Vince Laird

SPORTS BETTING LINE MOVEMENT FINAL SUBMISSION
DATA SCIENCE - CAREER TRACK

Introduction

Sports betting lines and odds move and change throughout the day, and it would be useful to be able to predict this line movement. This will help people betting on sports make an informed decision on the optimal time of when to place their wager.

Dataset

The initial dataset consists of just under 2.1 million JSON documents containing odds information collected every minute for all MLB baseball games from 4/1/2017-7/2/2017, though this can be reduced to 330K record with the removal of duplicates. This covers the first 1,230 games of the 2017 MLB season. The JSON documents were copied locally weekly from a database running on MongoDB, and these were compiled to perform the capstone project.

Key data contained in the dataset included: time-based betting lines (favorite and underdog), odds, and teams. This data was cleaned to determine opening odds, closing odds, and odds 8 hours prior to the start of each contest.

Data cleaning

Several steps were taken to clean the data. Whitespace was removed so that the columns of data were consistent. All odds other than moneyline odds (i.e. the odds for each team to win the contests) were dropped. Columns were added with timestamps that could be compared and used in calculations. Duplicate rows were removed. Data was sorted and subsets were created for each team.

Once the data was subset by team, a data frame was created with one row for every minute between the minimum and maximum time where odds were collected for that team. This was necessary to standardize the time between when odds values were observed. In the original dataset, there was no consistency between when odds were updated - it varied between 3-6 minutes.

Because it is possible for a team to have odds available for multiple games at the same time, the odds for the most recent game were utilized in the analysis and the odds for the game that was "further out" (in terms of contest start time) were dropped. Also, in the event that no odds were available for a team at a certain time (teams have off days), the last known moneyline odds values were used for these missing times. Helper data frames were utilized to map other info to each team's data frame, such as game number and contest start time. The final data frame for each team was also imported into R.

Odds values themselves also needed to be standardized, for two reasons: 1) odds values are not continuous, and 2) a team's moneyline odds from one game to the next game are also not continuous.

To make the odds continuous, a simple calculation was done to the actual values: if odds were less than zero, 200 was added to the value. This is because odds values "break" between 100 and -100. When a team is said to have odds of +100, that is the same as saying "even", where if you bet and win you will double your money. -100 is the equivalent of that.

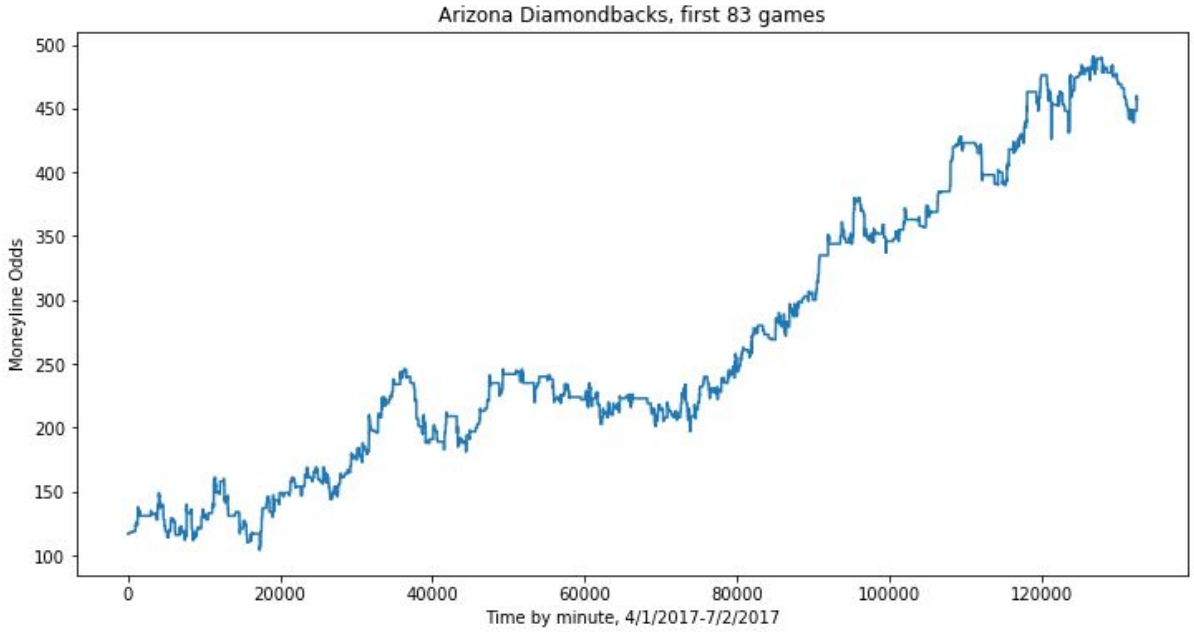
To make the odds continuous for an individual team between games, an offset value was determined and added to all subsequent games. This is necessary because a team could be a heavy underdog one day and a heavy favorite the next day, depending on who they are playing, who is pitching, etc. For example, if a team ended the day with moneyline odds of +200 (a $1 bet yields a $2 return), and the next day they opened as -200 (a $1 bet yields a $0.50 return), the offset would be the difference, which would be 200 (as we are adding 200 to the value of -200, making that value 0, and the difference between the original closing odds of +200 and 0 is 200).

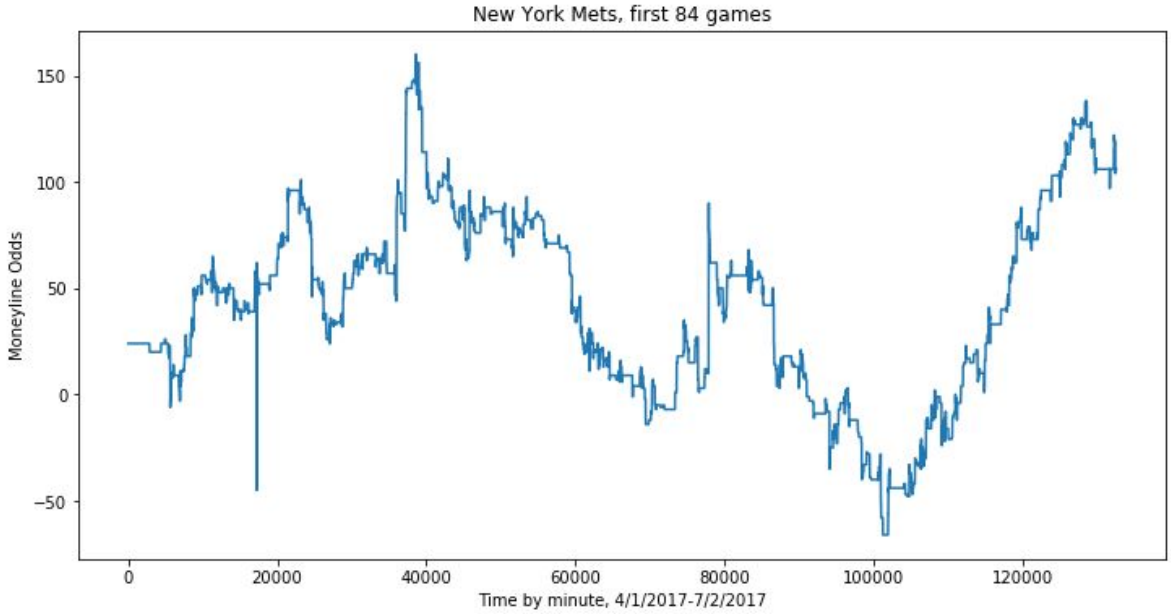
Subsetting the data by team was a key component of the analysis. Trends emerged when looking at how odds changed for particular teams over the entire season and over multiple

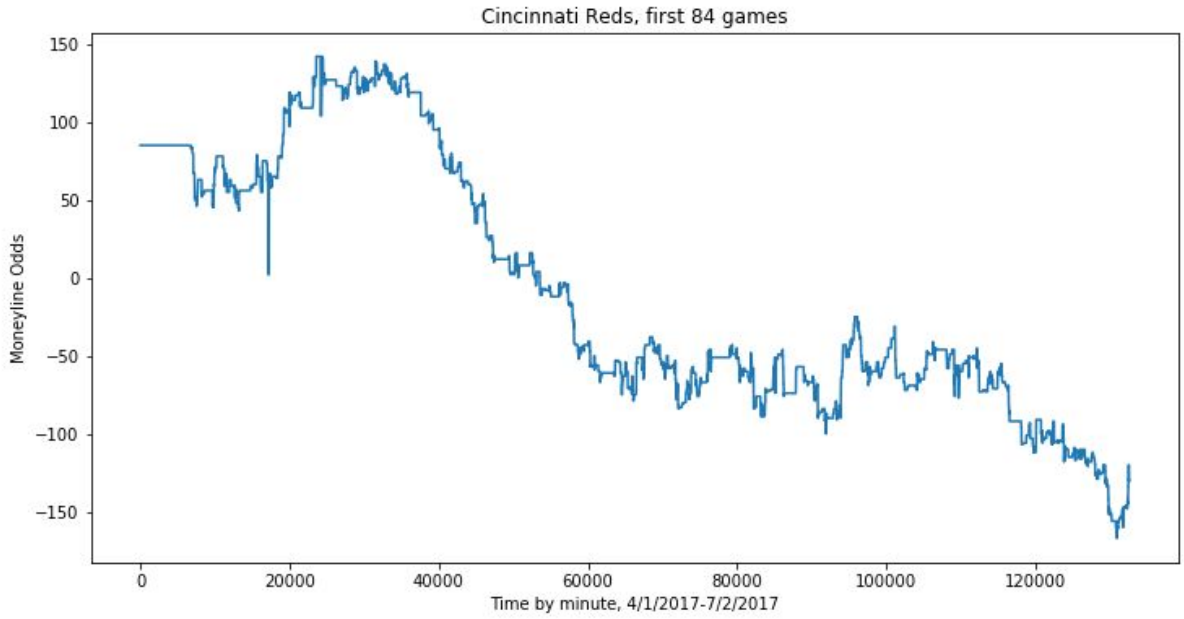
game periods, and these were the trends which proved to be most useful when determining the ideal time to wager on MLB contests.

Data exploration

Once a data frame was created that contained odds values for all times and all teams, the odds were plotted and viewed to try to get a feel for how best to approach the problem.







The plots for each team varied greatly, but the three plots above give an example of the types of things that were observed. The plot for Arizona's odds shows what appears to be an upward trend; the plot for the Mets looks random except for the last 15%, which shows an upward trend; the plot for Cincinnati shows a downward trend, followed by a long period where the odds don't seem to move strongly in either direction.

Simple trend

The first evaluation method utilized was subtracting odds at different points in time to determine a trend, then simply assuming that trend would continue. The time period utilized could have been anything, I simply picked eight hours prior to each contest. Based on individual games, using the three teams mentioned above in the plots (Arizona, N.Y. Mets, Cincinnati), the results were poor. An example of this method:

Arizona, game 1: odds opened at 117. Eight hours prior to game 1, odds were at 126. Assume this (upward) trend will continue, and wait till right before the game begins to bet. Result: Odds closed at 131, upward trend did continue, and this method would have a net gain of 5 (131 - 126).

The total results were: Arizona, -218. N.Y. Mets, -117. Cincinnati, -71.

Results were better when the games were paired together. I opted to pair games by three; teams often play each other in three game series, and trends extending for multiple games in the plots. Example:

Arizona, game 3: odds for game 1 opened at 117. Eight hours prior to game 3, odds (with offset included) were at 129. Assume this (upward) trend will continue, and wait till right before the game begins to bet. Result: Odds for game 3 closed at 116, upward trend did not continue, and this method would have a net loss of 13 (116 - 129).

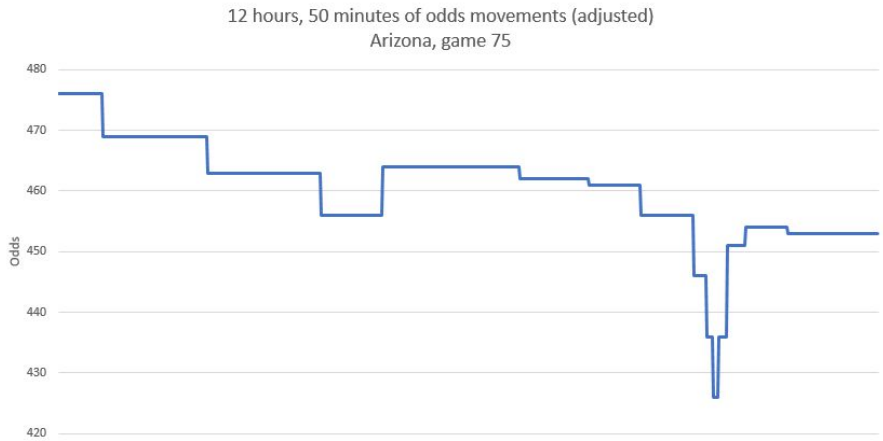
The total results for this method were: Arizona, 118. N.Y. Mets, 105. Cincinnati, -17.

Overall, this three-game pairing simple trend method resulted in a gain of 206. Based on the total number of games this covered (248, with 3 games dropped due to rainout or incomplete odds for games on the last day of the date range), the average gain from this method was 0.83 - not worth much, but better than nothing. For reference, the difference between the opening and closing odds for these three teams per game was 11.7.

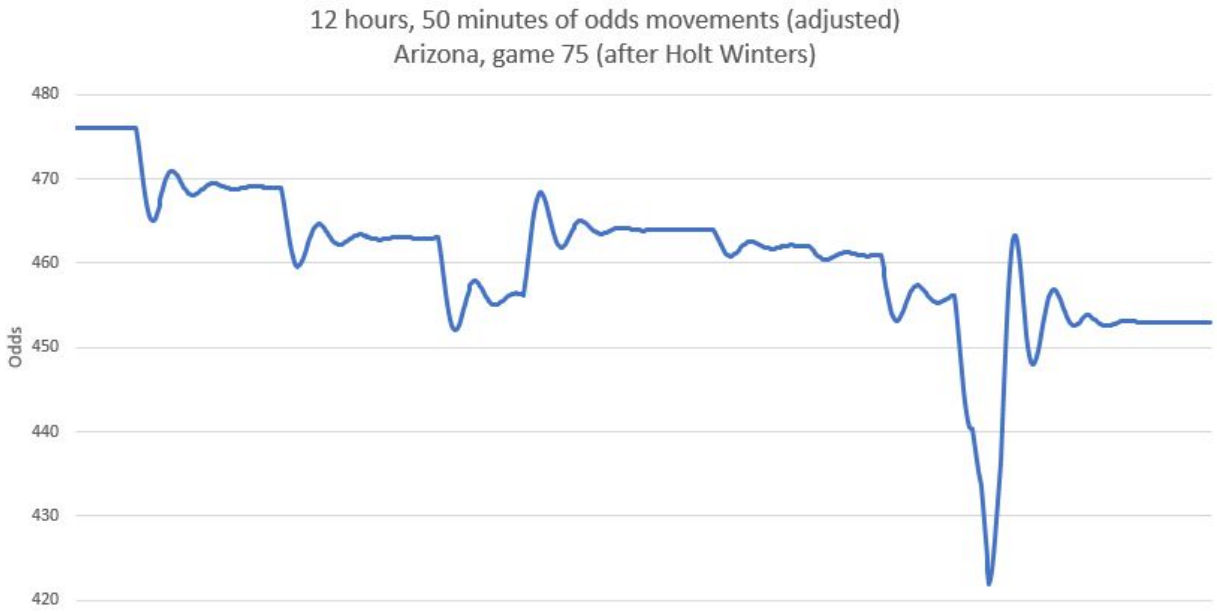
Bsts: Bayesian structural time series

The second evaluation method was bsts. This method was inspired by an excellent blog post on bsts models by Kim Larsen (link in references).

Because the data has jumps, triple exponential smoothing, also known as the Holt Winters method, was utilized in an attempt to smooth out the data. Here is an example of a period of jumps in the Arizona odds data:

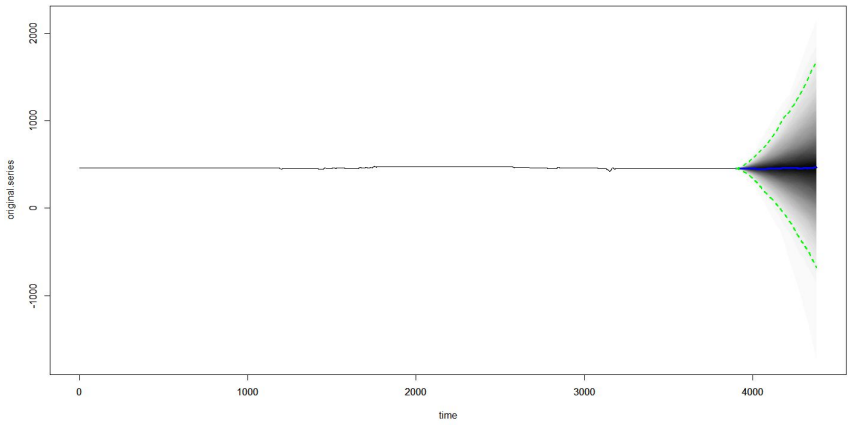


Here is a plot of the same data, post Holt Winters:



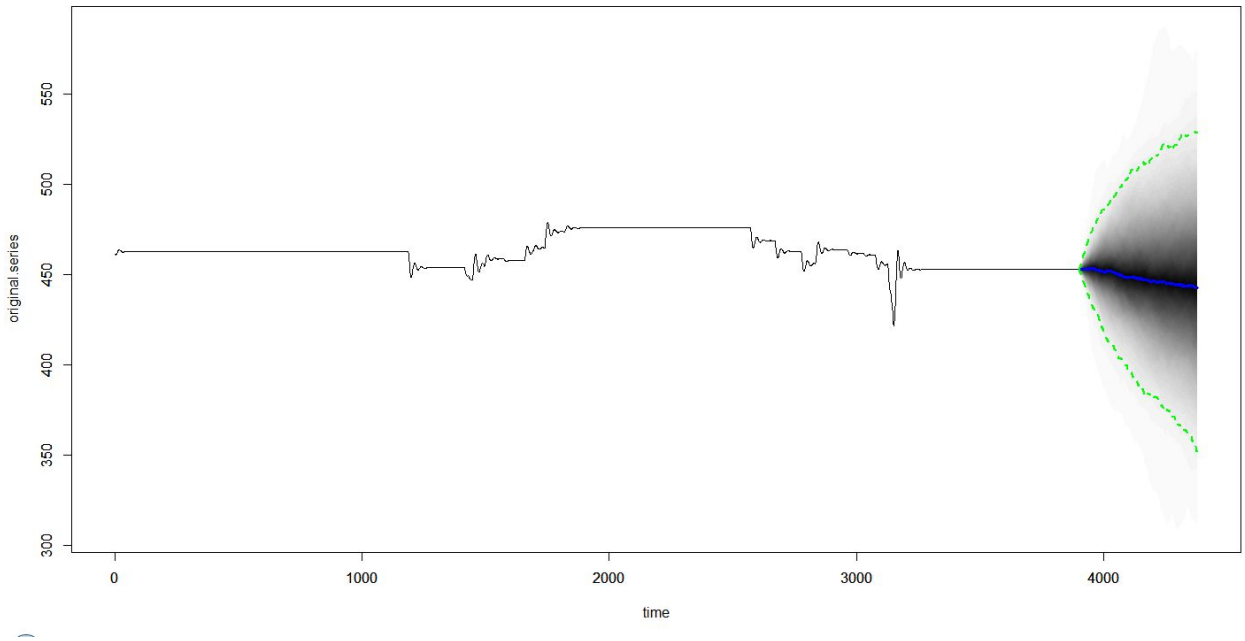
To evaluate bsts, R was used as the bsts package was only available for unix based systems at the time this capstone was done, and I do not have a unix based OS.

The first results from the bsts package were not realistic, due to the fact that different criteria must be used when projecting a large number of periods ahead. Because we are using data for prior to eight hours before each contest begins, and our time series is by minute, and our forecast horizon is 480 periods. When using bsts in R with a local linear trend, the spread was much too great; below is a bsts model using Arizona, games 74-76, with forecasts for odds movements eight hours prior to the start of game 76:



The final period had odds between over 2,000 and under -1,700: not realistic.

Changing the bsts model criteria from a local linear trend to a semilocal linear trend was the solution to this problem. Here is the same plot with this change:



Two functions were created to carry out the bsts modeling. 200 Markov chain Monte Carlo samples were used; initially, 1,000 MCMC samples were used for each game, but this caused the functions to take in excess of two hours for one team. The final period of the horizon forecast was evaluated to see which percentage was higher, the number of MCMC samples greater than the odds value eight hours prior to contest start, or the number of samples less than this value. Using this method to determine whether or not to bet immediately or wait, the results for our three sample teams:

Arizona, 22. N.Y. Mets, 59. Cincinnati, -47. Note, these results vary due to the results of new MCMC samples each time the functions are executed.

The bsts model did not perform as well as the simple trend method. When the MCMC draws favored one side over their other (i.e. more than 60% of the draws indicated that the odds would either be greater or less than they were eight hours prior to the start time of the contest), the bsts model was correct 57% of the time. However, this situation only occurred in 26% of games. The results of this method by team:

Arizona, 36 (12-for-22). N.Y. Mets, 2 (12-for-21). Cincinnati, 17 (13-for-22).

One other evaluation of this model was done, and that was to simply take the mean of the final forecast horizon period and compare to the odds eight hours prior to the contest start. In terms of the bsts evaluations methods, this yielded better results than looking at the count of MCMC draws above vs. below the odds value prior to contest end, but still not as good as the simple trend method. Here are the results for this method:

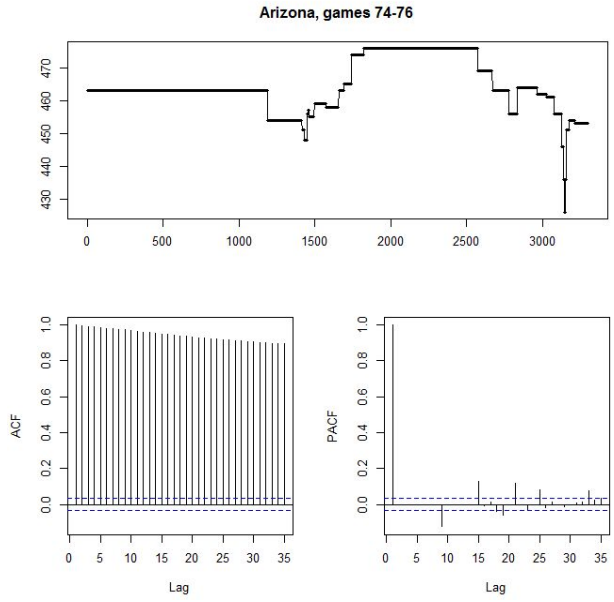
Arizona, -24. N.Y. Mets, 27. Cincinnati, 103

Note: the bsts library in R threw an sdy error on a small number of games for the N.Y. Mets and Cincinnati data frames (total of 4 games). This was likely due to a small number of contests having the same odds leading up to the game start time (no odds value changes). These games were removed from the analysis.

ARIMA

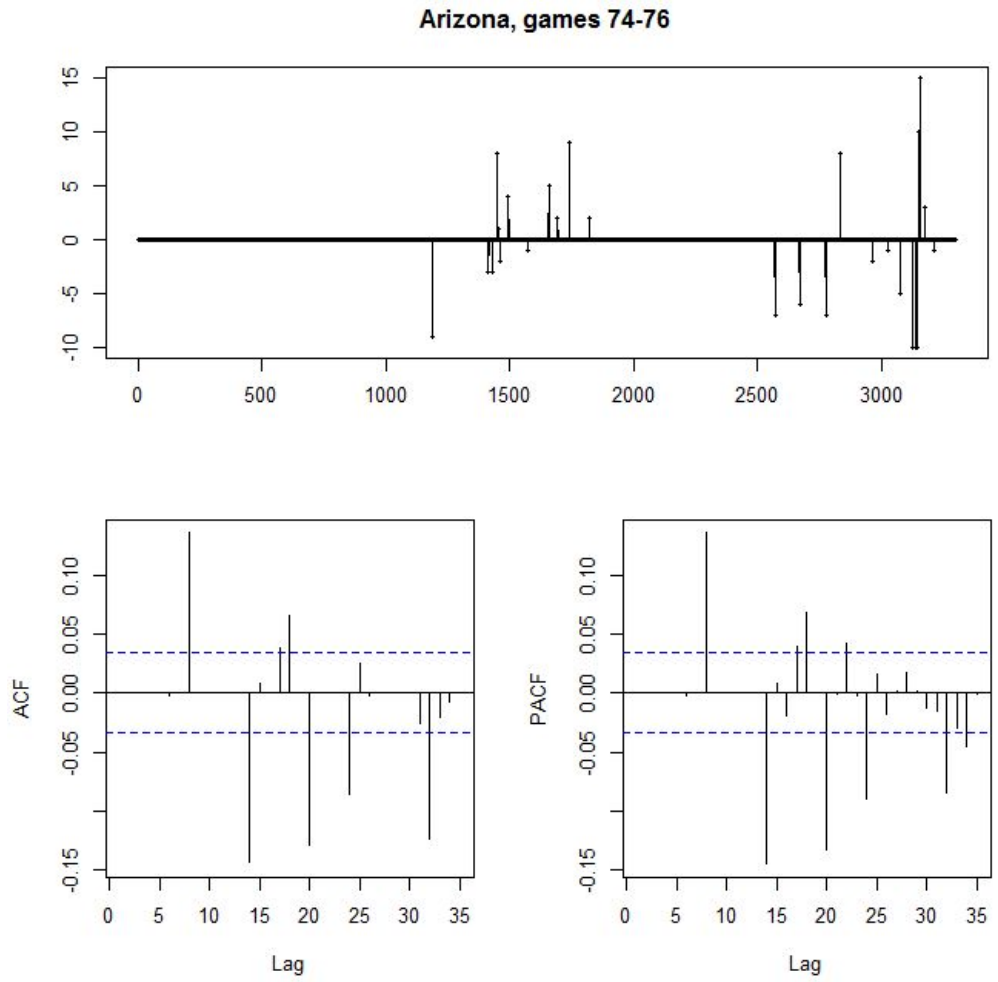
ARIMA modeling was also done, but this method proved difficult and the results were poor.

The first challenge of using ARIMA was to determine the candidate models and parameters to use for the model. Prior to using any differencing functions, the time series and ACF/PACF plots are shown below for a sample three-game period:



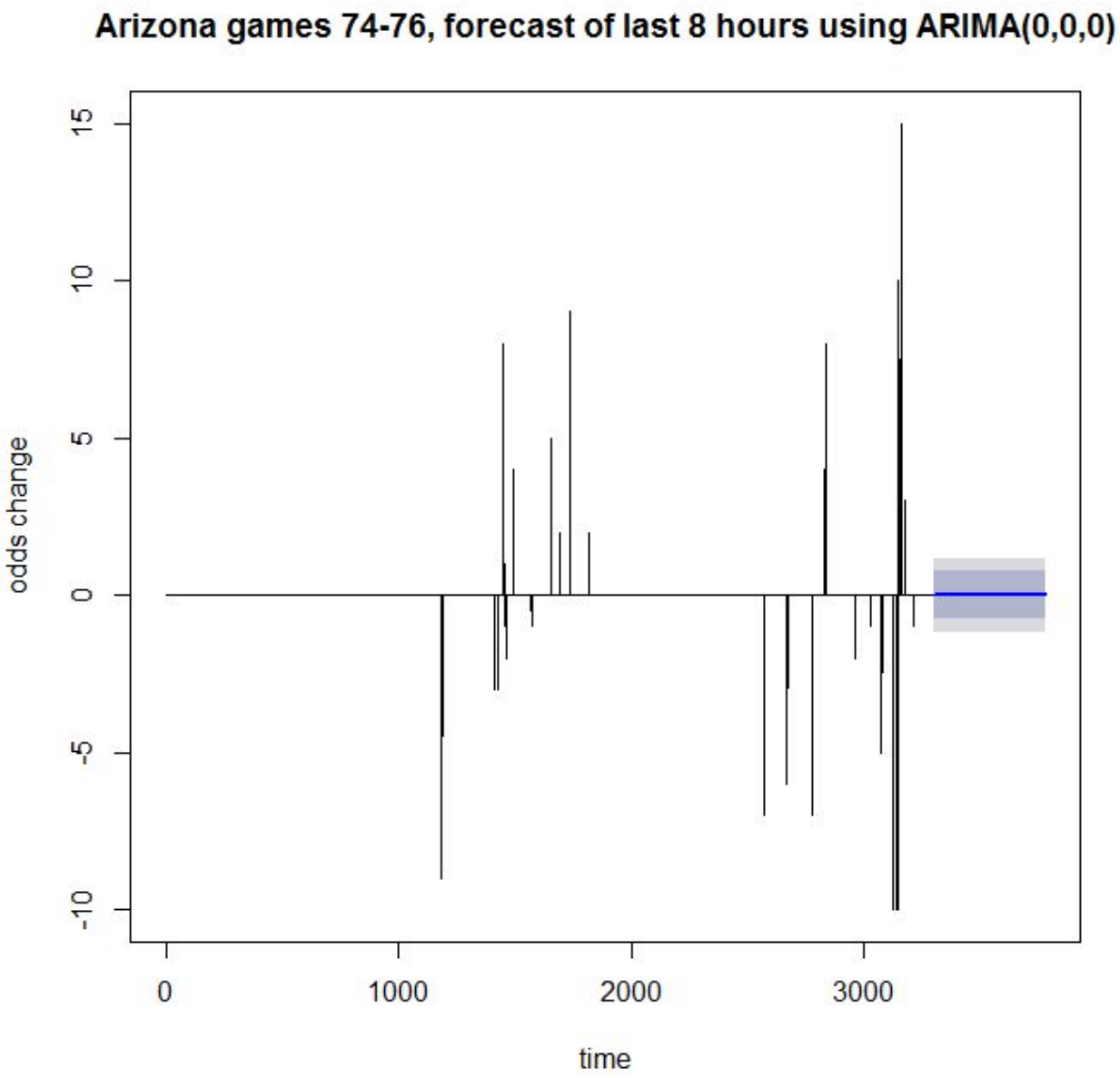
The above ACF is decaying very slowly, and remains well above the insignificance range (dotted blue lines). This is indicative of a non-stationary series.

Differencing the time series resulted in the following tables:



After multiple trials of different ARIMA models, I was unable to reduce the spikes in the ACF/PACF plots outside of the insignificant range. Using auto ARIMA in R lead to the suggested parameters of (0, 0, 0), which is the white noise model. Also, with the large periods in the time series when there aren't any movements at all, followed by sharp movements, I was unable to get any type of realistic results using ARIMA with several different parameter settings.

On the following page is a forecast using the ARIMA model, which was simply a straight line.



Conclusion

The sheer amount of variables that impact odds line movements made modeling and forecasting these movements much more difficult than anticipated. Aside from time periods where odds may not change at all, for every MLB game, major factors can and do change odds on a seemingly random basis. Leading up to each contest, any of the following may occur at random times: lineup changes, pitching changes, weather changes, large

amounts of money being bet at different sports books on a particular team, injuries, players being given days off, etc.

Multi-game trends have been observed in some cases for some teams. Viewing these trends by team and making the determination that if odds have increased or decreased over the past few days is the easiest way to forecast future changes; simply assume whatever trend appears to be present based on plots is going to continue. This method did prove right more often than wrong, though the advantage of knowing this isn't much.

References

Blog post on bsts model vs. ARIMA:

http://multithreaded.stitchfix.com/blog/2016/04/21/forget-arima/

Information on ARIMA modeling:

https://www.otexts.org/fpp/8