsts 100  
## 8 New Zealand Daemons 60  
## 10 Australia Centurions 60  
## 2 Malaysia Tornadoes 50

# [RESULT] Results Ladder: Each team’s points for the full season (ordered by points)  
season2016Set <- subset(seasonTable, subset=(season == 2016))  
season\_no\_matches <- length(season2016Set$season)  
buildResultsLadder(season2016Set, season\_no\_matches)

## teamName totalPoints  
## 9 Wales Longshots 203  
## 3 Durhamstrang 173  
## 2 Turkey Knights 163  
## 12 New Zealand Daemons 153  
## 5 England Nightmares 150  
## 10 France Hippogriffs Firsts 143  
## 7 Scotland Squirrels 123  
## 13 France Thestrals 120  
## 11 Ireland Griffins 103  
## 16 Japan Falcons 63  
## 8 Denmark Horntails 60  
## 4 Pakistan Swans 50  
## 6 Malaysia Tornadoes 50  
## 14 Australia Centurions 50  
## 15 Germany Werewolves 50  
## 1 United States Chimeras 30

# Pick the season who’s last digit corresponds to 3.  
season2013Set <- subset(seasonTable, subset=(season == 2013))  
season\_no\_matches <- length(season2013Set$season)  
buildResultsLadder(season2013Set, season\_no\_matches)

## teamName totalPoints  
## 1 United States Chimeras 183  
## 2 Malaysia Tornadoes 173  
## 6 France Hippogriffs Firsts 153  
## 9 Wales Longshots 130  
## 12 Japan Falcons 113  
## 7 Scotland Squirrels 103  
## 13 France Thestrals 103  
## 5 England Nightmares 100  
## 4 Denmark Horntails 80  
## 3 Durhamstrang 70  
## 8 New Zealand Daemons 63  
## 11 Ireland Griffins 40  
## 14 Germany Werewolves 40  
## 10 Australia Centurions 30

# QUESTION 3

## An investigation into home game advantage is planned for the coming months. Write a function that returns a count of the number of home games each team has had, the total number of games each team has played in, the total points scored in a home game and the total number of points the team has scored. Provide these results in a data frame. Run the function on the provided data and show the resulting table output.

Function below with associated getTotalPoints() I’ve created to handle point/goal calculations. Scroll to next code snippet to test. code

## Please note: normally, I would use more of a functional programming approach (centralisation and usage of modules), but for time efficiency, I have not fully done this.   
homeGameCalculations <- function (cf) {  
 # numeric vars  
 n\_matches = nrow(competition.results.df)  
 # data vars  
 team.statistics <- data.frame(matrix(ncol=5, nrow=0))  
 colnames(team.statistics) <- c("team", "home.games", "total.games", "total.points.homegame", "total.points")  
  
 for(i in 1:n\_matches) {  
 match <- competition.results.df[i,]  
 homeTeam <- match$homeTeam  
 awayTeam <- match$awayTeam  
   
 # indexes  
 homeTeam.idx <- which(team.statistics$team == homeTeam)  
 awayTeam.idx <- which(team.statistics$team == awayTeam)  
 nextIdx <- length(team.statistics$team)  
   
 # Hometeam stat calculations  
 homeIdx <- ifelse(nextIdx == 0, 1, ifelse(length(homeTeam.idx) != 0, homeTeam.idx, nextIdx+1))  
   
 # Awayteam stat calculations  
 awayIdx <- ifelse(nextIdx == 0, 2, ifelse(length(awayTeam.idx) != 0, awayTeam.idx, nextIdx+2))  
   
 team.statistics[homeIdx, 1] <- homeTeam   
 team.statistics[awayIdx, 1] <- awayTeam   
   
 # The number of HOME games each team has had  
 totalHomeGames <- ifelse(is.na(team.statistics[homeIdx, 2]), 1, team.statistics[homeIdx, 2] + 1)  
 team.statistics[homeIdx, 2] <- totalHomeGames   
 team.statistics[awayIdx, 2] <- ifelse(is.na(team.statistics[awayIdx, 2]), 0, team.statistics[awayIdx, 2])  
   
 # The total number of games each team has played in  
 totalHomeTeamGames <- ifelse(is.na(team.statistics[homeIdx, 3]), 1, team.statistics[homeIdx, 3] + 1)  
 team.statistics[homeIdx, 3] <- totalHomeTeamGames   
 totalAwayTeamGames <- ifelse(is.na(team.statistics[awayIdx, 3]), 1, team.statistics[awayIdx, 3] + 1)  
 team.statistics[awayIdx, 3] <- totalAwayTeamGames   
   
 # The total points scored in a home game  
 totalHomeGamePoints <- ifelse(is.na(team.statistics[homeIdx, 4]), 1, team.statistics[homeIdx, 4] + match$homeGoals)  
 team.statistics[homeIdx, 4] <- totalHomeGamePoints  
 team.statistics[awayIdx, 4] <- ifelse(is.na(team.statistics[awayIdx, 4]), 0, team.statistics[awayIdx, 4])  
   
 # total points overall  
 totalPointsForMatch <- 0 # initialise variable here  
 whoGotSnitch <- match$snitch  
 totalPointsForMatch <- getTotalPoints(homeTeam, awayTeam, match$homeGoals, match$awayGoals, totalHomeGamePoints, team.statistics[awayIdx, 5], whoGotSnitch)  
   
 team.statistics[homeIdx, 5] <- totalPointsForMatch[1]  
 team.statistics[awayIdx, 5] <- totalPointsForMatch[2]  
 }  
   
 print(team.statistics)  
  
}  
  
getTotalPoints <- function(homeTeam, awayTeam, homeTeamGoals, awayTeamGoals, homeTeamPoints, awayTeamPoints, whoGotSnitch) {  
 homeTeamFinalPoints <- 0  
 awayTeamFinalPoints <- 0  
   
 # convert goals to points  
 homeTeamFinalPoints <- homeTeamGoals \* 10  
 awayTeamFinalPoints <- awayTeamGoals \* 10  
  
 # check NA on awayTeamPoints  
 awayTeamPoints <- ifelse(is.na(awayTeamPoints),0,awayTeamPoints)  
   
 # award points: win, draw  
 if(homeTeamPoints > awayTeamPoints) {  
 homeTeamFinalPoints <- homeTeamPoints + 3  
 awayTeamFinalPoints <- awayTeamPoints  
 } else if(homeTeamPoints == awayTeamPoints) {  
 homeTeamFinalPoints <- homeTeamPoints + 1  
 awayTeamFinalPoints <- awayTeamPoints + 1  
 } else if(homeTeamPoints < awayTeamPoints){  
 homeTeamFinalPoints <- homeTeamPoints  
 awayTeamFinalPoints <- awayTeamPoints + 3  
 } else {  
 # do nothing  
 }  
   
 # snitch bonus points  
 if(homeTeam == whoGotSnitch) {  
 homeTeamFinalPoints <- homeTeamFinalPoints + 50  
 } else if (awayTeam == whoGotSnitch) {  
 awayTeamFinalPoints <- awayTeamFinalPoints + 50  
 }  
  
 total.match.points <- c(homeTeamFinalPoints, awayTeamFinalPoints)  
 total.match.points  
}  
  
competition.results.df <- read.csv("competitionResults.csv")  
   
# Run the function on the provided data and show the resulting table output.  
homeGameCalculations(competition.results.df)

## team home.games total.games total.points.homegame  
## 1 United States Chimeras 164 328 1210  
## 2 Malaysia Tornadoes 164 328 1234  
## 3 Durhamstrang 164 328 1209  
## 4 Denmark Horntails 164 328 1311  
## 5 England Nightmares 164 328 1151  
## 6 France Hippogriffs Firsts 164 328 1265  
## 7 Scotland Squirrels 164 328 1211  
## 8 New Zealand Daemons 164 328 1282  
## 9 Wales Longshots 164 328 1303  
## 10 Australia Centurions 164 328 1240  
## 11 Ireland Griffins 164 328 1149  
## 12 Japan Falcons 164 328 1184  
## 13 France Thestrals 164 328 1148  
## 14 Germany Werewolves 164 328 1268  
## 15 <NA> NA NA NA  
## 16 Turkey Knights 138 276 1052  
## 17 <NA> NA NA NA  
## 18 Pakistan Swans 138 276 1029  
## 19 <NA> NA NA NA  
## 20 Norway Unspeakables 93 186 702  
## 21 <NA> NA NA NA  
## 22 South Africa Snidgets 93 186 681  
## 23 <NA> NA NA NA  
## 24 Iceland Quidlings 59 118 439  
## 25 <NA> NA NA NA  
## 26 India Direwolves 59 118 452  
## 27 <NA> NA NA NA  
## 28 Portugal Manticores 21 42 127  
## 29 <NA> NA NA NA  
## 30 China Nifflers 21 42 118  
## total.points  
## 1 1673  
## 2 1284  
## 3 1259  
## 4 1311  
## 5 1204  
## 6 1595  
## 7 1267  
## 8 1559  
## 9 1412  
## 10 1514  
## 11 1308  
## 12 1452  
## 13 1360  
## 14 1536  
## 15 NA  
## 16 1052  
## 17 NA  
## 18 1079  
## 19 NA  
## 20 752  
## 21 NA  
## 22 681  
## 23 NA  
## 24 489  
## 25 NA  
## 26 452  
## 27 NA  
## 28 127  
## 29 NA  
## 30 168

## Testing function

# PUT HERE

# QUESTION 4

## Provide a table containing the team statistics for all seasons. Including, but not limited to the following: number of games, number of wins, winning percentage, points scored, points conceded, number of snitch catches, goals scored, number of tournament wins. Order the teams in terms of total points scored.

library(expss)

## Loading required package: maditr

##   
## To select rows from data: rows(mtcars, am==0)

s1 <- data.frame(matrix(nrow=2,ncol=2))  
colnames(s1) <- c("test","test1")  
s1[2,1] <- "a"  
s1[2,2] <- "a\_test1"  
  
s\_1 <- list("2012", s1)  
  
s2 <- data.frame(matrix(nrow=2,ncol=2))  
colnames(s1) <- c("test","test1")  
s2[2,1] <- "b"  
s2[2,2] <- "b\_test1"  
  
  
s\_1 <- list("2012", s1)  
s\_2 <- list("2013", s2)  
nnn <- list()  
  
nnn[[1]] <- s\_1  
# print(length(nnn))  
#print(nnn[[2]][2])

# Provide a table containing the team statistics for all seasons.   
c\_data <- read.csv("competitionResults.csv")  
seasonList <- list()  
n\_matches <- length(c\_data$Season)  
  
 for(i in 1:1) { #TODO : only for one item  
 match <- c\_data[i,]  
 homeTeam <- match$homeTeam  
 awayTeam <- match$awayTeam  
 season <- match$Season  
   
 ###############################  
 # Season data  
 ###############################  
 # if seasonList does't have season, create, else extract subset  
 seasonSubset <- data.frame(matrix(nrow=0,ncol=9))  
 colnames(seasonSubset) <- c("team","no.games","no.wins","win.percentage","points.scored","points.conceded","total.snitches","goals","tournament.wins")  
 seasonExistsIndex <- 0  
 homeTeamExistsIndex <- 0  
 awayTeamExistsIndex <- 0  
  
 if(length(seasonList) > 0) {  
 for(s in length(seasonList)) {  
 if(season == seasonList[[s]][1]) {  
 seasonSubset <- seasonList[[s]][2]  
 # get match from season subset  
 for(ss in 1:length(seasonSubset$no.games)) {  
 if(seasonSubset$team == match$homeTeam) {  
 homeTeam <- seasonSubset  
 homeTeamExistsIndex <- ss  
 }  
 if(seasonSubset$team == match$awayTeam) {  
 awayTeam <- seasonSubset  
 awayTeamExistsIndex <- ss  
 }  
 }  
 seasonExistsIndex <- s  
 }  
 }  
 }  
   
 ###############################  
 # Match statistics  
 ###############################  
 # calculate  
 # Number of games  
 totalHomeGames <- ifelse(homeTeamExistsIndex == 0, 1, seasonSubset[homeTeamExistsIndex, 2] + 1)  
 totalAwayGames <- ifelse(awayTeamExistsIndex == 0, 1, seasonSubset[awayTeamExistsIndex, 2] + 1)  
  
 # Number of points  
 totalHomePoints <- ifelse(homeTeamExistsIndex != 0, ifelse(is.na(seasonSubset[homeTeamExistsIndex, 5]),match$homeGoals \* 10,seasonSubset[homeTeamExistsIndex, 5]), match$homeGoals \* 10)  
 totalAwayPoints <- ifelse(homeTeamExistsIndex != 0, ifelse(is.na(seasonSubset[homeTeamExistsIndex, 5]),match$homeGoals \* 10,seasonSubset[homeTeamExistsIndex, 5]), match$homeGoals \* 10)  
  
 # Number of wins  
 totalHomeWins <- ifelse(homeTeamExistsIndex == 0 && totalHomePoints > totalAwayPoints, seasonSubset[homeTeamExistsIndex, 3] + 1, seasonSubset[homeTeamExistsIndex, 3])  
 totalAwayWins <- ifelse(awayTeamExistsIndex == 0 && totalHomePoints < totalAwayPoints, seasonSubset[awayTeamExistsIndex, 3] + 1, seasonSubset[awayTeamExistsIndex, 3])  
   
 # Points conceded  
 homePointsConceded <- ifelse(homeTeamExistsIndex != 0, ifelse(is.na(seasonSubset[homeTeamExistsIndex, 6]), totalAwayPoints, seasonSubset[homeTeamExistsIndex, 6] + totalAwayPoints), totalHomePoints)  
 awayPointsConceded <- ifelse(awayTeamExistsIndex != 0, ifelse(is.na(seasonSubset[awayTeamExistsIndex, 6]), totalHomePoints, seasonSubset[awayTeamExistsIndex, 6] + totalHomePoints), totalAwayPoints)  
  
 # colnames(seasonSubset) <- c("team","no.games","no.wins","win.percentage","points.scored","points.conceded","total.snitches","goals","tournament.wins")  
   
 # Winning percentage  
 totalHomeWinPerc <- totalHomeWins / totalHomeGames  
 totalAwayWins <- totalAwayWins / totalAwayGames  
   
 # no of snitch catches, goals scored, number of tournament wins.  
  
 ###############################  
 # Update/Insert season  
 ###############################  
 # insert partial data into season data  
   
 h.team.idx = ifelse(homeTeamExistsIndex != 0, homeTeamExistsIndex, length(seasonSubset$team) + 1)  
 seasonSubset[h.team.idx,1] <- homeTeam  
 seasonSubset[h.team.idx,2] <- totalHomeGames  
 seasonSubset[h.team.idx,3] <- totalHomeWins  
 seasonSubset[h.team.idx,4] <- 0  
 seasonSubset[h.team.idx,5] <- totalHomePoints  
 seasonSubset[h.team.idx,6] <- homePointsConceded  
 seasonSubset[h.team.idx,7] <- 0  
 seasonSubset[h.team.idx,8] <- 0  
 seasonSubset[h.team.idx,9] <- 0  
   
 a.team.idx = ifelse(awayTeamExistsIndex != 0, awayTeamExistsIndex, length(seasonSubset$team) + 1)  
 seasonSubset[a.team.idx,1] <- awayTeam  
 seasonSubset[a.team.idx,2] <- totalAwayGames  
 seasonSubset[a.team.idx,3] <- totalAwayWins  
 seasonSubset[a.team.idx,4] <- 0  
 seasonSubset[a.team.idx,5] <- totalAwayPoints  
 seasonSubset[a.team.idx,6] <- awayPointsConceded  
 seasonSubset[a.team.idx,7] <- 0  
 seasonSubset[a.team.idx,8] <- 0  
 seasonSubset[a.team.idx,9] <- 0  
  
 # insert season subset back into season  
 if(seasonExistsIndex != 0) {  
 seasonList[[seasonExistsIndex]][[1]] <- season  
 seasonList[[seasonExistsIndex]][[2]] <- seasonSubset  
 }else {  
 seasonList[[(length(seasonList) + 1)]] <- list(season, seasonSubset)  
 }  
 }  
  
# TODO: I couldn't easily add subheaders for seasons, so I'm outputting the entire list instead  
 # for(i in length(seasonList)) {  
 # print(seasonList[[i]][2][[1]])  
 # }  
   
 print(seasonList)

## [[1]]  
## [[1]][[1]]  
## [1] 2012  
##   
## [[1]][[2]]  
## team no.games no.wins win.percentage points.scored  
## 1 United States Chimeras 1 NA 0 80  
## 2 Malaysia Tornadoes 1 NA 0 80  
## points.conceded total.snitches goals tournament.wins  
## 1 80 0 0 0  
## 2 80 0 0 0

# TODO: Teams ordered in terms of total points scored.