Los Angeles Lakers Shot Chart Analysis

Stats 416 Final Project

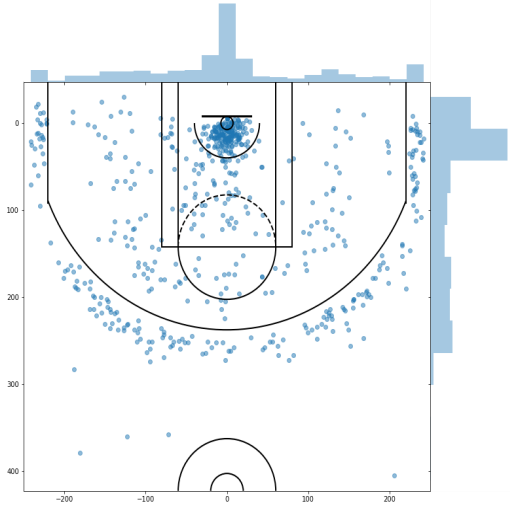
Andrew Sang

12/3/2019

Project Description

Basketball is a team sport that is played by millions of people around the world. In a regulation game, there are two teams of 5 players each, and their goal is to be able to shoot a basketball into a hoop that is elevated off of the ground. The professional league in the United States is called the National Basketball Association (NBA). One of the most popular and winningest teams is the Los Angeles Lakers. For this final project, the spatial locations for each shot that was attempted by a Los Angeles Laker during the first 7 regular season games of the 2019-2020 NBA season was collected.

A representation of this with an outline of the court is below.



Data Scraping Process

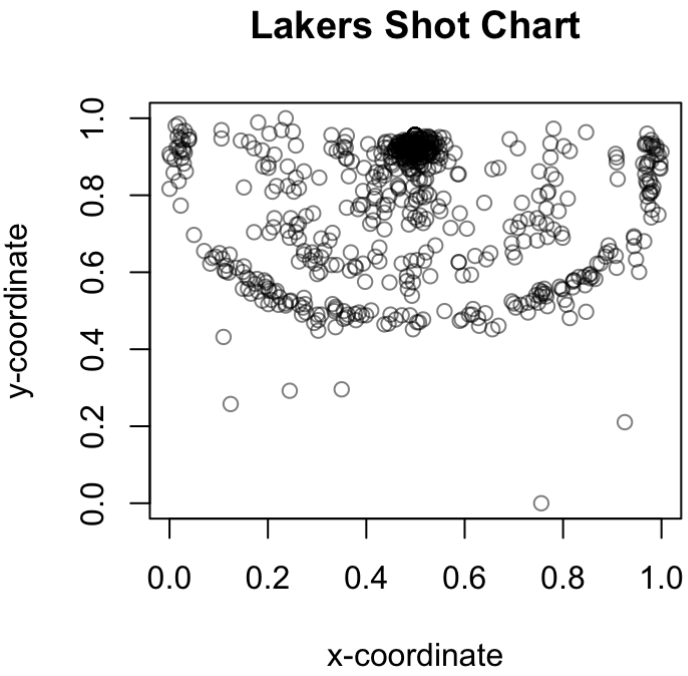
The NBA stores many advanced metrics and detailed data that is collected during games on their site at stats.nba.com. Additionally, there are multiple resources on Github (https://github.com/slieb74/NBA-Shot-Analysis/blob/master/nba_shots_scraper.py), which allow one to pull the information programmatically. An existing script that pulls information from the NBA API in order to pull the information was modified and used for this project. Unfortunately, after running the script a couple of times, access to the API became limited/blocked and I wasn't able to pull more data.

Data Description

The dataset represents shot attempts for 7 games, which resulted in a total of 625 shot attempts. Each shot attempt has many pieces of information, but the piece that was most interesting was the loc_x and loc_y fields. The locations of the shot attempts was normalized into a 1x1 window.

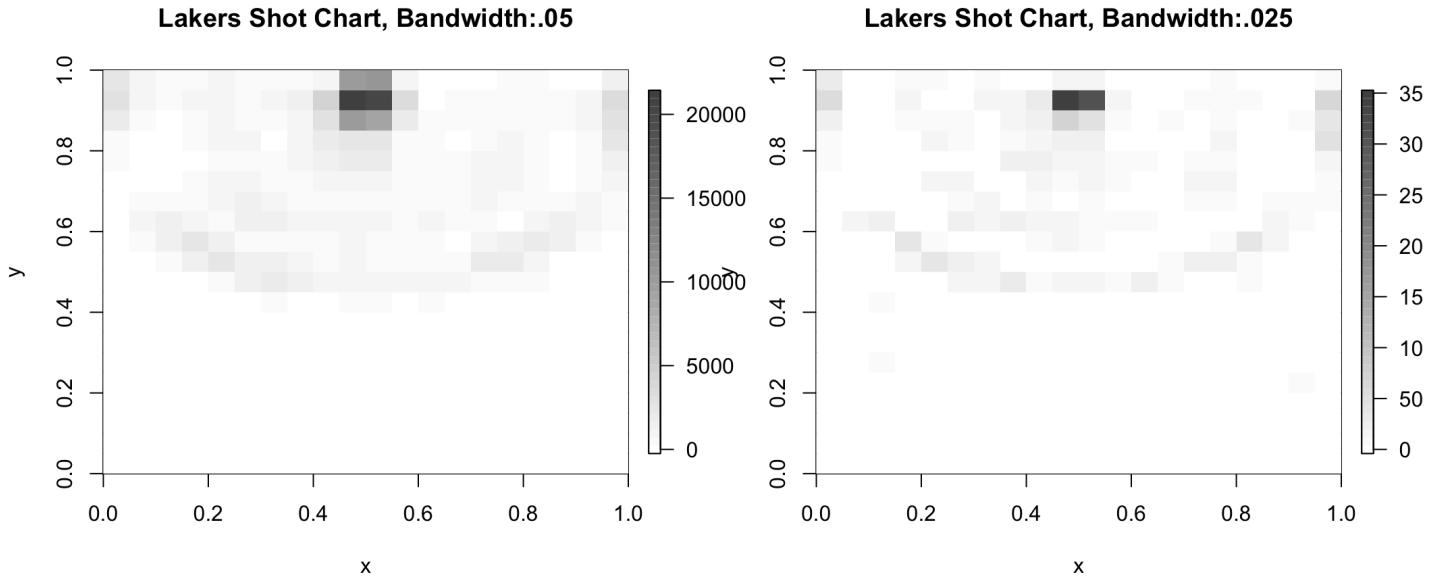
Each point here represents a shot attempt. In a real game, there are 2 baskets, and each team shoots at one basket only. Throughout the course of a game, the teams will rotate which basket they are shooting towards every quarter. For this dataset, only 1 basket is represented, and it's the basket that the team is shooting towards.

A plot of the points with their normalized locations are below.



Kernel Smoothing

Kernel smoothing for point processes is a method that allows the end-user to simplify the plot of the point process. A good way of describing the technique laid out by Professor Schoenberg is that it's similar to "smashing peas". An important parameter for kernel smoothing is the bandwidth, which controls the amount of smoothing that is applied.

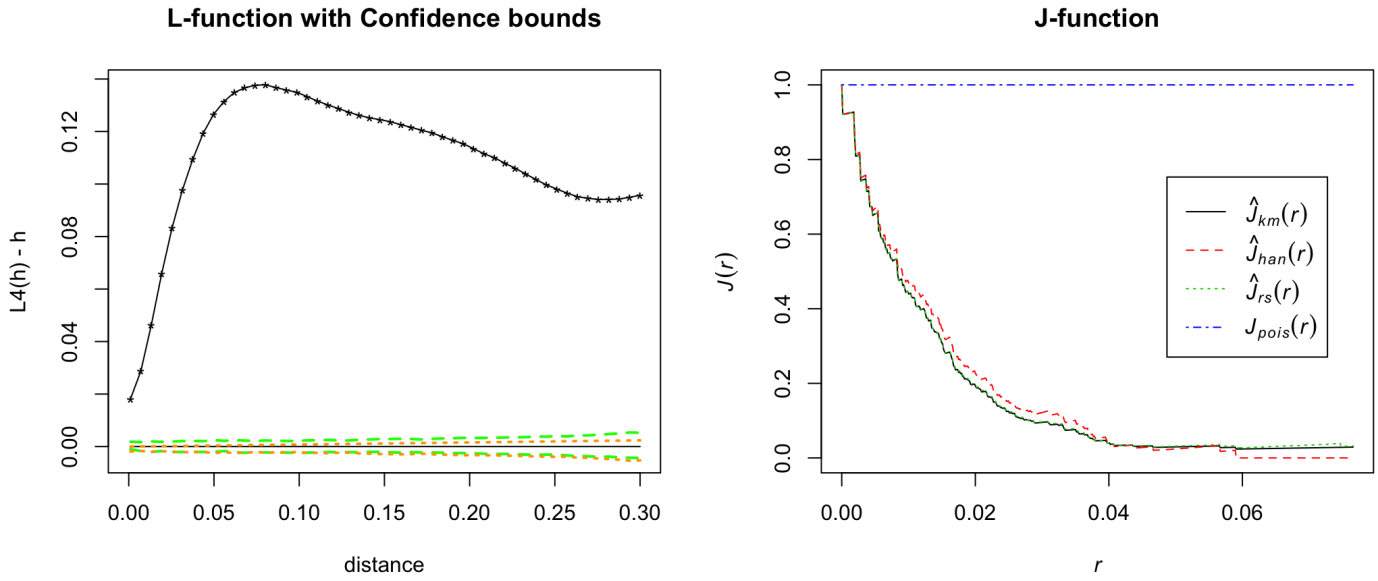


The two plots above have varying bandwidths. A smaller bandwidth means that we are smoothing by less. Looking at the two plots, the second one does a sligthly better job in representing the point process.The main reason is that the figure on the left has a moderate amount of grey in the two pixels above the basket, which is statistically accurate but not functionally accurate. In a game of basketball, we wouldn't expect many shots to come from above the basket. Additionally, the plot on the right does a slightly better job in showing the gap of points right inside of the 3 point line. As expected, there is a great deal of grey (representing a large lambda or intensity of points) close to the basket.

Looking at the data and the kernel smoothing plots, it does seem like this is an inhomogenous poisson process (as there does seem to be very different intensities of points in the total space). This seems to manifest itself in many points close to the basket, as previously mentioned. In addition to this, there are very few shots far from the basket. This can be observed when looking at the lack of intensity below y=0.4. This effect makes sense in terms of basketball, as you generally have better accuracy, the closer you are to the basket. All else equal, a shot closer to the basket has a higher chance of scoring.

L and J functions

In addition to the kernel smoothing plots, we can also look at functions like the K and L functions, which can tell us about the amount of clustering or inhibition for a given radius.

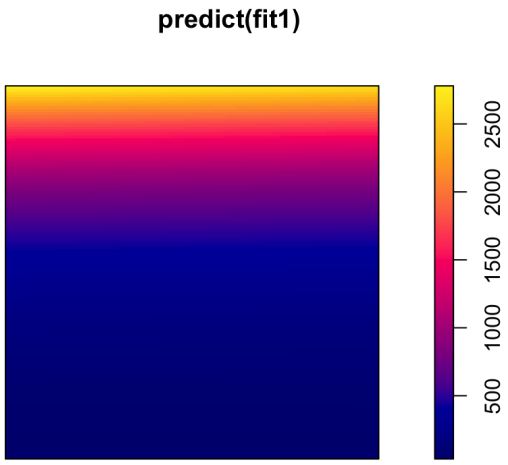


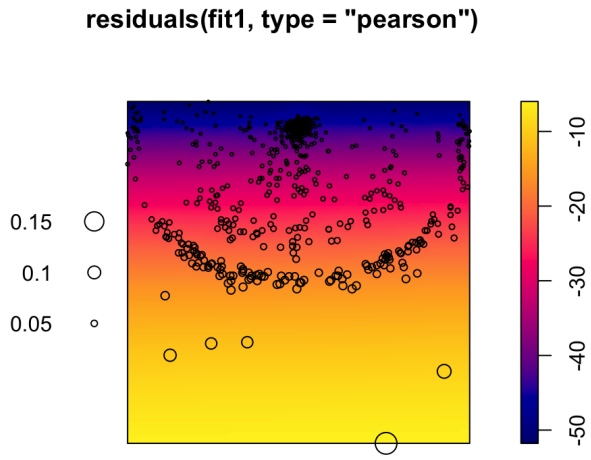
Looking at both plots, it seems like there is clustering present in both. In the L-function plot on the left-hand side, the value is much greater than 0 and higher than the simulated and theoretical bands. The bands as well as 0 represent no clustering or inhibition. Because the plot is so much greater than both, it seems like there is definite clustering present. Additionally, the J-function shows something similar where the plot is much below 1, which indicates that there may be clustering present.

First Model

For modeling, the point process was modeled as an inhomogeneous Poisson model. The ppm function from the spatstat library was utilitzed. The first model had the form $\lambda_\theta(x, y) = e^{\theta_0+\theta_1 x+\theta_2 y}$.

```
## Nonstationary Poisson process
##
## Log intensity:  ~x + y
##
## Fitted trend coefficients:
## (Intercept)           x           y
##  3.58893265 -0.04845452  4.35852696
##
##                Estimate      S.E.    CI95.lo   CI95.hi Ztest       Zval
## (Intercept)  3.58893265 0.1766170  3.2427697 3.9350956   *** 20.3204245
## x           -0.04845452 0.1385896 -0.3200852 0.2231761       -0.3496259
## y            4.35852696 0.2013356  3.9639165 4.7531374   *** 21.6480720
```



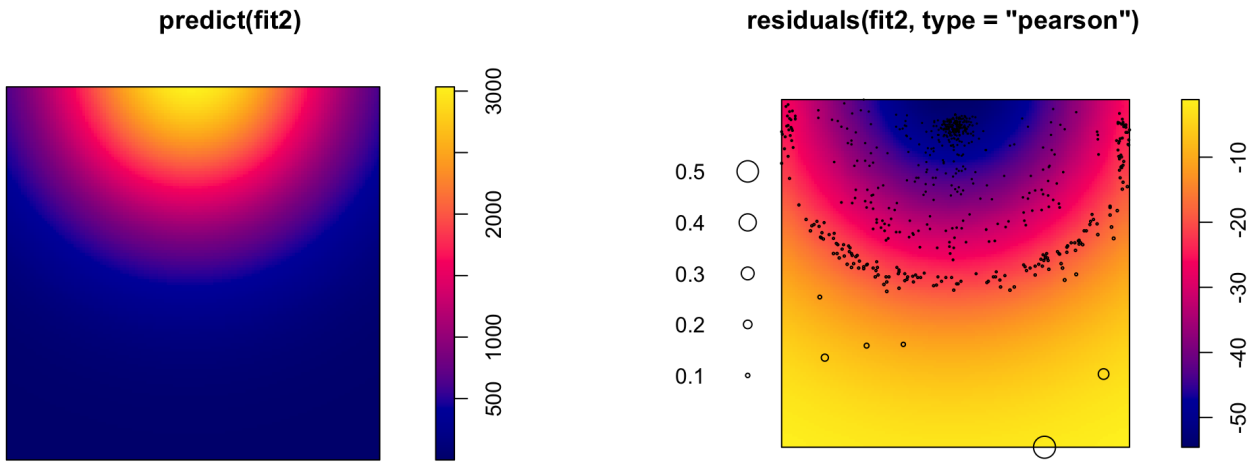


The coefficients on the model indicate that there is little movement in lambda when moving to the left or right, by looking at the relatively small coefficient on the x feature. Additionally, the relatively large coefficient on the y feature, tells us that lambda is greater for larger values of y. The trend and residual plots above show lambda as well as the Pearson residuals for the model. We can assume that the large concentration of shots near the basket largely influenced the model's fit. This simple 2 parameter (3 including the intercept) has an issue with the points around the three point arch towards the middle of the plot, as the model predicts a low lambda but sees more points than expected.

Second Model

The second model that was fitted was a more complex inhomogeneous Poisson model. The model had the form $\lambda_\theta(x, y) = e^{\theta_0+\theta_1 x+\theta_2 y+\theta_3 xy+\theta_4 x^2+\theta_5 y^2}$.

```
## Nonstationary Poisson process
##
## Log intensity:  ~x + y + I(x^2) + I(x * y) + I(y^2)
##
## Fitted trend coefficients:
## (Intercept)           x           y      I(x^2)    I(x * y)      I(y^2)
##   0.2886069   6.1818164  11.5786992  -6.3530594   0.1224843  -5.4103078
##
##                Estimate      S.E.    CI95.lo   CI95.hi Ztest       Zval
## (Intercept)  0.2886069 0.6549560 -0.9950833  1.572297        0.4406508
## x            6.1818164 1.0465110  4.1306926  8.232940   ***  5.9070728
## y           11.5786992 1.5479304  8.5448115 14.612587   ***  7.4801164
## I(x^2)      -6.3530594 0.6615121 -7.6495993 -5.056520   *** -9.6038451
## I(x * y)     0.1224843 1.0197973 -1.8762816  2.121250        0.1201066
## I(y^2)      -5.4103078 1.0414486 -7.4515095 -3.369106   *** -5.1949830
```



The second model has many more coefficients. The plot here seems to do a much better job in predicting the points around the basket, as well as displays decreasing likelihood as the distance increases from the basket.

```
##            pearson   logLik deviance       AIC
## model1 -0.82823696 3736.828 2967.880 -7467.656
## model2  0.06178036 3807.684 2826.168 -7603.368
```

The second model seems to fit much better based on the model metrics like Pearson Residuals (closer to 0), logLikelihood (higher), deviance (lower), and AIC (lower).

Comments on Inhomogeneity vs. Causal Clustering

For this dataset, it seems that the aggregation of points in the data is due to inhomogeneity (instead of causal clustering). The conclusion was drawn due to the fact that if we had a different 7 games of data from the same team and observed the data, I think that we'd see a very similar distribution of points. I would expect to see many points near the basket and then fewer shot attempts as the distance increases to the basket. Additionally, I would expect to see a lack of points directly inside of the three point line. Thus, it's more of a function of the space compared to something about the point that truly causes another point to occur.

That being said, the introduction of sports analytics in basketball has shown that when factoring at the expected value of a shot attempt = (point value: 2 or 3) x accuracy (shot taken / shot attempts), it is optimal to maximize the number of layups, dunks, and threes while minimizing the number of jump shot attempts taken inside of the three point line. This has caused a shift in the distribution of shot attempts over the last few decades of the nba (source: https://shottracker.com/articles/the-3-point-revolution (https://shottracker.com/articles/the-3-point-revolution)). This might indicate that there is some causal clustering in the sense that we might expect to see that successful or winning teams' shot distributions may have some effect on other teams over time.

Next Steps

I was not able to explore it further, but it would be interesting if we had more data around each point to be able to layer in the expected value of a shot attempt. Would historical information about the accuracy at a particular location multiplied by whether it was a 3 point or 2 point attempt lead to a better fitting model?

It would also be interesting to research different (non-Poisson models) that could take into account the three point line to better predict the regions right inside and outside of the three point line.