1341 Final Project

Andrew Windsheimer

12/10/2021

#Import data  
############################  
scores <- read.csv("nhl\_scores.csv")  
  
scores[scores=="Chicago Blackhawks"] <- "Chicago Black Hawks"  
scores[scores=="Dallas Stars"] <- "Minnesota North Stars"  
scores[scores=="Carolina Hurricanes"] <- "Hartford Whalers"  
scores[scores=="Colorado Avalanche"] <- "Quebec Nordiques"  
scores[scores=="Arizona Coyotes"] <- "Winnipeg Jets"  
  
names(scores)[[1]] <- "id"  
  
pre\_season <- scores[scores$year<=1982 | (scores$year==1983 & scores$month<10),]  
season\_schedule <- scores[(scores$year==1983 & scores$month>=10)|(scores$year==1984 & scores$month<=4 & scores$game\_type=="r"),]  
  
team\_info <- read.csv("nhl\_initial\_elos.csv")  
  
names(team\_info)[[1]] <- names(season\_schedule)[[1]] <- "team"  
############################  
  
# Obtain list of unique conference names and unique division names  
conferences = na.omit(unique(team\_info$conference))  
divisions = na.omit(unique(team\_info$division))  
  
weight = 7.4  
hfa = 55.7522039863812  
  
# Determine preseason ELOs  
############################  
# Iterate through all games in the sport's history up to season being simulated  
for(i in 1:nrow(pre\_season)) {  
 # Find indices corresponding to home and away teams for current game  
 home\_index = which(team\_info$team == pre\_season$home\_team[i])  
 away\_index = which(team\_info$team == pre\_season$away\_team[i])  
  
 # Find home and away team Elo ratings  
 home\_elo = team\_info$rating[home\_index]  
 away\_elo = team\_info$rating[away\_index]  
  
 # Calculate home team win probability  
 win\_prob = 1 / (10^((away\_elo - (home\_elo + hfa\*pre\_season$neutral[i]))/400) + 1)  
  
 # Calculate actual margin of victory - must be positive  
 score\_diff = abs(pre\_season$home\_score[i] - pre\_season$away\_score[i])  
  
 # Determine home team result  
 if(pre\_season$home\_score[i] > pre\_season$away\_score[i]) {  
 home\_result = 1 # Home team wins  
 } else if(pre\_season$home\_score[i] < pre\_season$away\_score[i]) {  
 home\_result = 0 # Home team loses  
 } else {  
 home\_result = 0.5 # Tie  
 }  
  
 # Calculate amount each team's Elo rating is adjusted by  
 home\_elo\_adjustment = weight \* log(score\_diff + 1) \* (home\_result - win\_prob)  
  
 # Adjust Elo ratings - add point to winner and subtract points from loser  
 team\_info$rating[home\_index] = team\_info$rating[home\_index] + home\_elo\_adjustment  
 team\_info$rating[away\_index] = team\_info$rating[away\_index] - home\_elo\_adjustment  
  
 # Adjust Elo ratings at end of season to regress 1/3 of the way towards 1500  
 if(i < nrow(scores) && scores$season[i+1] > scores$season[i]) {  
 for(j in 1:nrow(team\_info)) {  
 if(scores$season[i] >= team\_info$inaugural\_season[j]) {  
 team\_info$rating[j] = team\_info$rating[j] - (team\_info$rating[j] - 1500)/3  
 }  
 }  
  
 # Identify all teams that existed at beginning of following season  
 existing\_teams = team\_info[which(team\_info$inaugural\_season <= (scores$season[i] + 1)),]  
  
 # Calculate amount each team's Elo rating must be adjusted by to make mean 1500  
 expansion\_adjustment = -1\*(mean(existing\_teams$rating) - 1500)  
  
 # Perform expansion adjustment on teams that existed at beginning of following season  
 for(j in 1:nrow(team\_info)) {  
 if((scores$season[i] + 1) >= team\_info$inaugural\_season[j]) {  
 team\_info$rating[j] = team\_info$rating[j] + expansion\_adjustment  
 }  
 }  
 }  
}  
############################  
  
#Run season simulations  
set.seed(99)  
# Specify season to simulate  
iterations = 10000  
  
team\_info = team\_info[which(team\_info$conference != 'NA'),]  
summary = data.frame(matrix(0, ncol = 8, nrow = nrow(team\_info)))  
colnames(summary) = c("team", "average\_points", "playoffs", "division\_titles", "div\_semi\_winner", "div\_final\_champ", "conference\_championships", "championships")  
summary$team = team\_info$team  
  
histories = data.frame(matrix(0, ncol = nrow(team\_info), nrow = iterations))  
colnames(histories) = team\_info$team  
  
team\_info[,c("team", "rating")]

## team rating  
## 1 Boston Bruins 1578.074  
## 2 Buffalo Sabres 1539.047  
## 3 Hartford Whalers 1409.570  
## 4 Montreal Canadiens 1559.431  
## 5 Quebec Nordiques 1499.244  
## 6 Chicago Black Hawks 1531.412  
## 7 Detroit Red Wings 1418.871  
## 8 Minnesota North Stars 1525.035  
## 9 St. Louis Blues 1458.988  
## 10 Toronto Maple Leafs 1457.545  
## 11 New Jersey Devils 1403.009  
## 12 New York Islanders 1605.022  
## 13 New York Rangers 1520.586  
## 14 Philadelphia Flyers 1550.334  
## 15 Pittsburgh Penguins 1414.398  
## 16 Washington Capitals 1511.922  
## 17 Calgary Flames 1492.356  
## 18 Edmonton Oilers 1595.837  
## 19 Los Angeles Kings 1454.736  
## 20 Vancouver Canucks 1493.126  
## 21 Winnipeg Jets 1470.559

for(i in 1:iterations) {  
 season\_stats = team\_info[,which(colnames(team\_info) != "inaugural\_season")]  
 season\_stats$points = 0  
 season\_stats$rand = runif(nrow(team\_info))  
   
 for(j in 1:nrow(season\_schedule)) {  
 # Find indices corresponding to home and away teams for current game  
 home\_index = which(season\_stats$team == season\_schedule$home\_team[j])  
 away\_index = which(season\_stats$team == season\_schedule$away\_team[j])  
   
 # Find home and away team Elo ratings  
 home\_elo = season\_stats$rating[home\_index]  
 away\_elo = season\_stats$rating[away\_index]  
  
 # Calculate home team win and tie probabilities  
 tie\_prob = (1/(sqrt(4\*pi))) \* exp(-((away\_elo - (home\_elo + hfa\*season\_schedule$neutral[j]))^2/160000))  
 win\_prob = 1 / (10^((away\_elo - (home\_elo + hfa\*season\_schedule$neutral[j]))/400) + 1) - 0.50\*tie\_prob  
 u = runif(1)  
   
 if(u < win\_prob + 0.4375\*tie\_prob) { # Home team wins in regulation/OT  
 season\_stats$points[home\_index] = season\_stats$points[home\_index] + 2  
 } else if(u > win\_prob + .5625\*tie\_prob) { # Away team wins in regulation/OT  
 season\_stats$points[away\_index] = season\_stats$points[away\_index] + 2  
 } else { # Tie after overtime: assumption is 12.5% chance  
 season\_stats$points[home\_index] = season\_stats$points[home\_index] + 1  
 season\_stats$points[away\_index] = season\_stats$points[away\_index] + 1  
 }  
  
 # Calculate actual margin of victory - must be positive  
 score\_diff = abs(season\_schedule$home\_score[j] - season\_schedule$away\_score[j])   
   
 # Determine home team result  
 if(season\_schedule$home\_score[j] > season\_schedule$away\_score[j]) {   
 home\_result = 1 # Home team wins  
 } else if(season\_schedule$home\_score[j] < season\_schedule$away\_score[j]) {   
 home\_result = 0 # Home team loses  
 } else {   
 home\_result = 0.5 # Tie  
 }  
   
 # Calculate amount each team's Elo rating is adjusted by  
 home\_elo\_adjustment = weight \* log(score\_diff + 1) \* (home\_result - win\_prob)  
   
 # Adjust Elo ratings after game has been simulated to get team's new strength  
 season\_stats$rating[home\_index] = season\_stats$rating[home\_index] + home\_elo\_adjustment  
 season\_stats$rating[away\_index] = season\_stats$rating[away\_index] - home\_elo\_adjustment  
 }  
   
 summary$average\_points = summary$average\_points + season\_stats$points  
   
 #No wild cards needed in 1983-84 format: the top four teams are taken from each division, then played in 1 vs 4, 2 vs 3 format.   
   
 division\_winners = data.frame(matrix(ncol = 6, nrow = 0))  
 colnames(division\_winners) = c("team", "conference", "division", "rating", "points", "rand")  
  
 non\_division\_winners = data.frame(matrix(ncol = 6, nrow = 0))  
 colnames(non\_division\_winners) = c("team", "conference", "division", "rating", "points", "rand")  
  
 for(div in divisions) {  
 div\_standings = season\_stats[which(season\_stats$division == div),]  
 div\_standings = div\_standings[order(-div\_standings$points, -div\_standings$rand),]  
 division\_winners = rbind(division\_winners, div\_standings[1:4,])  
 non\_division\_winners = rbind(non\_division\_winners, div\_standings[5:nrow(div\_standings),])  
 }  
   
 division\_winners = division\_winners[order(division\_winners$conference, division\_winners$division, -division\_winners$points, -division\_winners$rand),]  
   
 for(j in 1:nrow(division\_winners)) {  
 index = which(season\_stats$team == division\_winners$team[j])  
 summary$playoffs[index] = summary$playoffs[index] + 1  
 if(j %% 4 == 1) { # Only increment division winners by 1 in division titles  
 summary$division\_titles[index] = summary$division\_titles[index] + 1  
 }  
 }  
  
 #First round was five games, the rest were seven  
 games\_per\_round = c(5, 7, 7, 7)  
   
 playoff\_bracket = data.frame(matrix(-Inf, ncol = 6, nrow = 16))  
 colnames(playoff\_bracket) = c("team", "conference", "division", "rating", "points", "rand")  
 next\_round = NULL  
   
 playoff\_bracket$division[4] = playoff\_bracket$division[3]  
 playoff\_bracket$division[8] = playoff\_bracket$division[7]  
 playoff\_bracket$division[12] = playoff\_bracket$division[11]  
 playoff\_bracket$division[16] = playoff\_bracket$division[15]  
   
 playoff\_bracket = division\_winners  
  
 playoff\_bracket$seed = rep(1:4,4)  
  
 # Divisional rounds  
 for(round in 1:2) {  
 for(j in 1:4) {  
 for(k in 1:(nrow(playoff\_bracket)/8)) {  
 high\_seed\_index = 0.25\*nrow(playoff\_bracket)\*j-(0.25\*nrow(playoff\_bracket)-k)  
 low\_seed\_index = 0.25\*nrow(playoff\_bracket)\*j-(k-1)  
 high\_seed\_elo = playoff\_bracket$rating[high\_seed\_index]  
 low\_seed\_elo = playoff\_bracket$rating[low\_seed\_index]  
 high\_seed\_home\_win\_prob = 1 / (10^((low\_seed\_elo - (high\_seed\_elo + hfa))/400) + 1)  
 low\_seed\_home\_win\_prob = 1 / (10^((high\_seed\_elo - (low\_seed\_elo + hfa))/400) + 1)  
 win\_probs = c(rep(high\_seed\_home\_win\_prob, ceiling(games\_per\_round[round]/2)), 1-rep(low\_seed\_home\_win\_prob, floor(games\_per\_round[round]/2)))  
 u = runif(games\_per\_round[round])  
 high\_seed\_wins = sum(u < win\_probs)/games\_per\_round[round]  
   
 if(high\_seed\_wins > 0.50) {  
 next\_round = rbind(next\_round, playoff\_bracket[high\_seed\_index,])  
 } else{  
 next\_round = rbind(next\_round, playoff\_bracket[low\_seed\_index,])  
 }  
 }  
 }  
   
 playoff\_bracket = next\_round  
 playoff\_bracket = playoff\_bracket[order(playoff\_bracket$division, playoff\_bracket$seed),]  
   
 if(round==1) {  
 for(team in playoff\_bracket$team) {  
 index = which(season\_stats$team == team)  
 summary$div\_semi\_finalist[index] = summary$div\_semi\_finalist[index] + 1  
 }  
 } else {  
 for(team in playoff\_bracket$team) {  
 index = which(season\_stats$team == team)  
 summary$div\_finalist[index] = summary$div\_finalist[index] + 1  
 }  
 }  
   
 #peek into seasons where the pens made the division finals  
 # if(!identical(which(playoff\_bracket$team=="Pittsburgh Penguins"), integer(0))){  
 # print(division\_winners)  
 # }  
   
 next\_round = NULL  
 }  
   
   
 # Conference championships  
 playoff\_bracket = playoff\_bracket[order(playoff\_bracket$conference, playoff\_bracket$seed, -playoff\_bracket$points, -playoff\_bracket$rand),]  
  
 for(j in 1:2) {  
 high\_seed\_index = 2\*j-1  
 low\_seed\_index = 2\*j  
 high\_seed\_elo = playoff\_bracket$rating[high\_seed\_index]  
 low\_seed\_elo = playoff\_bracket$rating[low\_seed\_index]  
 high\_seed\_home\_win\_prob = 1 / (10^((low\_seed\_elo - (high\_seed\_elo + hfa))/400) + 1)  
 low\_seed\_home\_win\_prob = 1 / (10^((high\_seed\_elo - (low\_seed\_elo + hfa))/400) + 1)  
 win\_probs = c(rep(high\_seed\_home\_win\_prob, ceiling(games\_per\_round[length(games\_per\_round)]/2)), 1-rep(low\_seed\_home\_win\_prob, floor(games\_per\_round[length(games\_per\_round)]/2)))  
 u = runif(games\_per\_round[3])  
 high\_seed\_wins = sum(u < win\_probs)/games\_per\_round[3]  
   
 if(high\_seed\_wins > 0.50) {  
 next\_round = rbind(next\_round, playoff\_bracket[high\_seed\_index,])  
 } else{  
 next\_round = rbind(next\_round, playoff\_bracket[low\_seed\_index,])  
 }  
 }  
   
 playoff\_bracket = next\_round  
 playoff\_bracket = playoff\_bracket[order(playoff\_bracket$division, playoff\_bracket$seed),]  
 next\_round = NULL  
   
 # Stanley Cup Finals  
 playoff\_bracket = playoff\_bracket[order(-playoff\_bracket$points, -playoff\_bracket$rand),]  
   
 high\_seed\_elo = playoff\_bracket$rating[1]  
 low\_seed\_elo = playoff\_bracket$rating[2]  
 high\_seed\_home\_win\_prob = 1 / (10^((low\_seed\_elo - (high\_seed\_elo + hfa))/400) + 1)  
 low\_seed\_home\_win\_prob = 1 / (10^((high\_seed\_elo - (low\_seed\_elo + hfa))/400) + 1)  
 win\_probs = c(rep(high\_seed\_home\_win\_prob, ceiling(games\_per\_round[length(games\_per\_round)]/2)), 1-rep(low\_seed\_home\_win\_prob, floor(games\_per\_round[length(games\_per\_round)]/2)))  
 u = runif(games\_per\_round[4])  
 high\_seed\_wins = sum(u < win\_probs)/games\_per\_round[4]  
   
 if(high\_seed\_wins > 0.50) {  
 champion = playoff\_bracket[1,]  
 } else{  
 champion = playoff\_bracket[2,]  
 }  
  
 for(team in playoff\_bracket$team) {  
 index = which(season\_stats$team == team)  
 summary$conference\_championships[index] = summary$conference\_championships[index] + 1  
 }  
   
 index = which(season\_stats$team == champion$team)  
 summary$championships[index] = summary$championships[index] + 1  
 histories[i,] = season\_stats$points  
}  
  
summary$average\_points = summary$average\_points/iterations  
summary$actual\_points = team\_info$points  
summary$resid = summary$actual\_points - summary$average\_points  
  
summary

## team average\_points playoffs division\_titles  
## 1 Boston Bruins 100.6374 9999 6087  
## 2 Buffalo Sabres 93.4936 9982 2195  
## 3 Hartford Whalers 58.5016 183 0  
## 4 Montreal Canadiens 87.2136 9912 773  
## 5 Quebec Nordiques 88.5231 9924 945  
## 6 Chicago Black Hawks 82.4713 9829 3385  
## 7 Detroit Red Wings 64.1504 5253 88  
## 8 Minnesota North Stars 86.7515 9944 5895  
## 9 St. Louis Blues 70.5315 8137 451  
## 10 Toronto Maple Leafs 67.1020 6837 181  
## 11 New Jersey Devils 50.9098 19 0  
## 12 New York Islanders 105.3911 10000 7306  
## 13 New York Rangers 87.9772 9965 435  
## 14 Philadelphia Flyers 95.5199 9997 1817  
## 15 Pittsburgh Penguins 53.1672 48 0  
## 16 Washington Capitals 87.8283 9971 442  
## 17 Calgary Flames 78.0247 9228 45  
## 18 Edmonton Oilers 109.4570 10000 9909  
## 19 Los Angeles Kings 65.1296 4273 5  
## 20 Vancouver Canucks 75.5270 8804 31  
## 21 Winnipeg Jets 71.6922 7695 10  
## div\_semi\_winner div\_final\_champ conference\_championships championships  
## 1 0 0 1921 946  
## 2 0 0 1362 581  
## 3 0 0 1 1  
## 4 0 0 218 56  
## 5 0 0 944 376  
## 6 0 0 430 62  
## 7 0 0 77 14  
## 8 0 0 1144 303  
## 9 0 0 194 26  
## 10 0 0 35 0  
## 11 0 0 0 0  
## 12 0 0 2886 1546  
## 13 0 0 231 74  
## 14 0 0 988 407  
## 15 0 0 0 0  
## 16 0 0 1449 681  
## 17 0 0 560 134  
## 18 0 0 7176 4739  
## 19 0 0 26 1  
## 20 0 0 236 43  
## 21 0 0 122 10  
## actual\_points resid  
## 1 104 3.3626  
## 2 103 9.5064  
## 3 66 7.4984  
## 4 75 -12.2136  
## 5 94 5.4769  
## 6 68 -14.4713  
## 7 69 4.8496  
## 8 88 1.2485  
## 9 71 0.4685  
## 10 61 -6.1020  
## 11 41 -9.9098  
## 12 104 -1.3911  
## 13 93 5.0228  
## 14 98 2.4801  
## 15 38 -15.1672  
## 16 101 13.1717  
## 17 82 3.9753  
## 18 119 9.5430  
## 19 59 -6.1296  
## 20 73 -2.5270  
## 21 73 1.3078