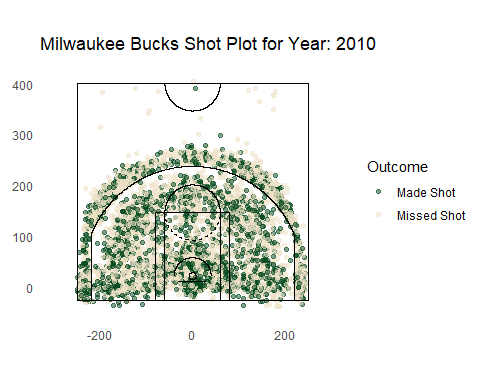
Stat 345 Midterm

Gracie Funk

2025-03-30

### Overview:

* The shot charts I have created for my project show the 2010 to 2020 shot locations for the Milwaukee Bucks
* Each dot on the animation represents a shot taken by a player
  + Each green dot is a made shot
  + Each beige dot is a missed shot
* I have also added a court to the animation to help understand exactly where each shot was taken
  + The dimensions of the court has been scaled to align to the dimensions of each shot taken



### 

### Key Findings:

* Overall, less shots are being taken over time
* More shots are being made right next to the basket
* Three point shots are more concentrated at the top of the arc rather than the sides
* Less shots are being taken inside the arc and are moving outside the arc

### More Details:

* Looking at how the total amount of shots being taken over each season, we can see that the number has decrease for both inside and outside of the three point line
  + This is probably due to the Milwaukee Bucks focusing more on accuracy than total number of shots taken
* Looking near the hoop, we can see more shots being taken right near it as well as more being made which can be seen by the growing number of green dots over the years
* Looking at specifically 3-point shots, we can see that the Milwaukee Bucks over time have started to focus these shots near the top of the 3-point arc, rather than the sides -We also see more attempts at these types of shots over time
* Looking at the overall location of the shots, we see a movement from the inside of the arc towards the outside of the arc, showing that over time the Milwaukee Bucks has taken focus away from 2-point shots and putting more effort into taking 3-point shots

### My code:

# Code for the Milwaukee Buck's shot data for the year 2010 - 2020  
# You will need the following packages:  
library(nbastatR) #NBA Data  
library(ggplot2) #Graphing package  
library(gganimate) #Animation for plot  
library(devtools) #Devtools  
devtools::install\_github("abresler/nbastatR", force = TRUE)  
  
# Code to increase the connection buffer size:  
Sys.setenv(VROOM\_CONNECTION\_SIZE = 131072 \* 2)  
  
# Code that generates a basketball court (credits to github user edkupfer (https://gist.github.com/edkupfer/6354404#file-ggplot-nba-halfcourt)):  
court <- ggplot(data = data.frame(x = 1, y = 1), aes(x, y)) +  
 #court box:  
 geom\_path(data = data.frame(x = c(-25, -25, 25, 25, -25), y = c(0, 47, 47, 0, 0))) +  
 #solid half of the free throw semicircle:  
 geom\_path(data = data.frame(x = c(-6000:(-1)/1000, 1:6000/1000), y = c(19 + sqrt(6^2 - c(-6000:(-1)/1000, 1:6000/1000)^2))), aes(x = x, y = y)) +  
 #dashed half of the free throw semicircle:  
 geom\_path(data = data.frame(x = c(-6000:(-1)/1000, 1:6000/1000), y = c(19 - sqrt(6^2 - c(-6000:(-1)/1000, 1:6000/1000)^2))), aes(x = x, y = y), linetype = 'dashed') +  
 #the key:  
 geom\_path(data = data.frame(x = c(-8, -8, 8, 8, -8), y = c(0, 19, 19, 0, 0))) +  
 #the box inside of the key:  
 geom\_path(data = data.frame(x = c(-6, -6, 6, 6, -6), y = c(0, 19, 19, 0, 0))) +  
 #the restricted area semicircle:  
 geom\_path(data = data.frame(x = c(-4000:(-1)/1000, 1:4000/1000), y = c(5.25 + sqrt(4^2 - c(-4000:(-1)/1000, 1:4000/1000)^2))), aes(x = x, y = y)) +  
 #half court semicircle:  
 geom\_path(data = data.frame(x = c(-6000:(-1)/1000, 1:6000/1000), y = c(47 - sqrt(6^2 - c(-6000:(-1)/1000, 1:6000/1000)^2))), aes(x = x, y = y)) +   
 #rim:  
 geom\_path(data = data.frame(x = c(-750:(-1)/1000, 1:750/1000, 750:1/1000, -1:-750/1000), y = c(c(5.25 + sqrt(0.75^2 - c(-750:(-1)/1000, 1:750/1000)^2)), c(5.25 - sqrt(0.75^2 - c(750:1/1000, -1:-750/1000)^2)))), aes(x = x, y = y)) +  
 #backboard:  
 geom\_path(data = data.frame(x = c(-3, 3), y = c(4, 4)), lineend = 'butt') +  
 #three-point line   
 geom\_path(data = data.frame(x = c(-22, -22, -22000:(-1)/1000, 1:22000/1000, 22, 22), y = c(0, 169/12, 5.25 + sqrt(23.75^2 - c(-22000:(-1)/1000, 1:22000/1000)^2), 169/12, 0)), aes(x = x, y = y)) +  
 #making a fixed ratio 1:1  
 coord\_fixed() +  
 #making it a blank background  
 theme\_void()  
  
# Code that creates a function which takes in a certain team and seasons and outputs shot data   
shot\_data <- function(team, season) {  
 teams\_shots(teams = team, seasons = season, season\_types = "Regular Season", measures = "FGA")   
}  
  
# Code for my specific focus on the Milwaukee Buck's 2010 - 2020 shot data   
years <- c(2010:2020)  
bucks\_shot\_data <- lapply(years, function(year) {  
 shot\_data("Milwaukee Bucks", year) })  
bucks\_combined\_shot\_data <- do.call(rbind, bucks\_shot\_data) #This line combines all shot data from shot\_data\_list into one data frame to make plotting easier  
  
# Code for creating the plots and then combining them into one animation  
plots <- ggplot(bucks\_combined\_shot\_data, aes(x = locationX, y = locationY, color = typeEvent)) + #generates plot with out data  
 geom\_point(alpha = 0.5) + #adds dots with slight transparency for better reading  
 labs(title = "Milwaukee Bucks Shot Plot for Year: {round(frame\_time,0)}", x = " ", y = " ") + #adds titles  
 theme\_minimal() + theme(panel.grid = element\_blank()) + #minimal theme and getting rid of the grid   
 xlim(-300, 300) + # x-axis limits  
 ylim(-47, 423) + # y-axis limits  
 annotation\_custom(ggplotGrob(court), xmin = -300, xmax = 300, ymin = -47, ymax = 423) + #adding the court to the plots  
 coord\_fixed(ratio = 1.1) + #setting the proportions of the plot to ensure they line up with our data  
 scale\_color\_manual(name = "Outcome", values = c("Made Shot" = "#00471B", "Missed Shot" = "#EEE1C6")) + #adding a key and team colors to missed/made shots  
 transition\_time(bucks\_combined\_shot\_data$yearSeason) #adding the transition to have a plot and frame for each season  
  
animate(plots, fps = 1, nframes = 10) #creating the animation with 1 second per plot with 10 years of data (2010 - 2020)

### 4-Point Line

I decided to explore where the NBA should draw a 4-point line. I did this by creating a function which allows you to input a distance for the line and see how the expected value of at the proposed distance, and how certain Milwaukee Bucks players would perform with your proposed line distance. Using my function you can look at how a player’s point total for the season would change. Therefore, with my function you can evaluate if you think the line distance is fair or not based on the expected value and changes in player’s point totals.

library(dplyr)  
  
#redefining my bucks 2010-2020 shot data  
shot\_data <- do.call(rbind, bucks\_shot\_data)  
  
#this function takes in your input of shot data from a team and the 4-point line distance you want to test and outputs the expected value   
calculate\_expected\_value <- function(shot\_data, line\_distance) {  
 shot\_data <- shot\_data[complete.cases(shot\_data[, c("locationX", "locationY")]), ] #only letting there be cases with both coordinates  
 shot\_data$shot\_distance <- sqrt((shot\_data$locationX / 10)^2 + (shot\_data$locationY / 10)^2) #adding a column to shot\_data called shot\_distance which uses the PT to calculate shot distance for each shot attempt  
  
 #adding a column to shot\_data called point\_value which has the value of each shot (2 for 2-pts, 3 for 3-pts, 4 for our 4-pt line shot)  
 shot\_data$point\_value <- ifelse(   
 shot\_data$shot\_distance <= line\_distance, 2,  
 ifelse(shot\_data$shot\_distance > line\_distance & shot\_data$shot\_distance <= (line\_distance + 4), 4, 3)  
 )  
  
 #calculating the shooting percentage for each point value   
 shooting\_percentage <- aggregate(isShotMade ~ point\_value, data = shot\_data, FUN = function(x) sum(x) / length(x))  
 expected\_value <- sum(shooting\_percentage$isShotMade \* shooting\_percentage$point\_value) #expected value calculation  
 return(expected\_value)  
}  
  
#this function calculates the total amount of points made in a season  
calculate\_season\_points <- function(shot\_data) {  
 made\_shots <- shot\_data[shot\_data$isShotMade == TRUE, ]  
 season\_points <- sum(case\_when(  
 made\_shots$typeShot == "2PT Field Goal" ~ 2,  
 made\_shots$typeShot == "3PT Field Goal" ~ 3,  
 made\_shots$typeShot == "4PT Field Goal" ~ 4,  
 TRUE ~ 0  
 ))  
   
 return(season\_points)  
}  
  
#putting in my own line distance  
line\_distance <- 30  
  
#printing the expected value from my function with this distance  
expected\_value <- calculate\_expected\_value(shot\_data, line\_distance)  
print(paste("Expected value with proposed 4-point line at", line\_distance, "feet:", expected\_value))

## [1] "Expected value with proposed 4-point line at 30 feet: 1.88153417298266"

#testing on a Milwaukee Bucks player  
player\_name <- "Brook Lopez"  
player\_shot\_data <- shot\_data[shot\_data$namePlayer == player\_name, ]  
season\_points\_without\_4pt <- calculate\_season\_points(player\_shot\_data)  
player\_shot\_data\_with\_4pt <- player\_shot\_data  
player\_shot\_data\_with\_4pt$shot\_distance <- sqrt((player\_shot\_data\_with\_4pt$locationX / 10)^2 + (player\_shot\_data\_with\_4pt$locationY / 10)^2)  
player\_shot\_data\_with\_4pt$typeShot <- ifelse(player\_shot\_data\_with\_4pt$shot\_distance > line\_distance, "4PT Field Goal", player\_shot\_data\_with\_4pt$typeShot)  
season\_points\_with\_4pt <- calculate\_season\_points(player\_shot\_data\_with\_4pt)  
percentage\_change <- ifelse(season\_points\_without\_4pt == 0, NA,   
 ((season\_points\_with\_4pt - season\_points\_without\_4pt) / season\_points\_without\_4pt) \* 100)  
  
  
#print outs for each outcome  
if (is.na(percentage\_change)) {  
 outcome <- paste(player\_name,"'s season point total cannot be determined due to zero baseline points.")  
} else if (percentage\_change > 0) {  
 outcome <- paste(player\_name,"'s season point total would increase by", round(percentage\_change, 2), "%")  
} else if (percentage\_change < 0) {  
 outcome <- paste(player\_name,"'s season point total would decrease by", abs(round(percentage\_change, 2)), "%")  
} else {  
 outcome <- paste(player\_name,"'s season point total would remain unchanged")  
}  
  
print(outcome)

## [1] "Brook Lopez 's season point total would increase by 0.44 %"

Looking at this output from my function, I can see that a 4-point line at 30 feet we have an expected value of 1.88. This means that a player is expect to score 1.9 points per shot at this line distance. The reason this number is below 4 is because expected value takes into account the actual probability of making a shot from 30 feet. Using my example of Brook Lopez from the Milwaukee Bucks, we can expect that his season point total would increase by .44% with this 30 foot 4-point line.