In [this previous post](https://statisticaloddsandends.wordpress.com/2019/04/23/probability-of-winning-a-best-of-7-series/), I explored the probability that a team wins a best-of-n series, given that its win probability for any one game is some constant p. As one commenter pointed out, most sports models consider the home team to have an advantage, and this home advantage should affect the probability of winning a series. In this post, I will explore this question, limiting myself (most of the time) to the case of n = 7.

To be as general as possible, instead of giving a team just one parameter p, I will give a team two parameters: p_Hfor the probability of winning at home, and p_Afor winning away. In general we will have p_H > p_A, although that is not always the case. (For example, in the NBA 2018/19 regular season, all but 2 teams had more home wins than road wins, the 2 teams being the Miami Heat and the Chicago Bulls.) In my simulations, I will allow p_Hand p_Ato take on any values between 0 and 1.

The random variable denoting the number of wins in a best-of-7 series will depend on whether the team starts at home or not. (We will assume our team will alternate being playing at home and playing away.) If starting at home, the number of wins has distribution \text{Binom}(4, p_H) + \text{Binom}(3, p_A); if starting away, the distribution would be \text{Binom}(3, p_H) + \text{Binom}(4, p_A).

We can modify the sim\_fn function from the previous post easily so that it now takes in more parameters (home\_p, away\_p, and start indicating whether the first game is at “home” or “away”):

sim\_fn <- function(simN, n, home\_p, away\_p, start) {

if (n %% 2 == 0) {

stop("n should be an odd number")

}

if (start == "home") {

more\_p <- home\_p

less\_p <- away\_p

} else {

more\_p <- away\_p

less\_p <- home\_p } mean(rbinom(simN, (n+1)/2, more\_p) + rbinom(simN, (n-1)/2, less\_p) > n / 2)

}

The following code sets up a grid of home\_p,away\_p and start values, then runs 50,000 simulation runs for each combination of those values:

home\_p <- seq(0, 1, length.out = 51)

away\_p <- seq(0, 1, length.out = 51)

df <- expand.grid(home\_p = home\_p, away\_p = away\_p, start = c("home", "away"),

stringsAsFactors = FALSE)

set.seed(111)

simN <- 50000

n <- 7

df$win\_prob <- apply(df, 1, function(x) sim\_fn(simN, n, as.numeric(x[1]),

as.numeric(x[2]), x[3]))

How can we plot our results? For each combination of (home\_p, away\_p, start) values, we want to plot the series win probability. A decent 2D representation would be a heatmap:

library(tidyverse)

ggplot(df, aes(x = home\_p, y = away\_p)) +

geom\_raster(aes(fill = win\_prob)) +

labs(title = paste("Win probability for best of", n, "series"),

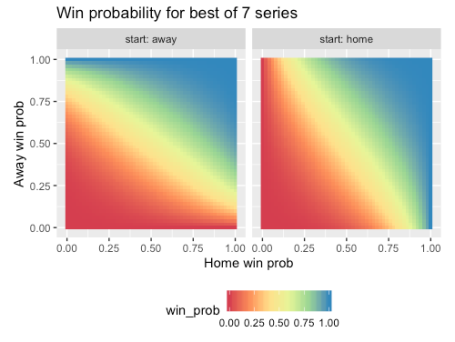
x = "Home win prob", y = "Away win prob") +

scale\_fill\_distiller(palette = "Spectral", direction = 1) +

facet\_wrap(~ start, labeller = label\_both) +

coord\_fixed() +

theme(legend.position = "bottom")



While the colors give us a general sense of the trends, we could overlay the heatmap with contours to make comparisons between different points easier. Here is the code, the geom\_text\_contour function from the metR package is for labeling the contours.

library(metR)

ggplot(df, aes(x = home\_p, y = away\_p, z = win\_prob)) +

geom\_raster(aes(fill = win\_prob)) +

geom\_contour(col = "white", breaks = 1:9 / 10) +

geom\_text\_contour(col = "#595959") +

labs(title = paste("Win probability for best of", n, "series"),

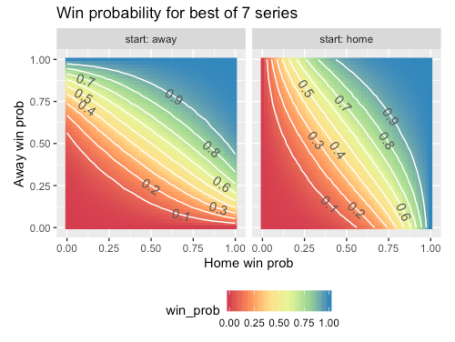
x = "Home win prob", y = "Away win prob") +

scale\_fill\_distiller(palette = "Spectral", direction = 1) +

facet\_wrap(~ start, labeller = label\_both) +

coord\_fixed() +

theme(legend.position = "bottom")



For a full 3D surface plot, I like to use the plotly package. This is because it is hard to get information from a static 3D plot: I want to be able to turn the plot around the axes to see it from different angles. plotly allows you to do that, and it also gives you information about the data point that you are hovering over. The code below gives us a 3D surface plot for the start = "home" data. (Unfortunately WordPress.com does not allow me to embed the chart directly in the post, so I am just showing a screenshot to give you a sense of what is possible. Run the code in your own R environment and play with the graph!)

library(plotly)

df\_home <- df %>% filter(start == "home")

win\_prob\_matrix <- matrix(df\_home$win\_prob, nrow = length(away\_p), byrow = TRUE) plot\_ly(x = home\_p, y = away\_p, z = win\_prob\_matrix) %>%

add\_surface(contours = list(

z = list(

show = TRUE,

usecolormap = TRUE,

highlightcolor = "#ff0000",

project = list(z = TRUE)

)

)

) %>%

layout(

title = paste("Win probability for best of", n, "series (start at home)"),

scene = list(

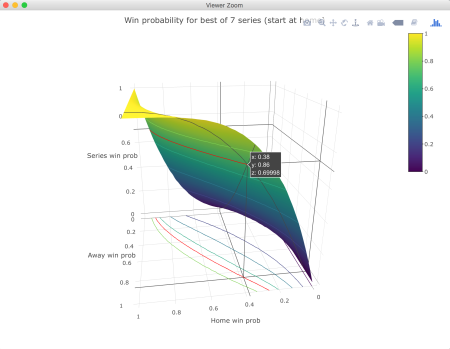
xaxis = list(title = "Home win prob"),

yaxis = list(title = "Away win prob"),

zaxis = list(title = "Series win prob")

)

)



***How much better is it to start at home than away?***

To answer this question, we will subtract the probability of winning a series when starting away from that when starting at home. The more positive the result, the more “important” starting at home is.

# difference in starting at home vs. away

df2 <- df %>% spread(start, value = win\_prob) %>%

mutate(home\_adv = home - away)

# raster plot

ggplot(df2, aes(x = home\_p, y = away\_p)) +

geom\_raster(aes(fill = home\_adv)) +

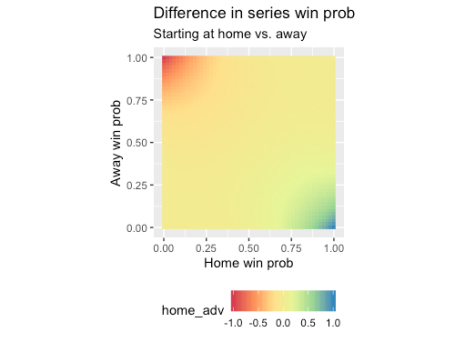
labs(title = "Difference in series win prob (starting at home vs. away)",

x = "Home win prob", y = "Away win prob") +

scale\_fill\_distiller(palette = "Spectral", direction = 1) +

coord\_fixed() +

theme(legend.position = "bottom")



It looks like most of the variation is in the corners. Drawing contours at manually-defined levels helps to give the center portion of the plot more definition (note the “wigglyness” of the contours: this is an artifact of Monte Carlo simulation):

breaks <- c(-0.5, -0.2, -0.1, -0.05, -0.02, 0, 0.02, 0.05, 0.1, 0.2, 0.5)

ggplot(df2, aes(x = home\_p, y = away\_p, z = home\_adv)) +

geom\_raster(aes(fill = home\_adv)) +

geom\_contour(col = "white", breaks = breaks) +

geom\_text\_contour(breaks = breaks,

col = "#595959", check\_overlap = TRUE) +

labs(title = "Difference in series win prob",

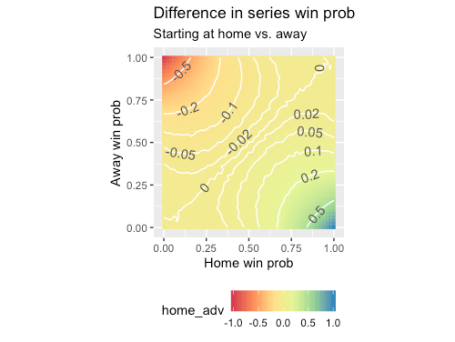
subtitle = "Starting at home vs. away",

x = "Home win prob", y = "Away win prob") +

scale\_fill\_distiller(palette = "Spectral", direction = 1) +

coord\_fixed() +

theme(legend.position = "bottom")



***When does being at home tip you over (or back under) the 50% threshold?***

That is, when does starting at home make you more likely to win the series than your opponent, but starting away makes your opponent the favorite? This can be answered easily by performing some data manipulation. In the plot, “Better” means that you are the favorite at home but the underdog away, while “Worse” means the opposite.

threshold <- 0.5

df2$home\_impt <- ifelse(df2$home > threshold & df2$away < threshold, 1,

ifelse(df2$home < threshold & df2$away > threshold, -1, NA))

df2$home\_impt <- factor(df2$home\_impt)

df2$home\_impt <- fct\_recode(df2$home\_impt,

"Better" = "1", "Worse" = "-1")

df2 %>% filter(!is.na(home\_impt)) %>%

ggplot(aes(x = home\_p, y = away\_p)) +

geom\_raster(aes(fill = home\_impt)) +

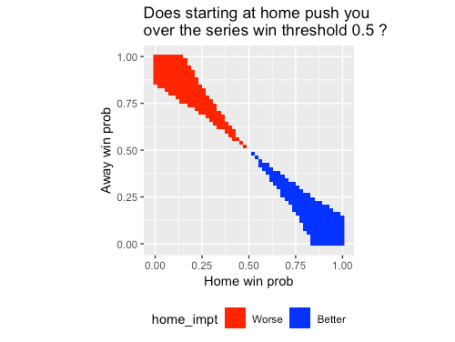
labs(title = paste("Does starting at home push you \nover the series win threshold", threshold, "?"),

x = "Home win prob", y = "Away win prob") +

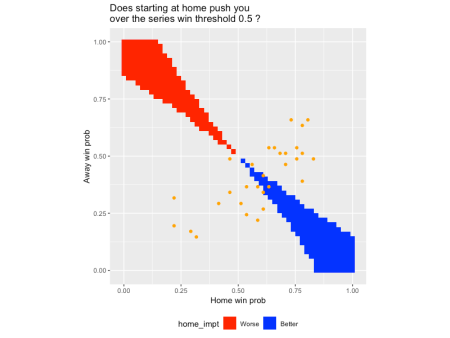
scale\_fill\_manual(values = c("#ff0000", "#0000ff")) +

coord\_fixed() +

theme(legend.position = "bottom")

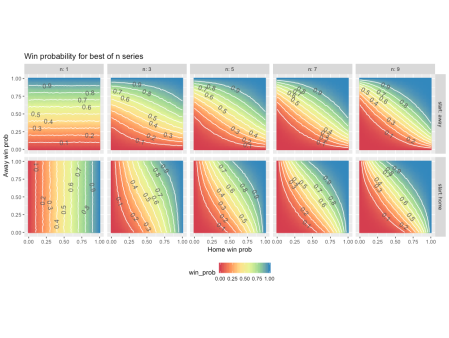


The blue area might not look very big, but sometimes teams do fall in it. The plot below is the same as the above, but overlaid with the 30 NBA teams (home/away win probabilities based on the regular season win percentages). The team squarely in the blue area? The Detroit Pistons. The two teams on the edge? The Orlando Magic and the Charlotte Hornets.

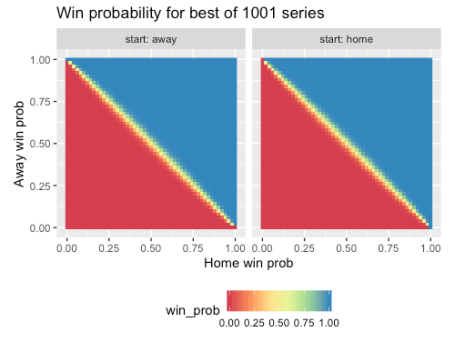


How does the length of the series affect the series win probability?

In the analysis above, we focused on best-of-seven series. What happens to the series win probabilities if we change the length of the series. Modifying the code for the first plot above slightly (and doing just 10,000 simulations for each parameter setting), we can get the following plot:



There doesn’t seem to be much difference between best-of-7 and best-of-9 series. What happens as ngrows large? The plot for best-of-1001 series is below, try to figure out why this is the case!



Code for this post (in one file) can be found [here](https://github.com/kjytay/misc/blob/master/blog/2019-04-25_best_of_n_2.R).