UC Davis

Option 2: Fouling Out

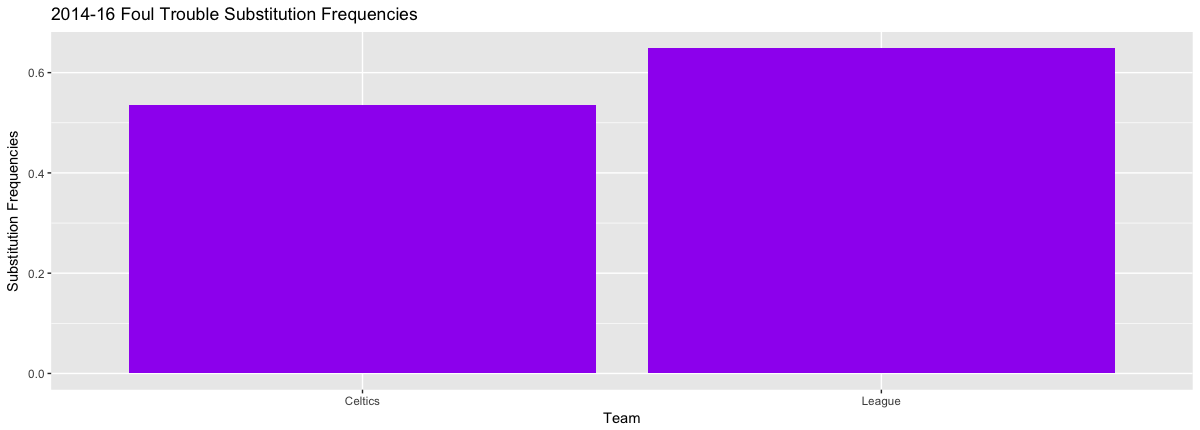
By: Andrew Willmore, Colin DeCure, Nimal Subramanian, Yoav Kaliblotzky

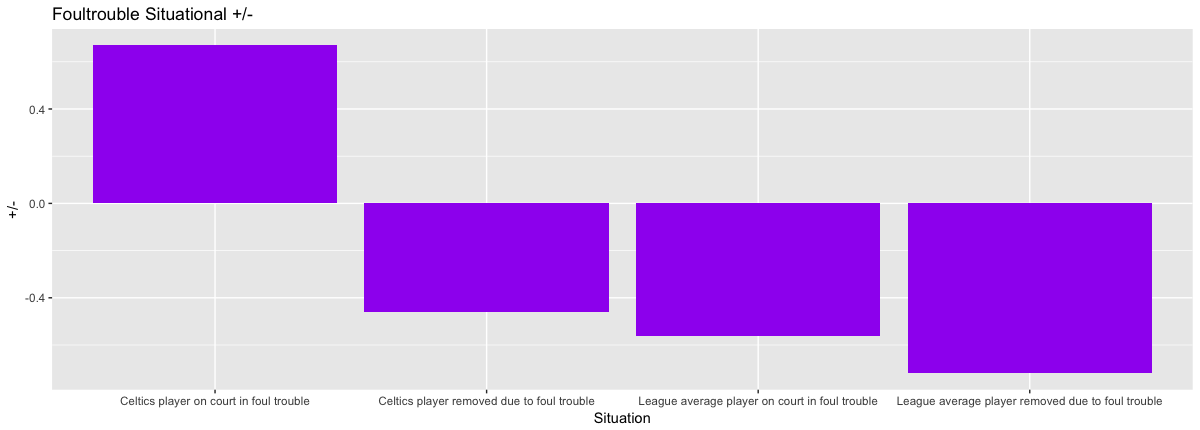
The problem of handling players in foul trouble is one that parallels very closely with the issue of bullpen management in baseball. In baseball, many teams will save their best reliever for the ninth inning. However this mindset is beginning to be challenged by a few of the league’s better teams (ex. the Indians’ usage of Andrew Miller as a relief ace). There are a few key differences when comparing the dilemmas in that 1) relievers need rest and can’t pitch in every game, and as a result can’t appear in every high leverage situation over the course of a season; 2) closers make much more money than non-closers on average, and would likely be adverse to the idea of transitioning into a different role; 3) once a reliever is removed from a game, they can’t be brought back in. That being said, this decision is very similar to that of handling foul trouble in that it challenges the notion that the end of the game is the most important aspect and it is therefore important to have a team’s best players on the court/field for the last quarter or inning. However, what if the game is actually won or lost in a high leverage situation earlier in the game? What if the best player is saved for later in the game, at which point the game is already out of hand? It is possible that situations like these could be avoided by keeping a team’s best players on the floor for as long as possible. With this comparison in mind, we attempted to discern whether, and in what situations to remove players as they begin to accumulate fouls. In order to do so, we looked at 1) how teams perform with a player in foul trouble on the court in comparison to team performance with a player in foul trouble off the court and 2) how win probabilities change throughout a game.

The first aspect we attempted to gauge was the rate at which players in foul trouble end up fouling out. We defined foul trouble as the current quarter number plus one. For example, three fouls in the second quarter would constitute a player being in foul trouble. Using the play by play data from 2014-2016, we found 16,939 cases of players in foul trouble and 801 instances of players fouling out, resulting in a foul out rate of 4.73%. We then compared this league average foul-out rate to that of the Brad Stevens-led Celtics from 2014-2016. The Celtics totaled 714 cases of foul trouble in that timespan with 21 foul-outs, yielding a foul-out rate of 2.94%. We would have expected the Celtics to have a higher foulout rate based upon Stevens’ philosophy when it comes to handling players in foul trouble, but this was not actually the case. The Celtics had a lower foul-out rate than the league average by over a percentage point, despite having a slower trigger when removing players in foul trouble. This observation could be partially explained by the Celtics’ success under Stevens (.573 win percentage since 2014) and could be attributed to simply playing more disciplined basketball. Even so, it remains an interesting result that actively removing players in foul trouble may not actually help in preventing players from fouling out.

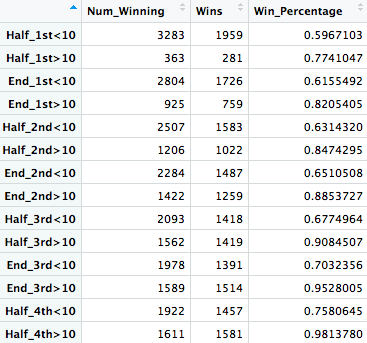
Our next step was to look at quality of play broken down into foul trouble situations. We first formed two categories for players in foul trouble: those who were subbed out immediately upon picking up their last foul, and those who weren’t. We then compared on-court plus-minus stats for the two situations to understand how team performance changes for each scenario. For this step in our analysis, we used the play by play data from the 2014-15 and 2015-16 seasons. The league wide substitution rate for players in foul trouble as defined by our previously stated definition was 64.89%, meaning approximately 65% of players who picked up a foul putting them in foul trouble were subsequently removed from the game. The average on-court plus-minus for situations in which players were removed due to foul trouble came out to be -.72. In comparison, the average plus-minus for situations in which players in foul trouble remained in the game was -.56. These values were both significantly different from the null hypothesis of a +/- of 0 at 95% confidence. The values also were not significantly different from each other, meaning league-wide there was little difference in team plus-minus between leaving in a player in foul trouble and taking him out. However, these results differed greatly with those obtained when considering only Celtics players in foul trouble.

As was expected, the Celtics substituted players in foul trouble at a much lower rate than the rest of the league: 53.56% compared to the league average of 64.98%. Additionally, the plus-minus for the team when a player in foul trouble was taken out was -.46, which did not significantly differ from that of the league average. However, the plus-minus for the team when a player in foul trouble remained in the game was .67, which did differ significantly from the league average at 95% confidence. This result showed that not only did the Celtics perform significantly better than the league average when leaving a player in who was in foul trouble, but they actually had a positive plus-minus when doing so. Of course the Celtics had an above .500 winning percentage, and as a result their baseline plus-minus was above 0. That said, it is still a significant result that their plus-minus was of that magnitude higher than their plus-minus when taking out a foul trouble player. This information can be summarized in the graphs below.





Our final step was to quantify how win probability changes throughout a game. In order to do so, we broke each quarter into halves for a total of 14 game segments. We then used game results from the 2014, 2015, and 2016 seasons to estimate win probabilities for each game segment depending on whether the winning team was up by less than 10 points, or more than 10 points but less than 25 points. The results can be visualized in the chart below.



One observation that immediately stands out is that win percentage and time appear to have a linear relationship. Win percentage appears to increase by about 3% per half quarter for both point differentials. Also obvious is the fact that falling behind by more than 10 in any segment of the game dramatically reduces a team’s ability to come back. According to our projected win percentages, a team winning by more than 10 at the end of the first quarter actually has a higher win probability than a team winning by less than 10 half way through the fourth quarter. These results imply that remaining within 10 points at any moment in the game is crucial to a team’s chances of winning. It also shows there is no significant uptick in win probability for a lead in the fourth quarter, based upon the linear relationship between win probability and game segment. This result would appear to diverge from the notion that fourth quarter minutes are more important and would therefore require saving players for later in the game.

The results of our analysis appear to back the claims and methods of Brad Stevens when handling players in foul trouble. We believe that coaches should abandon the traditional definition of foul trouble (current quarter plus one, as defined previously) and give players a longer leash when considering removing them in foul trouble. According to our win probability chart, the game is often lost before the fourth quarter when one team establishes a lead that is simply too much for the other to overcome. It may sound simplistic, but based upon our results, coaches should aim to stay competitive for as long as possible if losing and attempt to pull away early if winning, even if it means pushing players a foul or two further into foul trouble. Interestingly enough, our results also seem to show that even when coaches have a longer leash for players in foul trouble, the players are not assured to foul out with a higher frequency. If the Celtics are any indication, players not concerned with picking up an extra foul will actually provide better minutes when left on the floor.

When considering our results with the baseball closer analogy from the beginning of this report, they make sense. Teams will often deploy their best relievers as closers because games that are close at the end of the game will often happen to coincide with a closer’s availability. Almost every game will have a high leverage moment in which the result hangs in the balance, but given that relievers cannot throw in every game or even half of a team’s games, deploying top relievers as closers guarantees having a team’s best reliever available in something of a high leverage situation. Were rest not an issue, baseball teams would likely deploy their best relievers earlier in games in more crucial situations. In contrast, basketball players are afforded the luxury of not needing such rest between games. If a game needs saving or even putting away in the third quarter, a coach should not remove one of his better players simply based upon the idea that fourth quarter minutes are more important.

Code Appendix

win\_pr <- function(data, matrix, period, score1, score2){

#Visitor < 10

for(i in 1:nrow(data)){

if(data[i,"period"] == period){

if(data[i, "visitor\_score"] > data[i, "home\_score"]) {

if(data[i,"visitor\_score"] - data[i, "home\_score"] < 10){

matrix[score1,"Num\_Winning"] = matrix[score1,"Num\_Winning"] + 1

for(j in 1:3936){

if(data[i,"game\_id"] == schedule[j,"game\_id"]){

if(schedule[j,"visitor\_team\_points"] > schedule[j, "home\_team\_points"]){

matrix[score1,"Wins"] = matrix[score1,"Wins"] + 1

break

}

for(i in 1:nrow(data)){

if(data[i,"period"] == period){

if(data[i, "home\_score"] > data[i, "visitor\_score"]) {

if(data[i,"home\_score"] - data[i, "visitor\_score"] < 10){

matrix[score1,"Num\_Winning"] = matrix[score1,"Num\_Winning"] + 1

for(j in 1:3936){

if(data[i,"game\_id"] == schedule[j,"game\_id"]){

if(schedule[j,"home\_team\_points"] > schedule[j, "visitor\_team\_points"]){

matrix[score1,"Wins"] = matrix[score1,"Wins"] + 1

break

}

for(i in 1:nrow(data)){

if(data[i,"period"] == period){

if(data[i, "visitor\_score"] > data[i, "home\_score"]) {

if(data[i,"visitor\_score"] - data[i, "home\_score"] >= 10 & data[i,"visitor\_score"] - data[i, "home\_score"] <=25){

matrix[score2,"Num\_Winning"] = matrix[score2,"Num\_Winning"] + 1

for(j in 1:3936){

if(data[i,"game\_id"] == schedule[j,"game\_id"]){

if(schedule[j,"visitor\_team\_points"] > schedule[j, "home\_team\_points"]){

matrix[score2,"Wins"] = matrix[score2,"Wins"] + 1

break

}

for(i in 1:nrow(data)){

if(data[i,"period"] == period){

if(data[i, "home\_score"] > data[i, "visitor\_score"]) {

if(data[i,"home\_score"] - data[i, "visitor\_score"] >= 10 & data[i,"home\_score"] - data[i, "visitor\_score"] <= 25){

matrix[score2,"Num\_Winning"] = matrix[score2,"Num\_Winning"] + 1

for(j in 1:3936){

if(data[i,"game\_id"] == schedule[j,"game\_id"]){

if(schedule[j,"home\_team\_points"] > schedule[j, "visitor\_team\_points"]){

matrix[score2,"Wins"] = matrix[score2,"Wins"] + 1

break

}

return(matrix)

}

win\_prob <- win\_pr(half2016,win\_prob, 1, "Half\_1st<10", "Half\_1st>10")

for(i in 1:nrow(win\_prob)){

win\_prob[i,"Win\_Percentage"] <- win\_prob[i,"Wins"]/win\_prob[i,"Num\_Winning"]

}

ggplot(subfreq, aes(x = Team, y = `Substitution Freq`)) + geom\_col(fill = "purple") + ggtitle("2014-16 Foul Trouble Substitution Frequencies") + ylab("Substitution Frequencies")

ggplot(plusminus, aes(x = Situation, y = `Team Plus/Minus`)) + geom\_col(fill = "purple") + ylab("+/-") + ggtitle("Foultrouble Situational +/-")

foultroub14 <- subset.data.frame(foul\_stats14, foul\_trouble == "True")

foulout14 <- subset.data.frame(foul\_stats14, foul\_num == 6)

foul\_oncourt14 <- subset.data.frame(foultroub14, taken\_out == "False")

foul\_offcourt14 <- subset.data.frame(foultroub14, taken\_out == "True")

foul\_oncourt14.2 <- foul\_oncourt14 %>% as\_tibble %>% mutate(plus\_minus = on\_court\_team\_points\_scored - on\_court\_team\_points\_allowed)

foul\_offcourt14.2 <- foul\_offcourt14 %>% as\_tibble %>% mutate(plus\_minus = off\_court\_team\_points\_scored - off\_court\_team\_points\_allowed)

foulout14.2 <- foulout14 %>% as\_tibble %>% mutate(plus\_minus = off\_court\_team\_points\_scored - off\_court\_team\_points\_allowed)

mean(foul\_offcourt14.2$plus\_minus)

mean(foul\_oncourt14.2$plus\_minus)

mean(foulout14.2$plus\_minus)

def getTime(clock\_time):

minute = int(clock\_time[:2])

second = float(clock\_time[3:])

return (12.0 - minute - (second / 60))

no\_foul = [25,30,12,19,18,13,16,11,26,17]

for year in ['14-15','15-16']:

c.execute('''SELECT game\_id,description,player\_id,player\_id2,action\_type,

stat\_value, stat\_value2,stat\_value3,message\_type,

event\_number, game\_clock, period, offensive\_team\_id,

team\_id,visitor\_score,home\_score,

wall\_clock\_as\_int FROM `PBP\_{}`

ORDER BY game\_id ASC, event\_number ASC'''.format(year))

current\_row = c.fetchone()

current\_game = None

home\_team = 0

visiting\_team = 0

player\_dict = {}

current\_period = 0

current\_lineup = [0,0,0,0,0,0,0,0,0,0]

played\_in\_game = []

scores = {}

keeping\_track = 0

while current\_row:

# Game has changed, reset players that played in the game and figure out teams

if not current\_game or current\_game != current\_row[0]:

# if current\_game:

# break

current\_game = current\_row[0]

schedule\_cursor.execute('SELECT game\_id, visitor\_id, home\_id FROM SCHEDULE where game\_id = {}'.format(current\_game))

s = schedule\_cursor.fetchone()

visiting\_team = s[1]

home\_team = s[2]

scores = {}

visit\_or\_home = {visiting\_team: 14, home\_team: 15}

scores[visiting\_team] = 0

scores[home\_team] = 0

played\_in\_game = []

current\_period = 0

current\_lineup = [0,0,0,0,0,0,0,0,0,0]

if current\_period != current\_row[11]:

lineup\_change = True

current\_period = current\_row[11]

lineup\_cursor.execute('SELECT \* FROM `LINEUPS\_{}` WHERE game\_id = {} and period = {}'.format(

year, current\_game, current\_period))

lineup = lineup\_cursor.fetchone()

if not lineup:

print('Error: Game does not exist, where game\_id = {}, period = {}'.format(

current\_game, current\_period))

continue

for i in range(10):

if current\_period > 1:

player\_dict[(current\_game,current\_lineup[i])].subbedOut((current\_period - 1) \* 12)

current\_lineup[i] = lineup[i+4]

if not (current\_game, current\_lineup[i]) in player\_dict:

if (i < 5):

player\_dict[(current\_game, current\_lineup[i])] = data\_classes.PlayerGame(

gid=current\_game,tid=visiting\_team,

pid=current\_lineup[i],

start\_time=((current\_period - 1) \* 12))

player\_dict[(current\_game, current\_lineup[i])].offTeamScored(scores[visiting\_team])

player\_dict[(current\_game, current\_lineup[i])].offOppScored(scores[home\_team])

else:

player\_dict[(current\_game, current\_lineup[i])] = data\_classes.PlayerGame(

gid=current\_game,tid=home\_team,

pid=current\_lineup[i],

start\_time=((current\_period - 1) \* 12))

player\_dict[(current\_game, current\_lineup[i])].offOppScored(scores[visiting\_team])

player\_dict[(current\_game, current\_lineup[i])].offTeamScored(scores[home\_team])

else:

player\_dict[(current\_game,current\_lineup[i])].subbedIn((

current\_period - 1) \* 12)

if not current\_lineup[i] in played\_in\_game:

played\_in\_game.append(current\_lineup[i])

#print('New lineup,{}'.format(current\_lineup))